Computing and Informatics, Vol. 37, 2018, 1293–1312, doi: 10.4149/cai_2018_6_1293

NEW METHODS OF UNCHEATABLE GRID COMPUTING

Jianhua Yu

School of Mathematical Sciences, South China Normal University
Guangzhou, P. R. China
e-mail: yujianhuascnu@126.com

Yuan Li

Department of Mathematics, Winston-Salem State University
NC 27110, USA
e-mail: liyu@wssu.edu

Abstract. Grid computing is the collection of computer resources from multiple locations to reach a common goal. According to the task publisher's computing power, we will classify the deceptive detection schemes into two categories, and then analyze the security of deceptive detection schemes based on the characteristics of computational task function. On the basis of double check, we proposed an improved scheme at the cost of time sacrifice called the secondary allocation scheme of double check. In our scheme, the security of double check has been greatly strengthened. Finally, we analyzed the common problem of High-Value Rare Events, improved the deceptive detection scheme due to [1], and then put forward a new deceptive detection scheme with better security and efficiency. This paper is revised and expanded version of a paper entitled "Deceptive Detection and Security Reinforcement in Grid Computing" [2] presented at 2013 5$^{\text{th}}$ International Conference on Intelligent Networking and Collaborative Systems, Xi'an city, Shanxi province, China, September 9–11, 2013.

Keywords: Distributed, deceptive detection, ringers, grid computing

Mathematics Subject Classification 2010: 68Q85

1 INTRODUCTION

The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. Grids are a form of distributed computing whereby a "super virtual computer" is composed of many networked loosely coupled computers acting together to perform large tasks. Grid computing combines computers from multiple administrative domains to reach a common goal to solve a single task, and may then disappear just as quickly. By grid computing, we can use the computer's idle capacity of tens of thousands of volunteers from all over the world through the Internet and analyze the electrical signal from outer space and find the hidden black holes, at the same time, explore the possible existence of alien wisdom life; still, we can look for more than 10 million digital mason prime numbers; and we can also search for and find more effective drugs to against HIV and to complete the project which needs surprisingly large amount of calculation.

Past few years have seen a tremendous growth in grid computing with its effect being felt in the biotechnology industry, entertainment industry and financial industry, etc. [3, 4, 5, 6, 7, 8, 9]. For example, the Search for Extra-Terrestrial Intelligence (SETI@home) project [6], which distributes to thousands of users the task of analyzing radio transmissions from space, has a collective performance of tens of teraflops. Another Internet computation, the GIMPS project directed by Entropia.com, has discovered world-record prime numbers. Future projects include global climate modeling and fluid dynamics simulation. There are also cryptographic protocols which allow to implement provably optimal systems in theory [10, 11, 12, 13, 14]. However, these algorithms are often computationally expensive in practice.

With the number of participants who join the grid computing increases, inevitably, there will be a lot of cheating. To gain more profits, some people may provide some false certificates to prove that he/she himself/herself has provided a lot of resources, including the computing time or the computing power, etc.; and there are also some people who will directly return some false results. In addition, some people can get more useful information from the calculated target. For example, in the search for an effective approach of grid computing, a participant finds an effective treatment in the grid computing. However, he does not provide it to the supervisor of the grid computing, but sell it to another hospital, in order to gain more profits. Therefore, we need to deal with two things in the grid computing:

1. How to detect the participant cheating;
2. In certain cases, how to hide the results which the task publisher really needs.

2 RELATED WORK

With the rapid development of grid computing, the cheating behavior in the grid computing received more and more attention, and there exists many research achievements [1, 11, 15, 16, 17, 18]. In order to detect the behavior of fraud in the

grid computing, Golle and Mironov proposed ringers scheme [15], which can protect against coalitions of lazy cheaters provided that the computational tasks all involve the Inversion of a One-Way Function (IOWF) f for a given value y, as in the distributed.net attacks on cryptographic functions. In the ringer scheme, during the initialization stage for each participant, the supervisor randomly selects several inputs x_i that will be assigned to that participant and computes $f(x_i)$ for each one. Then, in addition to the value y that the supervisor wishes to invert, the supervisor also sends to that participant all the "ringers" the supervisor has computed for him. The participant must report the pre-images of all the ringers (as well as the pre-image of y if he was lucky enough to discover it). That is, the participant needs to compute f on x for all x in his input domain D and return the pre-image of y if found, and he also has to return all the ringer pre-images he finds. By remembering the ringers for each participant, the supervisor can easily verify whether each participant has found all his ringers or not. If he has, then the supervisor is assured with reasonable probability that the participant has indeed conducted all his computations. In the ringers scheme, the supervisor mixes some previous computation results with the corresponding input and sends them to the participant. Golle and Mironov have confirmed that in Ringers scheme, the participant's cheating success opportunity is very small, but the use of computational task function must be one-way. If participants collaborate with each other, they will know the number of the ringers in the scheme. Therefore, Golle and Mironov proposed the subsequent Hybrid Scheme.

Szajda et al. extend the ringers scheme [17] to deal with other general classes of computations, including optimization and Monte Carlo simulations. They propose effective ways to choose ringers for those computations. To find wrong results created by lazy participants or cheaters, today many systems simply replicate the work units and use the majority rules to decide the correctness of results [20]. However, this approach often implies a waste of CPU cycles. For tasks where the verification is less expensive than redoing the computations, alternative approaches are preferable.

Golle and Mironov [15] propose a solution where a server secretly precomputed the results for a set of input values, the so called ringers. These ringers are then interspersed among the ordinary input values of a work-unit. If a lazy client does not compute the entire unit, it is likely to miss a ringer which can easily be detected by the server. The disadvantages of this approach are the need for precomputation, the redundant computation that inherently occurs, and the fact that a cheater is still likely to be undetected when cheating on a very small fraction of input values only. Szajda et al. have generalized this approach in [17].

Du et al. [18] present a commitment-based sampling scheme for cheater detection in grid computations based on Merkle trees. A server selects some samples which are assigned to a participant. The participant has to commit to its results which are subsequently checked. The drawback of their approach is the additional burden on the server which has to recompute some work-units itself.

In [1] Du and Goodrich introduce the deceptive detection scheme named Searching for high-value rare events. In their criterion-expansion scheme, under the circumstance of one-to-one correspondence in the computational task function, it has very good safety, and it can defeat three kinds of cheating models put forward in this article, but under the circumstance of more-to-one correspondence in the computational task function, the participant will be able to position the supervisor to find the value of the data according to the characteristics of some of the data. In this case, the solution cannot resist the second and the third deceptive model in this article. At the same time, the criterion-expansion detection scheme cannot resist the collusion of the participants. In the criterion-reduction scheme of [1], it lacks the necessary honesty testing steps, and it even cannot judge whether the participant honestly computes all the data on the mission. Hence, there will be some serious defects in terms of security.

Based on the scheme in [1], we present a new scheme using the Monte Carlo simulation method. This scheme can resist the semi-honest cheater and hoarding cheater. This scheme is available for one-to-one function and more-to-one function. At the same time, it can resist the semi-honest cheater and hoarding cheater. But most of schemes are available for one-to-one function.

We also design an other new scheme. In the new scheme, we add some variables which satisfy the criterion Y to the domain of the function in the computational task. For example, we are looking for signs of life in outer space in grid computing. Since the value of the sign of life in outer space data is very very rare, the participant can easily judge the criterion Y according to these characteristics, and therefore cannot hide it. If we add appropriately some biological signs of life on earth and make the domain of Y expand appropriately, thus let a rare event become not rare, and even if participants find the valuable data to satisfy the standard of value Y, they are still unable to distinguish whether these data come from outer space or from earth. Thus, it protects rare events.

In most of cheating detection schemes, the computation task X had not been dealt with. In order to achieve the purpose of hiding Y, they only used new criterion Y' to replace Y. We present a new method. Let Y be y, we can construct a new set C, for $\forall x \in C, f(x) = y$. In this way, we expand the task to $X \cup C$. Compared with[1], this new scheme has greatly improved the security.

3 THE MODEL

3.1 Problem Definition

In grid computing, when a supervisor releases a computational task, he/she needs to compute $f(x)$, $\forall x \in X = \{x_1, x_2, \ldots, x_n\}$ and finally picks out valuable data from the results of x_i and $f(x_i)$. Due to the limited computation ability of the supervisor, he/she will assign tasks to n participants. The task set of every P_i $(i = 1, 2, \ldots, m)$ is D_i. Finally, each participant will return the valuable data to the supervisor in their respective task set.

Here we will introduce some basic terminology in the grid computing.

- Honesty Ratio r: For $\forall x \in D$, the participant needs to compute $f(x)$. But in fact, the participant may only compute $f(x)$ for $x \in D' \subset D$. Define honesty ratio r as $\frac{|D'|}{|D|}$.
- Honesty Participant Ratio p: In grid computing, the ratio of the honest participants in all of the participants should be considered.
- Valuable Events: Data that the supervisor is interested in.
- Honesty Return Ratio h: h is the proportion of correct valuable events which were returned by P_i.
- Criterion y: The criterion can be a specific value or a range, and it can also be a literal description language, etc. Using it, the supervisor and participants will check whether $f(x)$ satisfies the set of the valuable events.
- Pseudo-Valuable Events: The supervisor selects some data which satisfy the criterion to hide the true criterion.
- High-Value Rare Events: The valuable data that satisfy the supervisor's criterion Y is rare, and it even does not exist. For example, in the process of searching for rare blood type, the rare blood type is the rare event.
- Ringer: A ringer is a value chosen by the supervisor in the domain of f.
- $Pr(r)$: Assume that the participant is assigned a task that consists of computing $f(x)$ for all $x \in D$, where $D = \{x_1, \ldots, x_n\}$. If a participant computes the function f only on $x \in D'$, where $D' \subseteq D$, we define the honesty ratio r as the value of $\frac{|D'|}{D}$. When the participant is fully honest, the honesty ratio is $r = 1$; otherwise $r < 1$.

Let $Pr(r)$ be the probability that a participant with honesty ratio r can cheat without being detected by the supervisor. Let $C_{cheating}$ be the expected cost of the required task. We say a grid computing is uncheatable if one of the following or both inequalities are true:

$$Pr(r) < \varepsilon, \text{ for a given } \varepsilon(0 < \varepsilon \leq 1) \text{ or } C_{cheating} > C_{task}.$$

In grid computing, what the supervisor mainly faces is how to ensure the following four conditions established:

1. Every participant P_i will compute $f(x)$ for all data in D_i;
2. Every participant P_i cannot know which data is the valuable data that the supervisor is looking for;
3. The supervisor is able to distinguish the value of the returned data correctly;
4. Even if the participants act in collusion with each other, they cannot break the effective implementation of the task.

In grid computing, the organizer of the computation is the supervisor, at the same time there are a lot of untrusted participants. The Supervisor will allocate computing tasks.

The supervisor can distribute one or more computing tasks $X = \{x_1, x_2, \ldots, x_n\}$ according to the actual situation. The computing task of P_i is D_i, and all of them is $X = D_1 \cup D_2 \cup \ldots \cup D_n$.

For a specific computation, it is performed by two functions described below.

- A computational task function $f : X \to T$ defined on a finite domain X. The goal of the computation is to evaluate f on all $x \in X$. For the purpose of distributing the computation, the supervisor partitions X into subsets. The evaluation of f on subset D_i is assigned to participant P_i.
- A screening function S. The screener is a function that takes as input a pair of the form $((x, f(x)); y)$ for $x \in X$, and returns a string $s = S((x, f(x)); y)$, where y represents the criterion. S is intended to screen for "valuable" outputs of f that are reported to the supervisor by means of the string s.
- A payment scheme P. The payment scheme is a publicly known function P that takes as input a string s from participant i and outputs the amount due to that participant. We require that P may be efficiently evaluated. Specifically, one evaluation of P should equal a small constant number of evaluations of f. In many articles, they use the P function, but we do not use it.

3.2 Three Deceptive Models

This article uses three similar kinds of deceptive models proposed by Du and Goodrich [1]. We assume each participant is given a domain $D \subset X$, and his/her task is to compute $f(x)$ for all $x \in D$.

1. **Semi-Honest Cheater Model.** In this model, the participant follows the supervisor's computations with one exception: for $x \in \breve{D} \subset D$, the participant uses $\breve{f}(x)$ as the result of $f(x)$. Function $\breve{f}$ is usually much less expensive than function f; for instance, $\breve{f}$ can be a random guess. The goal of the cheating participant in this model is to reduce the amount of computations, such that it can maximize its gain by "performing" more tasks during the same period of time.
2. **Hoarding Cheater Model.** In this model, the participant conducts all the required computations. However, the participant will keep the computation results if the results are valuable. However, the participant will keep the computation results if the results are valuable. For example, if the computation is to search for a rare event, a "lucky" participant who has found a rare event might report a negative result because of the value of such a result. This type of cheating behavior is a cheating on the screening function S.
3. **Malicious Cheater Model.** In this model, the behaviors of the participant can be arbitrary, or even be hostile. For example, the participant P does all the

correct calculation $f(x)$ to all of the data in D, but he deliberately returns to the supervisor wrong value screening results, to achieve the purpose of disorder and confuse the supervisor to work normally, the participant may be the supervisor of the competitors.

4 DECEPTIVE DETECTION MODEL

Whether the supervisor has the computing power is decided the way they test cheating. On this basis, we divide the deceptive detection model into two forms.

4.1 The Supervisor Has Not Computing Power

Because the supervisor has not ability to compute, so it is given a task data x, the supervisor is unable to determine the accuracy of $f(x)$. The supervisor can only verify according to the participant. For this case, we can adopt the method of double check.

Kuhn et al. [21] consider a grid framework consisting of a server and a potentially large number of clients. A client is the logic entity with which the server interacts. The server sends its work-units (or tasks or jobs) to clients, which return the corresponding set of results. A participant is a user who has registered an account for the project. He/she may use one or more computers (or machines) working for him/her in the project. Moreover, a computer can correspond to one or several clients. The computational resources of the clients are heterogeneous.

The server distributes two different kinds of tasks to the clients. Work-units are the main computational tasks of the grid computing framework; checking units require the client to perform a number of checks for the different results. They assume that the main incentive for participation in the project are credit points: There may be websites listing the credit points earned by the different users, or there may even be ways to convert credit points into real money, by a lottery, for example. In their system, a client earns credit points for computing both work-units and checking units. The number of points is thereby proportional to the amount of work, such that a participant is indifferent between the two tasks.

It is of prime importance that the credit points be earned honestly. Their algorithms' goal is to make sure that participants only get the credits they really deserve, and that the system is not flooded with wrong results. They distinguish between two kinds of clients: good clients and bad clients (or cheaters). They consider a harsh model where all cheaters form a single coalition. Today's systems such as Seti@Home are reported to have roughly 1 % cheaters.

A checking algorithm which identifies dishonest behavior can achieve a higher effectiveness if it interplays with a mechanism to punish cheaters. They assume that a wrong result or an improper check implies that the corresponding client is a cheater. Being debunked as a cheater basically implies that the corresponding client's user loses all its credits, and the corresponding account is closed.

1. **Double-Check.** The supervisor will assign the subtask to two or more different participants, and then compare whether the returned results are consistent with each participant's. If it is consistent, the supervisor believes that the participant is honest, otherwise, allocates the task again.

   Security Analysis. If there is no collusion between participants, double check scheme has a good security; if the participants of the same task are in collusion with each other, they will return the same results to the supervisor. The supervisor will still believe they are honest participants. So double check scheme cannot defeat the joint attack of the participants.

   In order to resist the joint attack, we can adopt the deformation mode of double check.

2. **Deformation 1.: The secondary allocation in double check scheme.** The supervisor will randomly assign the subtask D to the participant A firstly. After it returns the results, the supervisor will randomly assign the same task D to another participant B. For determining honesty of the participant A and B, the supervisor compares conformance of the returned data from A and B.

3. **Deformation 2: The samples testing in double check scheme.** The computing task X is divided into n parts, i.e. $X = D_1 \cap D_2 \ldots \cap D_n$. The participant P_i has computing task D_i. The supervisor randomly selects $d_i \in D_i$ as the honesty sample data of P_i. When all the results are returned, the supervisor will assign $d_1, d_2, \ldots, d_n$ randomly to n new participants (at this stage, adopting the secondary allocation in double check scheme to verify these n new participants' honest behaviors). The supervisor will compare the sample data with returned values of P_1, P_2, ..., P_n, if the returned value of P_j is consistent with the sample data, the participant is honest.

   Security and efficiency analysis.

1. The two deformations of double check improve the security of the scheme. Because A does not know another participant B, he will compute honestly. Otherwise, the supervisor will know he is a dishonest participant.

2. Because the samples of double check are only a small part of X, the deformation 2 which is relative to the former has a very good improvement in efficiency.

3. Both double check and two deformations require higher honest proportion p, otherwise progress of completing the task is too slow, the price is too big. Because the supervisor needs to repeat each computing task of data distribution, computing resources will be wasted.

4. The two deformation schemes in security have improved, but at the same time caused the stagnation on time.

4.2 The Supervisor Has Computing Power

The supervisor has a certain ability to compute $f(x)$, but this kind of ability is limited. Because of the large workload, the supervisor cannot independently complete all the verification.

[18] puts forward a kind of uncheatable grid computing based on binary tree. When the supervisor uses m samples, the participant which honesty ratio is r can deceive the chance of success

$$Pr(cheating - succeeds) = [r + (1 - r)q]^m,$$

for $x \in D - D^{'}$, q is the probability that the participant correctly guesses the value of $f(x)$.

Combining both cases of $x \in D^{'}$ and $x \in D - D^{'}$, for one sample x, the probability that the participant can prove its honesty on sample x is $(r + (1 - r)q)$. Therefore, the probability that the participant can prove its honesty on all m samples is $(r + (1 - r)q)^m$.

To keep the probability of successful cheating below a small threshold ε, the sample size m should be

$$m \geq \frac{\log \varepsilon}{\log(r + (1 - r)q)}.$$

But this detection method can only detect whether the participant computes correctly the $f(x)$, $\forall x \in D$. The participant will still be able to deceive the screener function. So anti-cheat binary tree method can only detect semi-honest cheater.

5 THE HIDDEN CRITERION

The supervisor wants to search for valuable data from a large database. He can adopt the method of grid computing and ask participants to help to search. The supervisor has the criterion Y, but he cannot disclose it to participants. Otherwise the participant will obtain valuable data. From thousands of drugs, for example, the supervisor wants to find a formulation for the treatment of certain disease effectively, but does not want to let participant know effective standard of the formulation.

In order to avoid leaking the criterion Y, the supervisor constructs a new criterion to replace Y. So the supervisor can hide Y and the valuable data $x \in \{x \mid f(x) = Y, x \in D\}$. In general, there are two kinds of typical processing method for criterion Y.

5.1 Criterion Expansion

Golle and Mironov put forward the basic ringer scheme [15]. The basic ringer scheme hides the criterion by the method of criterion-expansion method. Let the supervisor's criterion be $Y = \{y\}$, we are looking for valuable x which satisfies $f(x) = y$, $x \in D$. The supervisor and participant will perform the following steps.

1. The supervisor chooses for participant P_i uniformly independently at random n values x_1^i, x_2^i, ..., x_n^i in D_i, and also computes the corresponding images: $y_j^i = f(x_j^i)$.
2. The screener S_i is defined as follows. On input $(k, f(k))$, test whether $f(k)$ belongs to the set $\{y, y_1^i, y_2^i, \ldots, y_n^i\}$. If so output the string k, otherwise output the empty string.
3. The secret key K_i is the set x_1^i, x_2^i, ..., x_n^i, which we call the set of ringers.
4. The supervisor checks that s_i contains all the ringers in K_i plus possibly x such that $f(x) = y$. If so, the participant is honest.

Security Analysis.

1. If the criterion y included in the set $\{y, y_1^i, y_2^i, \ldots, y_n^i\}$ is not hidden, participant P_i can relatively easy to determine y.
2. Only for f is a one-way function. If f is not a one-way function, the participant can solve the preimages set of x_1^i, x_2^i, ..., x_n^i directly from the set of $\{y, y_1^i, y_2^i, \ldots, y_n^i\}$ and does not need to compute tasks in each function value $f(x)$. But the supervisor cannot find any cheating behavior.
3. Participant P_i knows the number of the elements in the ringer set. If participant P_i found all elements in the ring set, he needs not to do the rest of the calculation.
4. This scheme cannot prevent collusion. By comparing $\{y, y_1^i, y_2^i, \ldots, y_n^i\}$ with $\{y, y_1^j, y_2^j, \ldots, y_n^j\}$, P_i and P_j can gain the criterion y and its corresponding valuable x.

After the basic ringer scheme, Golle and Mironov put forward two improvement schemes: Bogus ringer scheme and Hybrid scheme.

5.2 Criterion Reduction

Assuming the supervisor has computing task $f(x), x \in X$ and criterion $Y = \{y \mid y = f(x)$, and satisfies the constraints of $y_1, y_2, \ldots, y_t\}$.

1. The supervisor partitions all participants into n parts ($n < t$, n is safety parameter). Each part has the ability to complete computing $f(x)$, $x \in X$ alone.
2. The supervisor distributes computing task $f(x)$, $x \in X$ and criterion $\{y_i\}$ to part i. Part i eventually returns the set S_i, for $\forall x \in S_i$, $y_i = f(x)$.
3. The supervisor gets the set $S = S_1 \cap S_2 \cap \ldots \cap S_n$. He can verify each participant's behavior.
4. The supervisor's criterion is $y' = \{y \mid f(x) = y, x \in S$ and satisfies the constraints of $y_{n+1}, y_{n+1}, \ldots, y_t\}$.

In the process of allocating the task by supervisor, just sent t constrained conditions to n parts, even if all participants are in collusion with each other, they also cannot get to judge criterion Y.

Because of $|S| \ll |X|$, the supervisor can easily gain the valuable data.

6 SEARCHING FOR HIGH-VALUE RARE EVENTS

Du and Goodrich [1] introduce the searching for high-value rare events scheme.

They propose a more practical set of grid computations – *data filtering* problems. In data filtering problems they are given a large set X of data instances and a Boolean filtering function f. The supervisor is interested in all the elements x of X such that $f(x) = 1$. Usually, the function f will involve some internal scoring function on each input x along with a threshold value such that if x scores above this value, then x is considered rare and interesting. This class of problems includes the SETI@home application, where X consists of extraterrestrial signals that are scored against what are considered to be patterns of intelligence.

By their very nature, it is not obvious which of the inputs in X will score positive for the filter f (for otherwise there would be no motivation for them to go to the trouble of using a grid computing environment to solve this problem). For example, a casual examination of the signals that have scored highest so far in the SETI@home scoring function does not yield any obvious patterns; to the naked eye they all appear as noise. Thus, for data filtering applications such as this, employing an input chaff injection scheme is easy.

To inject input chaff into the set of tasks, the supervisor needs only to have a set of instances Y such that determining if any member y_i is not in X is at least as difficult as computing $f(y_i)$. (The supervisor may not need to explicitly construct Y if he/she has a way of choosing elements from Y probabilistically.) Then the supervisor can randomly inject members of Y into the task sets $D \subset X$ for each participant (with some probability p) to provably obfuscate the rare events. For example, a true input x in the SETI@home application could be transformed into chaff simply by adding a pattern of intelligence to it.

1. The supervisor randomly selects m inputs $x_1, \ldots, x_m$ from the input domain X. Note that X is the global input domain, each participant only conducts tasks for a subset of X.
2. The supervisor generates m chaff by computing $c_i = \text{hash}(f(x_i))$, for $i = 1, \ldots, m$.
3. The supervisor sends the list $C = \{\text{hash}(y), c_1, \ldots, c_m\}$ to all the participants. C should be permuted to hide hash(y).
4. For any input x assigned to each participant, the participant computes hash$(f(x))$ and compares the results with the list C. If a match occurs, the participant sends x back to the supervisor; otherwise x is discarded.

5. The supervisor can verify whether a returned x value is an actual rare event or chaff, by a simple lookup in C (say, by storing the elements of C in a hash table). The supervisor also checks whether the participant whose tasks include chaff has returned the chaff or not. This way, the cheater can be caught.

This scheme cannot resist collusion of participants in a criterion expansion type and cannot detect the semi-honest cheater in a detection scheme. Here we introduce two schemes in [1].

6.1 Scheme I (Criterion Expansion)

1. The supervisor randomly selects n inputs $x_1, \ldots, x_n$ from the input domain X. Note that X is the global input domain, each participant only conducts tasks for a subset X.
2. The supervisor generates n ringers by computing

$$c_i = \text{hash}(f(x_i)),$$

for $i = 1, \ldots, n$.

3. The supervisor sends the list

$$C = \{\text{hash}(y), c_1, \ldots, c_n\}$$

to all the participants. C should be permuted to hide hash(y).

4. For any input x assigned to each participant, the participant computes hash$(f(x))$ and compares the results with the list C. If a match occurs, the participant sends x back to the supervisor, otherwise x is discarded.
5. The supervisor can verify whether a returned x value is an actual rare event or ringer, by a simple lookup in C (say, by storing the elements of C in a hash table). The supervisor also checks whether the participant whose tasks include ringer has returned the ringer or not. This way, the cheater can be caught.

If f is a one-to-one function, this scheme has a good security. Even if all participants in the scheme are in collusion with each other, we can also hide criterion Y and valuable data x and $f(x)$. First of all, every participant gets the same set

$$C = \{\text{hash}(y), c_1, c_2, \ldots, c_n\},$$

even if participants are in collusion with each other, they also cannot get more information from C. Next, when f is the one-to-one function, the preimages of

$$\{f(x_1), f(x_2), \ldots, f(x_n), y\}$$

have only one, respectively, $x_1, x_2, \ldots, x_n, x$. The participant is unable to distinguish between valuable x and pseudo valuable data $x_1, x_2, \ldots, x_n$. But when all

participants are in collusion, they can know whether the valuable data exists. We can improve the security of this scheme. In step 3, change the original criterion to

$$C = \{\text{hash}(y), c_1, c_2, \ldots, c_n, c_1^{'}, c_2^{'}, \ldots, c_t^{'}\},$$

in which, $c_1^{'}, c_2^{'}, \ldots, c_t^{'}$ are some random numbers that supervisor adds. If the valuable data exists, hash(y) and $c_1, c_2, \ldots, c_t$ will have only one preimage, respectively; if the rare valuable data does not exist, hash(y) and $c_1^{'}, c_2^{'}, \ldots, c_t^{'}$ will have not any preimage. As long as n and t are privately owned by the supervisor, every participant cannot determine whether the valuable data exists.

When f is a more-to-one function, if all participants are in collusion with each other, participants can find the hash(y) based on the number of preimages of c_i and hash(y). Usually, every c_i has a lot of preimages in X. However, the preimages of hash(y) are rare. Participants will find criterion hash(y) and valuable data. This scheme cannot resist the second and third deceive models.

6.2 Scheme II (Criterion Reduction)

1. The supervisor computes $h(y)$, and let $\hat{y}$ be the first k bits of the result, where k is a security parameter. The supervisor sends $\hat{y}$ to participants along with the task assignments.
2. For each assigned input x, a participant computes $f(x)$, and checks whether the first k bits of $h(f(x))$ equal $\hat{y}$. If true, the participant returns x and $h(f(x))$ to the supervisor; otherwise, discards x.
3. The supervisor verifies whether $h(f(x)) = h(y)$. If false, x is just ringer; else, x is a rare event.

Apparently, in the single compression method there exist larger defects. In the b) of scheme 2, participant P_i may only compute the part of the data in D_i, but the supervisor cannot detect this kind of deception. It can easily cause the loss of rare event.

By the above analysis, when the task function f is one-to-one function in the scheme I, it has a good security and can resist the three deceptive models. But when f is a more-to-one function, participants will be able to find the rare event that the supervisor is looking for. So the scheme I cannot resist the second and third deception models. The scheme II lacks the necessary honesty test steps, and even cannot determine whether participant do the honest computation for all the data in the task. Thus, it also has a lot of defects on security.

6.3 Three New Schemes

In this section, we propose three new schemes.

1) New Scheme I.

Firstly, we introduce Monte Carlo simulation [9, 16]. It is a technique that employs random numbers to solve problems in which time plays no substantive role. The technique involves simulating a random experiment a large number, say N, of times and recording the number of times, say C, that an event of interest occurs. The law of large numbers asserts that if N is large, the ratio C/N should be a good point estimate of the probability of the event occurring.

As a simple example, consider the problem of finding the area of a region S contained in the square $U \equiv [0,1] \times [0,1]$ in the xy-plane. Using Monte Carlo simulation, one can choose N points from a uniform distribution in U, and count the number of points, C, that lie in S. The approximation for the area would then be C/N.

This example is not well suited for a large scale distributed computation, but serves as an illustration of how the seeding technique can be applied to Monte Carlo simulations in general. The supervisor chooses a particular implementation for the random number generator (ensuring portability) and some number k of seeds. Before any tasks are assigned, an initial run of N/k replications is computed using one of the seeds s' chosen arbitrarily. This seed becomes the ringer for the remaining task assignments. Participants are then sent the code for the generator along with k seeds (including s'), and are instructed to run N/k replications with each of the seeds, returning the area estimate corresponding to each seed. An adversary cannot determine which of the k seeds is the ringer, and therefore cannot return results for fewer than k seeds without raising suspicion. The returned results can be checked for validity using the initial run generated with s'. In effect, the supervisor has managed to provide a measure of assurance while performing only $1/k$ of the work.

By their very nature, Monte Carlo simulations provide a form of redundancy because, provided the number of replications is sufficiently large, each task should return an estimate similar to the other tasks. However, seeding as described here augments the redundancy by enhancing the resistance to collusion.

We can use Monte Carlo simulations to verify whether the participant is honest. This scheme is available for one to one function and more to one function. At the same time, it can resist the semi-honest cheater and hoarding cheater.

Let n be the number of participants in the scheme. P_i $(i = 1, 2, \ldots, n)$ is a participant, and D_i is P_i' task set. Obviously $X = D_1 \cup D_2 \cup \ldots D_n$.

1. The supervisor randomly selects n inputs x_1, x_2, ..., x_n from the input domain X.
2. The supervisor computes $c_i = \text{hash}(f(x_i))$, for $i = 1, \ldots, n$ and $\text{hash}(y)$. Let $\hat{y}$ be the first k bits of $\text{hash}(y)$.
3. The supervisor randomly selects $D_i' \subset D_i$. For each input $x \in D_i'$, the supervisor computes $f(x)$, and checks whether the first k bits of $h(f(x))$

equal $\hat{y}$. If true, the supervisor adds x in $D_i^{''}$. Let p_i be the number $\frac{|D_i^{''}|}{|D_i^{'}|}$, if $p_i < \varepsilon$ (ε is a parameter), we select k again and repeat b). Otherwise, the supervisor can estimate the number of which satisfies that the first k bits of $h(f(x)), x \in D_i$ equal $\hat{y}$ is $p_i|D_i^{'}|$.

4. The supervisor sends $\hat{y}$ and $C = \{c_1, c_2, \ldots, c_n\}$ to participants along with the task assignments.
5. For each assigned input x, participant P_i computes $f(x)$, and checks whether the first k bits of $h(f(x))$ equal $\hat{y}$ or $h(f(x))$ equals C_i. If true, P_i returns x and $h(f(x))$ to the supervisor. Actually, the results which P_i returns are divided into two sets. One is $R_i = \{(x, h(f(x))) \mid x \in D_i$, the first k bits of $h(f(x))$ equal $\hat{y}\}$ and the other is

$$R_i^{'} = \{\{x \in D_i \mid \text{hash}(f(x)) = c_1\}, \ldots, \{x \in D_i \mid \text{hash}(f(x)) = c_n\}\}.$$

6. The supervisor checks whether $h(f(x))$ equals $h(y)$, $x \in R_i$. If true, x is the rare event. The supervisor also checks whether $\frac{|R_i|}{p_i|D_i^{'}|}$ approximately equals 1 and

$$x_i \in \{\{x \in D_i \mid h(If(x)) = c_i\}, \ldots, \{x \in D_i \mid h(f(x)) = c_n\}\}.$$

If true, P_i is a honest participant.

Security Analysis.

Because R_i contains more elements, p_i is unable to determine the real rare event. Hence whether for the one-to-one or more-to-one function, it can well hide the valuable data. Because $R_i^{'}$ must contain all the random inputs in D_i, P_i must compute all $f(x)$ for $x \in D_i$. So this scheme can resist semi-honest cheater and hoarding cheater. And at the same time, the validation of equation $\frac{|R_i|}{p_i|D_i^{'}|} \approx 1$ ensures the small possibility of missing rare events. On average, the feasibility and the security of the scheme are both relatively high.

2) New Scheme II.

Like most of schemes of grid computing, the above scheme is unable to resist malicious cheater. Because the supervisor did not verify the elements in R_i which were returned by participant P_i in order to destroy the supervisor's work, malicious cheater P_i obtains R_i, but he/she does not return the real R_i to the supervisor. So he/she constructs a new set $R_i^{'}$ in which the first k bits of every element equals $\hat{y}$, and $|R_i^{'}| = |R_i|$. In this way, the supervisor cannot find the cheat of P_i and loses some high-value rare events. For example, x_0 is a high-value rare element which is obtained by malicious cheater P_i, i.e., $f(x) = y, x_0 \in D_i$. P_i uses (x_0, A) to replace $(x_0, \text{hash}(x_0))$ (the first k bits of A equal $\hat{y}$ and

$A \neq \text{hash}(x_0)$. According to $A \neq \text{hash}(y)$, the supervisor thinks that x_0 is not a value element, so the high-value element will be lost.

In the following we present a simple improved scheme. Using the sample test of R_i, it can resist the three deception models, and the security is enhanced.

Let n be the number of participants in the scheme. P_i $(i = 1, 2, \ldots, n)$ is a participant, and D_i is the task of set of P_i, and

$$X = D_1 \cup D_2 \cup \ldots \cup D_n.$$

1. The supervisor randomly selects n inputs $x_1, x_2, \ldots, x_n$ from the input domain X.
2. The supervisor computes $c_i = \text{hash}(f(x_i))$, for $i = 1, \ldots, n$ and $\text{hash}(y)$. Let $\hat{y}$ be the first k bits of $\text{hash}(y)$.
3. The supervisor randomly selects $D_i^{'} \subset D_i$. For each input $x \in D_i^{'}$, the supervisor computes $f(x)$, and checks whether the first k bits of $h(f(x))$ equal $\hat{y}$. If true, the supervisor adds x in $D_i^{''}$. Let p_i be the number $\frac{|D_i^{''}|}{|D_i^{'}|}$, if $p_i < \varepsilon$ (ε is a parameter), we select k again and repeat b). Otherwise, the supervisor can estimate the number of which satisfies that the first k bits of $h(f(x)), x \in D_i$ equal $\hat{y}$ is $p_i|D_i^{'}|$.
4. The supervisor sends $\hat{y}$ and $C = \{c_1, c_2, \ldots, c_n\}$ to participants along with the task assignments.
5. For each assigned input x, participant P_i computes $f(x)$, and checks whether the first k bits of $h(f(x))$ equal $\hat{y}$ or $h(f(x))$ equals C_i. If true, P_i returns x and $h(f(x))$ to the supervisor. Actually, the results which P_i returns are divided into two sets. One is $R_i = \{(x, h(f(x))) \mid x \in D_i$, the first k bits of $h(f(x))$ equal $\hat{y}\}$ and the other is

$$R_i^{'} = \{\{x \in D_i \mid \text{hash}(f(x)) = c_1\}, \ldots, \{x \in D_i \mid \text{hash}(f(x)) = c_n\}\}.$$

6. The supervisor checks whether $h(f(x))$ equals $h(y)$, $x \in R_i$. If true, x is the rare event. The supervisor also checks whether $\frac{|R_i|}{p_i|D_i^{'}|}$ approximately equals 1 and

$$x_i \in \{\{x \in D_i \mid h(f(x)) = c_i\}, \ldots, \{x \in D_i \mid h(f(x)) = c_n\}\}.$$

   If true, P_i is a honest participant.
   If true, the supervisor will carry out sample test.
7. The supervisor randomly chooses m elements from R_i. Let

$$S = \{(s_1, A_1), (s_2, A_2), \ldots, (s_m, A_m)\}$$

be the set which is constructed by m elements and their computing values. The supervisor verifies whether $h(f(s_i))$ equals A_i for $i = 1, \ldots, m$. If true, P_i is honest. So the supervisor checks whether $f(f(x))$ equals $h(y)$, $x \in R_i$. If true, x is the rare event.

Theorem 1. When s samples are sampled by the supervisor in this scheme, the probability that a participant with honesty return ratio h can cheat successfully is

$$Pr(\text{cheating succeeds}) = h^s.$$

Proof. Since the sample number is very big, it is an independent and identically distributed probability. When the probability of one correct sample is h, the probability of s correct samples is h^s. □

Therefore, the probability that the supervisor finds the cheat of a malicious cheater P_i is $1 - h^s$.

To keep the probability of unsuccessful cheating above a big threshold ε, the sample size s should be

$$s > \frac{\log(1-\varepsilon)}{\log h}.$$

Table 1 shows how large s should be for different honesty return ratio h, given $\varepsilon = 0.9$ or 0.99. In fact, if $s > 500$, the probability that the supervisor finds the cheat of malicious cheater P_i is greater than 0.99.

h	$t = 0.9$	$t = 0.99$
0.5	4	7
0.8	11	21
0.9	22	44
0.99	230	459

Table 1.

3) New Scheme III.

In most of cheating detection schemes, the computation task X had not been dealt with. In order to achieve the purpose of hiding Y, they only used new criterion Y' to replace Y.

We present a new method. Let Y be $\{y\}$, we can construct a new set C, for $\forall x \in C$, $f(x) = y$. In this way, we expand the task to $X \cup C$.

The new scheme is described in the following:

1. The supervisor constructs a new collection C, for $\forall x \in C$, $f(x) = y$, and $C = C_1 \cup C_2 \cup \ldots \cup C_n$.

2. The supervisor computes hash(y), and let $\hat{y}$ be the first k bits, where k is a safety parameter. The supervisor sends $\hat{y}$ to participant P_i along with the task $D_i \cup C_i$, $i = 1, \ldots, n$.
3. For each assigned input x, participant P_i computes $f(x)$, and checks whether the first k bits of $h(f(x))$ equal $\hat{y}$. If true, P_i returns x and $h(f(x))$ to the supervisor.

Let $R_i = \{(x, h(f(x))) \mid x \in D_i \cup C_i$, the first k bits of $h(f(x))$ equal $\hat{y}\}$.

- The supervisor checks whether C_i is contained in R_i. If true, P_i is honest.
- The supervisor checks whether $h(f(x))$ equals $h(y)$. If true, x is the rare event.

If C_i is contained in R_i, P_i is honest. So P_i must compute every $f(x)$, $x \in D_i \cup C_i$. If not, the supervisor will find his dishonesty. Due to a participant cannot distinguish the data from D_i and C_i, the rare events can be disguised. In conclusion, the scheme can resist all three cheat modes and a collusion attack.

7 CONCLUSION

In this article, according to the computing power of the supervisor, we propose the deceptive detection schemes under two different circumstances, and combine the characteristics of the task function f to analyze the security of the deceptive detection. Based on the technology of double check, we proposed an improved scheme at the sacrifice of time, i.e. the secondary allocation scheme of double check. We reinforced the security of double check greatly. Finally, we analyzed the common problem of High-Value Rare Events, improved the deceptive detection scheme due to Du and Goodrich [1], and then put forward a new deceptive detection scheme with better security and efficiency.

REFERENCES

[1] Du, W.—Goodrich, M. T.: Searching for High-Value Rare Events with Uncheatable Grid Computing. Proceedings of the 3rd Applied Cryptography and Network Security Conference (ACNS), New York, NY, USA, June 2005, pp. 122–137, doi: 10.1007/11496137_9.

[2] Yu, J.—Wang, X.: Deceptive Detection and Security Reinforcement in Grid Computing. 2013 5th International Conference on Intelligent Networking and Collaborative Systems, 2013, pp. 146–152, doi: 10.1109/INCoS.2013.30.

[3] Climate Prediction Web Site. Available at: http://www.climateprediction.net.

[4] Distributed.net Web Site. Available at: http://www.distributed.net.

[5] IBM Grid Computing. Available at: http://www-1.ibm.com/grid/aboutgrid/what-is.shtml.

[6] SETI@Home. Available at: `http://setiathome.berkeley.edu`.

[7] The Smallpox Research Grid. Available at: `http://www-3.ibm.com/solutions/lifesciences/research/smallpox`.

[8] The Great Internet Mersenne Prime Search. Available at: `http://www.mersenne.org/prime.htm`.

[9] Search for Extraterrestrials' or Extra Cash. Available at: `http://www.dallasnews.com/technology/1202ptech9pcs.htm`.

[10] Law, A. M.—Kelton, W. D.: Simulation Modeling and Analysis. McGraw-Hill, 3rd edition, 2000.

[11] Aiello, W.—Bhatt, S.—Ostrovsky, R.—Rajagopalan, S. R.: Fast Verification of Any Remote Procedure Call: Short Witness-Indistinguishable One-Round Proofs for NP. In: Montanari, U., Rolim, J. D. P., Welzl, E. (Eds.): Automata, Languages and Programming (ICALP 2000). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1853, 2000, pp. 463–474, doi: 10.1007/3-540-45022-x_39.

[12] Anderson, D. P.: BOINC: A System for Public-Resource Computing and Storage. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, Washington, DC, USA, November 2004, pp. 4–10, doi: 10.1109/grid.2004.14.

[13] Beigel, R.—Margulis, G.—Spielman, D. A.: Fault Diagnosis in a Small Constant Number of Parallel Testing Rounds. Proceedings of the Fifth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA '93), 1993, pp. 21–29, doi: 10.1145/165231.165234.

[14] Brin, S.—Page, L.: The Anatomy of Large-Scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems, Vol. 30, 1998, No. 1–7, pp. 107–117, doi: 10.1016/s0169-7552(98)00110-x.

[15] Golle, P.—Mironov, I.: Uncheatable Distributed Computations. In: Naccache, D. (Ed.): Topics in Cryptology (CT-RSA 2001). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2020, 2001, pp. 425–440, doi: 10.1007/3-540-45353-9_31.

[16] Golle, P.—Stubblebine, S. G.: Secure Distributed Computing in a Commercial Environment. Proceedings of the 5th International Conference on Financial Cryptography, British West Indies, February 2001, pp. 289–304, doi: 10.1007/3-540-46088-8_23.

[17] Szajda, D.—Lawson, B.—Owen, J.: Hardening Functions for Large Scale Distributed Computations. Proceedings 2003 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, May 2003, pp. 216–224, doi: 10.1109/SECPRI.2003.1199338.

[18] Du, W.—Jia, J.—Mangal, M.—Murugesan, M.: Uncheatable Grid Computing. Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS), Hachioji, Tokyo, Japan, March 2004, pp. 4–11, doi: 10.1109/icdcs.2004.1281562.

[19] Minsky, Y.—van Renesse, R.—Schneider, F. B.—Stoller, S. D.: Cryptographic Support for Fault-Tolerant Distributed Computing. Proceedings of the Seventh ACM SIGOPS European Workshop: System Support for Worldwide Applications (EW 7), Connemara, Ireland, September 1996, pp. 109–114, doi: 10.1145/504450.504472.

[20] Kahney, L.: Cheaters Bow to Peer Pressure. Wired Magazine, February 15, 2001. Available at: https://www.wired.com/2001/02/cheaters-bow-to-peer-pressure/.

[21] Kuhn, M.—Schmid, S.—Wattenhofer, R.: Distributed Asymmetric Verification in Computational Grids. Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing, Miami, Florida USA, April 2008, pp. 1–10, doi: 10.1109/ipdps.2008.4536244.

Jianhua Yu is Ph.D. student at the South China Normal University. He is Lecturer at the School of Mathematical Sciences, South China Normal University. His research interests include issues related to cryptography, computer network, and mathematical modeling. He is author of a great deal of research studies published in national and international journals and conference proceedings.

Yuan Li received his Ph.D. in number theory at the State University of New York. He is Associate Professor at the Department of Mathematics, Winston-Salem State University. His research interests include computational number theory and cryptography.

Computing and Informatics, Vol. 37, 2018, 1313–1338, doi: 10.4149/cai_2018_6_1313

FINGERPRINT BASED DUPLICATE DETECTION IN STREAMED DATA

Amritpal Singh, Shalini Batra

Department of Computer Science and Engineering
Thapar University
Patiala, Punjab, India
e-mail: amritpal.singh203@gmail.com, sbatra@thapar.edu

Abstract. In computing, duplicate data detection refers to identifying duplicate copies of repeating data. Identifying duplicate data items in streamed data and eliminating them before storing, is a complex job. This paper proposes a novel data structure for duplicate detection using a variant of stable Bloom filter named as FingerPrint Stable Bloom Filter (FP-SBF). The proposed approach uses counting Bloom filter with fingerprint bits along with an optimization mechanism for duplicate detection. FP-SBF uses d-left hashing which reduces the computational time and decreases the false positives as well as false negatives. FP-SBF can process unbounded data in single pass, using k hash functions, and successfully differentiate between duplicate and distinct elements in $O(k+1)$ time, independent of the size of incoming data. The performance of FP-SBF has been compared with various Bloom Filters used for stream data duplication detection and it has been theoretically and experimentally proved that the proposed approach efficiently detects the duplicates in streaming data with less memory requirements.

Keywords: Duplicate detection, stable Bloom filter, d-left hashing, FingerPrint bits, streaming data

1 INTRODUCTION

With the exponential increase in data generation resources, paradigm of analytics has changed from static to dynamic processing especially in data mining applications. Advancements in application areas of IoT [4] and cloud computing [3] have shifted the focus of big data analytics towards streaming data analytics. In appli-

cations like network monitoring, sensor networks, data management in web applications, etc. [11], data is generated in various forms which include IP of system in network applications, URL of web page visited, sensor readings, unique user name used by user in social networking sites, etc. [26].

On-line monitoring of data streams and eliminating duplicates is an important issue in stream analytics. For redundant data elimination, primary focus should be on differentiating between duplicate and distinct element in the data stream. Detecting duplicates in streams becomes more difficult because of unbounded nature of data coming at high rate and necessity of processing data in one pass by using limited amount of memory [7].

Conventional databases and traditional data mining techniques are efficient for stored data analytics but for in-streamed data, where data is ariving continously, it is not feasible to store the data into a database and then perform mining operations since all such applications demand time bound query output.

In applications where efficiency is more important than accuracy, use of probabilistic approaches and approximation algorithms can serve as a key ingredient in data processing. Probabilistic methodologies provide quick answers with an allowable error rate compared to deterministic approaches that give exact matches, and which are slow and memory consuming [20].

1.1 Standard Bloom Filter

Bloom Filter (BF) [8], a space efficient probabilistic data structure, is used to represent a set S of n elements from a universe U. It consists of an array of m bits, denoted by $BF[1, 2, \ldots, m]$, initially all set to 0. To describe the elements in the set, the filter uses k independent hash functions $h_1, h_2, \ldots, h_k$ with their value ranging between 1 to m; assuming that these hash functions independently map each element in the universe to a random number uniformly over the defined range. For each element $x \in S$; the bits $BF[h_i(x)]$ are set to 1 s.t. $1 < i < k$. Given an item y, its membership is checked by examining the BF whether the bits at positions $h_1(y)$, $h_2(y)$, $\ldots$, $h_k(y)$ are set to 1. If all $h_i(y)$ $(1 < i < k)$ are set to 1, then y is considered to be part of S, otherwise y is definitely not a member of S. The accuracy of a Bloom filter, (f_p) depends on the filter size m, the number of hash functions k, and the number of elements n. User can predefine false positive error according to application's requirement.

$$f_p = (1 - e^{-kn/m})^k. \tag{1}$$

1.2 Application Domains

Some real time problems related to different domains are explored in this section, where duplicate detection is required to perform analysis on the streaming data.

Duplicate item identification is a common problem faced by social networking sites like Twitter, Facebook, Instagram, etc. where multiple copies of same event

(tweet or post) are generated for same input by multiple users, which keep on appearing continuously from different sources on the user's screen. In such scenarios, the primary need is to identify duplicate events and group them together to improve the user's experience [21]. Another important event related to duplicate detection is URL crawling. Search engines regularly crawl the Web to enlarge their collections of Web pages. While scanning a URL, search engine's task is to identify the new web pages and add them to its repository. So, the basic task in this scenario is comparing each scanned URL with all existing URLs in its database to identify duplicate URLs [18]. In network monitoring applications, selecting distinct IP addresses is a task associated with duplicate detection. In networks, a particular server is checked for unique hits. This analytics facilitates in understanding the pattern of traffic which helps in efficient allocation of network resources [13]. Web advertising is easy and most effective way to publicize a product where advertisers pay web site publishers for number of clicks on their advertisements. Fake clicks may be generated (by using scripts) to increase the profit of the publisher. To detect the duplicate users in clicks is thus associated with duplicate detection task [22].

We propose a novel duplicate detection technique using a variant of stable Bloom filter [13] named as FingerPrint Stable Bloom Filter (FP-SBF). The proposed approach uses stable Bloom filter with fingerprint bits and optimization mechanism which reduces the computational time and decreases the false positives as well as false negatives using d-left hashing. Optimized deletion mechanism is used to evict the data from Bloom filter to make more space for incoming data. It uses constant space irrespective of the size of incoming data and accommodates more number of elements before reaching the saturation stage. Based on above mentioned improvements in existing approach, the proposed Bloom filter outperforms the stable Bloom filter in detecting the duplicates in streaming data, both theoretically and experimentally.

The plan of this paper is as follows: Section 2 provides literature study on Bloom filter and its variants. Section 3 states the duplicate detection problem and challenges associated with it. In Section 4, proposed approach is discussed in detail. Section 5 provides experimental results on both real datasets and synthetic data, comparing existing approaches with the proposed approach. Section 6 discuses use case scenarios for proposed approach. Finally, Section 7 concludes the paper with possible future extensions in this area.

2 RELATED WORK

2.1 Duplicate Detection

Duplicate detection is the task of detecting unique entries in unbounded data streams. Duplicate elimination is a crucial intermediate step in data processing and analytics of the incoming streams. The problem of finding approximate duplicate items has been studied in the contexts of data management and Web applications. Data streams are generally unbounded and traditional DBMS approaches of accom-

modating whole stream in the memory leads to very high memory utilization. We need to provide results for the query in minimum time with less computational cost and minimum memory requirements.

There are many solutions for the duplicate detection problem [21] but our main focus is on the solutions based on usage of Bloom filters for efficient detection of duplicate datasets in streaming data. In window based approaches, for every new input arriving from the stream, an old entry is evicted by adjusting the size of window. Recent work done in window based framework for duplicate detection using Bloom filter is by Metwally et al. [22]. In their work three type of window based approaches are defined: landmark window, sliding window and jumping window. In landmark window approach, incoming data is processed as disjoint portions of the stream, which are separated by landmarks. Landmarks can be defined either in terms of time, e.g. on a daily or weekly basis, or in terms of the number of elements observed. In landmark window approach, whole data from Bloom filter is deleted upon reaching the new landmark. Since individual element deletion is not performed in this approach, a simple Bloom filter is used. In sliding window approach size of window is not fixed, it grows as the incoming data from stream grows. The element deletion is also not allowed in this approach so simple Bloom filter is used in implementation. The full size of window is maintained to query old data. Querying process in sliding window has more computational cost as compared to landmark window. Landmark window requires less space as compared to sliding window but there are chances of some missed values in case of landmark window because upon reaching new landmark previous data is deleted. The jumping window is based on idea of dividing the individual element window into smaller sub windows and usage of counting Bloom filter to accommodate more number of elements. The latest sub window active for insertion is referred as jumping window or current window. As the latest sub-window crosses its threshold, jumping window is moved to next sub-window and oldest sub-window is deleted. So these window based approaches either need dynamic size that leads to extra computation and expensive query process or they remove a large chunk of data together which may lead to the higher false negatives [27].

Another solution is use of buffering and caching methods, used in many database, network applications, URL caching and web crawling. Some Bloom based buffering techniques include double buffering and A^2 buffering. In double buffering [10] two different buffers of size $\frac{m}{2}$ are maintained. As one filter crosses its threshold, insertion starts in second and when second filter is full, i.e., this filter reaches its threshold, then whole data from the first one is drained and insertion is again started in the first filter. For query process, first active filter, i.e., current filter in which insertion is being done recently, is checked and if query returns false then previous filter is scanned. In A^2 buffering [28] similar approach is used, in this, when active filter reaches its 50 % capacity, the insertion is started in both filters, and when active filter is full, its data is evicted by resetting it to zero. In buffering approaches element wise deletion is not allowed so a simple Bloom filter is used. Buffering approaches store data for short time with large redundancy. Handling data streams

with buffering leads to wastage of memory; further it is computationally costlier to check the threshold of Bloom filter at every step.

Recently variant of Bloom filter for duplicate detection in stream named as Stable Bloom Filter (SBF) has been proposed by Deng and Rafiei [13]. Counting Bloom filter with c bits in counter is used as the base of its implementation. Before inserting any element in the stream, it is checked that whether the incoming element is queried earlier in the stream or not. If query returns TRUE, i.e., values of counters corresponding to hash indexes are high, then element is marked as duplicate and element is not inserted, but if query returns FALSE indicating that the element is observed for the first time in the stream, element is added. Before insertion begins, SBF makes space for incoming elements by randomly decrementing the values of counter by one in each iteration. In insertion process, k hash functions are computed and correspondingly hash value counters are set to Max, where $Max = 2^c - 1$. The main advantage of this approach is that it provides query results on streaming data in $O(k)$, independent of the size of incoming data, using constant memory.

Another technique proposed for duplicate detection is Reservoir Sampling based Bloom Filter (RSBF), a hybrid of reservoir sampling and Bloom filter [14]. It uses k Bloom filters, each of size s bits. Insertion of an element is performed after query process. One hash function of each Bloom filter is reserved. If hash index associated with each Bloom filter is found HIGH then element is considered as duplicate and insertion is not performed. Else, insertion is performed by setting corresponding k bits HIGH in k Bloom filters. To accommodate streaming data in fixed size Bloom filters, random deletion is done by resetting k randomly selected bits to LOW. RSBF uses same amount of memory as SBF, has fast convergence rate and shows significant improvement in false negatives but it does not provide a significant improvement in false positives.

Streaming Quotient Filter (SQF) is another duplicate detection algorithm based on probabilistic data structures [15]. In proposed model, Dutta et al. use Quotient filter for approximate membership query instead of Bloom filter. SQF maintains a hash table and bucket associated with each element in hash table. Hashing is done at two levels: first element is mapped to hash table and then corresponding to hash table entry, hashing is done in bucket by using quotienting technique. SQF performs efficiently in detecting false positives and false negatives but the preprocessing time for converting data into binary form for quotienting and hashing at two levels increases the computational time many folds. Another drawback associated with SQF is that because of clustering, operations like insertion and searching are difficult to perform in parallel.

2.2 Counting Bloom Filter (CBF) and Its Variants

Counting Bloom filter, introduced by Fan et al. [17], uses a counter of c bits where range of counter is $\{1, 2^c - 1\}$. Based upon the hash indexes computed, i.e. $\overset{k}{_{i=1}} h_i(x) = \overset{k}{_{i=1}} H_i(x)$, insertion and deletion operations are performed on the counters.

Whenever an element is added or deleted from the CBF, the corresponding counters are incremented or decremented, respectively. CBF is primarily used to answer frequency queries. Although it can be efficiently used for applications where deletion operation is required, the memory overhead for CBF as compared to standard Bloom filter is significantly large and determining value of counter is a difficult process. Number of variants of CBF have been proposed to overcome such issues and maximize the storage of counting Bloom filter with minimum memory requirements and less false positive and false negative rates.

Some important variants of CBF are discussed in this section. In Spectral Bloom filter [12] value of smallest counter is increased when a new element is inserted and query process returns minimum count for an element from the counter selected. It is mainly used for storing the multi sets data and supports frequency query. Another improvement in CBF was done by improving hashing technique and called d-left counting Bloom filter [9], in which CBF with d-left hashing was used to calculate index values of hash functions by dividing Bloom filter into sub tables. Use of this efficient hashing in CBF leads to less number of collisions. It is used for element lookups and fingerprint matching applications. Deletable Bloom filter [24], another variant of CBF, optimizes the process of deletion by using probabilistic approach for element removal. It keeps the record of regions with high collisions, i.e., the regions where probability of deletion is quite high. The main aim of this variant is to minimize false negatives. It finds application in source routing to avoid loops, middle-box services like load balancer, firewalls, etc.

Recent variants of CBF include Variable Increment Counting Bloom Filter (VI-CBF) and FingerPrint based Counting Bloom Filter (FP-CBF). In VI-CBF [25] two different sets of hash functions are maintained, first set of hash functions, i.e. $H_{i=1}^{k}$, decide the indexes on which increment is performed, and another set of hash functions, i.e. $G_{i=1}^{k}$, decide the value from a set $D_L = \{L, L+1, \ldots, 2^L - 1\}$ by which increment is performed on the selected indexes. Same process is followed in deletion operation. It decreases the false positives by a huge factor as compared to standard CBF. Some limitations of VI-CBF are that implementation is more complex, calculations for two hash functions lead to extra computational cost and more bits are required to avoid overflow of counters.

Pontarekki et al. proposed FP-CBF [23], where each counter has some extra bits denoted as fingerprint bits. The main idea behind the use of these bits is to provide a second level check in querying operation and reduce false positive and false negatives. Here each index of d bits is divided into counter bits (c) and fingerprint bits ($f = d - c$). In insertion process, indexes for update are computed (based on the k hash functions used) and corresponding to these indexes all counters are incremented by one. A separate hash function H_{fp} is used to update the fingerprint cell of all selected indexes with XOR operation. In querying process, values of counters are checked and if all values are high then second level check is performed by matching H_{fp} value of queried element with fingerprint cells. If values of H_{fp} in all fingerprint cells match, query returns TRUE value. FP-CBF provides more accurate results with minimum computational effort.

2.3 d-Left Hashing

Hashing is another important aspect to determine the accuracy of a Bloom filter. Variants of CBF use advance hashing techniques like double hashing [1], multiple hashing [6], perfect hashing scheme [5], etc. Recently cuckoo hashing [16] has been used in CBF which has shown drastic decrease in error rates. Major concern in cuckoo hashing is that computational cost associated with swapping of elements is quite high and pre-processing cost in insertion of each element is also quite high. Another important hashing, d-left hashing refereed as perfect hashing technique, can be used to reduce the collisions by great extent. Further, it is easy to implement and requires less computational and pre-processing cost.

d-left hashing is a minimal perfect hashing approach, where a hash table of size M is divided into d sub tables of size m_d, where $m_d = \frac{M}{d}$. Each element in sub table is referred as bucket B_i and b denotes the maximum capacity of bucket. During insertion operation, k hash functions are used to compute the hash indexes and select the buckets. New item is placed in the bucket which is least loaded. If there is tie between two buckets then bucket on left side is selected to perform insertion operation.

For an element $u_i \in U$, hash functions are calculated as:

$$\forall_{i=1}^{k} \left(\kappa_i(u_i) = \ell_1(u_i) + i \times \ell_2(u_i) \ mod \ m\right). \tag{2}$$

Each hash function $\kappa_i(.)$ selects i^{th} bucket from a sub array. Double hashing is used for hash function generation, where two independent hash functions $\ell_1(x)$ and $\ell_2(x)$ are used to generate k hash functions such that $\forall i | i < k$. To get best results of this scheme, m should be prime, so size of array and number of buckets should be choosen in such a way that $\frac{m}{d}$ returns a prime number. This technique leads to less inter-hash function collision; further, usage of only two hash functions to generate all k hash functions decreases the computational overhead. Theoretical and experimental evidences prove that it helps in reduction of collision as a huge factor [9, 19].

3 PROBLEM STATEMENT

The main emphasis of the paper is to detect duplicate data sets in the data stream. The problem can be summarized as: Given an input stream $S = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$ with N elements; where $N \to \infty$, i.e. unbounded stream of data, identify whether x_i appears in S or not in a given space M, where $(M << N)$. Challenges associated with duplicate detection include:

1. Preprocessing effort for each element should be minimum for fast results.
2. Eviction of stale data is necessary as the memory size of the filter is fixed.
3. Response time for query should be minimum and independent of the size of data.

4 PROPOSED WORK

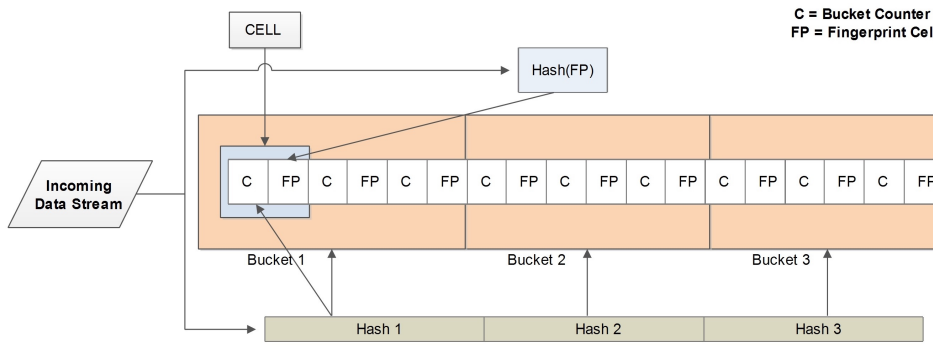


Figure 1. Structure of FP-SBF

To solve the duplicate detection task efficiently using minimum memory some approximation based processing techniques need to be applied on streaming data. A variant of Stable Bloom filter (SBF) named as FingerPrint Stable Bloom filter (FP-SBF) is proposed to detect duplicates in streaming data. FP-SBF is an array of indexing $[1, \ldots, m]$ of size M where each element of array is represented by d bits (i.e. $M = m \times d$). d bits of each element are further divided into two parts: c bits for buckets and f bits for fingerprint as indicated in Figure 5. Bucket bits (c bits) are used as counter, where range of counter values is $\Re_c \leftarrow \{1, \{Max = (2^c - 1)\}\}$ and $f = (d - c)$ bits are fingerprint bits, also called FingerPrint Cell (FPC). FP-SBF uses d-left hashing with $(k + 1)$ hash functions, where k hash functions are used to select the appropriate index from m elements and update bucket's counter to Max and one hash function H_{fp} is used to update fingerprint cell of each selected index. Since available memory space is fixed, old data should be evicted to make room for new data. Parameter ξ called Eviction Rate (ER) is used to control the eviction of stored data in Bloom filter.

The task of detecting duplicate from the streams using FP-SBF consists of following steps (Algorithm 1):

1. **Detection:** Query the existing data to find whether current data element x_i exists in the filter or not.
2. **Deletion:** Remove the data randomly by decrementing the values of buckets to make spaces for new incoming elements.
3. **Insertion:** Insert the data in the filter by updating the corresponding bucket and fingerprint cells, if the element is not present in the array.

Algorithm 1 Duplicate Detection in Streams (DDS) using FP-SBF

```
1: procedure DDS(FP-SBF[], S, H_k, H_fp) ▷ Check whether x_i element is present in the stream or not
2:     for ∀x_i ∈ S do
3:         if Detection(x_i) == TRUE then ▷ Duplicate detected
4:             Element appeared previously in the filter, no insertion required
5:         else
6:             if Random(ξ) > Threshold then ▷ Check for threshold and then insertion is performed
7:                 Delete()
8:             end if
9:             Insert(x_i)
10:        end if
11:    end for
12: end procedure
```

4.1 Detection

To detect an element x_i in FP-SBF, $k+1$ hash functions H_1^k and H_{fp} are used. Corresponding to k hash functions, indexes are generated, i.e. $h_1^k \leftarrow H_1^k$, and following checks are performed (Algorithm 2):

1. If any of the bucket corresponding to h_1^k is set to zero, duplicate detection mechanism returns false, i.e. $x_i \notin (x_1, \ldots, x_{i-1})$.
2. If all buckets are non-zero, $H_{fp}(x_i)$ is computed for x_i and all fingerprint cells of indexes h_1^k are checked. If $H_{fp}(x_i)$ is not found in any FPC, duplicate detection mechanism returns FALSE, i.e. $x_i \in (x_1, \ldots, x_{i-1})$.
3. If step 1. and 2. (mentioned above) are false then x_i is duplicate, i.e., it is previously seen in the stream $(x_1, \ldots, x_{i-1})$, hence no need to perform further operations.

Deletion and insertion operations are invoked when detection process fails, i.e., element is not found in the stored stream. False positives may occur in the detection due to hash function collision of two different elements, i.e., bucket set high by x_i returns detection results as TRUE, resulting in identification of a distinct element as duplicate element.

A false negative in duplicate detection on streamed data occurs when a duplicate element (x_i) is wrongly reported as a distinct element. Eviction process in FP-SBF leads to decrement the value of buckets to zero, i.e., for an element which is present in stream, the value of bucket is decremented to zero and this element is regarded as a distinct element although it is a duplicate element.

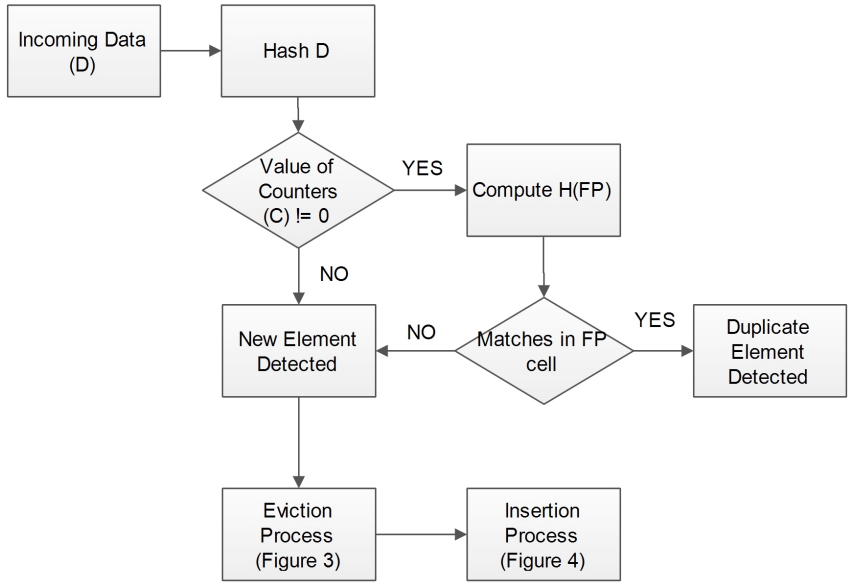


Figure 2. Flowchart of detection process in FP-SBF

4.2 Deletion or Eviction of Data

To accommodate unbound data in constant memory, it is necessary to evict the data at regular intervals. FP-SBF applies an optimized deletion approach which evicts the data quickly and decreases false positives (Algorithm 3).

Algorithm 2 Detection

procedure DETECTION($FP\text{-}SBF[], x_j, H_k, H_{fp}$)
  $\forall i|(1 < i < k)$ Calculate hash $h_{i=1}^{k}(x_j) \leftarrow H_{i=1}^{k}(x_j)$
  if $(\forall i|(1 < i < k)), [h_{i=1}^{k}(x_j) > 0]$ **then**
    $h_{fp}(x_j) \leftarrow H_{fp}(x_j)$
    if $(\forall i|(1 < i < k)), [FPC_i = h_{fp}(x_j)]$ **then**
      Return *TRUE*
    else
      Return *FALSE*
    end if
  else
    Return *FALSE*
  end if
end procedure

k indexes are selected randomly with probability p and decremented by a value z, s.t. $z \in (1, 2, \ldots, 2^c - 1)$. Deletion process is invoked after i iterations where i is selcted randomly. This process is controlled by Evicting Rate (ER) parameter $\xi()$ $\Re_\xi \rightarrow \{0, 1\}$). Steps followed to accomodate incoming data are:

1. In every iteration, check if $Random(\xi)$ returns a value more then defined threshold, where $Random(\xi)$ is a function which generates random values as per the value of ξ provided to it. If $Random(\xi)$ is greater than defined threshold then deletion is performed else steps 2. to 4. are skipped
2. If step 1. is false, i.e., threshold is not reached, select k index, i.e., $h_1^k(D)$ for deletion where probability of any cell being selected is $(\frac{p}{m})$.
3. Select a random value of z for each selected index $h_1^k(D)$ from a given range $\Re_z \rightarrow \{1, 2^c - 1\}$.
4. For each index selected in step 3. decrement the value of bucket by z.

Deletion frequency can be increased by increasing value of ξ. Fingerprint cell of selected indexes remains unaffected in the deletion process.

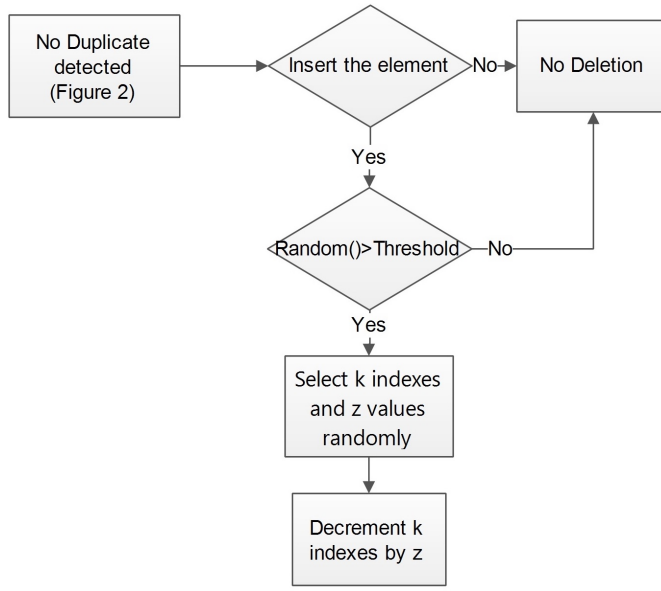


Figure 3. Flowchart of eviction process in FP-SBF

4.3 Insertion

To insert an element x_i, $k + 1$ hash functions are used. Steps followed for insertion of an element are (Algorithm 4):

Algorithm 3 Deletion Process

1: **procedure** DELETE($FP\text{-}SBF[], p_d, \Re_z$)
2: `Select` k `cells from m with probability` p
3: $L_{i=1}^{k}$ are the selected cells for decrement operation.
4: `Select a value to be decremented from each bucket`
5: $z \leftarrow Random(\Re_z)$
6: $(\forall i | (1 < i < k))\ Bucket[L_i] = (Bucket[L_i] - z)$
7: **end procedure**

1. Hashing is done to get the indexes, i.e. $h_1^k(x_i) \leftarrow H_1^k(x_i)$.
2. Buckets are updated:
$$Set(h_1^k(x_i)) = Max$$
where $Max = (2^c - 1)$.
3. Fingerprint cell is updated:
$$h_{fp}(x_i) \leftarrow H_{fp}(x_i).$$
$h_{fp}(x_i)$ is used to update the corresponding FPC of selected indexes. $FPC_i \leftarrow (FPC_i)$ XOR $(h_{fp}(x_i))$.

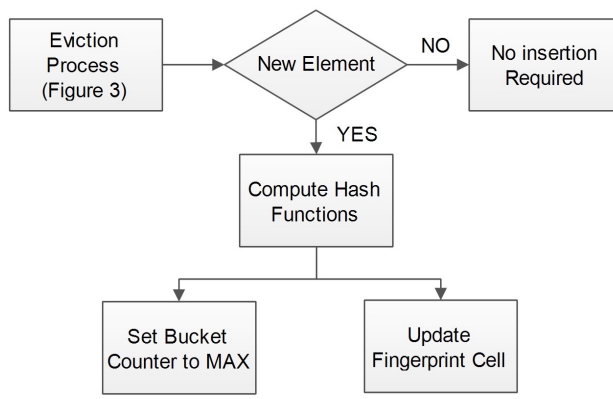


Figure 4. Flowchart of insertion in FP-SBF

4.4 Stable Property (SP)

Stable property assures that after ℓ iterations, the fraction of zeros in SBF/FP-SBF will be fixed, i.e., they will not depend on any parameter. In FP-SBF number of iterations ℓ required to achieve SP is depended on eviction rate ξ. It has been proved

Algorithm 4 Insertion in FP-SBF

1: **procedure** INSERT($FP\text{-}SBF[], x_j, H_k, H_{fp}$)
2: $(\forall i|(1 < i < k))$ Calculate hash $h_{i=1}^{k}(x_j) \leftarrow H_{i=1}^{k}(x_j)$
3: $(\forall i|(1 < i < k))Set(Bucket[h_i] \leftarrow Max)$
4: $h_{fp}(x_i) \leftarrow h_{fp}(x_j)$
5: $(\forall i|(1 < i < k))$ `do`
6: $FPC_i \leftarrow (FPC_i)$ XOR $h_{fp}(x_j)$
7: **end procedure**

theoretically and experimentally that SP plays an important role in determining false positive rate.

Theorem 1. In FP-SBF with m cells, each cell is updated to Max with a probability $p_i = (\frac{k}{m})$ and each cell is decremented by a value $z \in (1, \ldots, 2^c - 1)$ with a probability $p_d = (\frac{p}{m})$. The probability that cells become zero after N iterations as $N \to \infty$ is constant. i.e.

$$\lim_{N\to\infty} \sum_{i=1}^{N} Pr(ASBF_i = 0) \Rightarrow Constant$$

where $ASBF_N$ is value of cell at the end of N iterations.

Proof. For each element of stream, three possible operations are detection, deletion and insertion. Only deletion and insertion will effect the values of buckets. After N operations, a bucket can be set to $Max \leq N$ times and decremented with certain value $< N$ times. Let p_i be probability of a bucket to be selected for insertion given by $p_i = \frac{k}{m}$; p_d be probability of a bucket to be selected for deletion, i.e. $p_d = \frac{p}{m}$, and ξ be Evicting rate for FP-SBF. A_l denotes that no insertion has been performed in the bucket in recent l iterations ($l < N$); probability of event A_l is given by:

$$Pr(A_l) = (1 - p_i)^l p_i. \tag{3}$$

A_N denotes that no insertion is performed in any bucket in N iterations, its probability is:

$$Pr(A_N) = (1 - p_i)^N. \tag{4}$$

Since no insertion operation is performed in a particular bucket for N iterations, probability of having value zero in that bucket following event A_N is:

$$Pr(ASBF_N = 0|A_N) = 1. \tag{5}$$

P_0, probability that after deletion operation value of bucket is zero is:

$$P_0 = \xi \times p_d \times Pr(P_i = 0|(FP - SBF[i] = x)) \tag{6}$$

where value of $Pr(P_i = 0|(FP - SBF[i] = x))$ is dependent on the current value in the bucket and z value selected from Deletion Value Set $(1, 2^c - 1)$ is given by:

$$Pr(P_i = 0|(FP - SBF[i] = x)) = \frac{\Delta[(z \geq x)\&(z \in DVS)]}{\Delta(z \in DVS)}. \tag{7}$$

$\Delta()$ is function which counts the number of values of $z \in DVS$, to calculate the sample space and favorable events for deletion.

When a bucket follows event A_l, the probability that after N iterations where $N > l$ the decrement operation will reset the bucket value to zero is given by:

$$Pr(ASBF_N = 0|A_l) = \sum_{j=Max}^{l} \binom{l}{j} P_0^j (1 - P_0)^{l-j}. \tag{8}$$

Thus, for a random element, probability $Pr(ASBF_N = 0)$ that the bucket is zero after N iterations is given by:

$$\begin{aligned} Pr(ASBF_N = 0) = &\sum_{l=Max}^{N-1} [Pr(ASBF_N = 0|A_l) Pr(A_l)] \\ &+ Pr(ASBF_N = 0|A_N) Pr(A_N). \end{aligned} \tag{9}$$

In FP-SBF, if bucket is not set to *Max* in l iterations, more then one operations are required to decrement its value to zero, i.e., for $l \leq Max$ cell can be decreased to zero and for $l = N$ A_N event occurs. So from Equation (9) it is proved that $\lim_{N\to\infty} Pr(ASBF_N = 0)$. Hence, proposed FP-SBF follows stable property principle of standard SBF efficiently and with less computational complexity. □

Definition 1 (Convergence Rate (CR))**.** From the stable point property, each bucket has fixed probability of being set to *Max* and constant probability of being reset to zero after certain iterations. P_0, probability that a cell becomes zero is same for all buckets. Thus, expected number of zeros in FP-SBF converges exponentially and Convergence Rate (CR) can be derived using Equation (9):

$$CR = Pr(ASBF_N = 0) - Pr(ASBF_{N-1} = 0). \tag{10}$$

Definition 2 (Stable point)**.** It is the expected fraction of zeros in SBF, when data is unbounded. Using Theorem 1 probability of values in bucket being set to zero is constant, i.e.

$$\begin{aligned} Pr(ASBF_N = 0) = &\sum_{l=Max}^{N-1} [Pr(ASBF_N = 0|A_l) Pr(A_l)] \\ &+ Pr(ASBF_N = 0|A_N) Pr(A_N). \end{aligned} \tag{11}$$

Let H be the number of cells that are set to Max and L be the number of cells that are decremented to zero after certain number of iterations (from Theorem 1), then expected number of zeros in FP-SBF, i.e. stable point property of FP-SBF, is given by (Z_{SP}):

$$Z_{SP} = \left(\frac{1}{1 + \frac{1}{L(\frac{1}{H} - \frac{1}{m})}} \right)^{Max}. \tag{12}$$

The optimization in the decrement operation will help to achieve stable point in FP-SBF in less number of iterations with less computational complexity, which will further help to detect parameters like false positives and negatives at earlier stage.

4.5 False Positives (FPs) Analysis

Theorem 2. When SBF reaches a stable point, the FPs is given by:

$$FP = F_{FP\text{-}SBF} - RF_{FPC}$$

where $F_{FP\text{-}SBF}$ is error due to collision in buckets of FP-SBF and RF_{FPC} denotes the reduction factor in FPs due to fingerprint cells.

Proof. False positives are generated when distinct element in the stream is wrongly reported as duplicate. FPs are directly dependent on number of zeros at particular point in FP-SBF, i.e., when stable point is reached FPs can be estimated. Change in fingerprint cell of each bucket helps to improve FPs by providing a second level check. In case of collision in the buckets, $F_{FP\text{-}SBF}$ is dependent on the number zeros in the filter. More the number of zeros, less the collisions and less FPs. When stable point is reached, number of zeros becomes constant and after certain iterations, H buckets are set to Max and L are decremented to zero; $F_{FP\text{-}SBF}$ is given:

$$F_{FP\text{-}SBF} = \left(\frac{1}{1 + \frac{1}{L(\frac{1}{H} - \frac{1}{m})}} \right)^{Max}. \tag{13}$$

RF_{FPC} is the reduction factor in detection procedure due to fingerprint cells when all buckets corresponding to hash indexes are high. It acts like a second level check on these f bits. Since the possibility of FPs is one out of 2^f cases in fingerprint cells, it helps in reducing the FPs by a fixed factor. Assuming that insertion operation is performed I times on k indexes (equal to number of hash functions) and fingerprint cells are updated by XOR operation; RF_{FPC}, the probability of not having collision in f bits in m FPC's is:

$$RF_{FPC} = \left[\left(\frac{2^f - 1}{2^f} \right) \left(\frac{1}{m} \right) \binom{I}{1} \left(1 - \frac{1}{f} \right)^{I-1} \right]^k. \tag{14}$$

From Equations (13) and (14), FPs in FP-SBF are given by:

$$FPs = \left(\frac{1}{1+\frac{1}{L(\frac{1}{H}-\frac{1}{m})}}\right)^{Max} - \left[\left(\frac{2^f-1}{2^f}\right)\left(\frac{1}{m}\right)\binom{I}{1}\left(1-\frac{1}{f}\right)^{I-1}\right]^k. \quad (15)$$

Thus, the use of fingerprint cell helps to reduce the FPs by a significant factor and improve the accuracy of the duplicate detection system. □

4.6 False Negative Rate (FNs) Analysis

Theorem 3. At stable point FNs for FP-SBF are given by:

$$FNs = FR_{FP\text{-}SBF} + E_{FPC}$$

where $F_{FP\text{-}SBF}$ is error due to deletion operation on buckets of FP-SBF and E_{FPC} denotes the error in fingerprint cell due to the collision of H_{fp} function.

Proof. A false negative in duplicate detection on streamed data occurs when a duplicate element (x_i) is wrongly reported as distinct element. This happens during decrement operation when some buckets associated with hashed indexes of x_i are decremented to zero, before appearing in the stream or there is a mismatch in fingerprint cells when all the buckets have high value corresponding to x_i.

Suppose x_i appears second time in the stream after δ iterations and $h_{j=1}^{j=k}(x_i)$ denotes the corresponding hash indexes and p_{ij} is probability that a particular cell C_j is set to Max. In δ iterations, some bucket from $h_{j=1}^{j=k}(x_i)$ are reduced to zero. The probability of error due to buckets resetting $(FR_{FP\text{-}SBF})$ is same as in Equation (9); given by:

$$FR_{FP\text{-}SBF} = 1 - \prod_{j=1}^{k}(1 - Pr(ASBF_\delta = 0)) \quad (16)$$

and probability that after δ iterations a particular bit of FP-SBF is zero, i.e. $Pr(ASBF_\delta = 0)$, is given by:

$$\begin{aligned} Pr(ASBF_\delta = 0) = \sum_{l=Max}^{\delta-1} [Pr(ASBF_N = 0|A_l)Pr(A_l)] \\ + Pr(ASBF_N = 0|A_N)Pr(A_N). \end{aligned} \quad (17)$$

So FN is function of δ and p_{ij} given by:

$$FR_{FP\text{-}SBF} = 1 - \prod_{j=1}^{k}(1 - Pr(\delta_j, p_{ij})). \quad (18)$$

E_{FPC}, chance of false negatives due to error in H_{fp} hash function in fingerprint cell is due to the collision in changing bits because of XOR operation in δ iterations. In m size array, selected k FPCs are checked and mismatch in any of them leads to failure in detection process, i.e., a duplicate is reported as distinct element. After f iterations, chances that erroneously reported elements are 1 out of 2^f for δ iterations using k hash function is given by:

$$E_{FPC} = \left[\left(\frac{1}{2^f}\right) \left(\frac{1}{m}\right) \binom{\delta}{1} \left(1 - \frac{1}{f}\right)^{\delta-1} \right]^k. \tag{19}$$

FNs for FP-SBF is given by:

$$FNs = \left(1 - \prod_{j=1}^{k} (1 - Pr(ASBF_\delta = 0))\right) + \left[\left(\frac{1}{2^f}\right) \left(\frac{1}{m}\right) \binom{\delta}{1} \left(1 - \frac{1}{f}\right)^{\delta-1} \right]^k. \tag{20}$$

Use of FPCs shows great improvement in decreasing the FPs of FP-SBF in all cases. □

Theorem 4. For given inputs k, Max, f, FPs and H_{fp}; processing each data item of the stream requires $O(k+1)$ time which is independent of the size of Bloom filter and the incoming data stream.

Proof. In duplicate detection, the primary goal is minimization of error rates while using constant space, and time complexity in the detection process should be independent of the nature and size of data stream, i.e., there should be constant processing time for each element. First step in duplicate detection is to check whether an element is seen previously in the stream or not. For this first k hash functions are computed for buckets and then one hash function is calculated for fingerprint cell. For given parameters k, Max, f, FPs with constant values, there is no effect of element and the detection process is also independent of the size of Bloom filter (m) (Algorithm 1).

From Algorithm 1 and analysis performed above it is concluded that processing time in duplicate detection is only dependent on number of hash functions, i.e. $O(k+1)$. □

5 OBSERVATIONS AND ANALYSIS

The theoretical analysis has been provided in previous sections for some important parameters, i.e. Max, H, L, m, ξ, false negatives (FN) and false positives (FP); where H is number of cells set to Max in insertion operation (i.e. equal to number of hash functions), L denotes the number of cells selected for the decrement operation and m is fixed amount of memory used for the Bloom filter.

False positives can be bounded according to user specified requirements. Since the false negatives in the fixed amount of memory are depended on deletion operation

they cannot be bounded in specified limits. Two parameters, desired false positive (FP) rate and size of the Bloom filter (m) are taken as input from user and other parameters like Max, H, L are selected in such a way that false negatives are minimal. The parameter ξ is also user defined and helps to control the frequency of deletion operation.

For user defined FPs, m and for constant values of Max and H; L is defined as:

$$L = \left(\frac{1}{\left(\frac{1}{\left(1 - FPs^{1/H}\right)^{1/Max}} - 1 \right) (1/H - 1/m)} \right). \tag{21}$$

Equation (21) helps to find value of L, i.e. number of cells selected for decrement operation. From the value of L (calculated in Equation (21)) p_d, the probability of selecting cells for decrement, and P_0, probability that after certain decrement operations value of cell is zero, can be derived.

Parameter H is equal to the number of hash functions used (k). $E(FN)$ denotes the expected number of false negatives in the stream. Optimal value of H should be selected to minimize the FNs. With $\tilde{N}$ as the number of false negatives in a stream of N elements, $E(FN)$ is:

$$E(FN) = \sum_{i=1}^{\tilde{N}} \left(Pr(FNR_i) \right), \tag{22}$$

$$E(FN) = \sum_{i=1}^{\tilde{N}} \left(\left(1 - \prod_{j=1}^{k} (1 - Pr(ASBF_\delta = 0)) \right) + \left[\left(\frac{1}{2^f} \right) \left(\frac{1}{m} \right) \binom{\delta}{1} \left(1 - \frac{1}{f} \right)^{\delta - 1} \right]^k \right). \tag{23}$$

The value of Max is dependent on size of input data and the number of hash functions used. Optimal value of Max can be derived from Equation (23) by minimizing FNs. For efficient memory utilization Max should be set $2^c - 1$. The remaining bits that are not allocated to counter are used as fingerprint bits. For fixed amount of cells when value of Max is increased, the effectiveness of fingerprint cell is reduced.

The optimization in deletion process is controlled by user defined parameter ξ and $Rand(\xi)$ function is used to set the frequency of deletion operation; larger the value of ξ more frequently the deletion operation is performed and vise versa.

All the experiments have been performed on i7-3612QM CPU @ 2.10 GHz with 8 GB of RAM. To maintain the uniformity in the results *CityHash* 64 bit library [2] is used to calculate two hash functions in double hashing. Data set of 100 k elements with 70 % distinct entires and 30 % duplicate entires has been generated using R-studio.

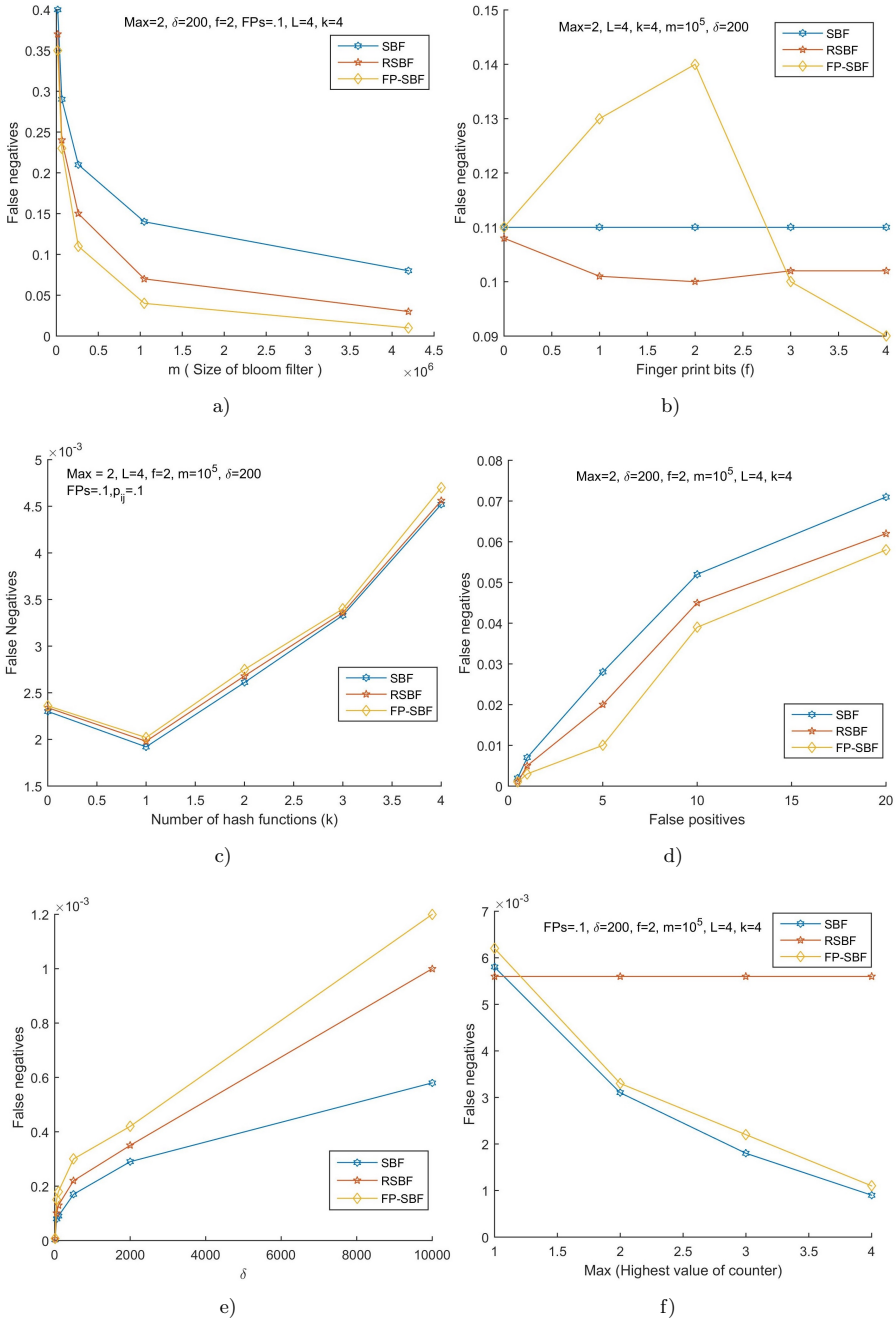


Figure 5. False negatives variation with different parameters

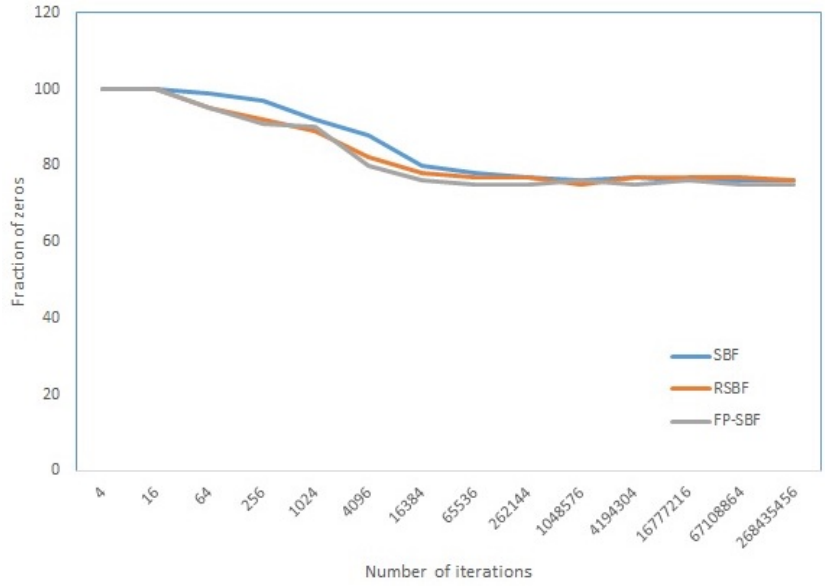


Figure 6. Stable point

Comparative analysis has been performed between SBF, RSBF and FP-SBF. All the experiments indicate that the proposed approach outperforms SBF and RSBF; both the approaches used for duplicate detection in the streamed data using Bloom filters.

Figure 5 shows the impact on false negatives with variations in size of Bloom filter (m), size of counter in bucket (Max), number of fingerprint bits (f), number of hash functions (k) and false positives (FP). All results have been evaluated using fixed values for the required parameters.

As shown in Figure 5 a), false negatives decrease with the increase in size of Bloom filter; the more the space, the less the effect of deletion operation and the less the false negatives. Figure 5 b) indicates the change in false negatives with respect to fingerprint bits in FP-SBF; since the results of SBF and RSBF are not effected by value of f, so false negatives remain the same. In FP-SBF first false negative increases when f is small; for $f = 3$, the accuracy is same as in the RSBF, but as the value of f increases, the acurracy of the proposed scheme increases. Figure 5 c) indicates that as the number of hash functions increases more positions need to be checked, increasing the chances of false negatives. With the change in predefined false positives the changes in false negatives are indicated by Figure 5 d) showing that the more the FPs, the bigger error is allowed; the less deletion operations, the less false negatives appear. δ denotes the average number of iterations between two similar items in the stream, effect of δ on false negatives is shown in Figure 5 e) which clearly shows that as the

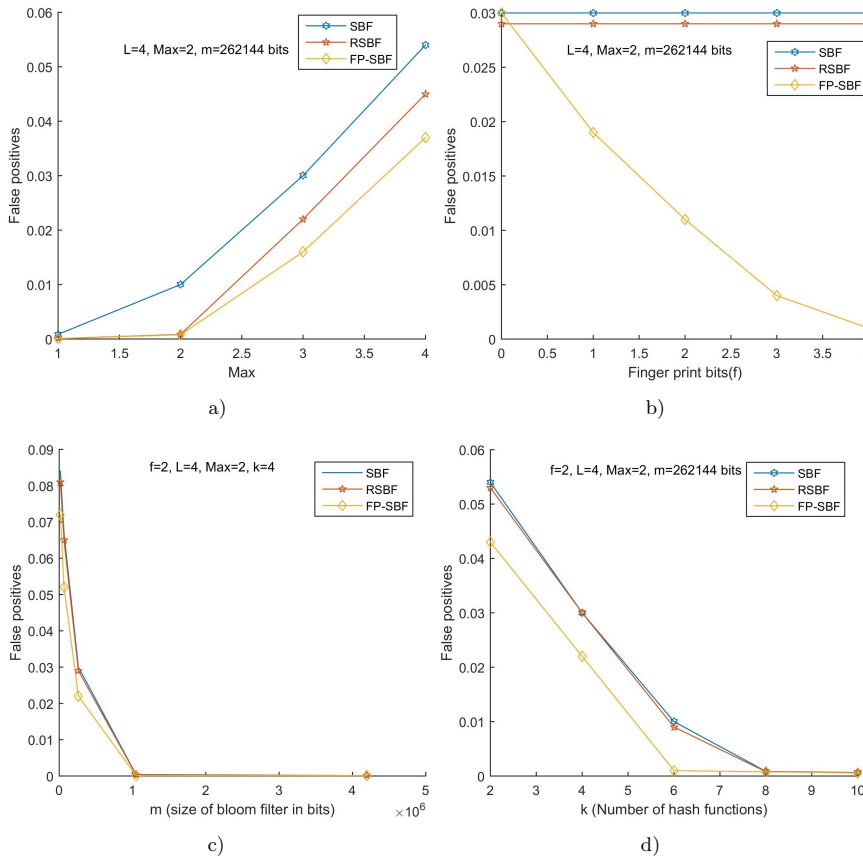


Figure 7. False positives variation with different parameters

value of δ increases, duplicates are detected after more iterations, hence more deletion operations are performed between two similar elements, so chances are high that a duplicate is detected as a distinct element. Figure 5 f) shows the change in false negatives with respect to bucket size *Max*; the larger the bucket size, the more operations are required to reset it to zero and the less false negatives appear.

Figure 6 depicts the number of iterations required to achieve stable point, i.e. constant fraction of zeros in the Bloom filter. Initially, the filter is empty so number of zero is 100 %. As the number of iterations increases, more insertion operations are performed and number of zeros is reduced; after some time deletion operation is also performed and later number of zeros becomes constant. Both RSBF and FP-SBF perform efficiently in achieving the stable point but FP-SBF has the advantage of controlling the deletion operation by using optimized

deletion process which reduces the number of iterations required to reach stable point.

Figure 7 provides the false positives analysis which shows that with the use of finger print bits false positives are drastically reduced in the FP-SBF, as compared to RSBF and SBF. In Figure 7 a) false positives increase with the size of counter in bucket, but the use of fingerprint bits in the FP-SBF helps to reduce it to a great extent. Figure 7 b) shows that the more the number of fingerprint bits used the less false positives appear in duplicate detection task. Figure 7 c) indicates that the larger the size of Bloom filter used, the less false positives appear. Number of hash functions is always a critical factor in the Bloom filter, Figure 7 d) shows that the more hash functions used, the less false positives appear.

6 DISCUSSION

FP-SBF uses fingerprint bits to improve the accuracy of the task and there are cases when false negatives increase with the reduction in false positives. Few use cases are discussed below to analyse the output in various scenarios.

- Parameter tunning in FP-SBF which can decrease the false positives to minimum level is:

$$MFP((Max > 3), (f > 3), (m > 5 \times 10^5), (k > 5)). \tag{24}$$

Case I – where FP-SBF can be used successfully – includes:

  - IoT data streams, where data is coming from number of sensors, FP-SBF can be used to check the identity of the sensors.
  - Detection of the first time user in real time data streams of online shopping platforms for promotional strategies.
  - Detection of the active users in social networking websites like Twitter, Facebook, etc., from the number of posts registered in the given time span t.

- Parameter tunning in FP-SBF which can decrease the false negatives to minimum level is:

$$MFN((\delta < 1\,000), (Max > 4), (f > 4), (m > 4 \times 10^6), (2 < k < 5)). \tag{25}$$

Case II – where FP-SBF might not be the best option – includes:

  - Medical domain, where life saving critical decisions need to be taken.
  - Surveillance and monitoring of real time data for security and safety purposes.
  - Online detection of financial frauds like credit card fraud, money laundering activities, illegal betting, etc.

7 CONCLUSION

Duplicate detection task on streaming data in one pass is among the most important tasks associated with in-stream analytics. To accommodate unbounded data and detect the duplicate in stream, the proposed framework FP-SBF uses advanced stable Bloom filter with fingerprint bits to decrease error rate. d-left hashing has been used which leads to less collisions and update minimum number of counters in insertion operation. Further d-left hashing improves the accommodation capacity of Bloom filter and improve accuracy in results. A randomized approach in deletion process is used to reduce the computational overhead as compared to existing technique. Results achieved clearly indicate that the proposed framework performs efficiently for duplicate detection problem and further parameters can be tunned according to specific application's requirement.

Acknowledgment

The first author would like to acknowledge the financial support given to him by the Department of Computer Science and Technology under the Department of Electronics and Information Technology (DeitY) to complete his doctoral studies.

REFERENCES

[1] Korwar, A.: Bloom Filters. `http://www.cse.iitk.ac.in/users/arpk/articles/BloomFilters.pdf`. [Online; May 2010].

[2] Pike, G.—Alakuijala, J.: Introducing CityHash. `https://opensource.googleblog.com/2011/04/introducing-cityhash.html`. [Online; April 11, 2011].

[3] What Is Cloud Computing? `http://aws.amazon.com/what-is-cloud-computing/`. [Online; 2013].

[4] Alur, R.—Berger, E.—Drobnis, A. W.—Fix, L.—Fu, K.—Hager, G. D.—Lopresti, D.—Nahrstedt, K.—Mynatt, E.—Patel, S. et al.: Systems Computing Challenges in the Internet of Things. arXiv preprint arXiv:1604.02980, 2016.

[5] Antichi, G.—Ficara, D.—Giordano, S.—Procissi, G.—Vitucci, F.: Blooming Trees for Minimal Perfect Hashing. IEEE 2008 Global Telecommunications Conference (IEEE GLOBECOM 2008), 2008, pp. 1–5, doi: 10.1109/glocom.2008.ecp.305.

[6] Azar, Y.—Broder, A. Z.—Karlin, A. R.—Upfal, E.: Balanced Allocations. SIAM Journal on Computing, Vol. 29, 1999, No. 1, pp. 180–200, doi: 10.1137/s0097539795288490.

[7] Babcock, B.—Babu, S.—Datar, M.—Motwani, R.—Widom, J.: Models and Issues in Data Stream Systems. Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '02), ACM, 2002, pp. 1–16, doi: 10.1145/543613.543615.

[8] Bloom, B. H.: Space/Time Trade-Offs in Hash Coding with Allowable Errors. Communication of the ACM, Vol. 13, 1970, No. 7, pp. 422–426, doi: 10.1145/362686.362692.

[9] Bonomi, F.—Mitzenmacher, M.—Panigrahy, R.—Singh, S.—Varghese, G.: An Improved Construction for Counting Bloom Filters. In: Azar, Y., Erlebach, T. (Eds.): Algorithms – ESA 2006. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4168, 2006, pp. 684–695, doi: 10.1007/11841036_61.

[10] Chang, F.—Feng, W.-C.—Li, K.: Approximate Caches for Packet Classification. Proceedings of the Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), 2004, Vol. 4, pp. 2196–2207, doi: 10.1109/infcom.2004.1354643.

[11] Chen, H.—Chiang, R. H. L.—Storey, V. C.: Business Intelligence and Analytics: From Big Data to Big Impact. MIS Quarterly: Management Information Systems, Vol. 36, 2012, No. 4, pp. 1165–1188, doi: 10.2307/41703503.

[12] Cohen, S.—Matias, Y.: Spectral Bloom Filters. Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD '03), ACM, New York, NY, USA, 2003, pp. 241–252, doi: 10.1145/872757.872787.

[13] Deng, F.—Rafiei, D.: Approximately Detecting Duplicates for Streaming Data Using Stable Bloom Filters. Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data (SIGMOD '06), ACM, New York, NY, USA, 2006, pp. 25–36, doi: 10.1145/1142473.1142477.

[14] Dutta, S.—Bhattacherjee, S.—Narang, A.: Towards Intelligent Compression in Streams: A Biased Reservoir Sampling Based Bloom Filter Approach. Proceedings of the 15th International Conference on Extending Database Technology (EDBT '12), ACM, 2012, pp. 228–238, doi: 10.1145/2247596.2247624.

[15] Dutta, S.—Narang, A.—Bera, S. K.: Streaming Quotient Filter: A Near Optimal Approximate Duplicate Detection Approach for Data Streams. Proceedings of the VLDB Endowment, Vol. 6, 2013, No. 8, pp. 589–600, doi: 10.14778/2536354.2536359.

[16] Fan, B.—Andersen, D. G.—Kaminsky, M.—Mitzenmacher, M. D.: Cuckoo Filter: Practically Better Than Bloom. Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies (CoNEXT '14), ACM, 2014, pp. 75–88, doi: 10.1145/2674005.2674994.

[17] Fan, L.—Cao, P.—Almeida, J.—Broder, A. Z.: Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. IEEE/ACM Transactions on Networking (TON), Vol. 8, 2000, No. 3, pp. 281–293, doi: 10.1109/90.851975.

[18] Kapoor, A.—Arora, V.: Application of Bloom Filter for Duplicate URL Detection in a Web Crawler. 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC), 2016, pp. 246–255, doi: 10.1109/cic.2016.042.

[19] Kirsch, A.—Mitzenmacher, M.: Less Hashing, Same Performance: Building a Better Bloom Filter. Random Structures and Algorithms, Vol. 33, 2008, No. 2, pp. 187–218, doi: 10.1002/rsa.20208.

[20] Liberty, E.—Nelson, J.: Streaming Data Mining. Presented at Princeton University by Yahoo Research Group, online 2012.

[21] Merlin, J. S.—Mary, A. V. A.: An Approach for Quick and Efficient Detection of Duplicate Data-Survey. International Journal of Applied Engineering Research, Vol. 11, 2016, No. 5, pp. 3430–3432.

[22] Metwally, A.—Agrawal, D.—El Abbadi, A.: Duplicate Detection in Click Streams. Proceedings of the 14th International Conference on World Wide Web (WWW '05), ACM, New York, NY, USA, 2005, pp. 12–21, doi: 10.1145/1060745.1060753.

[23] Pontarelli, S.—Reviriego, P.—Maestro, J. A.: Improving Counting Bloom Filter Performance with Fingerprints. Information Processing Letters, Vol. 116, 2016, No. 4, pp. 304–309, doi: 10.1016/j.ipl.2015.11.002.

[24] Rothenberg, C. E.—Macapuna, C. A. B.—Verdi, F. L.—Magalhaes, M. F.: The Deletable Bloom Filter: A New Member of the Bloom Family. IEEE Communications Letters, Vol. 14, 2010, No. 6, pp. 557–559, doi: 10.1109/lcomm.2010.06.100344.

[25] Rottenstreich, O.—Kanizo, Y.—Keslassy, I.: The Variable-Increment Counting Bloom Filter. IEEE/ACM Transactions on Networking, Vol. 22, 2014, No. 4, pp. 1092–1105, doi: 10.1109/tnet.2013.2272604.

[26] Singh, M. P.—Hoque, M. A.—Tarkoma, S.: Analysis of Systems to Process Massive Data Stream. arXiv preprint arXiv:1605.09021, 2016.

[27] Wei, J.—Jiang, H.—Zhou, K.—Feng, D.—Wang, H.: Detecting Duplicates over Sliding Windows with RAM-Efficient Detached Counting Bloom Filter Arrays. 2011 IEEE Sixth International Conference on Networking, Architecture and Storage (NAS '11), 2011, pp. 382–391, doi: 10.1109/nas.2011.37.

[28] Yoon, M.: Aging Bloom Filter with Two Active Buffers for Dynamic Sets. IEEE Transactions on Knowledge and Data Engineering, Vol. 22, 2010, No. 1, pp. 134–138, doi: 10.1109/tkde.2009.136.

Amritpal Singh received his Ph.D. degree from Thapar Institute of Engineering and Technology (TIET), Punjab, India, with a minor in big data analysis and probabilistic data structures in 2018. He is working as Lecturer with Computer Science Department at TIET, Punjab, India. He served both industry and academia. His research interest includes probabilistic data structures, machine learning and big data.

Shalini Batra received her Ph.D. degree in computer science and engineering from Thapar University, Patiala, India, in 2012. She is currently working as Associate Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, India. She has guided many research scholars leading them to Ph.D. and M.E./M.Tech. She authored more than 60 research papers published in various conferences and journals. Her research interests include machine learning, web semantics, big data analytics and vehicular ad-hoc networks.

Computing and Informatics, Vol. 37, 2018, 1339–1362, doi: 10.4149/cai_2018_6_1339

LEARNED SPATIO-TEMPORAL TEXTURE DESCRIPTORS FOR RGB-D HUMAN ACTION RECOGNITION

Zhengyuan Zhai, Chunxiao Fan, Yue Ming

Beijing University of Posts and Telecommunications
Beijing Key Laboratory of Work Safety Intelligent Monitoring
Xitucheng Road 10
100 876 Beijing, China
e-mail: {zhaiyuan, cxfan@bupt.edu.cn}, myname35875235@126.com

Abstract. Due to the recent arrival of Kinect, action recognition with depth images has attracted researchers' wide attentions and various descriptors have been proposed, where Local Binary Patterns (LBP) texture descriptors possess the properties of appearance invariance. However, the LBP and its variants are most artificially-designed, demanding engineers' strong prior knowledge and not discriminative enough for recognition tasks. To this end, this paper develops compact spatio-temporal texture descriptors, i.e. 3D-compact LBP(3D-CLBP) and local depth patterns (3D-CLDP), for color and depth videos in the light of compact binary face descriptor learning in face recognition. Extensive experiments performed on three standard datasets, 3D Online Action, MSR Action Pairs and MSR Daily Activity 3D, demonstrate that our method is superior to most comparative methods in respects of performance and can capture spatial-temporal texture cues in videos.

Keywords: 3D pixel differences vectors, compact binary face descriptor, feature fusion, human action recognition, RGB-depth videos

Mathematics Subject Classification 2010: 68Txx

1 INTRODUCTION

As an important field of computer vision, human action recognition (HAR) has acquired many researchers' attention and been extensively used in our real life,

such as human-computer interaction, smart video surveillance and assisted living. Researchers mainly focus on recognizing actions from common videos in the past, facing with the challenge of variable illuminations, cluttered background and partial occlusions. Lately, due to the prevalence of low-cost depth sensors like Kinect, color and depth data are more easily to access simultaneously. Complementary to color images, depth images are robust to the change in lighting conditions and background, also can provide 3D structure of the object. Color images can offer more color, texture and appearance information than depth images. Consequently, more researchers are paying attention to recognize actions with color and depth images, namely RGB-D action recognition [7, 12, 20, 27, 28].

It has been shown that texture features are always important descriptors for the task of recognition, like the Local Binary Pattern (LBP) [1]. However, there are two issues when employing such features to recognize action from depth images. For one thing, most texture feature descriptors [2, 3, 4, 5, 17] require engineers' strong prior knowledge to determine the threshold, which is a constant during the process of encoding texture information such that it does not work for different datasets. For another, we cannot obtain discriminative LBP feature from depth images due to the lack of texture information. To solve the mentioned problems, we respectively develop 3D-compact local binary patterns (3D-CLBP) and local depth patterns (3D-CLDP) descriptor for color and depth images in this paper built on the Compact Binary Face Descriptor (CBFD) learning [22], which allows us to learn local binary patterns from raw pixels automatically. Furthermore, two fusion features are presented for RGB-D action recognition with developed 3D-CLBP and 3D-CLDP descriptor, thus they can simultaneously possess the texture characteristic of color and depth data.

As depicted in Figure 1, both color and depth videos are divided into non-overlap space-time volumes. The pixel differences vectors (PDVs) of volumes with same spatial locations and through the overall time are first extracted, then used to learn a spatial projection. To obtain compact and robust binary coding, we project all PDVs into binary vectors by learned spatial projections, and aggregate those binary vectors to low-dimensional LBP features. We further improve the discriminability of those LBP features by jointly employing the sparse coding and spatial-temporal pyramid pooling. In the end, we adopt the feature-level and decision-level fusion to simultaneously capture texture cues from color and depth data.

The main contributions of our work can be summarized as below:

1. Considering that the difference between central pixel and neighboring pixels may change both spatially and temporally when motion occurs, we propose a method to extract PDVs from a spatio-temporal volume using the difference of the former and later frames' neighboring pixels (central pixel) to current frame's central pixel (neighboring pixels).
2. Our 3D-CLBP and 3D-CLDP descriptors are extensions of CBFD, which straightly learn spatial thresholds from raw spatio-temporal pixels, making them

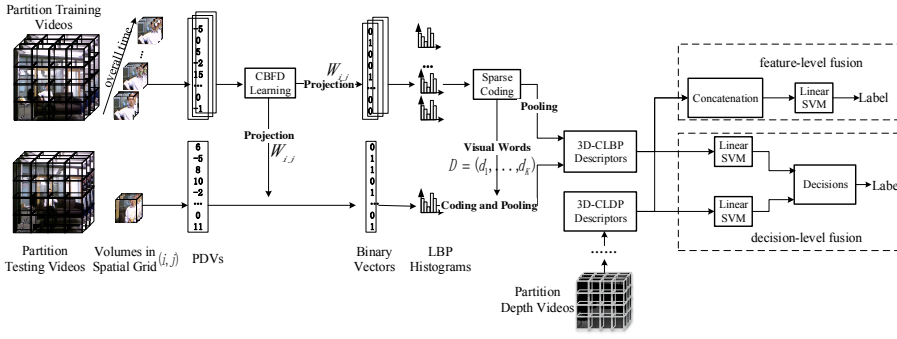


Figure 1. The pipeline of our developed method for action recognition with color and depth data. The pipeline presents the process of developing our 3D-CLBP descriptors for color videos. Likewise, we can develop our 3D-CLDP descriptors for videos in depth channel. In the end, two fusion techniques are used to achieve the task of action recognition.

more selective and suitable for different datasets than hand-crafted descriptors since we do not require engineers' strong prior knowledge.

3. We investigate different fusion methods to explore simultaneously preserving texture cues in color and depth data, further validate these methods in experiment to demonstrate the complementary nature of RGB and depth information in action recognition task.

The rest of our paper is organized as follows. The related work is given in Section 2. Section 3 describes detailed process of our compact binary codes learning in videos. The introduction of our classification and fusion methods are contained in Section 4. Section 5 reports the experiment parameters tuning and experiments' results on three datasets. Conclusions of this paper are given in Section 6.

2 RELATED WORK

This section first reviews some related researches on action recognition using LBP-like features from depth channel. We also briefly investigate some existing fusion works utilizing the data from color and depth channel.

Li et al. [6] first reported the work of action recognition from depth sequences. Afterwards, a variety of descriptors have been proposed, such as Spatio-Temporal features [8, 9, 11, 12, 20], Shape-motion features [13, 15, 16], and Texture descriptors [17, 18, 19]. Among those, texture descriptors are robust to subjects' clothing, appearance, and can capture substantial texture variations in the video. However, there is absence of abundant color and texture cues in depth images such that most extended LBP descriptors developed for color videos are not available in depth images sequence. To this end, the work [17] first projected an entire depth video into Depth Motion Maps (DMMs) from three projection views (front, side and top), then

employed the LBP descriptor on these projected DMMs. Later, Bulbul [18] respectively calculated the LBP and Edge Oriented Histograms (EOHs) within overlapping and non-overlapping blocks on DMMs to extract local texture and dense shape information. The DMMs can provide adequate texture information for action recognition and well solve the variation of different videos' duration. Nevertheless, the work [19] indicated that the DMMs are presentation of entire depth video, cannot possess the motion and appearance information in temporal. Toward this problem, they extracted space-time auto-correlation of gradients as a complementary feature to conquer the loss of temporal information in the process of generating DMMs. Instead of computing LBP feature from DMMs, the work in [21] put forward the Gradient-LBP (G-LBP) descriptor to encode facial information from 2D depth images. The common fault of above works is that these existing LBP-like descriptors are all hand-crafted, which demands engineers' strong prior knowledge. To eliminate the defect of hand-crafted LBP descriptors, Lu [22] proposed a compact binary face descriptor (CBFD) learning for face recognition. With CBFD learning, obtained binary codes can evenly distribute at each bin and contain more discriminative information than hand-crafted descriptors. Enlightened by this, we attempt to extend the CBFD learning in 2D images to 3D videos in this paper.

In the light of complementary nature of color and depth information, some earlier works using both color and depth data in different recognition tasks can be found in those works. For instance, Ni et al. [7] derived Depth-layered multi-channel Spatio-Temporal Interest Points (STIPs) and 3D Motion History Images (3D-MHIs) from primitive STIPs and MHIs to fuse color and depth information for activity recognition. They also show a fusion framework to localize complex activity in videos by integrating information from grayscale and depth images in [23]. Zhu [24, 25] investigated some previous depth features developed for HAR, further combined these features and STIPs-based feature in color channel with various fusion schemes. Considering that features from RGB and depth channel share some similar structure, the works [26, 27, 28] explored the relationship between visual and depth features with different learning methods, which projected visual and depth features into a common subspace. The difference between them is that the work [26] projects Local Flux Feature (LFF) extracted from RGB and depth channel into a hamming subspace, while the works [27] and [28] learn the projection with label information. Similar to above learning methods, the works [29] and [30] respectively utilized graph-based genetic programming (RGGP) and regularized reconstruction independent component analysis deep network to build the relationship between RGB and depth modality for recognition tasks. Besides, Jia et al. [31] treated action data as fourth-order tensor, and discovered the correlation between RGB and depth modalities with cross-modality regularized transfer learning. Zhang and Parker [32] detected STIPs in saliency maps constructed by color and depth videos, then calculated 4-dimensional color-detph (CoDe4D) orientation histogram descriptor on each interest point. Results of above methods fully highlight that combining color and depth data can benefit to the task of action recognition, leading us to obtain selective representation for action recognition with our developed color and

depth feature. Moreover, Kong and Fu [36] proposed the bilinear heterogeneous information machine (BHIM) to learn cross-modal features for RGB-D action recognition, which captures heterogeneous visual and depth information simultaneously. While Shahroudy et al. [41] utilized a deep autoencoder-based nonlinear common component analysis network to discover the shared and informative componets of RGB and depth data for an action. With the development of deep learning, many convolution neural networks (ConvNets) based methods were proposed for action recognition and obtained promising recogniton performance. Karpathy [37] and Tran [39] respectively utilize a deep 3-dimensional convolutional network (3D ConvNet) to recognize actions in video. Simonyan and Zisserman [38] proposed a two stream framework which employ two ConvNets to respectively extract features from appearance and motion streams then fuse the results for recognition. To deal with excessive computational cost when applied 3D ConvNet on long video sequences, the temporal segment network (TSN) [40] was proposed based on long-range temporal structure modeling. However, those ConvNets based methods need a great deal of training samples and have no evident advantage over traditional methods for our concerned small RGB-D datasets in this paper.

3 COMPACT BINARY CODES LEARNING

We first elaborate the motivation of our 3D pixel difference vectors (PDVs) extraction, then present how to compute discriminative 3D-PDVs from a spatio-temporal volume in this section. In the end, we extend the Compact Binary Face Descriptor (CBFD) learning in face images to videos.

3.1 Motivation and 3D PDVs

For a video, previous methods always calculate the differences between central pixel (or neighboring pixels) of current frame and neighboring pixels (or central pixel) of frames before (after) it, then compare the differences of both to obtain binary values. Such methods can miss the information of current frame since the pixel of current frame will be easily counteract when it is greater or less than corresponding pixels before and after the current frame. Moreover, we observe that the pixel at central location (x, y) of current time t may shift to neighboring location $(x + \Delta x, y + \Delta y)$ at time $t - \Delta t$ or $t + \Delta t$ when motions happen and vice versa. This stimulates us to find a better way to capture textures dynamic changes during motions occurrence.

Let $\{V_n^c, V_n^d\}$, $n = 1, 2, \ldots, N_0$ be N_0 color-depth video pairs in the dataset. For a video V_n^c (or V_n^d), it's often divided into some space-time volumes $\{V_{ij,k}^c, i = 1, \ldots, M; j = 1, \ldots, N; k = 1, \ldots, F\}$ of fixed size, like $20 \times 20 \times 5$ in our experiment, by $M \times N \times F$ grids. To obtain discriminative pixel differences vector $PDV_{ij,k}$ of a spatio-temporal volume $V_{ij,k}^c$, our method not only calculates the differences between current frames' central pixel and the former (latter) frames' neighboring pixels, also measures the differences between the former (latter) frames' central pixel

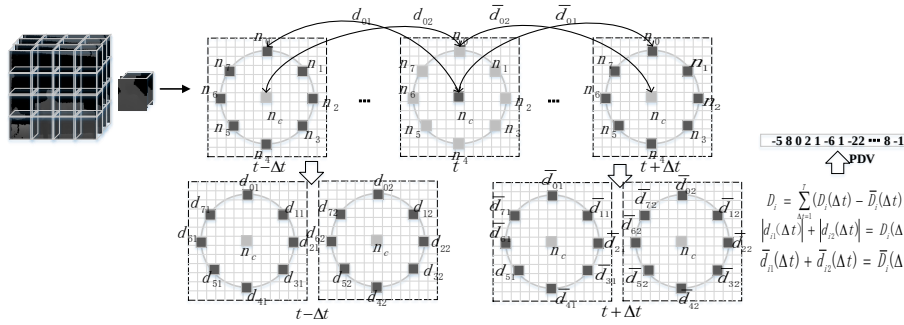


Figure 2. Illustration of our method to extract PDVs from a volume both spatially and temporally. Here we just show the process of calculating the frames at time $t - \Delta t$ and $t + \Delta t$ to current frame at time t. The same process can be used for depth data after computing the gradient of each frame.

and current frames' neighboring pixels, so that it can better capture the dynamic change of texture both spatially and temporally.

As depicted in Figure 2, given a $20 \times 20 \times 5$ volume V, central point n_c and its neighbors n_i, $i = 0, \ldots, 7$ at time t, $t - \Delta t$ and $t + \Delta t$, $\Delta t = 1, 2$. We first compute the pixel differences between n_c at time t and n_i at time $t - \Delta t$, $t + \Delta t$, respectively written as

$$d_{i1}(\Delta t) = I_{n_c}(t) - I_{n_i}(t - \Delta t), \quad \overline{d_{i1}}(\Delta t) = I_{n_c}(t) - I_{n_i}(t + \Delta t), \quad \Delta t = 1, 2 \tag{1}$$

where $I_{n_c}(t)$ denotes as the pixel value of n_c at time t. Likewise, we measure the pixel differences between n_c at time $t - \Delta t, t + \Delta t$ and n_i at time t, which can be represented as

$$d_{i2}(\Delta t) = I_{n_i}(t) - I_{n_c}(t - \Delta t), \quad \overline{d_{i2}}(\Delta t) = I_{n_i}(t) - I_{n_c}(t + \Delta t), \quad \Delta t = 1, 2. \tag{2}$$

Upon obtaining $d_{i1}(\Delta t)$, $d_{i2}(\Delta t)$, $\overline{d_{i1}}(\Delta t)$, $\overline{d_{i2}}(\Delta t)$, $i = 0, \ldots, 7$ at time $t - \Delta t$, $t + \Delta t, \Delta t = 1, 2$, the distance of current frame to frames before and after it, respectively denoted as D_i and $\overline{D_i}$, can be defined following

$$D_i = \sum_{\Delta t=1}^{2} (|d_{i1}(\Delta t)| + |d_{i2}(\Delta t)|), \quad \overline{D_i} = \sum_{\Delta t=1}^{2} (|\overline{d_{i1}}(\Delta t)| + |\overline{d_{i2}}(\Delta t)|). \tag{3}$$

Then, the difference between D_i and $\overline{D_i}$ of each pixel in volume V forms the final PDV. In respect of depth videos, we first compute the gradient information of each frame, then extract PDVs from depth videos with Equations (1), (2) and (3).

3.2 CBFD Learning

The CBFD has been proven to be more effective than conventional LBP descriptors in face recognition due to learning binary codes automatically instead of manually designing an encoding method. Herein, we extend CBFD learning in 2D images to 3D videos. Given all PDVs $\{PDV_{ij,k}^{c_m}, m = 1, \ldots, N_1\}$ from N_1 training color videos, we aim to train $M \times N$ projections ω_{ij}, $i = 1, \ldots, M$, $j = 1, \ldots, N$. For different color videos c_m, the number of PDVs k with a spatial grid (i, j) may be different and large. To improve computation efficiency, we randomly sample some $PDV_{ij,k}^{c_m}$ to train a projection $\omega_{ij} \in \mathbb{R}^{8\times 8}$. For simplicity, training PDVs set for ω_{ij} denotes as $\{pdv_{ij,k}, k = 1, \ldots, N'\}$, $pdv_{ij,k} \in \mathbb{R}^{8\times d}$, it can be projected and quantized to binary codes $b_{ij,k} = \left[b_{ij,k}^{(1)}, \ldots, b_{ij,k}^{(8)}\right]$, with $b_{ij,k}^{(8)} \in \{0, 1\}^{d\times 1}$ as below:

$$b_{ij,k} = 0.5 \times (\mathrm{sgn}(pdv_{ij,k}^{\mathrm{T}}\omega_{ij}) + 1)$$

where $\mathrm{sgn}(x)$ is written as the sign function, equaling to 1 if $0 \leq x$ and -1 otherwise.

To get more discriminative and compact binary codes $b_{ij,k}$ for a volume, the CBFD learning imposes three important criterions (i.e. evenly distributed binary codes, less redundancy and less missed information in the learned binary codes) on objective function. The optimization objective function can be formulated as follows:

$$\begin{aligned}\min_{\omega_{ij}} J_1(\omega_{ij}) + \lambda_1 * J_2(\omega_{ij}) + \lambda_2 * J_3(\omega_{ij}) = &-\sum_{k=1}^{N'} \|b_{ij,k} - \mu_k\|^2 \\ &+ \lambda_1 \sum_{k=1}^{N'} \left\|(b_{ij,k} - 0.5) - pdv_{ij,k}^{\mathrm{T}}\omega_{ij}\right\|^2 \\ &+ \lambda_2 \left\|\sum_{k=1}^{N'}(b_{ij,k} - 0.5)\right\|^2 \end{aligned} \tag{4}$$

In the above formula, N' indicates the number of PDVs extracted from training videos in spatial grid (i, j), μ_k serves as the mean of all training PDV's k^{th} binary code, updating in each iteration, λ_1 and λ_2 are two parameters to balance the effect of different terms in the objective function. The physical meaning of different terms and solution can be referred to [22].

4 CODING AND CLASSFICATION

In this section, we describe the idea of sparse coding and show how to aggregate learned LBP features into our final 3D-CLBP and 3D-CLDP descriptors based on the spatial-temporal pooling. We also introduce our classification and fusion techniques in this section.

4.1 Sparse Coding and Spatial-Temporal Pyramid Pooling

Instead of assigning each feature vector to the nearest visual word learned by k-means clustering in Bag of Words (BOW), the sparse coding enables a linear and sparse combination of all learned atoms in dictionary. This approach reduces the quantization error in the process of approximating a crude feature vector. To well keep more information of low-level features, the work [13] employs the coefficient-weighted differences between each visual word and a primitive feature vector. Consider a set of extracted features $P = (p_1, \ldots, p_M)^{\mathrm{T}} \in \mathbb{R}^{M \times N}$ and coefficients of features $U = (u_1, \ldots, u_M)^{\mathrm{T}} \in \mathbb{R}^{M \times K}$. The process of sparse coding can be represented by:

$$\min_{D,U} \sum_{m=1}^{M} \left(\left\| p_m - D^{\mathrm{T}} u_m \right\|^2 + \lambda \left\| u_m \right\|_1 \right), \quad \text{subject to } \|d_k\|^2 \leq 1, \quad \forall k = 1, \ldots, K$$

where $D = (d_1, \ldots, d_K)^{\mathrm{T}} \in \mathbb{R}^{K \times N}$is the learned dictionary with K visual words, and λ is the induced parameter of sparsity regularization. A detailed process of solving above optimization problem can be referred to [13]. Once learned the sparse coefficient u_{ik} of feature p_i to the k^{th} atom in dictionary, we can employ the coefficient-weighted difference $u_{ik}(p_i - d_k)$ as the coding.

In Section 3, we divide a video into some 3D volumes of fixed size, leading to different number of volumes with different videos. Hence, the final discriminative feature vector of different videos has different length, which cannot serve as the input of our classifier. The common solution is to perform spatial average pooling and temporal max (or sum) pooling for each video, which partition a video into various space-time grids. Let a space-time denoted by ST_t, which may be the entire video or a partitioned subsequence. vol_i is a small $20 \times 20 \times 5$ volume involved in ST_t and p_i is the feature of vol_i. $|ST_t|$ indicates the number of volumes in ST_t. Then the spatial average pooling and temporal max pooling with all partition are formulized as below:

$$v_k(t) = \frac{1}{|ST_t|} \sum_{vol_i \subset ST_t} u_{ik}(p_i - d_k), \quad v_k = \left(\max_t \{v_{k1}(t)\}, \ldots, \max_t \{v_{kN}(t)\} \right).$$

We concatenate all pooled vectors v_k from K visual words to form the distinctive vector $V = (v_1, \ldots, v_K)$ of KN dimensions as 3D-CLBP and 3D-CLDP descriptors.

4.2 Classification and Fusion

We employ widely used Support Vector Machine (SVM) as our action classifying framework. To deal with our sparse data, we use SVM with linear kernel as our classifier. The solver of linear SVM can be given in LIBLINEAR [33]. For data fusion, there are two ways: early fusion (sensor and feature level) and late fusion

(rank, score, and decision level). We apply feature-level fusion and decision-level fusion to combine our 3D-CLBP and 3D-CLDP descriptors.

1. Feature-level fusion. We simply stack the 3D-CLBP and 3D-CLDP descriptor into a composite vector for classification. To reduce the complexity, we perform PCA with all 3D-CLBP and 3D-CLDP descriptors before concatenation.

2. Decision-level fusion. Different from the straightforward fusion, the decision-level fusion considers each of our feature as the input to a SVM classifier, then merges the results using the confidence scores generated by two individual SVM classifiers. Denoting $f_q(x)_k$ as the q^{th} classifiers' confidence scores predicting x to the k^{th} label, the posterior probability associated with q^{th} classifier $p_q(y_k|x)$ can be written as below:

$$p_q(y_k|x) = \frac{1}{1+\exp(-f_q(x)_k)}.$$

Then we employ different decision rules to combine the two classifiers' results, such as Sum rule, Maximum rule shown in [25]. For Sum rule, Product rule, Maximum rule and Minimum rule, we respectively assign the final label y^* to the k^{th} label as Equations (5):

$$\begin{aligned} P(y_k|x) &= \sum_{q=1}^{2} \alpha_q p_q(y_k|x), \\ P(y_k|x) &= \prod_{q=1}^{2} p_q(y_k|x)^{\alpha_q}, \\ P(y_k|x) &= \max_{q=1,2} p_q(y_k|x), \\ P(y_k|x) &= \min_{q=1,2} p_q(y_k|x), \\ y^* &= \operatorname*{argmax}_{k=1,2,\ldots,C} P(y_k|x). \end{aligned} \tag{5}$$

5 EXPERIMENTAL

In this section, we conduct experiments on three common RGB-D action datasets, with our developed 3D-CLBP and 3D-CLDP feature, and two fusion features. We first introduce the datasets and their settings, then illustrate experiments setup including parameter setting and tuning. Finally, we compare our proposed methods with other state-of-the-art methods.

5.1 Evaluation Datasets

We use three RGB-D action databases and their settings following previous work to evaluate the performance of our developed features, i.e. 3D Online Action, MSR Action Pairs, and MSR Daily Activity. Some sampled video frames of these datasets are illustrated in Figure 3.

3D Online Action (3Donline) dataset is an RGB-D action dataset including 7 human-object interactions in the living room: drinking, eating, using laptop, reading cellphone, making phone call, reading book and using remote, where each action is performed by 16 subjects twice. The dataset has three parts: same-environment, cross-environment and multiple unsegmented actions. We evaluate our method with same-environment actions and follow the experiment setting employed in [14], which adopts the first 8 subjects as training and the last 8 subject as testing.

The MSR Pairs Action dataset (MSRpairs) contains six pairs of actions: lift a box/place a box, pick up a box/put down a box, push a chair/pull a chair, put on a backpack/take off a backpack, stick a poster/remove a poster, and wear a hat/take off a hat. These paired-activities are performed by 10 subjects and each subject performs each activity 3 times. Thus, there are totally 720 color and depth sequences with the resolution of 480×640 and 240×320, respectively. The dataset is challenging since paired-activities are very similar but the motion happens in different temporal order. We employ the first five actors for training and the rest for testing as described in [15].

The MSR Daily Activity dataset (MSRdaily) totally has 720 sequences including color and depth (each 360) with 16 daily activities: drink, eat, read book, call cellphone, write on paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lie down on sofa, walk, play guitar, stand up and sit down. In the dataset, RGB videos are offered with a resolution of 480×640, while each depth frame has a resolution of 240×320. Each daily activity is carried out by 10 subjects twice in the posture of standing or sitting, leading to large spatial and scaling changes. We follow the experiment setup in [16], taking subjects 1, 3, 5, 7, 9 as training and subjects 2, 4, 6, 8, 10 as testing.

5.2 Parameter Settings and Tuning

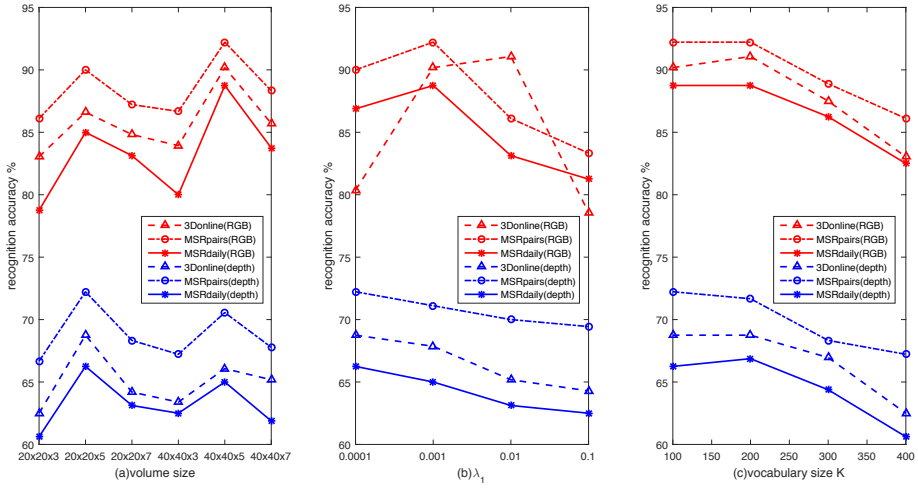
As aforementioned in Section 3, all videos (color and depth channel) are divided into multiple volumes, so we first adopt different volume sizes from $\{20\times20\times3, 20\times20\times5, 20\times20\times7, 40\times40\times3, 40\times40\times5, 40\times40\times7\}$ to test recognition performance over all experiment datasets, then employ the best volume size to tune other parameters. For the computation of neighboring pixels' differences we always choose 8 points with best neighborhood radius size R from $\{1, 2, 3\}$. Actually, we set $R = 1$ to get a fair result comparison with crude LBPs. λ_1 and λ_2 respectively denotes the weight of regularization term in Equation (5), we set λ_1 and λ_2 as the same value here and choose the best one from $\{0.0001, 0.001, 0.01, 0.1\}$ following [22]. To explore the impact of visual words size k in sparse coding, we tune k from $\{100, 200, 300, 400\}$

a) b) c)

Figure 3. Some sampled color images and corresponding depth images from three experimental datasets: a) sampled frames of "read book" with different performers in 3D Online Action dataset, b) sampled frames of the "push a chair/pull a chair" pair from MSR Action Pairs dataset, c) sampled frames of activities with different performers' postures in MSR Daily Activity 3D dataset

with 10 fold-cross validations. The value of regularization parameter λ in sparse coding is set to 0.15 as [13]. Besides, we employ a space-time pyramid of $4 \times 3 \times 7$ grids to pool our features.

Figure 4 a) shows recognition performance with different volume size over all datasets (RGB and depth channel). It can be observed that our approach obtains the best recognition performance with volume size $40 \times 40 \times 5$ and $20 \times 20 \times 5$ respectively over all RGB and depth datasets since color and depth images respectively have resolutions of 480×640 and 240×320. Moreover, we find that the performance is more sensitive to the choice of volume size in temporal scale than spatial scale, and a moderate temporal size can lead to good recognition accuracy. The explanation here is that the small time interval cannot provide sufficient information for a motion, when the time interval is too long some local variations are missing. The recognition performance over experiment datasets using different λ_1 values is depicted in Figure 4 b). From this figure, we observe that the choice of best λ_1 value for RGB channel generally depends on the characteristic of dataset. For example, the 3Donline dataset with significant intra-class variations, like subjects' clothing and motion style, achieves the best recognition performance when using $\lambda_1 = 0.01$; the MSRpairs and MSRdaily datasets have little intra-class variations obtaining best performance with $\lambda_1 = 0.001$. Another interesting observation is that our approach over all depth datasets is not very sensitive to the value of λ_1,



a) b) c)

Figure 4. Performance comparison with different parameter values over all experiment datasets (RGB and depth channel) with our developed descriptors. a) performance comparison with different volume sizes, with $\lambda_1 = 0.001$ (RGB channel), 0.0001 (depth channel), $K = 100$; b) performance comparison with different λ_1 values, with volume size $= 40 \times 40 \times 5$ (RGB channel), $20 \times 20 \times 5$ (depth channel), $K = 100$; c) performance comparison with different K values, with volume size $= 40 \times 40 \times 5$, $\lambda_1 = 0.001$ (RGB channel), volume size $= 20 \times 20 \times 5$, $\lambda_1 = 0.0001$ (depth channel).

which intuitively illustrates the absence of texture information in depth images. We compare the recognition performance over all datasets with different number of visual words K in Figure 4 c). As presented in the figure, the performance over all datasets generally reaches best under a moderate vocabulary size, i.e. $K = 200$. Because small vocabulary size often leads to incorrectly assigning different features to a same visual word, and similar features are assigned to different visual words when vocabulary size is too large. However, we adopt $K = 100$ in our experiment since the performance is nearly the same when $K = 100$ and $K = 200$, but the dimension of feature will enlarge $59 \times 100 \times 84$, further resulting in higher computation complexity.

5.3 Experiment Results and Comparison

To evaluate our proposed method, we choose some previous representative RGB and depth features as our comparison, like Improved Dense Trajectories (IDTs)-based, DMMs-based features and STIPs-based fusion technique. Specifically, we implement our experiment over the ORGBD and MSRpair dataset using released source codes with default parameter settings for IDTs, DMMs and STIPs.

IDTs-based features [34] first sample feature points in each frame and track them with dense optical flow tracking technique. Then, some low-level feature (MBH, HOG or HOF) is encoded to IDTs-based feature with spatio-temporal pyramids, i.e. IDTs-MBH, IDTs-HOG, IDT-HOF.

DMMs-based features [17] including DMMs-LBP, DMMs-HOG and DMMs-EOH are built on depth motion maps generated by accumulating motion energy of projected depth maps from front, side and top view. Different from 3D features, DMMs-based features encode the motion characteristics of an action from 2D images.

STIPs-based features employ Harris 3D detector, cuboid detector or Hessian detector to detect interest points from videos or depth maps, and extract local feature descriptors from each detected interest point location. The common STIPs-based features are STIPs-HOG, STIPs-HOG3D and STIPs-HOF, etc.

In addition to the above features, we also investigate and report other methods using these three datasets. Detailed comparison results are presented in Table 1.

Channel + Methods + Classifier	Accuracy
RGB + IDTs-HOG/HOF + SVM [34]	77.68 %
RGB + STIPs-HOG + SVM [35]	79.46 %
Depth + depthHarris3D-DCSF + SVM [11]	61.70 %
Depth + Ordelet + AdaBoosting [14]	71.40 %
Depth + DMMs-LBP + KELM [17]	69.64 %
Both + STIPs-HOG3D + SVM [25]	91.07 %
Both + DSSCA-SSLM [41]	94.6 %
RGB + 3D-LBP + linearSVM	62.50 %
RGB + 3D-CLBP + linearSVM	90.18 %
Depth + 3D-LDP + linearSVM	37.50 %
Depth + 3D-CLDP + linearSVM	68.75 %
Both + feature-level fusion + SVM	93.75 %
Both + SVM + decesion-level(Sum) fusion	88.39 %
Both + SVM + decesion-level(Maximum) fusion	89.29 %
our 3D-CLBP + IDTs-FV + decesion-level fusion	**94.64** %

Table 1. Average recognition accuracy comparison of our method and previous approaches over the 3Donline dataset

5.3.1 Experiments Results and Comparison on 3Donline

The average recognition accuracy of feature-level fusion with our 3D-CLBP and 3D-CLDP descriptor achieves 93.75 % over the 3Donline dataset, as shown in Table 1. We compare our proposed method with some baseline methods, like IDTs-based, STIPs-based and DMMs-based methods in Table 1. The table presents that our 3D-CLBP descriptor with linearSVM in RGB channel obtains the average accuracy of 90.18 % significantly outperforming the IDTs-HOG/HOF and STIPs-HOG features. Although the average recognition accuracy of our 3D-CLDP descriptor

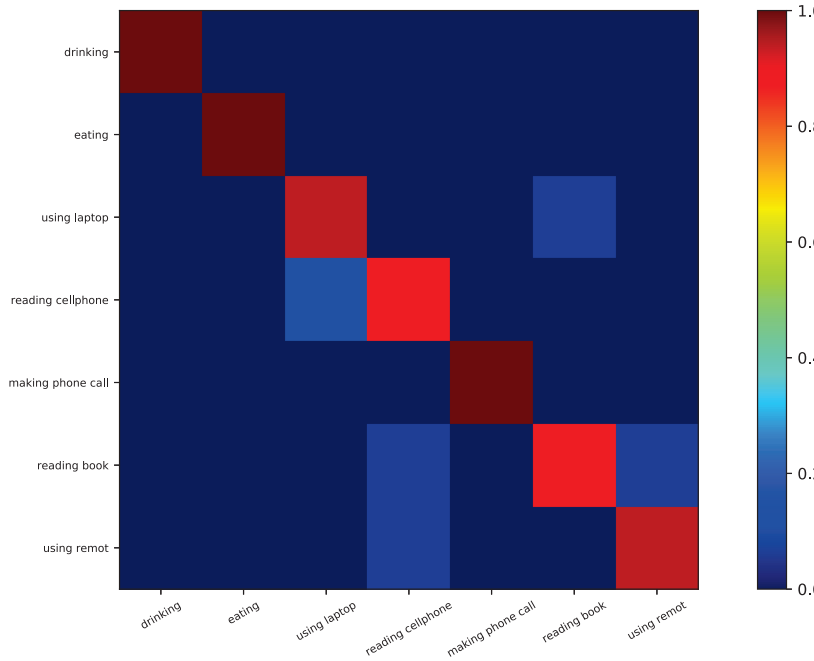


Figure 5. Fusion recognition results over the 3Donline dataset with our method

with linearSVM in depth channel is slightly below the Ordelet and DMMs-LBP method, it is higher than depth Harris3D-based method since Harris3D cannot detect sufficient spatio-temporal interest points in depth channel for partial occlusion in actions "using laptop", "reading cellphone" and "reading book". To demonstrate the superior performance of developed 3D-CLBP and 3D-CLDP, we also conduct experiments using primitive 3D-LBP and 3D-LDP with pre-set threshold 20 and 0.5, which obtains average recognition accuracy of 62.50 % and 37.50 %. Moreover, the feature-level, decision-Sum and decision-Maximum fusion with 3D-CLBP and 3D-CLDP are compared with STIPs-based fusion technique in [25]. And we obtain a promising result 93.75 % with feature fusion, around 2.7 % more than the STIP-HOG3D feature. For decision fusion, we obtain better recognition accuracy of 89.29 % with Maximum rule than weighted Sum rule in that our 3D-CLDP descriptor cannot work well on this dataset. To further improve the performance of our method, we combine the classify results of our 3D-CLBP descriptor and IDT-features with Fisher Vector (FV), and achieves the best performance of 94.64 %. This recognition result is also obtained by the deep shared-specific component analysis (DSSCA) network with structured sparsity learning machine (SSLM), which employed the deep convolutional network to extract modality-specific components of the modalities. The experiment results with our 3D-CLBP features and IDT

descriptors are shown in Figure 5. It is observed that our method can recognize the action "drinking", "eating" and "making phone call", where interactive objects have distinctive texture characteristic. While the action "reading cellphone" and "reading book" not having distinctive interactive objects are recognized with some wrong actions. Above observation indicates that our 3D-CLBP descriptor is capable of encoding texture information.

Channel + Methods + Classifier	Accuracy
RGB + IDTs-HOG/HOF + SVM [34]	**100** %
RGB + STIPs-HOG + SVM [35]	81.67 %
Depth + Skeleton-LOP + SVM [10]	63.33 %
Depth + SNV + linearSVM [13]	98.89 %
Depth + HON4d + SVM [15]	93.33 %
Depth + DMMs-LBP + KELM [17]	78.89 %
Both + STIPs-HOG3D + SVM [25]	95.0 %
Both + DRRL + linearSVM [28]	99.44 %
Both + BHIM [36]	**100** %
Both + DSSCA-SSLM [41]	**100** %
RGB + 3D-LBP + linearSVM	67.78 %
RGB + 3D-CLBP + linearSVM	92.22 %
RGB + 3D-LDP + linearSVM	45.0 %
Depth + 3D-CLDP + linearSVM	72.23 %
Both + feature-level fusion + SVM	92.78 %
Both + decesion-level(Sum) fusion + SVM	93.89 %
Both + decesion-level(Maximum) fusion + SVM	91.67 %
Both + decesion-level(Minimum) fusion + SVM	86.11 %
our 3D-CLBP + IDTs-FV + decesion-level fusion	97.78 %

Table 2. Comparison of average recognition accuracy on the MSRpairs dataset with our method and some works

5.3.2 Experiments Results and Comparison on MSRpairs

Table 2 shows comparison results of our proposed approach with previous baseline methods over the MSRpairs dataset. From this table, it can be seen that the IDTs-based method in RGB channel obtains best recognition accuracy of 100 %, because actions in MSRpairs dataset have distinct motion direction. The same reason can also account for good performance of Depth super normal vector (SNV) in [13]. When compared to most mehods in RGB and Depth channel, we cannot obtain more satisfactory recognition performances with 3D-CLBP and 3D-CLDP descriptor due to encoding local texture changes but being incapable of capturing holistic changes in temporal-dependent sequences. Another observation is that the performance of 3D-CLDP descriptor reaches 72.23 % lower than DMMs-LBP's 78.89 %, which explicitly illustrates DMMs can better encode local texture than depth images. Nevertheless, the recognition performance of developed features are still far above

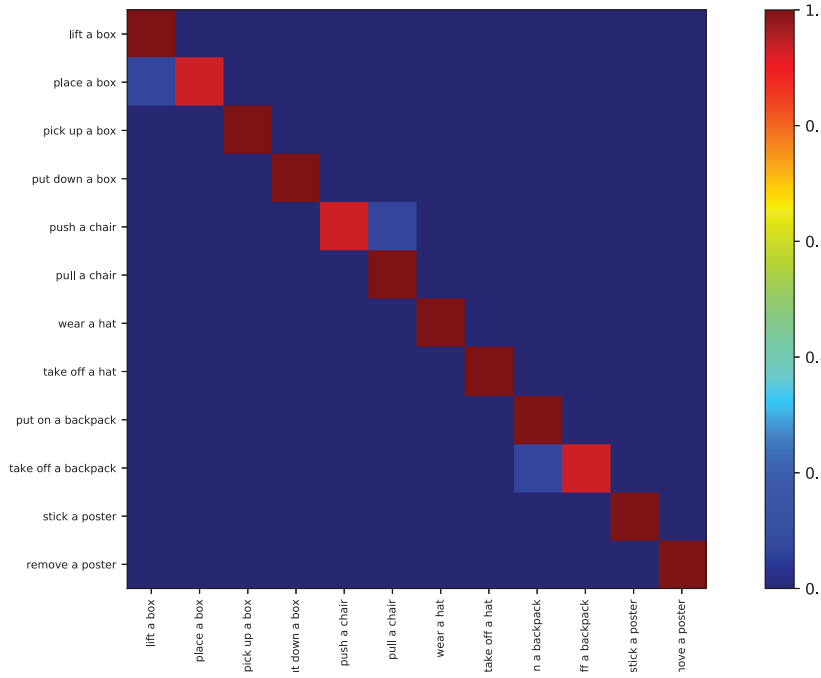


Figure 6. Obtained fusion recognition results over the MSRpairs dataset with our method

crude 3D-LBP and 3D-LDP features. Towards fusion methods using both RGB and depth data, the recogniton accuracy of proposed 3D-CLBP and 3D-CLDP with different fusion strategies reaches merely up to 93.89 %, which is lower than discriminative relational representation learning (DRRL), BHIM and lately DSSCA-SSLM. Even when fusing the results of our 3D-CLBP and effective IDTs-FV method, the fusion methods still cannot recognize all pairs actions which can be realized by just utilizing the IDTs method. It indicates that our proposed 3D-CLBP features with SVM mistake some actions with a larger probability. The reason is that some pairs actions in this dataset are extremely similar except the begining and end of a performed action, and they further generate same 3D PDVs with our proposed pixels difference computation method. This also accounts for the phenomenon that we do not obtain some encouraging experimental results using proposed method on this dataset comparing with the 3Donline dataset. In the end, we compare different decision fusion methods and gain better result with weighted Sum rule (balance parameter $\alpha = 0.8$ of two modality) than Maximum and Minimum rule. It well demonstrates that features in both RGB and Depth channel are important for the recognition task and have different impact. Figure 6 presents the experiment results by combining our 3D-CLDP with IDTs features and Fisher Vector. As shown in this figure, the fusion methods can recognize almost all pairs where temporal order and

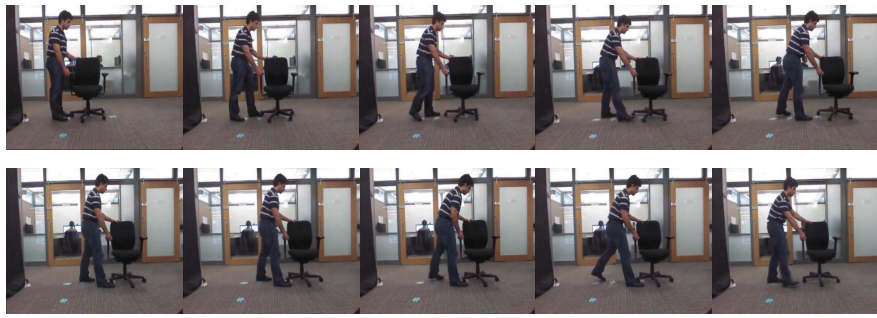


Figure 7. Sampled images except the begining and end of pair actions "push a chair" and "pull a chair" in MSRpairs dataset. Those images are very similar in terms of performed pose, leading to same PDVs and further incorrect identification

motion direction are significantly different, such as "pick up a box" and "put down a box" except some pair actions performed with some similar pose as illustrated in Figure 7.

5.3.3 Experiments Results and Comparison on MSRdaily

We compare the average recognition accuracy of our developed methods over the MSRdaily dataset with some baseline methods using color or depth images in Table 3. It can be observed that 3D-CLBP descriptor obtaining 88.75 % with RGB data is obviously superior to the performance of IDTs-based feature and an extended LTP, i.e. Center-Symmetric Motion LTP (CS-Mltp) feature [20], indicating our 3D-CLBP descriptor is more efficient than hand-crafted LTP. But the 3D-CLDP feature cannot get satisfactory result with depth data when compared with DMMs-LBP feature and some representative depth features, such as SNV, HON4d and Orderlet. To better evaluate the performance of our method over the MSRdaily dataset, we also compare our fusion features with some previous fusion methods, including STIPs-based fusion, LFF, DRRL, a deep learing model (RGGP) and BHIM methods. The results demonstrate that the feature-level and decesion-level fusion methods outperform those fusion approaches, increasing by 0.7 % to 6.9 %. However, the best performance of our fusion methods (namely feature-level fusion) only achieves 92.5 %, respectively 2.5 % and 5 % lower than the joint heterogeneous features learning (JOULE) model and lately DSSCA-SSLM method. This is because those two methods select discriminative features with learning models instead of merging features or recognition results directly in this paper. Another important observation in Table 3 is that our weighted decesion fusion features perform better than developed feature only using color or depth data, which highlights the importance of combing color and depth cues for recognition task. Besides, we analyze and compare fusion methods with different decision levels to explain the complementary nature of both modalities. The fused recognition results of this dataset with our 3D-CLBP and

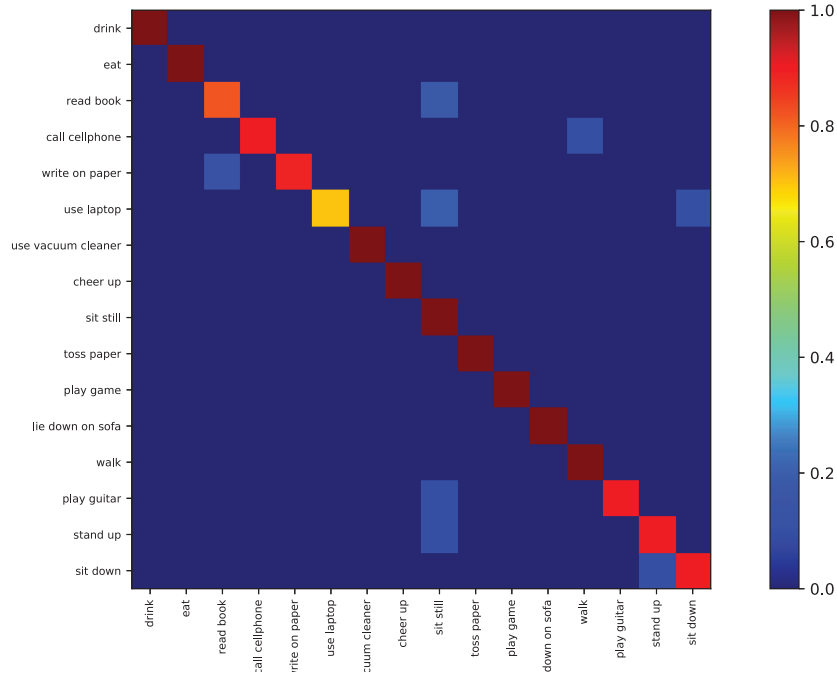


Figure 8. Fusion recognition results using proposed 3D-CLBP and IDTs-FV method over the MSRdaily dataset

a) b) c)

Figure 9. Incorrect identified actions from MSRdaily datasets. The action "read book" performed in sitting or standing way, i.e. column a), are identified as action "sit still" or "stand still", i.e. column b). Column c) enumerates two misidentified actions performed with seriously occlusion.

Channel + Methods + Classifier	Accuracy
RGB + CS-Mltp + SVM [20]	65.63 %
RGB + IDTs-HOG/HOF + SVM [34]	60.63 %
Depth + SNV + linearSVM [13]	86.25 %
Depth + HON4d + SVM [15]	80.00 %
Depth + Ordelet + SVM [16]	85.75 %
Depth + DMMs-LBP + KELM [17]	72.50 %
Both + STIPs-HOF/skeleton + 1NN [24]	89.29 %
Both + LFF-SPP + NN [26]	89.80 %
Both + DCP-DDP + JOULE-SVM [27]	95.0 %
Both + DRRL + linearSVM [28]	87.50 %
Both + RGGP(Deep model) [29]	85.60 %
Both + CoDe4D + SVM [32]	86.25 %
Both + BHIM [36]	86.88 %
Both + DSSCA-SSLM [41]	**97.5** %
RGB + 3D-CLBP + linearSVM	88.75 %
Depth + 3D-CLDP + linearSVM	66.25 %
Both + feature-level fusion + SVM	92.5 %
Both + decesion-level(Sum) fusion + SVM	90.63 %
Both + decesion-level(Product) fusion + SVM	91.87 %
our 3D-CLBP + IDTs-FV + decesion-level fusion	93.75 %

Table 3. Comparison of average recognition accuracy on the MSRdaily dataset using our method and other methods

IDTs-FV method is shown in Figure 8. From this figure, we can observe that our feature can correctly recognize all actions with acute motions, like "drink","eat", "cheer up","toss paper", "play game", "lie down on sofa","walk"and so on. But in terms of those actions with little motion such as "read book", "write on paper" and "play guitar", they may be wrongly identified as "sit still" in that those actions are performed in a sitting or standing way as shown in Figure 9 a). Thus, when those actions performed with no distinct pose, the calculated PDVs may be the same as "sit down still" or "stand up still" Figure 9 b). This demonstrates that our 3D-PDVs focus on the change of appearance rather than appearance information when motion happens. Besides, there are much wrong identification happen between "use laptop"and "sit still". The cause of this phenomenon can be interpreted as these actions are performed with seriously occlusion as demonstrated in Figure 9 c). We also notice that "sit still", "stand up"and "sit down" may be wrongly identified as other two actions since they share same action atomic.

6 CONCLUSIONS

This paper extends compact binary face descriptors learning in 2D images to 3D videos, which automatically learn discriminative binary representations for action recognition with color and depth videos. To this end, we develop a method to ex-

tract 3D pixels difference vectors (PDVs) from spatio-temporal volumes, then learn spatial projections with some PDVs extracted from the same spatial grid, and further project those PDVs into low-dimension binary codes. Moreover, we employ the sparse coding and spatial-temporal pooling to obtain discriminative representation of a video. In the end, we investigate different fusion methods to check the validity of combining color and depth data for action recognition. Extensive experiments performed on three standard benchmarks demonstrate that our method is superior to most methods being compared on 3D Online Action and MSR Daily Activity 3D datasets. However, we cannot obtain satisfying results on MSR Action Pairs dataset. Hence, combining the framework of deep learning or skeleton positions, to learn more discriminative descriptor for action recognition is our further research.

REFERENCES

[1] Ojala, T.—Pietikainen, M.—Maenpaa, T.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, 2002, No. 7, pp. 971–987, doi: 10.1109/tpami.2002.1017623.

[2] Kellokumpu, V.—Zhao, G.—Pietikäinen, M.: Human Activity Recognition Using a Dynamic Texture Based Method. In: Everingham, M., Needham, C. (Eds.): Proceedings of the British Machine Vision Conference (BMVC), Vol. 1, 2008, pp. 88.1–88.10, doi: 10.5244/c.22.88.

[3] Yeffet, L.—Wolf, L.: Local Trinary Patterns for Human Action Recognition. 2009 IEEE 12th International Conference on Computer Vision (ICCV '09), Kyoto, 2009, pp. 492–497, doi: 10.1109/iccv.2009.5459201.

[4] Mattivi, R.—Shao, L.: Human Action Recognition Using LBP-TOP as Sparse Spatio-Temporal Feature Descriptor. In: Jiang, X., Petkov, N. (Eds.): Computer Analysis of Images and Patterns (CAIP '09). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5702, 2009, pp. 740–747, doi: 10.1007/978-3-642-03767-2_90.

[5] Gupta, R.—Patil, H.—Mittal, A.: Robust Order-Based Methods for Feature Description. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10), San Francisco, June 2010, pp. 334–341, doi: 10.1109/cvpr.2010.5540195.

[6] Li, W.—Zhang, Z.—Liu, Z.: Action Recognition Based on a Bag of 3D Points. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Workshops (CVPRW '10), San Francisco, June 2010, pp. 9–14, doi: 10.1109/cvprw.2010.5543273.

[7] Ni, B.—Wang, G.—Moulin, P.: RGBD-HuDaAct: A Color-Depth Video Database for Human Daily Activity Recognition. 2011 IEEE International Conference on Computer Vision Workshops (ICCV 2011 Workshops), Barcelona, November 2011, pp. 1147–1153.

[8] Ming, Y.—Ruan, Q.—Hauptmann, A. G.: Activity Recognition from RGB-D Camera with 3D Local Spatio-Temporal Features. In: 2012 IEEE International Conference on Multimedia and Expo (ICME '12), Melbourne, July 2012, pp. 344–349, doi: 10.1109/icme.2012.8.

[9] Vieira, A. W.—Nascimento, E. R.—Oliveira, G. L.: STOP: Space-Time Occupancy Patterns for 3D Action Recognition from Depth Map Sequences. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (Eds): Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7441, 2012, pp. 252–259, doi: 10.1007/978-3-642-33275-3_31.

[10] Wang, J.—Liu, Z.—Wu, Y.—Yuan, J.: Mining Actionlet Ensemble for Action Recognition with Depth Cameras. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12), Providence, June 2012, pp. 1290–1297, doi: 10.1109/cvpr.2012.6247813.

[11] Xia, L.—Aggarwal, J. K.: Spatio-Temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera. Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13), Portland, June 2013, pp. 2834–2841, doi: 10.1109/cvpr.2013.365.

[12] Koppula, H. S.—Saxena, A.: Learning Spatio-Temporal Structure from RGB-D Videos for Human Activity Detection and Anticipation. Proceedings of the 30$^{\text{th}}$ International Conference on Machine Learning – Volume 28 (ICML '13), Atlanta, June 2013, pp. 792–800, doi: 10.1177/0278364913478446.

[13] Yang, X.—Tian, Y. L.: Super Normal Vector for Activity Recognition Using Depth Sequences. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14), Columbus, June 2014, pp. 804–811, doi: 10.1109/cvpr.2014.108.

[14] Yu, G.—Liu, Z.—Yuan, J.: Discriminative Orderlet Mining for Real-Time Recognition of Human-Object Interaction. In: Cremers, D., Reid, I., Saito, H., Yang, M. H. (Eds): Computer Vision – ACCV 2014. Springer, Cham, Lecture Notes in Computer Science, Vol. 9007, 2014, pp. 50–65, doi: 10.1007/978-3-319-16814-2_4.

[15] Oreifej, O.—Liu, Z.: HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences. Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13), Portland, June 2013, pp. 716–723, doi: 10.1109/cvpr.2013.98.

[16] Wang, J.—Liu, Z.—Wu, Y.—Yuan, J.: Learning Actionlet Ensemble for 3D Human Action Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 36, 2014, No. 5, pp. 914–927, doi: 10.1109/tpami.2013.198.

[17] Chen, C.—Jafari, R.—Kehtarnavaz, N.: Action Recognition from Depth Sequences Using Depth Motion Maps-Based Local Binary Patterns. 2015 IEEE Winter Conference Applications of Computer Vision (WACV '15), Waikoloa, January 2015, pp. 1092–1099, doi: 10.1109/wacv.2015.150.

[18] Bulbul, M. F.—Jiang, Y.—Ma, J.: DMMs-Based Multiple Features Fusion for Human Action Recognition. International Journal of Multimedia Data Engineering and Management (IJMDEM), Vol. 6, 2015, No.4, pp. 23–39, doi: 10.4018/ijmdem.2015100102.

[19] Chen, C.—Liu, M.—Zhang, B.—Han, J.—Jiang, J.—Liu, H.: 3D Action Recognition Using Multi-Temporal Depth Motion Maps and Fisher Vector. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16), New York, July 2016, pp. 3331–3337.

[20] Luo, J.—Wang, W.—Qi, H.: Spatio-Temporal Feature Extraction and Representation for RGB-D Human Action Recognition. Pattern Recognition Letters, Vol. 50, 2014, No. 3, pp. 139–148, doi: 10.1016/j.patrec.2014.03.024.

[21] Huynh, T.—Min, R.—Dugelay, J. L.: An Efficient LBP-Based Descriptor for Facial Depth Images Applied to Gender Recognition Using RGB-D Face Data. In: Park, J. I., Kim, J. (Eds.): Computer Vision – ACCV 2012 Workshops. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7728, 2012, pp. 133–145, doi: 10.1007/978-3-642-37410-4_12.

[22] Lu, J.—Liong, V. E.—Zhou, X.—Zhou, J.: Learning Compact Binary Face Descriptor for Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, 2015, No. 10, pp. 2041–2056, doi: 10.1109/tpami.2015.2408359.

[23] Ni, B.—Pei, Y.—Moulin, P.—Yan, S.: Multilevel Depth and Image Fusion for Human Activity Detection. IEEE Transactions on Cybernetics, Vol. 43, 2013, No. 5, pp. 1383–1394, doi: 10.1109/tcyb.2013.2276433.

[24] Zhu, Y.—Chen, W.—Guo, G.: Evaluating Spatiotemporal Interest Point Features for Depth-Based Action Recognition. Image and Vision Computing, Vol. 32, 2014, No. 8, pp. 453–464, doi: 10.1016/j.imavis.2014.04.005.

[25] Zhu, Y.—Chen, W.—Guo, G.: Fusing Multiple Features for Depth-Based Action Recognition. ACM Transactions on Intelligent Systems and Technology (TIST), Vol. 6, 2015, No. 2, Art. No. 18, doi: 10.1145/2629483.

[26] Yu, M.—Liu, L.—Shao, L.: Structure-Preserving Binary Representations for RGB-D Action Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 38, 2016, No. 8, pp. 1651–1664, doi: 10.1109/tpami.2015.2491925.

[27] Hu, J.-F.—Zheng, W.-S.—Lai, J.—Zhang, J.: Jointly Learning Heterogeneous Features for RGB-D Activity Recognition. Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15), Boston, June 2015, pp. 5344–5352.

[28] Kong, Y.—Fu, Y.: Discriminative Relational Representation Learning for RGB-D Action Recognition. IEEE Transactions on Image Processing, Vol. 25, 2016, No. 6, pp. 2856–2865, doi: 10.1109/tip.2016.2556940.

[29] Liu, L.—Shao L.: Learning Discriminative Representations from RGB-D Video Data. Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (AAAI'13), Bellevue, July 2013, pp. 1493–1500.

[30] Wang, A.—Lu, J.—Cai, J.—Cham, T.-J.—Wang, G.: Large-Margin Multi-Modal Deep Learning for RGB-D Object Recognition. IEEE Transactions on Multimedia, Vol. 17, 2015, No. 11, pp. 1887–1898, doi: 10.1109/tmm.2015.2476655.

[31] Jia, C.—Kong, Y.—Ding, Z.—Fu, Y. R.: Latent Tensor Transfer Learning for RGB-D Action Recognition. Proceedings of the 22nd ACM International Conference on Multimedia (MM'14), Orlando, November 2014, pp. 87–96, doi: 10.1145/2647868.2654928.

[32] Zhang, H.—Parker, L. E.: CoDe4D: Color-Depth Local Spatio-Temporal Features for Human Activity Recognition from RGB-D Videos. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 26, 2016, No. 3, pp. 541–555, doi: 10.1109/tcsvt.2014.2376139.

[33] Fan, R.-E.—Chang, K.-W—Hsieh, C.-J.—Wang, X.-R.—Lin, C.-J.: LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research, Vol. 9, 2008, No. 9, pp. 1871–1874.

[34] Wang, H.—Kläser, A.—Schmid, C.—Liu, C.-L.: Dense Trajectories and Motion Boundary Descriptors for Action Recognition. International Journal of Computer Vision, Vol. 103, 2013, No. 1, pp. 60–79, doi: 10.1007/s11263-012-0594-8.

[35] Peng, X. J.—Wang, L. M.—Wang, X.—Qiao, Y.: Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice. Computer Vision and Image Understanding, Vol. 150, 2016, No. 3, pp. 109–125, doi: 10.1016/j.cviu.2016.03.013.

[36] Kong, Y.—Fu, Y.: Bilinear Heterogeneous Information Machine for RGB-D Action Recognition. Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15), Boston, June 2015, pp. 1054–1062, doi: 10.1109/cvpr.2015.7298708.

[37] Karpathy, A.—Toderici, G.—Shetty, S.—Leung, T.—Sukthankar, R.—Li, F.-F.: Large-Scale Video Classification with Convolutional Neural Networks. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14), Columbus, June 2014, pp. 1752–1732, doi: 10.1109/cvpr.2014.223.

[38] Simonyan, K.—Zisserman, A.: Two-Stream Convolutional Networks for Action Recognition in Videos. Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14), Montreal, December 2014, pp. 568–576.

[39] Tran, D.—Bourdev, L.—Fergus, R.—Torresani, L.—Paluri, M.: Learning Spatiotemporal Features with 3D Convolutional Networks. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV'15), Santiago, December 2015, pp. 4489–4497, doi: 10.1109/iccv.2015.510.

[40] Wang, L. M.—Xiong, Y. J.—Wang, Z.—Qiao, Y.—Lin, D. H.—Tang, X. O.—Van Gool, L.: Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. Proceedings of the 14th European Conference on Computer Vision (ECCV'16), Amsterdam, October 2016, pp. 20–36, doi: 10.1007/978-3-319-46484-8_2.

[41] Shahroudy, A.—Ng, T. T.—Gong, Y. H.—Wang, G.: Deep Multimodal Feature Analysis for Action Recognition in RGB+D Videos. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, 2018, No. 5, pp. 1045–1058, doi: 10.1109/tpami.2017.2691321.

Zhengyuan Zhai is working toward his Ph.D. degree in the Beijing Key Laboratory of Work Safety Intelligent Monitoring at the Department of Electronic Engineering, Beijing University of Posts and Telecommunications. His research interests are computer vision, human activity recognition, machine learning, deep learning and privacy-preserving. He has some papers in international conferences and journals about these areas.

Chunxiao Fan is currently Professor and the Director of Center for Information Electronic and Intelligence System. She served as a member of ISO/IEC JTC1/SC6 WG9, ASN.1 (since 2006) and Chinese Sensor network working group. She also was elevated to evaluation expert of Beijing Scientific and Technical Academy Awards. Her research interests include heterogeneous media data analysis, internet of things, data mining, communication software, and so on. In recent years, she has been Director of several Nation Science Foundation Projects. She published more than 30 papers in international journals and conferences, authored and edited three books and authorized several patents for inventions.

Yue Ming received her B.Sc. degree in communication engineering, her M.Sc. degree in human-computer interaction engineering, and the Ph.D. degree in signal and information processing from Beijing Jiaotong University, China, in 2006, 2008, and 2013, respectively. She worked as a visiting scholar in Carnegie Mellon University, U.S., between 2010 and 2011. Since 2013, she has been working as a faculty member at Beijing University of Posts and Telecommunications. Her research interests are in the areas of biometrics, computer vision, computer graphics, information retrieval, pattern recognition, etc. She has authored more than 40 scientific papers in these areas.

Computing and Informatics, Vol. 37, 2018, 1363–1385, doi: 10.4149/cai_2018_6_1363

EXPLOITING THE USE OF COOPERATION IN SELF-ORGANIZING RELIABLE MULTIAGENT SYSTEMS

Sebnem Bora

Department of Computer Engineering
Ege University
35100 Izmir, Turkey
e-mail: sebnem.bora@ege.edu.tr

Abstract. In this paper, a novel and cooperative approach is exploited introducing a self-organizing engine to achieve high reliability and availability in multiagent systems. The Adaptive Multiagent Systems theory is applied to design adaptive groups of agents in order to build reliable multiagent systems. According to this theory, adaptiveness is achieved via the cooperative behaviors of agents and their ability to change the communication links autonomously. In this approach, there is not a centralized control mechanism in the multiagent system and there is no need of global knowledge of the system to achieve reliability. This approach was implemented to demonstrate its performance gain in a set of experiments performed under different operating conditions. The experimental results illustrate the effectiveness of this approach.

Keywords: Adaptive systems, availability, autonomous agents, redundancy, software reliability

Mathematics Subject Classification 2010: 68T42, 68M15

1 INTRODUCTION

Today, technology has become an integral part in the majority of our lives. From smart phones, to laptops to tablets – we are heavily connected to networks and systems; computers perform critical tasks in many areas of our lives every millisecond.

High reliability and availability are of the utmost importance to the majority of these computer systems. For example, such systems include nuclear reactor facilities, global air traffic control systems, and banking systems.

In order to build reliable systems, the programming is quite complex since such systems are usually software intensive. A promising approach to deal with complexity is to provide robustness, autonomy, and adaptation to a system by applying self-organizing algorithms. These algorithms are defined based on self-organization mechanisms inspired by nature. Ant and bee swarms, flocks of birds, and school of fish, the human immune system are typical examples of natural systems that exhibit properties inherent to self-organization. Swarms provide inspiration for mobile networks systems management [1], such as load-balancing and routing [2]. Solutions for the distribution of tasks to the available nodes and distribution of data between nodes in computational grids vary from techniques inspired by bee foraging behavior [3] to select the algorithms for executing small pieces of data, to business and market inspired techniques for dynamically changing the task assignment [4, 5]. The use of mobile agents makes us aware of network conditions. Mobile agents [6] can be supportive for intrusion detection and intrusion response in large-scale network infrastructures by following the behavior of the human immune system [7] and the ant foraging behavior [8]. Insect colonies based models are exploited in agent-based software for manufacturing control [9, 10]. PROSA is a representative example of the ant-like approach, where agents mimic the ants' behaviors [11, 12].

Reliability and availability are achieved via redundancy, i.e., duplication of critical functions exists in the system so that application software can reconfigure and maintain (continue to perform) their tasks in the presence of faults in the system. One technique that creates redundancy in a multiagent system (MAS) is adding extra computers or agents. In order to improve reliability and ensure the availability of the MAS organization, critical agents for the system's operation are replicated into groups. This paper presents a novel replication approach that includes resilience, self-organization, and adaptation as its primary properties. It employs a technique grounded in the Adaptive Multiagent Systems (AMAS) theory; therefore, replicating agents into groups and sharing limited resources among them are achieved without a central control or global knowledge about the MAS organization. These properties result from some simple behaviors of agents. The AMAS theory conceptualizes the design of an adaptive multiagent system and provides agents with adaptive capabilities [13]. According to this theory, adaptiveness is based on cooperative behaviors meaning that an agent seeks to help the most troubled agent in the system while achieving its goal. When agents encounter similar problems simultaneously, the program computes a degree of criticality in order to understand potential impacts of these problems on achieving its goal. Considering this criticality, the agent can determine what the most cooperative action should be taken.

The AMAS theory has been applied in various application domains such as dynamic ontologies [14], aircraft design [15], simulation of the functional behavior of a yeast cell [16], crisis management [17], bioprocesses control [18], ambient systems [19], product design [20], maritime surveillance [21, 22], and self-organizing

biological neural networks [23, 24, 25]. In this study, the AMAS theory is applied to the adaptive replication to effectively improve reliability in MAS organizations. Applying the AMAS theory to the model, in conjuncture with the software, creates a more reliable and adaptive MAS distribution because the system is lacking a centralized control. A key element that sets this study apart from previous research is that the adapting replica groups are achieved in a self-organized way without a centralized control. Agents applying this approach use local information and behave in cooperative ways to help the troubled neighbors. In previous studies, agents were usually associated with an adaptive replication manager which dynamically computed the criticalities of agents [26, 27, 28, 29, 30, 31, 32, 33, 34, 35]. The quantities that define the criticalities of agents are used for calculations to share the limited resources between the replica groups. The adaptive replication manager uses an observation service to collect global data about agents in the organization in order to calculate agents' criticalities.

In this study, the AMAS theory is adopted in the reliable MAS where each agent observes whether there is an increase or a decrease in its criticality. If there is an increase, it decides to increase the number of its replicas. However, during the decision process, the leader exploits only its local information when making a decision about the change in the agent's criticality.

This research thoroughly explains how the AMAS theory is applied to MAS in order to enhance reliability and availability. This approach is the first one that uses the AMAS theory for managing adaptive replication (not only in multiagent systems, but in general). A case study chosen to illustrate the AMAS theory is exemplified in the design of a library system which includes cooperative agents with local goals and partial views of their environment. In our conclusion some experimental results are presented to exhibit the effectiveness of this approach. The remaining sections of this paper are organized as follows. Section 2 explores related works; Section 3 provides background information detailing the AMAS theory; Section 4 introduces the agent-based model; Section 5 illustrates the experimental model developed for the study, presents data and provides the analysis and discusses the approach; and Section 6 summarizes the study.

2 RELATED WORK

There is a large number of multiagent platforms; however just a few offer reliability. Those multiagent platforms provide useful solutions of the problem of reliability in MAS. Since the approach presented in this study is a replication-based approach, scholarly articles and case studies associated with this topic published in computer science literature have been investigated.

In order to increase reliability in MAS, Fedoruk and Deters implemented transparent replication via proxies [36]. Their approach is a static replication approach which disregards the idea of changing replication techniques at run time. Thus, replication is only realized by a programmer before an application starts.

Guessoum et al. presented an adaptive multiagent architecture that was implemented with the DIMA [26] platform and DarX middleware [27, 28]. In DarX, software components can either be replicated or not, and it is possible to change the replication strategy at run time. DarX middleware must be integrated into any multiagent organization in order to improve reliability in the organization.

DimaX [29] is a fault-tolerant multiagent development platform that is the integration of DarX into DIMA. DimaX is founded on system level (DarX middleware); application level (agents); and monitoring level. In DimaX, the criticality of agents is calculated by using an interdependence graph. At the monitoring level, the control of replication is monitored by using an observation service [30, 31, 32].

Bora and Dikenelli proposed an approach which provided flexibility to multiagent organizations in terms of fault tolerance because the fault tolerance policies were implemented as reusable plan structures. Thus, whenever an agent needed to be made fault-tolerant, the action was performed by sending a request to that agent [33]. Bora and Dikenelli introduced a self-adaptive replication approach to exploit a feedback control loop and a proportional controller within a replication infrastructure [34, 35]. This approach was used to examine the criticality of specific agents. Moreover, Bora and Dikenelli presented a replication approach based on role concept for multiagent systems. They defined a "fault tolerant" role that is responsible for replicating instances of critical roles, coordination between critical role instances and satisfying all replication-based fault tolerance requirements [37].

In this study, the AMAS theory is adopted in the reliable MAS where each leader agent observes whether there is an increase or a decrease in its criticality. The leader exploits only its local information when it decides to change the agent's criticality. This approach is the first one that uses the AMAS theory for managing adaptive replication (not only in multiagent systems but in general). Further, the self-organizing replication approach in this study provided higher performance when compared to other self-adaptive replication approaches [30, 32, 35, 28], replica groups' monitoring costs not present in this state of the art increased due to the observation mechanisms and the need of global information when the self-adaptive replication approach was applied.

3 AMAS THEORY

The Adaptive Multi-Agent System (AMAS) theory was developed to act as the engine for any system to self-adapt itself to any changes encountered in a dynamic environment. It explains the cooperative relationship between the system's internal operations and its functional adequacy, thus ensuring that the cooperative system carries out the appropriate task it was designed for. It was proved that there is at least one cooperative internal medium system that accomplishes an equivalent function of any functionally adequate system in the same environment. In a cooperative internal medium system, the components of a system are always collaborating. This secures a symbiotic relationship and provides protection from Non Cooperative

Situations (NCS) which may be harmful for their cooperative situations. In regard to MAS, NCS represent situations that are against the cooperative social behaviors of an agent; thus, when identifying an NCS, an agent alters its relationships with other agents in order to return to a cooperative state [13].

If a multiagent system adopts AMAS, an agent can have two types of behavior:

1. nominal behavior; and
2. cooperative behavior,

which is divided into tuning, reorganization, and evolution. An agent with the cooperative behavior attempts to assist other agents; consequently, this agent has to detect and repair the NCS and avoid producing a new NCS [16].

A multiagent system applying AMAS includes agents that try to decrease the criticality of troubled agents by exhibiting local and cooperative behaviors. These behaviors maintain the system's ability to produce an adequate global function. Each agent calculates its own criticality, a function that calculates whether a signal is activated for a NCS or not by examining the criticality value and a threshold [22]. If the agent decides that a NCS has occurred, it exploits the situation by adapting itself to this new situation triggering tuning and reorganization behaviors. Tuning behaviors modify the parameters computed by nominal behaviors. If it fails to solve the NCS, it triggers the reorganization behavior by sending messages (feedback) and/or it informs the cooperative agents that, in turn, eliminate the NCS and transmit feedback.

Reorganization behaviors modify the agent's interaction with its environment or with other agents. If an agent cannot execute its nominal behavior because of NCS, experiences a failed tuning behavior and/or sends requests that cannot be served, then it may create a new communication link with a new agent. If the reorganization behaviors fail to solve the NCS, the evolution behaviors resolve the NCS by creating new agents or removing agents [16, 24].

While applying the AMAS theory, a programmer defines the agents of the system, the agents' nominal behaviors, any NCS the agents may encounter, and cooperative behaviors to overcome each NCS. In the following section, the nominal behaviors of reliable agents are explained in detail.

4 SELF-ORGANIZING RELIABLE AGENTS

The self-organizing reliable MAS consists of three variables:

1. the domain agents (Agent-0 to Agent-N) and their replicas;
2. the environment agents and their replicas; and
3. the environment.

Agents and their replicas run on computers and are grouped according to either active or passive replication approaches [38, 39]. The methods of building consistency in groups vary in active or passive replication approaches; nevertheless, in

both approaches, one replica is designated as the leader and is responsible for providing responses. If a leader fails, any replica can become a leader. Domain agents (Agent-0 to Agent-N) are leaders of their groups.

In addition to replication techniques, the replication degree, which provides the level of redundancy and assigns the number of replicas within a group, is essential in achieving redundancy in MAS while using replication policies. In static replication, both the replication degree and approach are set by a programmer during initialization. In adaptive replication, the group leader primarily decides on its replication degree based on system resources and its criticality. The agent's criticality is a numerical quantity that indicates its importance in the system. However, there is not just one general description and/or definition attached to an agent's criticality. It is the programmer who decides, after careful consideration, the agent's varying properties. In general, this process incorporates an observation service that transparently monitors the agents' behaviors as well as the availability of resources, and adaptively reconfigures the system according to the agents' criticalities according to data evaluation collected by the observation service [26, 27, 28, 29, 30, 31, 32, 33, 34, 35].

According to this design, when the leader performs a self-organizing replication, the group leader will either increase and/or decrease the replication degree without a central control (without an observation service or a replication manager) based on global knowledge of the environment. The leader collects data related to its communication activities and calculates the change in the value of its criticality. The value of an agent's criticality is determined by a quantity that indicates whether the importance of an agent to the system increases or not. If the leader agent's criticality increases, a request message is transmitted to the environment agent which provides resources (hosts in the environment) to agents for replication. Next, a new replica of that particular agent is created. On the other hand, if there is a gradual decrease in the agent's criticality, the replica must be removed after a certain period.

4.1 Nominal Behaviors of Reliable Agents

In this section, reliable agents' nominal behaviors are briefly described. These behaviors are implemented to form the infrastructure for achieving reliability and availability in a goal-oriented MAS architecture [40].

In a reliable self-organizing MAS, each agent performs its actions without a global function (goal) for the system; hence, a dependable (reliable and available) multiagent system emerges. Each reliable agent achieves specific goals, such as Group Communication, Failure Detection, Election of a New Leader, Recovery from Failure, and Adaptive Replication (shown in Figure 1), while being cooperative. In order to achieve these identified goals, their plans and the reusable services are realized. These services and plans are described in more detail in [41].

In order to improve replication in a self-organizing manner, the replication plan was modified and Self-Organizing Replication, Evolution, and Reorganization plans were created. To replicate new replicas, the leader executes the Replication plan

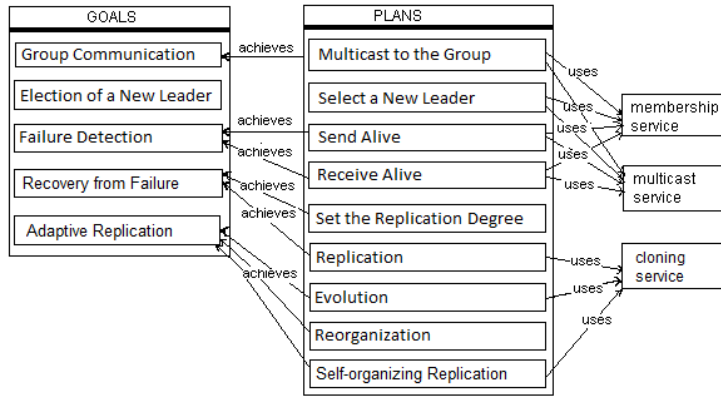


Figure 1. Goals and plans for the system's reliability

as many times as the number of replicas needed to be replicated, and asks the environment agent to locate a suitable host where new replicas can be placed.

For self-organizing replication, the critical agents must be identified. Each leader agent determines whether the change in the criticality occurs or not. The number of messages received by each agent, the degree of system's reliability, and the agent's role criticality are the example of certain metrics needed to calculate the agents' criticalities. In order to achieve the Adaptive Replication goal, the Self-Organizing Replication plan is executed.

4.2 Applying the AMAS Approach for Reliability

In order to achieve reliability in MAS, critical agents periodically monitor their criticalities. This data indicates the potential impact of the agent's failure in the MAS. We must consider two cases to ensure reliability in multiagent systems. Firstly, multiagent systems must have static organization structures so that critical agents can be identified and replication performed by the programmer before run time.

Secondly, if the agent's criticality cannot be determined before run time due to the multiagent systems' dynamic organization structures, critical agents can be dynamically evaluated at run time. In addition, metrics can be used for dynamically estimating and updating agents' criticalities in the MAS. Within the MAS organization, a role is defined as an abstract characterization of social agent's behavior in a specific domain and has a different operational impact in the MAS organization. The concept of a role represents the importance of an agent in an organization, and its dependencies from other agents, i.e., those affecting their criticalities in the multiagent organization.

The agent's dependency on another specific agent is yet another metric that indicates the agent's criticality. If for some reason, a critical agent that relies on an-

other agent fails, then goals will be difficult to achieve. The dependency on an agent is obtained from the number and performatives of the messages received. Messages, which are considered in terms of an agent's criticality in this work, contain performatives such as request, request-whenever, query-if, query-ref, and subscribe [37].

In this research, the change in the value of an agent's criticality is calculated by using precise formulas. If there is an increase in the value of agent's criticality, it signals the agent has become more critical than before and needs to have new replicas. The change in the agent's criticality in the current period, $\Delta Critic(t)$, is given below:

$$\Delta Critic(t) = Activity(t) - Activity(t-1) + C_{reliable}(t). \tag{1}$$

- $C_{reliable}(t)$: Is the contribution of the reliability of the system to the criticality in the current period. The $C_{reliable}$ is the difference between the number of failures in a group and half the number of replicas in the group in terms of a time interval. If the value of $C_{reliable}$ is larger than 0, then the contribution of the $C_{reliable}$ to the criticality is added. If the number of failures for a replica group is less than half the number of replicas, there is no point worrying about the system's reliability. In this case, the contribution of the reliability of the system is set to zero.
- $Activity(t)$: Is the degree of an agent's activity in the current period. $Activity(t-1)$ is determined for the last period and stored in the data structure. The value of the agent's role criticality and the ratio of the change in the number of messages sent to an agent for a specific role relative to the average number of messages over a number of periods are used for calculations of $Activity(t)$ as in the following equation.

$$Activity(t) = a * role(t) + b * Ratio\_req(t) \tag{2}$$

- a, b: Coefficients for contributions of the weight of the role and the average number of requests to the $Activity(t)$.
- $role(t)$: The value corresponding to the weight of the role in the current period. The weights of the roles are explicitly defined in the role ontology before the program starts.
- $Ratio\_req(t)$: The ratio of the change in the number of messages sent to an agent for a specific role to the average number of messages over a number of periods.

If $\Delta Critic(t) < 0$, it specifies the agent's criticality has decreased; however, if $\Delta Critic(t) > 0$, it signals the agent's criticality has increased. Next, the number of replicas of the group must be increased by 1. The agent asks the environment agent to provide a resource to create a new replica.

Following the calculations that determine the agent's criticality value, the leader of a replica group executes a *Self-Organizing Replication* plan in which a self-organizing replication is performed. The *Self-Organizing Replication* plan is given in Figure 2.

Algorithm 1 Self-organizing Replication

1. Calculate the Change in the Criticality, ΔCritic
2. **if** ΔCritic<0 $freq_{dec} \leftarrow freq_{dec} + 1$ **endif**
3. **if** ΔCritic>0 send("request","need a new replica", Environment Agent)
4. **if** Environment agent cannot provide any resource
5. $freq_a \leftarrow freq_a + 1$ and
6. **if** $freq_a > thresh_a$ send("request","need a new replica", Neighbour Agents)
7. **else** FILE← Agent's internal state as a serialized object and
8. Send FILE via RMI to the Environment Agent
9. $freq_a \leftarrow 0$
10. **endif**
11. **endif**

Figure 2. Algorithm for self-organizing replication

The first task of the *Self-Organizing Replication* plan is to determine the change in the agent's criticality, $\Delta Critic(t)$, by using the number of requests received over several periods, the number of failed replicas, and the weight of the role the agent has played (line 1 in Algorithm 1 in Figure 2).

If $\Delta Critic(t)$ is a positive integer, then a message containing a copy request is prepared and sent to the environment agent by using the send method (line 3 in Algorithm 1 in Figure 2). When the environment agent receives this message, it attempts to find a suitable host where new replicas will be placed. After an available host's address is obtained, the environment agent sends a request message to the agent to be replicated in order to convey the agent's internal state. As previously stated, the agent's internal state is serialized and written to a text file (line 7 in Algorithm 1 in Figure 2). Several Remote Method Invocation (RMI) messages are then sent to the environment agent to transfer both the agent's knowledge and internal state (line 8 in Algorithm 1). Thus, both sets of data are needed to commence the replication process.

The remote host's cloning server provides storage for the replicated agent. Multiple messages using RMI are sent to the cloning server by the environment agent to transfer the agent's knowledge and internal state received from the critical agent to be replicated. Upon receiving the RMI messages from the environment agent, the cloning server creates a new replica by using the contents of RMI messages. Next, the cloning server places the unserialized agent's state, libraries and source code to the selected paths and executes the agent's source code.

When the new replica is launched, it has identical state as the leader and contains the last view of the group. It then multicasts a JOIN message informing the other replicas that it has joined the group; subsequently, the other replicas register the new replica to their membership lists.

When $\Delta Critic(t)$ is a negative integer, it demonstrates the agent's criticality has decreased in the current period. If this occurs, the $freq_{dec}$ variable is increased by 1 so that it is possible to determine the number of periods in which the agent has experienced low levels of criticality (line 2 in Algorithm 1).

4.3 Identification of Non-Cooperative Situations

The proposed self-organizing replication architecture, in which replica agents can either be inserted or removed, is subject to NCS. Each NCS is identified by analyzing problematic stages of reliable multiagent systems.

Two kinds of NCS are identified in this work, the first being Unable to create a new replica. It occurs when the environment does not provide a resource to an agent that needs to replicate a new replica. The second NCS is Bad Message Density.

4.3.1 NCS: Unable to Create Replica

If an environment agent does not provide a resource to an $Agent_i$ that needs to replicate, it will send a message informing that there are no available resources. If an agent receives this message, it increases $freq_a$, which is termed a complain variable (see line 5 in Algorithm 1).

During the next period, the agent's criticality might increase. If so, it will then send the environment agent a new request for creating a new replica. The environment agent must find available resources for replication. If it fails, it informs the agent by sending a message containing *No available resources*. In this case, the Unable to create a replica NCS occurs and $freq_a$ is increased by 1 (line 5 in Algorithm 1). Concurrently, the environment agent waits to receive a message containing *Available resource:IP*. When it receives the message, it is able to provide a resource to $Agent_i$ and initialize the *Create a Replica* plan [41] in order to replicate a new replica of $Agent_i$.

4.3.2 Bad Message Density

The number of requests received by an agent exceeds a threshold value.

4.4 Cooperative Behaviours

In AMAS theory, an agent can exhibit two types of behaviour, nominal and cooperative. An agent's nominal behaviours in a reliable MAS are explained in Section 3.

As aforementioned, the cooperative behaviours are categorized into tuning, reorganization, and evolution.

One would presume an autonomous and cooperative agent is able to modify its replicas by adjusting its internal parameters by adopting a tuning behaviour. However, the tuning behaviour is not considered in this work since the consistency between replicas is one of the main issues in reliable systems. If the internal parameters of agent replicas are modified by triggering a tuning behaviour to overcome any NCSs, consistency between replicas cannot be continuously maintained. Since the tuning behaviour is not adopted in this approach, a cooperative agent adopts a reorganization behaviour in which it tries to change the way it interacts with others. The reorganization behaviours of reliable agents are implemented using the *Reorganization* plan.

The last category behaviour that may be adopted by a reliable agent is the evolution. In evolution behaviour, a replica can either be created or removed (e.g., if ineffective, it must leave the system). In these two last levels, the propagation of a problem to other agents is indeed possible if an agent is not able to execute its nominal behaviour.

The evolution behaviours of reliable agents are implemented using the *Evolution* plan (illustrated in Figure 3), which corresponds to the creation and/or removal of reliable agent's replicas. There is an underlying assumption that no replica can be created or removed simultaneously.

NCS is suppressed by executing the aforementioned plans as described in the following subsections. By executing two plans, an agent's criticality value will decrease, as these NCS increase the agent's criticality. The evolution behaviour of reliable agents will be explained first.

4.5 Suppression of "Unable to Create Replica" NCS

When an environment agent does not provide a resource to an $Agent_i$ and the complain variable $freq_a$ exceeds a predefined threshold $thresh_a$, the $Agent_i$ sends the feedback message *Need a new replica* to its neighbours. This message specifically requests an increase in the number of replicas (line 6 in Algorithm 1).

When an agent receives feedback from one or more neighbours (or from its environment), it may retro-propagate a feedback to its own neighbours. When the agent receives the *Need a new replica* message, the *Evolution* plan in Figure 3 is initialized. If the agent is a neighbour, it then stores the number of the *Need a new replica* messages (no_{fa_i}) in a data structure (line 1 in Algorithm 2 in Figure 3). If no_{fa_i} exceeds a certain value $thresh_c$ (line 2 in Algorithm 2), it tries to cooperate with the $Agent_i$.

If its criticality decreases over a certain number of periods (i.e. $freq_{dec}$ is larger than $thresh_{dec}$), it then kills one of its replicas (line 3 and 4 of the *Evolution* plan). After removing the replica, the other replicas in the group also delete this agent from their membership lists and it informs the environment agent (line 5 of Algorithm 2 in Figure 3). Upon receiving a new replica creation request, the environ-

Algorithm 2 Evolution

1. **if** receiving a message containing "need a new replica" $no_{fai} \leftarrow no_{fai} + 1$
2. **if** $no_{fai} > thresh_c$
3. **if** $freq_{dec} > thresh_{dec}$ and the number of replicas>1
4. delete one of the replicas and
5. send ("inform","available resource:IP",Environment Agent)
6. **else** send("request","need a new replica", Neighbour Agents) **endif**
7. **endif**
8. **endif**

Figure 3. Algorithm for the evolution behaviour

ment agent will then create a new replica in the host that has been released by the neighbour's replica. In this case, the number of replicas of $Agent_i$ increases by 1, thus the $Agent_i$ stops complaining and sets the value $freq_a$ to zero (line 9 of Algorithm 1 in Figure 2). If there is no decrease in the neighbour's criticality value of $Agent_i$ (i.e., it is still complaining) during a certain number of periods, it then forwards the *Need a new replica* message to its neighbours (line 6 in Algorithm 2).

4.6 Suppression of "Bad Message Density" NCS

When the number of requests received by $Agent_i$ exceeds a threshold value $thresh_m$, the $Agent_i$ sends the querying agent a message informing that the message density is inadequate.

When the number of messages received by the querying agent is high, the primary action of the *Reorganization* plan is activated by identifying the provision as the sender agent (i.e. $Agent_i$). Whenever $Agent_n$ has received this message, the value of no_{fm} is increased by 1 (line 2 in Algorithm 3 in Figure 4). If the value of no_{fm} exceeds a threshold $thresh_{fm}$, $Agent_n$ searches for a new agent in order to send its queries. Therefore, it asks the Directory Facilitator (DF) to send $Agent_j$ so that it can provide the same service as $Agent_i$ for its queries (line 3 in Algorithm 3 in Figure 4). In order to prevent interaction with $Agent_i$, $Agent_i$ is removed from the knowledge base when DF sends the identifier of $Agent_j$ (line 4 and 5 in Algorithm 3).

Next, when a new agent's identifier has been received from the Directory Facilitator (DF), the new agent's identifier is included to the knowledge base of $Agent_n$ in order to contact with $Agent_j$. Afterwards, it will be possible for it to send its queries to that agent (line 6 in Algorithm 3 in Figure 4). In this case, both the

agent's activity and criticality value decrease.

Algorithm 3 Reorganization

```
1. if receiving a message containing "the number of messages is high"
2.     no_fm ← no_fm +1
3.     if no_fm >thresh_fm send ("request","an Available Agent ", Directory Facilitator)
4.         if Directory Facilitator sends the Agent Identifier of an Available Agent
5.             Remove the Sender Agent that complains and
6.             send requests to the Available Agent
7.         endif
8.     endif
9. endif
```

Figure 4. Algorithm for the reorganization behavior

5 EXPERIMENTS

In order to evaluate the self-organizing replication based on AMAS theory, a library system was designed that included two specific agents – library assistant agents and user agents. The latter were designed to query library assistant agents. Each library assistant agent manages a different library and stores the library knowledge (i.e. bibliographical information) using the library ontology. Instances of this ontology hold the properties of all periodicals and books; for example, the title, the author(s), the ISBN number, and keywords related to documents.

In the case study, each user agent interacted with a user when it received a book and/or periodical request and then forwarded the request directly to all of the library assistant agents. Next, the library assistant agent executed a single plan to match the request to the document's ontology instance(s) and responded with the sources of the bibliographical information contained in a message. When the user agent received the response from the library assistant agents, it then selected a library where the source was located and provided the result to the user. In this case study, the user agents were dependent upon the library assistant agents since the library assistant agent was a critical and reliable agent for the library system's operation.

The library system was implemented by using Semantic Web Enabled Multiagent System Development Framework (SEAGENT) [42] and Java Version 1.5.0. The tests were run on a computer with Intel Core i7 CPU and 64 GB of RAM.

5.1 Costs of Reliability

In a reliable multiagent system, there must be multiple replicas of an agent. According to the preferred replication approach, those replicas may run concurrently, possibly in different environments. The cost of reliability of a multiagent system using a replication approach is the sum of the cost of replica creation/deletion, replica usage, and overheads incurred by the coordination of its replicas. Moreover, a constant number of resources in a system are reserved to provide redundancy for the performance of a certain task. However, usage of a constant number of resources in a system can be expensive. Varying the replication degree in a multiagent system can decrease the cost caused by replication of critical agents of the system. Adaptive replication techniques enable a multiagent system to change its replication degree in accordance with its environment.

In order to evaluate the costs of applying a self-organizing replication, a test environment was implemented, which included library assistant agent leaders ranging from 10 to 60, plus their replicas, and a user agent. In the first case, the library assistant agent leaders applied a static replication technique in the system. Next, the programmer manually deployed replicas of the leader agents in the system. When the static replication technique was applied, the number of replicas increased, doubling the number of the leaders.

In the second case, the library assistant agent leaders applied the self-organizing replication in the system. The programmer deployed the library assistant agent leaders in the system whereby they were automatically and dynamically replicated in accordance with their criticalities at runtime. The highest number of possible replicas in the system was set to two times that of the deployed leaders. The time of the test's sampling period was set to 400. In this test, the threshold value $thresh_a$ was set to 2, the threshold value $thresh_c$ was set to 1, and the threshold value $thresh_{dec}$ was set to 2.

In the third case, in order to compare the systems' costs, whether in static or self-organizing replication techniques, the systems' response times were compared to the control (i.e. response time of a system without using a replication approach). The user agents sent their requests to the library assistant agents without applying any replication technique. In this test, it was observed if the effects of self organizing replication influenced the overall performance of the system.

In the first and second case, semi-active replication was employed in the library systems. In order to measure the cost of the self-organizing replication approach, the user agent sent queries to the leaders. The number of requests sent to the leaders changed the agent's criticality accordingly. However, in all cases, the total number of requests sent to the organization were equal in every step of the tests. Response times, for queries were measured in all cases. The response time is the time it takes a querying agent to receive the reply after sending its request to a leader agent.

5.1.1 Test Results

The results of tests are illustrated as graphs in Figure 5. When the test environment included library assistant agent leaders ranging from 10 to 30, the systems' response times were very close to each other since the number of agents was not large in the systems and the computer performed all operation very fast. As illustrated by the graphs, the slopes of the average response times of the systems applying the static, self-organizing replication techniques, and no replication increased with the number agents, as seen from Figure 5.

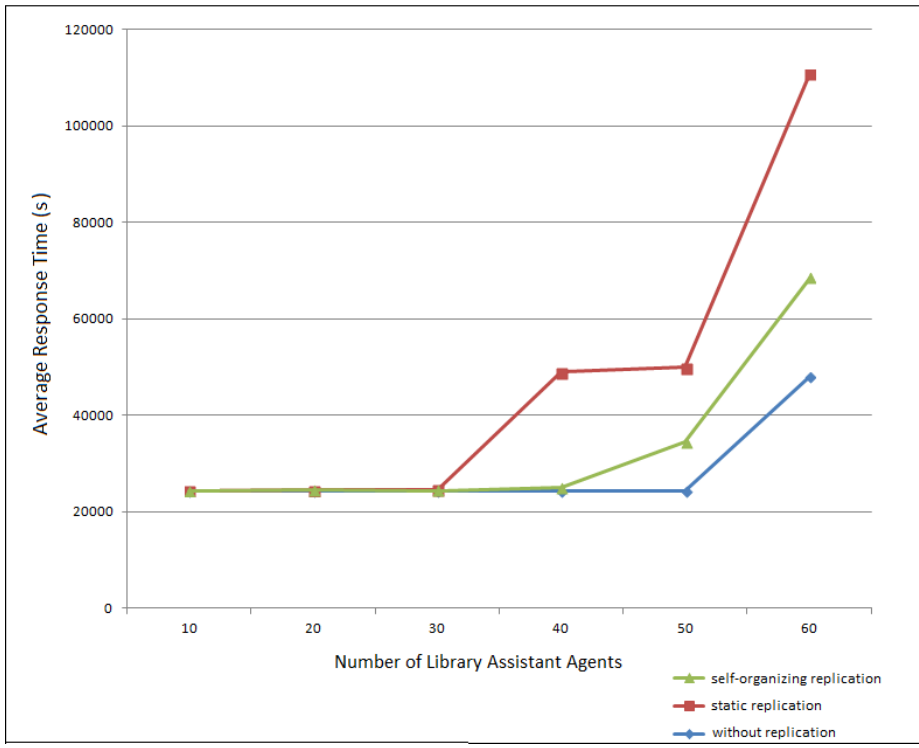


Figure 5. The effect of the self-organizing replication approach

The increase in response times was anticipated due to the fact that the leader of the group multicasts all incoming requests to the replicas; thus, the number of requests sent to the system increased. All replicas processed these requests. Finally, the number of messages exchanged also increased with the number of agents due to leader's multicasting of requests and heartbeat messages. Moreover, in SEAGENT, the communication module uses the RMI based communication infrastructure and all functionalities of internal architecture are based on threads. Therefore, when all agents were created in a single machine, then agents' threads were initialized and

the amount of time contributed by computation was slightly increased because of a computational load of the computer.

5.1.2 Discussion

According to the results, the self organizing replication approach is very promising and outperforms the static replication. As indicated in Figure 5, the system's response time, when the static replication is applied, is longer than the system's response time when the self-organizing replication is applied. Further, the system's response times, when the self-organizing replication is applied, are very close to the system's response time when no replication is applied. This result was expected, since the number of replicas in the system effects the system's response times, as mentioned in the previous sections.

When the self organizing replication technique is applied to the system, the number of replicas in each group may change with respect to the criticalities of the leader agents. The highest number of replicas in the system applying self organizing replication can equal the total number of replicas in the system applying static replication, for the same number of leaders. Sometimes, the agents may not receive any requests for a period of time or the number of requests they have received decreases. Thus, the degree of agents' activities will decrease as will the agents' criticalities during these periods. If a decrease in criticality is present for a certain period of time, the leader agent decreases the number of its replicas by removing useless replicas. In this case, the system's response time decreases as the number of replicas in the system decreases. If its criticality increases, it increases the number of its replicas in order to decrease its criticality. As the number of replicas in the system increases, the response time of the system also increases because the leader agent multicasts the received requests to its replicas and the replicas process the requests simultaneously.

However, when applying static replication, the programmer himself/herself decides on the number of replicas before executing the application. During execution, the number of replicas are fixed to a certain value in the organization ($2 \times$ number of leaders). In the system applying self organizing replication, the highest number of possible replicas can total the number of replicas in the system applying static replication. Although, the number of requests sent to the systems might be equal, the number of total replicas can change in accordance to the criticalities of the agents in self organizing replication. Therefore, the system's average response time when static replication is applied is longer than the system's average response time when self organizing replication is applied.

The most important advantage of the self organizing replication is the lack of centralized control. Indeed, the presence of a centralized control actually increases the system's response times since the centralized control mechanism needs to have global information of the system. Having the system's global information increases the number of messages sent and received; therefore, the response time also increases in the system applying a centralized self-adaptive reliable approach compared to the

system applying a static approach as explained in [35]. The agents in the system applying the self-organizing replication use the local information and behave in a cooperative manner in their system. Thus, the reliable self-organizing system takes advantage of applying the AMAS theory.

High reliability and availability are of the utmost importance to air traffic control (ATC) systems. The Advanced Automation System (AAS) [38] was a distributed real-time system that integrated all the services of the US air traffic control network. Critical services were replicated using either the active or passive approach, according to the application semantics and the hardware configuration. ATC systems are complex in the sense of complex systems. The control and knowledge are distributed in these systems and the flight data will be more complicated than the processed data at present to be processed by these systems. Therefore, adoption of new technologies and procedures must be taken into consideration when ATC systems are built. The self-organizing approach presented in this paper can be used in such systems in order to alleviate the effects of software failures and provide a great support to achieve high reliability and availability in ATC systems.

5.2 Evaluation of Suppression of "Bad Message Density" NCS

In order to observe whether a *Bad Message Density* NCS was suppressed or not, a user agent sent 80 queries to one of the library assistant agents with low processing capacity in each sampling period. Since the threshold $thresh_m$ was set to 25, the library assistant agent sent *The number of messages high* message to the user agent. The user agent receiving this message increased no_{fm} by 1. When the user agent detected that no_{fm} was larger than $thresh_{fm}$ (it was set to 1), it asked the DF agent to obtain a new library assistant agent's identifier that it could get the same service. The DF agent sent another library assistant agent's identifier to the user agent. The user agent removed the old library assistant agent's identifier from its knowledge base and added the new library assistant agent's identifier. As a result, the user sent its queries to the new agent. After receiving *The number of messages high* message, all operations were executed in a time frame ranging from 1 921–2 682 ms.

5.3 Evaluation of Robustness of the Self-Organizing Approach

A test bed consisting of five library assistant agent leaders and five user agents, and a failure simulator were designed and implemented in order to evaluate the robustness of the self-organizing replication approach. The failures ranging from 85 to 111 were injected into the system by a failure simulator which simply sent "kill" messages to the agents within a certain time frame. The agents receiving the "kill" messages stopped executing their threads. The majority of the kill messages were sent to the library assistant agents since they were the critical agents for the system's operation. A smaller number of kill messages were sent to the user agents. In this test, the number of available resources was set to 100; therefore, the maximum number of replicas in the whole multiagent system could be 100.

As shown in Figure 6, the number of failures injected to the multiagent system gradually increased, and the rate of success was observed and recorded. The rate of success is defined as the percentage of the replica groups that could accomplish their tasks in MAS within a certain time frame. The data in Figure 6 revealed the rate of success declined while the number of failures in multiagent system rose. This association was expected due to the fact that only 100 replicas existed in the multiagent system in order to tolerate failures.

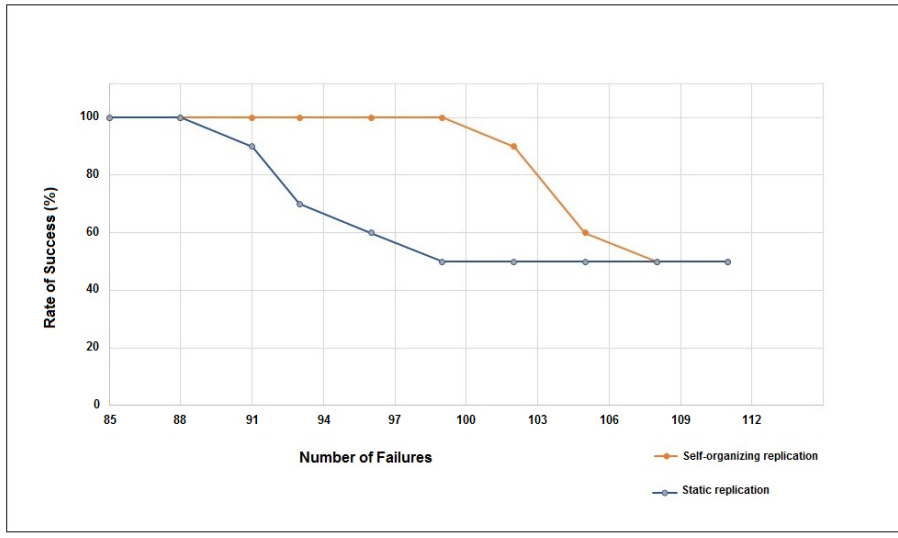


Figure 6. Robustness of the self-organizing replication approach

In the static replication approach, the replication degrees of the groups were fixed to certain numbers; therefore, the number of failures and agents' activities never changed the numbers of replicas in the groups. Since the majority of the failures influenced the library assistant agents, most of the library assistant agent groups failed to complete their tasks during these experiments; however, there existed the user agent groups in the multiagent system at the end of the experiments.

When the system employing self-organizing replication started, the replication manager determined the numbers of replicas of each library assistant agent leader and each user agent leader with respect to their criticalities. As the number of kill messages received by the replica groups linearly increased, each leader took into consideration the number of failures in order to calculate the change in the agent's criticality and it required a resource to create a new replica if the agent's criticality increased. Therefore, when the number of failures increased in the system, the self-organizing approach showed a stronger performance as compared to the static replication approach. From these experiments, it was observed that a replication approach required f+1 replica library assistant agents in order to survive up to

f library assistant agent crashes in the replica group. Moreover, the number of replicas in a multi-agent system should be at least equal to the number of library assistant agent leaders which were critical for the system's operation.

6 CONCLUSION

In this paper, a self-organizing approach based on AMAS theory was proposed for improving reliability in MAS. Based on observations, when the self organizing replication based on the AMAS theory was applied to the groups, the cost of reliability decreased. Therefore, this new approach to adaptive replication highly outperformed static replication.

In conclusion, the results indicated that the efficient replication was sustainable using this approach. Reliable systems can benefit from the self-organizing replication that utilizes resources more efficiently, and can also achieve greater flexibility. Some other future research opportunities can also be pursued to evaluate the performance of the various algorithms based on self-organization mechanisms inspired by nature. In the future, adaptive immune system concepts can be utilized to design and implement self-organizing reliable multiagent systems. Moreover, stigmergy and genetic algorithms can be used as the mechanisms to enhance reliability in multiagent systems. Especially, adaptive stigmergic mechanisms are good candidates for providing self-organization to reliable multiagent systems. Future work concerns a deeper analysis of those mechanisms, costruction of the reliable systems in different agent platforms, or implementation of a system that improves reliability in the cloud.

REFERENCES

[1] Brueckner, S. A.—Van Dyke Parunak, H.: Self-Organising MANET Management. In: Di Marzo Serugendo, G., Karageorgos, A., Rana, O. F., Zambonelli, F. (Eds.): Engineering Self-Organising Systems (ESOA 2003). Springer, Berlin, Heidelberg, Lecture Notes in Artificial Intelligence, Vol. 2977, 2004, pp. 20–35, doi: 10.1007/978-3-540-24701-2_2.

[2] Montresor, A.—Meling, H.—Babaoglu, O.: Messor: Load-Balancing Through a Swarm of Autonomous Agents. In: Moro, G., Koubarakis, M. (Eds.): Agents and Peer-to-Peer Computing (AP2PC 2002). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2530, 2003, pp. 125–137, doi: 10.1007/3-540-45074-2_12.

[3] Ko, S. Y.—Gupta, I.—Jo, Y.: Novel Mathematics-Inspired Algorithms for Self-Adaptive Peer-to-Peer Computing. Proceedings of the First International Conference on Self-Adaptive and Self-Organising Systems (SASO 2007), 2007, pp. 3–12, doi: 10.1109/SASO.2007.40.

[4] Li, Y.—Chen, F.-H.—Sun, X.—Zhou, M.-H.—Jiao, W.-P.—Cao, D.-G.—Mei, H.: Self-Adaptive Resource Management for Large-Scale Shared Clusters. Jour-

nal of Computer Science and Technology, Vol. 25, 2010, No. 5, pp. 945–957, doi: 10.1007/s11390-010-9379-0.

[5] Wrzesinska, G.—Maassen, J.—Bal, H. E.: Self-Adaptive Applications on the Grid. Proceedings of the 12th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '07), 2007, pp. 121–129, doi: 10.1145/1229428.1229449.

[6] Foukia, N.: IDReAM Intrusion Detection and Response Executed with Agent Mobility. Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS '05), Utrecht, The Netherlands, IEEE Press, New York, 2005, pp. 264–270, doi: 10.1145/1082473.1082513.

[7] Forrest, S.—Hofmeyr, S. A.—Somayaji, A.—Longstaff, T. A.: A Sense of Self for Unix Processes. Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, IEEE, 1996, pp. 120–128, doi: 10.1109/SECPRI.1996.502675.

[8] Holland, O.—Melhuish, C.: Stigmergy, Self-Organization, and Sorting in Collective Robotics. Artificial Life, Vol. 5, 1999, No. 2, pp. 173–202, doi: 10.1162/106454699568737.

[9] Buyurgan, N.—Meyyappan, L.—Saygin, C.—Dagli, C. H.: Real-Time Routing Selection for Automated Guided Vehicles in a Flexible Manufacturing System. International Journal of Manufacturing Technology Management, Vol. 18, 2007, No. 2, pp. 169–181, doi: 10.1108/17410380710722881.

[10] Clair, G.—Kaddoum, E.—Gleizes, M.-P.—Picard, G.: Self-Regulation in Self-Organising Multiagent Systems for Adaptive and Intelligent Manufacturing Control. Proceedings of the 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO '08), IEEE, 2008, pp. 107–116, doi: 10.1109/SASO.2008.19.

[11] Valckenaers, P.—Van Brussel, H.—Kollingbaum, M.—Bochmann, O.: Multi-Agent Coordination and Control Using Stigmergy Applied in Manufacturing Control. In: Luck, M., Mařík, V., Štěpánková, O., Trappl, R. (Eds.): Multi-Agent Systems and Applications (ACAI 2001). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2086, 2001, pp. 317–334, doi: 10.1007/3-540-47745-4_15.

[12] Van Brussel, H.—Wyns, J.—Valckenaers, P.—Bongaerts, L.—Peeters, P.: Reference Architecture for Holonic Manufacturing Systems: PROSA. Computers in Industry, Vol. 37, 1998, No. 3, pp. 255–274, doi: 10.1016/S0166-3615(98)00102-X.

[13] Capera, D.—George, J.-P.—Gleizes, M.-P.—Glize, P.: The AMAS Theory for Complex Problem Solving Based on Self-Organizing Cooperative Agents. Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises (WET ICE 2003), IEEE, 2003, pp. 383–388, doi: 10.1109/ENABL.2003.1231441.

[14] Ottens, K.—Gleizes, M.-P.—Glize, P.: A Multi-Agent System for Building Dynamic Ontologies. Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '07), ACM, 2007, Art. No. 227, doi: 10.1145/1329125.1329399.

[15] Combettes, S.—Sontheimer, T.—Rougemaille, S.—Glize, P.: Weight Optimization of Aircraft Harnesses. In: Demazeau, Y., Müller, J., Rodríguez, J., Pérez, J. (Eds.): Advances on Practical Applications of Agents and Multi-Agent Systems. Springer, Berlin, Heidelberg, Advances in Intelligent and Soft Computing, Vol. 155, 2012, pp. 229–232, doi: 10.1007/978-3-642-28786-2_26.

[16] Bernon, C.—Capera, D.—Mano, J.-P.: Engineering Self-Modeling Systems Application to Biology. In: Artikis, A., Picard, G., Vercouter, L. (Eds.): Engineering Societies in the Agents World IX (ESAW 2008). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5485, 2009, pp. 248–263, doi: 10.1007/978-3-642-02562-4_14.

[17] Lacouture, J.—Rodriguez, I.—Arcangeli, J.-P.—Chassot, C.—Desprats, T.—Drira, K.—Garijo, F.—Noel, V.—Sibilla, M.—Tessier, C.: Mission-Aware Adaptive Communication for Collaborative Mobile Entities. Handbook of Research on Mobility and Computing Evolving Technologies and Ubiquitous Impacts, 2011, pp. 1056–1076, doi: 10.4018/978-1-60960-042-6.ch064.

[18] Videau, S.—Bernon, C.—Glize, P.—Uribelarrea, J.-L.: Controlling Bioprocesses Using Cooperative Self-Organizing Agents. In: Demazeau, Y., Pěchoucěk, M., Corchado, J. M., Pérez, J. B. (Eds.): Advances on Practical Applications of Agents and Multiagent Systems. Springer, Berlin, Heidelberg, Advances in Intelligent and Soft Computing, Vol. 88, 2011, pp. 141–150, doi: 10.1007/978-3-642-19875-5_19.

[19] Guivarch, V.—Camps, V.—Péninou, A.: AMADEUS: An Adaptive Multi-Agent System to Learn a User's Recurring Actions in Ambient Systems. ADCAIJ Advances in Distributed Computing and Artificial Intelligence Journal, Vol. 1, 2013, No. 3, pp. 1–10.

[20] Kaddoum, E.—Georgé, J.-P.: Collective Self-Tuning for Complex Product Design. Proceedings of 2012 IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2012), IEEE, 2012, pp. 193–198, doi: 10.1109/SASO.2012.14.

[21] Brax, N.—Andono, E.—Gleizes, M.-P.: A Self-Adaptive Multi-Agent System for Abnormal Behavior Detection in Maritime Surveillance. In: Jezic, G., Kusek, M., Nguyen, N. T., Howlett, R. J., Jain, L. C. (Eds.): Agent and Multi-Agent Systems. Technologies and Applications (KES-AMSTA 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7327, 2012, pp. 174–185, doi: 10.1007/978-3-642-30947-2_21.

[22] Mano, J.-P.—Georgé, J.-P.—Gleizes, M.-P.: Adaptive Multi-Agent System for Multi-Sensor Maritime Surveillance. In: Demazeau, Y., Dignum, F., Corchado, J. M., Pérez, J. B. (Eds.): Advances in Practical Applications of Agents and Multiagent Systems (PAAMS 2010). Springer, Berlin, Heidelberg, Advances in Intelligent and Soft Computing, Vol. 70, 2010, pp. 285–290, doi: 10.1007/978-3-642-12384-9_34.

[23] Gürcan, Ö.: An Emergent Model for Mimicking Human Neuronal Pathways in Silico. Proceedings of the 12th European Conference on Artificial Life (ECAL 2013), MIT Press, 2013, pp. 1172–1173, doi: 10.7551/978-0-262-31709-2-ch180.

[24] Gürcan, Ö.—Bernon, C.—Türker, K. S.—Mano, J.-P.—Glize, P.—Dikenelli, O.: Simulating Human Single Motor Units Using Self-Organizing Agents. Proceedings of the 2012 IEEE Sixth International Conference on Self-

Adaptive and Self-Organizing Systems (SASO 2012), IEEE, 2012, pp. 11–20, doi: 10.1109/SASO.2012.18.

[25] Gürcan, Ö.—Türker, K. S.—Mano, J.-P.—Bernon, C.—Dikenelli, O.—Glize, P.: Mimicking Human Neuronal Pathways in Silico: An Emergent Model on the Efective Connectivity. Journal of Computational Neuroscience, Vol. 36, 2014, No. 2, pp. 235–257, doi: 10.1007/s10827-013-0467-3.

[26] Guessoum, Z.—Briot, J.–P.: From Active Objects to Autonomous Agents. IEEE Concurrency, Vol. 7, 1999, No. 3, pp. 68–76, doi: 10.1109/4434.788781.

[27] Guessoum, Z.—Briot, J. P.—Sens, P.—Marin, O.: Toward Fault-Tolerant Multi-Agent Systems. European Workshop on Modeling an Autonomous Agent in a Multi-Agent World (MAAMAW 2001), Annecy, France, 2001.

[28] Marin, O.—Bertier, M.—Sens, P.: DARX – A Framework for the Fault-Tolerant Support of Agent Software. 14th International Symposium on Software Reliability Engineering (ISSRE 2003), 2003, pp. 406–416, doi: 10.1109/ISSRE.2003.1251062.

[29] Faci, N.—Guessoum, Z.—Marin, O.: DimaX: A Fault Tolerant Multi-Agent Platform. Proceedings of the Fifth International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (ICSE '06, SELMAS '06), ACM, Shangai, China, pp. 13–20, 2006.

[30] Guessoum, Z.—Faci, N.—Briot, J.-P.: Adaptive Replication of Large-Scale Multi-Agent Systems – Towards a Fault-Tolerant Multi-Agent Platform. In: Garcia, A., Choren, R., Lucena, C., Giorgini, P., Holvoet, T., Romanovsky, A. (Eds.): Software Engineering for Multi-Agent Systems IV (SELMAS 2005). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3914, 2005, pp. 238–253, doi: 10.1007/11738817_15.

[31] Guessoum, Z.—Briot, J.-P.—Marin, O.—Hamel, A.—Sens, P.: Dynamic and Adaptive Replication for Large-Scale Reliable Multi-Agent Systems. In: Garcia, A., Lucena, C., Zambonelli, F., Omicini, A., Castro, J. (Eds.): Software Engineering for Large-Scale Multi-Agent Systems (SELMAS 2002). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2603, 2003, pp. 182–198, doi: 10.1007/3-540-35828-5_12.

[32] Guessoum, Z.—Ziane, M.—Faci, N.: Monitoring and Organizational-Level Adaptation of Multi-Agent Systems. Proceedings of the Third International Joint Conference on Autonomous Agents (AAMAS '04), Vol. 2, 2004, pp. 514–522.

[33] Bora, S.—Dikenelli, O.: Implementing a Multi Agent Organization That Changes Its Fault Tolerance Policy at Run-Time. In: Dikenelli, O., Gleizes, M. P., Ricci, A. (Eds.): Engineering Societies in the Agents World VI (ESAW 2005). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3963, 2006, pp. 153–167, doi: 10.1007/11759683_10.

[34] Bora, S.—Dikenelli, O.: Applying Feedback Control in Adaptive Replication in Fault Tolerant Multi-Agent Organizations. Proceedings of the Fifth International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (ICSE '06, SELMAS '06), ACM, Shangai, China, 2006, pp. 5–12, doi: 10.1145/1138063.1138066.

[35] Bora, S.—Dikenelli, O.: A Centralized Self-Adaptive Fault Tolerance Approach Based on Feedback Control for Multi-Agent Systems. Turkish Journal of Electrical

Engineering and Computer Sciences, Vol. 24, 2016, pp. 4707–4723, doi: 10.3906/elk-1405-58.

[36] Fedoruk, A.—Deters, R.: Improving Fault-Tolerance by Replicating Agents. Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy, 2002, pp. 737–744, doi: 10.1145/544862.544917.

[37] Bora, S.—Dikenelli, O.: Replication Based on Role Concept for Multi-Agent Systems. In: Aldewereld, H., Dignum, V., Picard, G. (Eds.): Engineering Societies in the Agents World X (ESAW '09). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5881, 2009, pp. 165–180, doi: 10.1007/978-3-642-10203-5_15.

[38] Cristian, F.—Dancey, B.—Dehn, J.: Fault-Tolerance in the Advanced Automation System. Proceedings of the 4th Workshop on ACM SIGOPS European Workshop (EW 4), ACM, 1990, pp. 6–17, doi: 10.1145/504136.504156.

[39] Elnozahy, E. N.—Zwaenepoel, W.: Replicated Distributed Processes in Manetho. Fault-Tolerant Computing. Digest of Papers, Twenty-Second International Symposium on FTCS-22, IEEE, 1992, pp. 18–27.

[40] Stollberg, M.—Rhomberg, F.: Survey on Goal-Driven Architectures. Technical Report DERI-TR-2006-06-04, DERI, Austria, 2006.

[41] Bora, S.: Implementing Fault-Tolerant Services in Goal-Oriented Multiagent Systems. Advances in Electrical and Computer Engineering, Vol. 14, 2014, No. 3, pp. 113–122, doi: 10.4316/AECE.2014.03015.

[42] Dikenelli, O.—Erdur, R. C.—Gumus, O. et al.: SEAGENT: A Platform for Developing Semantic Web Based Multi Agent Systems. Fourth International Joint Conference on Autonomous Agents (AAMAS '05), 2005, pp. 1271–1272, doi: 10.1145/1082473.1082728.

Sebnem Bora received her Ph.D. degree from Ege University, Turkey in 2006. She is currently Assistant Professor in the Computer Engineering Department at Ege University. Her research interests include dependable computing, self-adaptive systems, and agent-based modeling and simulation.

Computing and Informatics, Vol. 37, 2018, 1386–1410, doi: 10.4149/cai_2018_6_1386

A COOPERATIVE LOCAL SEARCH METHOD FOR SOLVING THE TRAVELING TOURNAMENT PROBLEM

Meriem Khelifa, Dalila Boughaci

LRIA-FEI-USTHB – Computer Science Department
BP 32 El-Alia Beb-Ezzouar, Algiers 16111, Algeria
e-mail: dboughaci@usthb.dz, khalifa.merieme.lmd@gmail.com

Abstract. Constrained optimization is the process of optimizing a certain objective function subject to a set of constraints. The goal is not necessarily to find the global optimum. We try to explore the search space more efficiently in order to find a good approximate solution. The obtained solution should verify the hard constraints that are required to be satisfied. In this paper, we propose a cooperative search method that handles optimality and feasibility separately. We take the traveling tournament problem (TTP) as a case study to show the applicability of the proposed idea. TTP is the problem of scheduling a double round-robin tournament that satisfies a set of related constraints and minimizes the total distance traveled by the teams. The proposed method for TTP consists of two main steps. In the first step, we ignore the optimization criterion. We reduce the search only to feasible solutions satisfying the problem's constraints. For this purpose, we use constraints programming model to ensure the feasibility of solutions. In the second step, we propose a stochastic local search method to handle the optimization criterion and find a good approximate solution that verifies the hard constraints. The overall method is evaluated on benchmarks and compared with other well-known techniques for TTP. The computational results are promising and show the effectiveness of the proposed idea for TTP.

Keywords: Sport scheduling, traveling tournament problem (TTP), optimization, constraints, search methods, stochastic local search method (SLS)

Mathematics Subject Classification 2010: 68xxx, 68Uxx, 90-08

1 INTRODUCTION

Constrained optimization involves minimizing or maximizing a certain objective function subject to a set of constraints. The main goal is to explore efficiently the search space in order to find a good approximate solution that verifies the hard constraints that are required to be satisfied. Constrained optimization problems are often difficult to solve, due to an eventual complex interaction between the goals of optimizing the objective function while satisfying the constraints.

Several methods have been proposed for solving constrained optimization problems. These methods include: the penalty function based method, the Lagrange multiplier method that can solve optimization problems with equality constraints, the augmented Lagrange multiplier for inequality constraints that combines the classical Lagrange method with the penalty function method, the quadratic programming methods (QP) that can solve optimization problems with a quadratic objective function and linear constraints, the gradient projection method for equality constraints and the gradient projection that can be extended to solve optimization problems with linear inequality constraints [26, 20].

We propose a cooperative search method for handling in two steps: *feasibility* and *optimality*. More precisely, the proposed search method consists of two main steps. In the first step, we search for feasible configurations satisfying the problem's constraints and ignore the optimization criterion. In the second step, we explore the feasible search space to handle the *optimization* problem and find the best solution. For this purpose, we propose to use constraints programming model, in the first step, to ensure the *feasibility* criterion of solutions. In the second step, we use a metaheuristic approach to handle the *optimization* criterion and find a good approximate solution that verifies the hard constraints.

To show the applicability of the proposed search method, we take as a case study the traveling tournament problem (TTP) which is a challenging sports scheduling problem [17, 24, 19]. The objective of the TTP is to find a double-round-robin tournament schedule that minimizes the total distance traveled by the teams and satisfies the related TTP constraints [11, 17, 24].

TTP is an interesting problem in both sports scheduling and combinatorial optimization. It is known to be an NP-hard problem which makes finding quality solutions in a short amount of time difficult [27]. TTP has attracted significant interest recently since a favorable TTP schedule can generate large incomes in the budget of managing the league's sport.

Several methods have been studied for the TTP. Among them, we cite: the branch and price method [16], the iterated local search method [7] that has been applied for a special case of TTP, so-called TTPPV (the traveling tournament problem with predefined venues). In [8], an integer programming formulation is proposed to the Max-MinTTP variant of TTP, in which the problem of minimizing the longest traveled distance is addressed. A hybrid approach combining a local search heuristic with an integer programming method was designed for TTP in [12]. Further, a simulated annealing algorithm that explores feasible and infeasible schedules us-

ing several structures of neighborhoods and compound movements is studied in [2]. A variable neighborhood search based method (VNS) is proposed in [18] and recently a harmony search method is studied for the mTTP variant (mirored traveling tournamant problem) [19].

In this work, first, we study two local search methods for TTP, which are simulated annealing (SA) and variable neighborhood search (VNS). The two methods are used as a first-step for finding a feasible solution that satisfies the problem constraints. As a second step, we propose a stochastic local search algorithm (SLS) to find a good approximate solution for TTP that minimizes the total distance traveled by the teams. The overall method is implemented, tested on benchmarks and compared with other well-known techniques for TTP.

Comparing our approach and the work of Anagnostopoulos et al. [2], we give the following differences.

- In our method, the search is limited to feasible solutions while Anagnostopoulos et al. explore both feasible and infeasible schedules [2].
- In our method, we handle the TTP problem in two main steps. 1) After generating an initial double round-robin configuration, we apply a local search method based on a constraint satisfaction problem encoding. We define a cost function to select feasible solutions, i.e., those having a zero value cost. 2) We propose a stochastic local search method (SLS) to further improve the solution. SLS is limited to feasible solutions. SLS minimizes the total distance traveled by the teams. The objective function used by SLS computes this total distance. There is no need to add a penalty function since we explore feasible configurations of the search space [2].
- Contrary to our methodology, Anagnostopoulos proposed a simulated annealing method (SATTP) in one step. This SATTP starts with a random initial solution obtained by using a simple backtrack search. The cost function combines travel distances and the number of violations. In Anagnostoploulos constraint violations are penalized [2].

The rest of this paper is organized as follows: Section 2 gives background on the traveling tournament problem (TTP). Section 3 presents in detail the proposed method applied to the TTP problem. Some numerical results are given in Section 4. Finally, Section 5 concludes the work and gives some perspectives.

2 PROBLEM DEFINITION

The traveling tournament problem (TTP) is the problem of scheduling a double round-robin tournament, while satisfying a set of related constraints and minimizing the total distance traveled by the teams [13, 11, 28, 21].

The problem can be stated as follows: let us consider n teams (n even and positive), a double round robin tournament is a set of games in which every team plays every other team exactly once at home and once away. A double round robin

tournament has $2 \cdot (n-1)$ slots. The distance between team cities are given by $n \cdot n$ symmetric matrix *Dis*, such that an element Dis_{ij} of *Dis* represents the distance between the homes of the teams t_i and t_j. The teams begin in their home city and must return there after the tournament.

The TTP is the problem of finding a feasible schedule that minimizes the distance traveled by the teams, and satisfies the following constraints:

Double round robin constraint (DRRT): that means that each team plays with every other team exactly twice, once in its home and once in the home of its opponent.

AtMost constraint: each team must play no more than U and no less than L consecutive games at home or away. Specifically, in our case, L is set to 1 and U to 3.

NoRepeat constraint: A game $t_i - t_j$ can never be followed in the next round by the game $t_j - t_i$.

1. The TTP contains the number of teams (denoted n) and the distance matrix (denoted *Dis*).
2. The output are: a double round robin tournament on the n teams respecting the three constraints AtMost, NoRepeat and DRRT, and where the total distance traveled by the teams is minimized.

Table 1 gives an example of a schedule for $n = 4$ teams. The negation sign means that the team plays away.

Round / **Team**	$Round_1$	$Round_2$	$Round_3$	$Round_4$	$Round_5$	$Round_6$
t_1	3	2	4	−3	−2	−4
t_2	−4	−1	−3	4	1	3
t_3	−1	4	2	1	−4	−2
t_4	2	−3	−1	−2	3	1

Table 1. Example of double round robin tournament with $n = 4$

This schedule specifies that the team t_1 has the following schedule: it successively plays against teams t_3 at home, t_2 at home, t_4 at home, t_3 away, t_2 away and t_4 away. The travel cost of team t_1 is: $Dis_{13} + Dis_{32} + Dis_{24} + Dis_{41}$.

We note that the travel costs of a schedule S is the sum of the travel cost of every team (denoted Travel-cost(S)).

3 THE PROPOSED APPROACH APPLIED TO TTP

We propose a new method for the TTP problem. The proposed method consists of two main steps. First, we start with an initial configuration satisfying the DRRT constraint. Then, we apply a local search method to generate a feasible configuration

that verifies the three constraints: AtMost, NoRepeat, and DRRT. For this step, we study two local search methods which are: variable neighborhood search (VNS) and simulated annealing (SA) where the role is to handle the feasibility criterion of solutions. In the second step, the feasible configuration found by the local search step is sent to the stochastic local search method (SLS) for minimizing the total distance traveled by the teams. Furthermore, we design a new technique which we call the aspiration technique. This technique is used to ensure that the three hard constraints are always satisfied when applying the SLS algorithm on the feasible search space.

3.1 The Initial Configuration

The search method starts with an initial configuration verifying the DRRT constraint. We create this configuration based on graph-theory modelling [9] as follows:

We have $n/2$ games per round and $2 \cdot (n-1)$ rounds. We number the vertices of the graph from 1 to n, where n is the number of teams. We put the top n in the center and the other vertices in a circle around the top n.

- In the first day, we organize a game between (Team 1 and Team n), (Team 2 and Team $n-1$), (Team 3 and Team $n-2$), and so on, up to the game between (Team $n/2$ and Team $n/2+1$).
- In the following day, we reproduce what happened in the previous day, making a simple clockwise rotation of the coupling.

Figure 1 shows an example of the creation of the initial configuration for the case of 6 teams.

$Round_1$	(t_1, t_6)	(t_5, t_2)	(t_4, t_3)
$Round_2$	(t_5, t_4)	(t_1, t_3)	(t_6, t_2)
$Round_3$	(t_1, t_5)	(t_2, t_4)	(t_6, t_3)
$Round_4$	(t_1, t_2)	(t_5, t_3)	(t_6, t_4)
$Round_5$	(t_5, t_6)	(t_1, t_4)	(t_2, t_3)

Table 2. Single round robin tournament with $n = 6$

For more details, Table 2 gives the schedule which is a single round robin tournament **(SRR)**. We note that in a single round-robin schedule, each team plays every other team once. When each team plays all others twice, this is called a double round-robin tournament (DRRT). As shown in Table 3, the double round robin tournament schedule can be obtained by adding the mirror of the SRR schedule.

3.2 Local Search Method for Feasible Schedules

After having created the initial DRRT schedule, we call the local search method in order to locate feasible configurations satisfying the three constraints: AtMost,

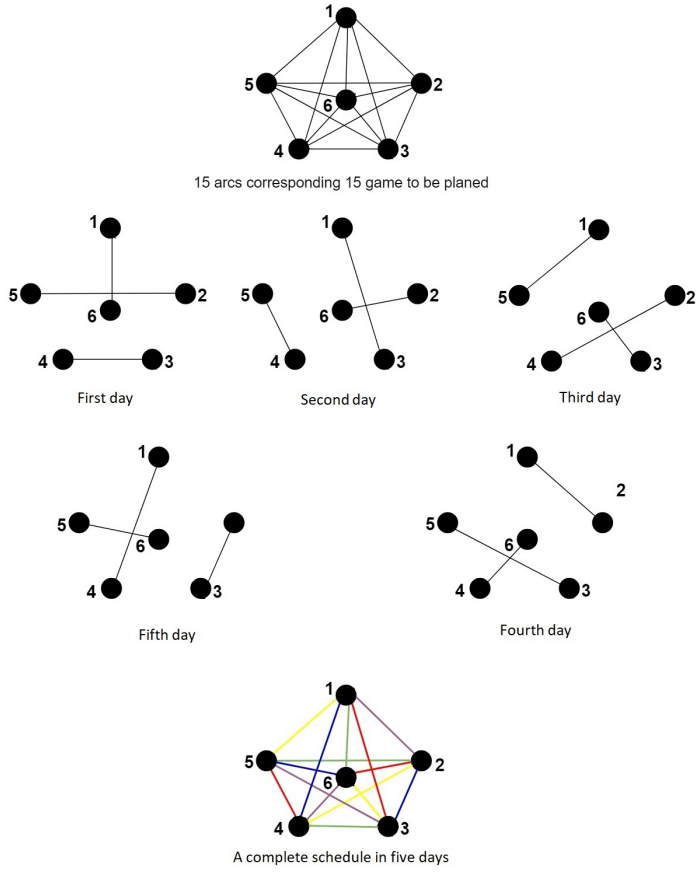


Figure 1. Creation of the initial configuration

$Round_1$	(t_1, t_6)	(t_5, t_2)	(t_4, t_3)
$Round_2$	(t_5, t_4)	(t_1, t_3)	(t_6, t_2)
$Round_3$	(t_1, t_5)	(t_2, t_4)	(t_6, t_3)
$Round_4$	(t_1, t_2)	(t_5, t_3)	(t_6, t_4)
$Round_5$	(t_5, t_6)	(t_1, t_4)	(t_2, t_3)
$Round_6$	(t_6, t_1)	(t_2, t_5)	(t_3, t_4)
$Round_7$	(t_4, t_5)	(t_3, t_1)	(t_2, t_6)
$Round_8$	(t_5, t_1)	(t_4, t_2)	(t_3, t_6)
$Round_9$	(t_2, t_1)	(t_3, t_5)	(t_4, t_6)
$Round_{10}$	(t_6, t_5)	(t_4, t_1)	(t_3, t_2)

Table 3. Mirror of single round robin tournament $n = 6$

NoRepeat, and DRRT. We study VNS and SA local search methods for building feasible schedules. We use a cost function to penalize configurations that violate the considered constraints. Further, we use neighborhood structures to explore the search space. The cost function, the neighborhood structures and the two methods are detailed in the following.

3.2.1 The Cost Function

The cost function consists of two terms. The first term permits to penalize configurations not satisfying the *AtMost* constraint. The second term is to penalize those not satisfying the *NoRepeat* constraint.

First, it is important to give the following useful notations in Table 4 to represent the constraints.

t_i	is the team number i where $i \in [1, n]$.
(t_i, t_j)	is the game t_i, vs. t_j in the home of t_i.
$Round_l$	is the round number l where $1 \leq l \leq \lvert Round \rvert$.
$R_{i,j}$	means that the match (t_i, t_j) is scheduled in a round $R_{i,j}$ where $1 \leq R_{i,j} \leq \lvert Round \rvert$, $\forall i, j \in \lvert T \rvert, i \neq j$. For example in Table 3, the match (t_5, t_1) is scheduled in $Round_8$ at home of t_5, $R_{5,1} = 8$, while the match (t_1, t_5) is scheduled in $Round_3$, $R_{1,5} = 3$.
S	is a DRRT schedule.

Table 4. Some useful notations and definitions

Now, we define the No-repeat constraint $f_{No\_repeat} \cdot occ\_norepeat(S, t_i, t_j)$. The function $occ\_norepeat(S, t_i, t_j)$ verifies, in a current schedule S, if the match (t_i, t_j) is followed in the next round by the match (t_j, t_i). This occurs when $R_{i,j} = R_{j,i} + 1$ or when $R_{j,i} = R_{i,j} + 1$.

$$occ\_norepeat(S, t_i, t_j) = \begin{cases} 1, & \text{if } (R_{i,j} = R_{j,i} + 1) \vee (R_{j,i} = R_{i,j} + 1), \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

For example, the schedule in Table 3 (denoted S_3), $occ\_norepeat(S_3, t_1, t_6) = 0$ because for the game (t_1, t_6), $R_{1,6} \neq R_{6,1} + 1$ and $R_{6,1} \neq R_{1,6} + 1$. The match (t_1, t_6) is played in $R_{1,6} = Round_1$ while the match (t_6, t_1) is scheduled in $R_{6,1} = Round_6$. This implies that $R_{1,6} \neq R_{6,1} + 1$.

The penalty $f_{No\_repeat}(S)$ is the total number of times that the game (t_i, t_j) is followed immediately by (t_j, t_i) in the schedule S.

$$f_{No\_repeat}(S) = \sum_{i=1}^{|T|} \sum_{j=i+1}^{|T|} occ\_norepeat(S, t_i, t_j). \quad (2)$$

Also, we define the At-Most constraint f_{At_Most}. $occ\_atmost\,(S, t_i, Round_l)$ is the number of times that the team t_i plays home or away games in three rounds successively from $Round_l$ ($Round_l, Round_l + 1, Round_l + 2, Round_l + 3$).

$$con\_atmost(S, t_i, Round_l) = \begin{cases} 1, & \text{if } occ\_atmost(S, t_i, Round_l) > 3, \\ & Round_l < |Round| - 3, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

For example, for the schedule S_3 in Table 3: $occ\_atmost\,(S_3, t_1, Round_1) = 1$, since the team t_1 is played 4 consecutive games at home in $Round_1$, $Round_2$, $Round_3$ and $Round_4$.

The penalty At-most $f_{At\_most}\,(S)$ is the total number of times the teams play more than three consecutive home games or three consecutive away games.

$$f_{At\_most}\,(S) = \sum_{i=1}^{|T|} \sum_{l=1}^{|Round|-3} con\_atmost\,(S, t_i, Round_l)\,. \tag{4}$$

The cost function is then defined by the sum of the two penalty constraints **NoRepeat** and **AtMost**:

$$\mathbf{Cost}(S) = f_{No\_repeat}(S) + f_{At\_most}(S). \tag{5}$$

The main goal of the local search method is to find a configuration of zero-cost value. This means finding a feasible configuration satisfying the three constraints: AtMost, NoRepeat, and DRRT.

3.2.2 Neighborhood Structures

We use three neighborhood structures which are detailed in the following:

- N_1: Swap Home. This move swaps the home/away roles of teams. For instance, when we take two teams t_i and t_j, the move $Swap\ Home(S, t_i, t_j)$ swaps the home/away roles of a game involving the teams t_i and t_j. If team t_i plays home against team t_j at $Round_k$ and away against team t_j at $Round_l$ then the move Swap Home (S, t_i, t_j) gives the same schedule as S, except that team t_i plays away against team t_j at $Round_k$, and home against t_j at $Round_l$. Table 5 displays an example of a move using the N_1 neighborhood structure.
- N_2: Swap Round. This move consists of swapping all games of a given pair of rounds. For example the move Swap Round $(S, Round_k, Round_l)$ swaps two given rounds ($Round_k$ and $Round_l$). Table 6 gives an example of schedule when applying the N_2 move.
- N_3 Swap Team. This move corresponds to swapping all opponents of a given pair of teams over all rounds, For example the move Swap Team (S, t_i, t_j) corresponds to swapping all opponents of teams t_i and t_j over all rounds. Table 7 shows an example of applying the N_3 move.

$Round_1$	(t_5, t_1)	(t_4, t_2)	(t_3, t_6)
$Round_2$	(t_4, t_1)	(t_3, t_5)	(t_2, t_6)
$Round_3$	(t_3, t_1)	(t_4, t_6)	(t_2, t_5)
$Round_4$	(t_6, t_1)	(t_2, t_3)	(t_4, t_5)
$Round_5$	(t_2, t_1)	(t_6, t_5)	(t_4, t_3)
$Round_6$	(t_1, t_5)	(t_2, t_4)	(t_6, t_3)
$Round_7$	(t_1, t_3)	(t_6, t_4)	(t_5, t_2)
$Round_8$	(t_1, t_4)	(t_5, t_3)	(t_6, t_2)
$Round_9$	(t_1, t_6)	(t_3, t_2)	(t_5, t_4)
$Round_{10}$	(t_1, t_2)	(t_5, t_6)	(t_3, t_4)

The application of the move Swap home away: N_1 (S, t_1, t_2):

$\Downarrow$

$Round_1$	(t_5, t_1)	(t_4, t_2)	(t_3, t_6)
$Round_2$	(t_4, t_1)	(t_3, t_5)	(t_2, t_6)
$Round_3$	(t_3, t_1)	(t_4, t_6)	(t_2, t_5)
$Round_4$	(t_6, t_1)	(t_2, t_3)	(t_4, t_5)
$Round_5$	(t_1, t_2)	(t_6, t_5)	(t_4, t_3)
$Round_6$	(t_1, t_5)	(t_2, t_4)	(t_6, t_3)
$Round_7$	(t_1, t_3)	(t_6, t_4)	(t_5, t_2)
$Round_8$	(t_1, t_4)	(t_5, t_3)	(t_6, t_2)
$Round_9$	(t_1, t_6)	(t_3, t_2)	(t_5, t_4)
$Round_{10}$	(t_2, t_1)	(t_5, t_6)	(t_3, t_4)

Table 5. Schedule before (up) and after (down) the application of home-away swap

The simulated annealing and variable neighborhood search (VNS) algorithms are used to search for configurations satisfying the three constraints. These methods are called for finding configurations with a cost function 5 equal to zero. Both SA and VNS use the cost function to compare, in terms of quality, two schedules S and S'. We note that the DRRT constraints are always satisfied since we started with an initial configuration satisfying the DRRT constraint.

3.2.3 SA for Feasible Configurations

SA starts with an initial DRRT schedule. Then it moves iteratively (using the N_1 move) from the current schedule to another one in the search space for finding lower cost solutions. When a new schedule with a lower cost is found, it replaces the current solution. When a new schedule of a higher cost is chosen, it replaces the current solution with some probability. This probability is decreased as the algorithm progresses (analogously to the temperature in physical annealing). The SA algorithm is sketched in Algorithm 1.

3.2.4 VNS for Feasible Configurations

VNS is a local search meta-heuristic proposed in 1997 by Mladenovic and Hansen. Various variants of VNS have been proposed since then, but the basic idea is a sys-

$Round_1$	(t_5, t_1)	(t_4, t_2)	(t_3, t_6)
$Round_2$	(t_4, t_1)	(t_3, t_5)	(t_2, t_6)
$Round_3$	(t_3, t_1)	(t_4, t_6)	(t_2, t_5)
$Round_4$	(t_6, t_1)	(t_2, t_3)	(t_4, t_5)
$Round_5$	(t_2, t_1)	(t_6, t_5)	(t_4, t_3)
$Round_6$	(t_1, t_5)	(t_2, t_4)	(t_6, t_3)
$Round_7$	(t_1, t_3)	(t_6, t_4)	(t_5, t_2)
$Round_8$	(t_1, t_4)	(t_5, t_3)	(t_6, t_2)
$Round_9$	(t_1, t_6)	(t_3, t_2)	(t_5, t_4)
$Round_{10}$	(t_1, t_2)	(t_5, t_6)	(t_3, t_4)

After applying the move Swap Round: N_2 $(S, Round_3, Round_5)$:

⇓

$Round_1$	(t_5, t_1)	(t_4, t_2)	(t_3, t_6)
$Round_2$	(t_4, t_1)	(t_3, t_5)	(t_2, t_6)
$Round_3$	(t_2, t_1)	(t_6, t_5)	(t_4, t_3)
$Round_4$	(t_6, t_1)	(t_2, t_3)	(t_4, t_5)
$Round_5$	(t_3, t_1)	(t_4, t_6)	(t_2, t_5)
$Round_6$	(t_1, t_5)	(t_2, t_4)	(t_6, t_3)
$Round_7$	(t_1, t_3)	(t_6, t_4)	(t_5, t_2)
$Round_8$	(t_1, t_4)	(t_5, t_3)	(t_6, t_2)
$Round_9$	(t_1, t_6)	(t_3, t_2)	(t_5, t_4)
$Round_{10}$	(t_1, t_2)	(t_5, t_6)	(t_3, t_4)

Table 6. Schedule before (up) and after (down) the application of Swap Round

tematic change of neighborhood combined with a local search [14, 22]. Unlike local search, VNS works on a set of different neighborhoods. In our study, we use three ($k = 3$) structures of neighborhood which are: N_1 (Swap Home), N_2 (Swap Round) and N_3 (Swap Team). At each iteration, we select among the three structures one to create neighbor solutions. The proposed VNS uses the deepest descent strategy (DDS) as a subroutine. More precisely, VNS starts with an initial DRRT (S) schedule and then tries to find a good solution in the whole neighborhood in an iterative manner. The DDS procedure is called for each candidate solution constructed by VNS method. As shown in Algorithm 2, DDS explores iteratively the search space of the given solution S and returns the best neighbor solution found in this space. VNS first uses the N_1 structures to create neighbor solutions. When there is no improvement, the neighborhood structure is mapped to N_2 and then to N_3 in the hope to create diverse and good neighbor solutions. As done with SA, VNS works in the same objective to find a feasible configuration. The overall process of VNS is repeated until a schedule with zero-cost is reached ($Cost(S) = 0$).

The VNS algorithm is sketched in Algorithm 3.

$Round_1$	(t_5, t_1)	(t_4, t_2)	(t_3, t_6)
$Round_2$	(t_4, t_1)	(t_3, t_5)	(t_2, t_6)
$Round_3$	(t_3, t_1)	(t_4, t_6)	(t_2, t_5)
$Round_4$	(t_6, t_1)	(t_2, t_3)	(t_4, t_5)
$Round_5$	(t_2, t_1)	(t_6, t_5)	(t_4, t_3)
$Round_6$	(t_1, t_5)	(t_2, t_4)	(t_6, t_3)
$Round_7$	(t_1, t_3)	(t_6, t_4)	(t_5, t_2)
$Round_8$	(t_1, t_4)	(t_5, t_3)	(t_6, t_2)
$Round_9$	(t_1, t_6)	(t_3, t_2)	(t_5, t_4)
$Round_{10}$	(t_1, t_2)	(t_5, t_6)	(t_3, t_4)

After the application of the move Swap Team: $N_3\ (S, (t_3, t_5))$:

$\Downarrow$

$Round_1$	(t_3, t_1)	(t_4, t_2)	(t_5, t_6)
$Round_2$	(t_4, t_1)	(t_3, t_5)	(t_2, t_6)
$Round_3$	(t_5, t_1)	(t_4, t_6)	(t_2, t_3)
$Round_4$	(t_6, t_1)	(t_2, t_5)	(t_4, t_3)
$Round_5$	(t_2, t_1)	(t_6, t_3)	(t_4, t_5)
$Round_6$	(t_1, t_3)	(t_2, t_4)	(t_6, t_5)
$Round_7$	(t_1, t_5)	(t_6, t_4)	(t_3, t_2)
$Round_8$	(t_1, t_4)	(t_5, t_3)	(t_6, t_2)
$Round_9$	(t_1, t_6)	(t_5, t_2)	(t_3, t_4)
$Round_{10}$	(t_1, t_2)	(t_3, t_6)	(t_5, t_4)

Table 7. Schedule before (up) and after (down) the application of Swap Team

3.3 Stochastic Local Search for TTP

As already mentioned, the two local search methods (SA and VNS) are used in the first step to handle the feasibility criterion. In the second step, we apply SLS on the feasible configuration to handle the optimization criterion and minimize the total distance traveled by the teams.

SLS [15, 3] is a local search method that combines diversification and intensification strategies to locate a good solution. The intensification phase selects the best neighbor solution while the diversification phase selects a random neighbor solution. The diversification phase is applied with a fixed probability $wp > 0$ and the intensification phase with a probability $1 - wp$. The process is repeated until a certain number of iterations called *maxiter* is reached.

To maintain that the three hard constraints are always satisfied in our SLS algorithm, we apply an aspiration technique. The latter is given in the next section.

Algorithm 1: The proposed SA for feasible configurations

Data: A DRRT schedule, $\alpha = 0.9$, the neighborhood structures N_1
Result: A feasible configuration that satisfies AtMost, NoRepeat and DRRT constraints

1 $S_0 \leftarrow$ an initial configuration verifying DRRT
2 $Temp \leftarrow$ an initial temperature
3 $S \leftarrow S_0$
4 **for** $(I = 1$ *to Maxiter*$)$ **do**
5 **if** $(Cost(S) \neq 0)$ **then**
6 $S' \leftarrow$ Choose random configuration using the neighborhood structure (N_1) on S
7 $r \leftarrow$ random number between 0 and 1
8 **if** $(r \prec e^{\frac{Cost(S)-Cost(S')}{Temp}})$ **then**
9 $S \leftarrow S'$
10 $Temp \leftarrow Temp \cdot \alpha$
11 **else**
12 Go to 13
13 Return the schedule S

Algorithm 2: DDS$(S, N_k(S))$

Data: A DRRT schedule S, the neighborhood structures $N_k(S)$
Result: an improved schedule

1 **repeat**
2 Choose $S' \in N_k(S)$ with $Cost(S') \leq Cost(S'')$, $\forall S'' \in N_k(S)$
3 **if** $(Cost(S') < Cost(S))$ **then**
4 $S \leftarrow S'$
5 **until** $Cost(S') \geq Cost(S), \forall S' \in N_k(S)$;
6 Return the schedule S

3.4 Aspiration Technique to Select the Best Neighbor

We propose a new technique which we called *aspiration technique* to filter the search space and keep only the feasible configurations. The aspiration technique permits to memorize information on moves leading to feasible neighbor configurations, starting from a current configuration. First, we explore the search space to locate feasible configurations (denoted CSC) with zero-cost according to the cost function (5) described in Section 3.2.1. Then among them, we take the best one having the minimum traveled distance.

Algorithm 3: The proposed VNS for feasible configurations

Data: A DRRT schedule, the three first Neighborhood structures N_k $(1 \leq k \leq 3)$

Result: A feasible configuration that satisfies AtMost, NoRepeat and DRRT constraints

```
1  S ← an initial configuration verifying DRRT
2  k ← 1
3  S ← local search(S)
4  for (I = 1 to Maxiter) do
5      if (Cost(S) ≠ 0) then
6          S' ← choose random solution (N_k(s))
7          S" ← Call DDS (S', N_k)
8          if (Cost((S")) < Cost((S)) then
9              S ← S"
10         else
11             if (k < 3) /* when there is no improvement, the neighborhood structure
                  is changed to the next one                                    */
12             then
13                 k ← k + 1
14             else
15                 k ← 1
16     else
17         Go to 18
18 Return the schedule S
```

For every neighborhood structure, the technique is illustrated as follows:

- For the N_1 neighborhood structure (Swap Home), we use a list of *NBgames* elements where *NBgames* is the number of the games ($NBgames = (n-1) \cdot n$). Each element of the list consists of two cells (see Figure 2). The first cell is the game (t_i, t_j) and the second is the cost of the move $N_1(S, (t_i, t_j))$ (denoted $C\_M(t_i, t_j)$). The latter is the variation of the cost value that must be applied if we swap the home/away roles of game (t_i, t_j) to create S'. If S' is the schedule obtained after applying N_1 on S in (t_i, t_j), i.e. $N_1(S, (t_i, t_j)) = S'$.

$$Cost(S') = Cost(S) + C\_M(t_i, t_j). \tag{6}$$

  With this technique, we can take the subset of swaps that gives feasible solutions with zero cost in time $O(|NBgames|)$.

- For the N_2 neighborhood structure (Swap Round), we use a matrix *Movr* of $|Round| \cdot |Round|$ elements where each element $Movr[Round_i, Round_j]$ represents the variation of the cost value that must be applied if a move exchanges two

Algorithm 4: The SLS method for TTP

Data: a TTP instance, *maxiter*, *wp*
Result: The best schedule S

```
Create an initial configuration (CS) verifying the DRRT constraint
Apply the local search method (VNS or SA) on CS to obtain a
  configuration (CSC) verifying the three constraints
S ← CSC
for (I = 1 to Maxiter) do
    if (r ≺ wp) then
        Apply the aspiration technique N1(S) to find a subset of
          neighborhoods with Cost equal to zero.
        Choose a random neighbor among them S' ∈ N1(S)
        S ← S'
    else
        Create Movr[ti, tj] using aspiration technique on N3(S)
        Select the best movement (ti, tj)
        S ← SwapTeam (S, ti, tj)
        Create a list (NBgames) using Aspiration technique, on N2(S)
        Select the best movement (ti, tj)
        S ← SwapHome (S, ti, tj)
Return the schedule S
```

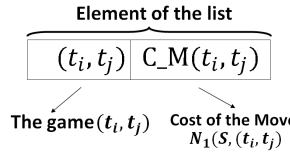


Figure 2. Structure of an element of our list

rounds: $Round_i$ and $Round_j$) to create the schedule S'. Thus the cost of $S' \in N_2(S)$ is obtained by the sum of the $Cost(S)$ and the value of the element $Movr[Round_i, Round_j]$: If S' is the schedule obtained after applying N_2 on S in $(Round_i, Round_j)$ (i.e. $N_2(S, (Round_i, Round_j)) = S'$).

$$Cost(S') = Cost(S) + Movr[Round_i, Round_j]. \tag{7}$$

We can get a subset of neighborhoods that have the cost-value equal to zero in time $\leq (O|N_2(S)|)$. We note that after every move, we update the aspiration matrix *Movr*.

- For N_3 (Swap Teams), we use a matrix *Movt* of $|T| \cdot |T|$ element where each element represents the variation of cost that must be applied if a move exchanges two plans of two teams (t_i, t_j). Thus the cost of S' (S after a move) is obtained

by the sum of the cost of S and the value of the element $Movt[t_i, t_j]$: If $S^{'}$ obtained after applying N_3 on S in (t_i, t_j), i.e. $N_3(S, (t_i, t_j)) = S^{'}$.

$$Cost(S^{'}) = Cost(S) + Movt[t_i, t_j]. \quad (8)$$

We can obtain a subset of feasible neighbor configurations with zero cost value in time equal to $O(|N_3(S)|)$. The best neighbor is obtained by browsing this subset in time $\leq O(|N_3(S)|)$. The matrix is updated after each move.

The SLS is sketched in Algorithm 4.

4 EXPERIMENTS

The source code is written in Java. The experiments are performed on a Core Duo (1.60 GHz) with 2 GB of RAM. We evaluate our method on five different sets of instances available at the website [1]. We validate our method on the most popular testbed that includes: the so-called *NLx* instances, Circular distance instance $CIRCx$, Super Instance, Galaxy and the CON instances [29, 25, 19]. The description of these instances is given as follows:

NLx instances are based on real data of the US National Baseball League, where x is an even number of teams.

CONx is the constant distance instances are characterized by a distance of one (1) between all teams.

SUPERx is based on Rugby League, a league with 14 teams from South Africa, New Zealand and Australia.

CIRCx instances: all teams are placed on a circle, with unit distances (distance of 1 between all adjacent nodes). The distance between two teams i and j with $i > j$ is then equal to the length of the shortest path between i and j which is the minimum of $i - j$ and $j - i + n$.

4.1 Parameter Tuning

The adjustment of the different parameters of the proposed algorithms is fixed by an experimental study. The set values are those for which a good compromise between the quality of the solution obtained by the algorithm and the running time of the algorithm is found. Due to the non-deterministic nature of the SLS method, for each instance, 20 runs have been considered, each of them for 10 000 iterations.

For the probability wp, a large wp ($wp > 0.6$) may cause of being trapped in local solutions, while a scriptsize wp ($wp < 0.3$) means every solution is chosen from the search space randomly, which may decrease explored more thoroughly the promising regions in search space. Therefore, we should use wp value between 0.3 and 0.6. Hence in our study the wp probability is set to 0.4.

For VNS, after twelve runs we observed that the average necessary number of iterations to find a CSC solution is about 8 000 iterations. Furthermore, in VNS the feasible schedule is found at each SLS run. The superiority of VNS is due to the good combination of the diversification and the intensification in the search space this by systematically changes the proposed neighborhood in two phases: firstly, descent to find a local optimum and finally, a perturbation phase to get out of the corresponding valley.

4.2 Numerical Results

In the following, we present the numerical results found by the proposed approach. First, we give in Table 8 (respectively in Table 9) the results found by SA (respectively VNS). The first column gives the number of teams which are instances with an even number of teams, from $n = 6$ up to $n = 36$, the column *Nbr_mov* represents the number of necessary moves to obtain a feasible configuration verifying the three constraints DRRT, AtMost, and NoRepeat. The column Time gives the CPU time in seconds to obtain the feasible configuration (the reported time is the best obtained time to find the feasible solution).

n	Nbr_mov	Time	n	Nbr_mov	Time
6	1	0.023	22	2 001	335.60
8	2	0.050	24	2 421	1 119.63
10	3	0.091	26	2 621	1 455.85
12	83	18.53	28	3 015	1 654.87
14	92	22.703	30	4 045	1 893.91
16	192	113.50	32	5 113	3 221.51
18	281	181.36	34	6 342	4 761.34
20	1 945	318.18	36	8 782	5 931.34

Table 8. The results found by SA

n	Nbr_mov	Time	n	Nbr_mov	Time
6	1	0.010	22	1 623	456.11
8	1	0.024	24	1 937	987.01
10	3	1.12	26	2 134	1 221.23
12	46	17.22	28	2 995	1 546.67
14	78	19.67	30	3 110	1 224.34
16	95	97.00	32	4 675	2 551.00
18	119	112	34	5 112	3 243.16
20	978	234.77	36	6 433	4 056.44

Table 9. The results found by VNS

We implemented two variants of our method: $\text{SLS}_{with\ SA}$ and $\text{SLS}_{with\ VNS}$. The first one is SLS combined with SA for a local search method. In the second one, we

use VNS instead of SA as a local search. Table 10 compares the time consumed by the two methods. The time is given in second.

As shown in Tables 8 and 9, both SA and VNS succeed in finding the feasible configuration (schedule satisfying DRRT, AtMost, and NoRepeat) for all the considered benchmarks (until n equals to 36 teams). The two methods VNS and SA are comparable in term of ability to find feasible configurations in comparable time. However, when we study the overall methods ($SLS_{with\ VNS}$) and ($SLS_{with\ SA}$), we can remark that VNS accelerates the search of solutions when it is integrated in SLS contrary to SA. We can see clearly that VNS is better than SA when it is integrated into SLS in term of total CPU time consuming. We draw Figure 3 to show this behavior.

Instance	**$SLS_{with\ SA}$**	**$SLS_{with\ VNS}$**
	Time	**Time**
CON4	9.89	**0.10**
CON6	24.80	**19.80**
CON8	55.47	**30.22**
CON10	104.21	**89.29**
CON12	161.04	**104.75**
CON14	344.64	**214.44**
CON16	629.99	**411.99**
CON18	1 145.00	**891.29**
CON20	1 887.12	**905.75**
CON22	2 008.05	**1 224.21**
CON24	4 507.58	**2 998.99**
NL4	98.11	**53.23**
NL6	341.89	**145.21**
NL8	967.34	**524.15**
NL10	1 064.31	**575.21**
NL12	2 681.38	**974.83**
NL14	3 671.10	**2 452.93**
NL16	7 343.24	**5 316.34**
CIRC4	123.36	**94.65**
CIRC6	370.62	**201.32**
CIRC8	612.55	**431.96**
CIRC10	1 449.50	**866.11**
Galaxy 4	160.71	**108.14**
Galaxy 6	430.11	**288.50**
SUPER4	1 941.67	**897.89**

Table 10. Time comparison between $SLS_{with\ SA}$ and $SLS_{with\ VNS}$

To better analyze the obtained results and demonstrate the performance of our approach we compare the proposed method to the best known solution and other the best-known techniques for TTP.

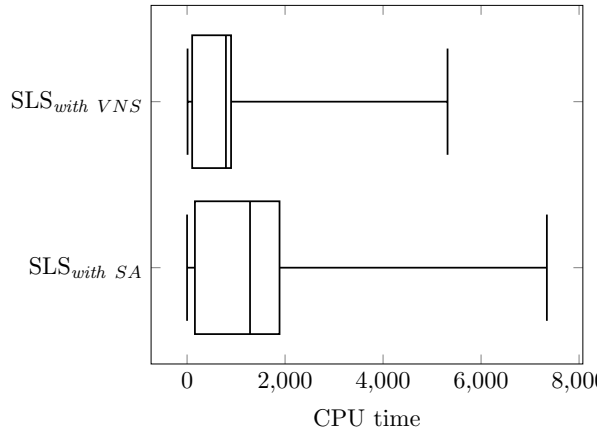


Figure 3. CPU time comparison between SLS$_{with\ SA}$ and SLS$_{with\ VNS}$

Table 11 gives the numerical results found by the overall approach. We give the CPU time (Time) in seconds (the reported time is the time needed to find the best solution), the best (Best) and the average solution (AVG) of twenty executions found by our method. We give the best known solutions (Best*) [1] for each instance and the gap between Best and Best*. The best results are in bold font. The proposed method SLS is compared with the best-known solutions Best* for TTP in order to show its performance in solving TTP.

$$Gap\,\% = \frac{\text{SLS}(Best) - Best*}{\text{SLS}(Best)} * 100. \tag{9}$$

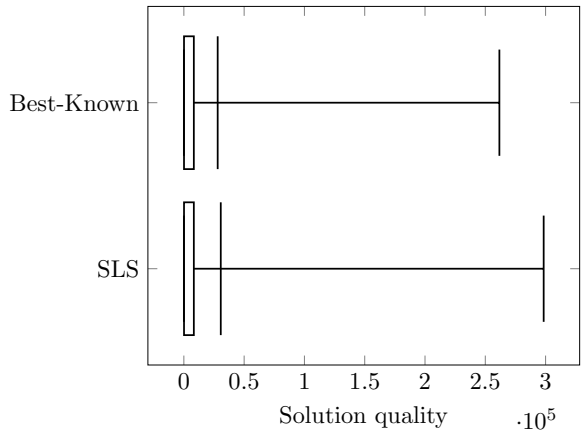


Figure 4. Comparison between SLS and Best-known

Instance	**Best***	**SLS**			**Gap %**
		Time(s)	**Best**	**Average**	
CON4	**17**	0.10	**17**	17	**0**
CON6	**43**	19.80	**43**	43	**0**
CON8	**80**	30.22	**80**	80	**0**
CON10	**124**	89.29	**124**	125	**0**
CON12	**182**	104.75	**182**	184	**0**
CON14	**252**	214.44	**252**	254	**0**
CON16	327	411.99	336	338	2.67
CON18	417	891.29	433	455	3.69
CON20	522	905.75	525	554	0.57
CON22	**628**	1 224.21	**628**	632	**0**
CON24	749	2 998.99	756	808	0.92
NL4	**8 276**	53.23	**8 276**	8 276	**0**
NL6	**23 916**	145.21	**23 916**	24 122	**0**
NL8	**39 947**	524.15	40 621	42 234	1.65
NL10	59 583	575.21	61 193	62 711	2.63
NL12	111 248	974.83	120 655	127 856	7.79
NL14	188 728	2 452.93	206 274	231 785	8.50
NL16	261 687	5 316.34	308 413	322 394	15.15
CIRC4	**20**	94.65	**20**	20	**0**
CIRC6	**64**	201.32	**64**	64	**0**
CIRC8	130	431.96	140	144	7.69
CIRC10	242	866.11	272	287	11.02
Galaxy 4	**416**	108.14	**416**	416	**0**
Galaxy 6	**1 365**	288.50	**1 365**	1 394	**0**
Galaxy8	**2 373**	1 007.01	**2 373**	2 648	**0**
Galaxy10	**3 676**	2 015.56	4 554	5 134	19.27
Galaxy12	7 034	2 897.14	7 354	8 005	4.35
Super 4	**63 405**	897.89	**63 405**	63 405	**0**
Super 6	**130 365**	1 425.11	**130 365**	130 365	**0**

Table 11. A comparison with best-known Best*

SLS succeeds in finding the optimum solutions for CON4, CON6, CON8, CON10, CON12, CON14, NL6, NL4, CIRC6, Galaxy4, Galaxy6, Galaxy8, Super4 and Super6. The gap between the best-known solutions and the results of our approach, in general, does not go above 15.15 % for NL instances and is between 1 and 3.67 % for CON instances. This demonstrates the effectiveness of the proposed approach in solving the traveling tournament problem. In addition to the numerical analysis, we draw the box plots diagrams to better visualize the distribution of cost value. The boxplot in Figure 4 shows that our method produces consistent results. The results are interesting and demonstrate the benefit of our approach for TTP.

4.3 A Comparative Study for NL Instances

Since the NLx family of instances is probably the most researched TTP benchmark-family and virtually all researches studying the TTP publish their computational results with NLx instances, in this section, we compare the proposed method to other well-known techniques for TTP on NL instances.

Table 12 compares SLS with some well-known techniques for TTP. The comparison is done with the following techniques: GA (which is a genetic algorithm with novel encoding scheme for representing a solution instance [6]), AIS_{TTP} (which is an immune-inspired algorithm based on the CLONALG framework [4]), $CTSA$ (which is a hybrid integer programming/constraint programming approach and a branch and price algorithm [23]), $AATTP$ (which is an approximation Algorithm for TTP [29]), $CPMT$ (which is a tabu search and simulated annealing [25]) and *ANT-HYP* (which is an ant based hyper-heuristic [5]).

The comparison with other techniques shows the efficiency of our method. In order to quantify this improvement we compute the performance ratio (PR) or the average gaps given as follows:

$$PR = \sum_{i=1}^{NBins} Gap_i / NBins \tag{10}$$

where the Gap_i is the gap between the best solution of our method and the best of the other techniques of the instance i. *NBins* is the total number of the considered instances.

The performance ratio between the proposed approach and $AATTP$ is equal to 8.68 % which means that our approach improves the results of $AATTP$ in average by 8.68 %. Also, our method gives better average than $CTSA$ with 0.76 % and improves the results of $CPMT$ and *ANT-HYP* method in average by 2.32 % and 3.61 %, respectively. Further, the proposed approach is able to perform better than GA results in average by 1.13 %, whereas our approach finds near solutions with deviation from AIS_{TTP} equal to -0.05 %.

4.4 A Comparative Study for CON Instances

For CON instances, the comparison is done with Tabu Search [10] since it achieves the best-known solution on almost all CON instances. Table 13 gives the results found by both SLS and Tabu search on CON instances. According to these numerical results, SLS succeeds in finding better results for all the checked instances compared to Tabu search. Furthermore, the performance ratio between our approach and Tabu search method is equal to 2.25 % which means that the proposed approach enhances the results of Tabu search in average by 2.25 %.

The superiority of our approach is explained by the good combination of the two main steps, which permits to explore efficiently the search space and locate good solutions.

Instance	**SLS**	**AATTP**	Gap	AIS_{TTP}	Gap	**CTSA**	Gap
NL4	**8 276**	—	—	—	—	—	—
NL6	**23 916**	—	—	—	—	24 467	**−2.30**
NL8	**40 621**	47 128	**−16.01**	40 156	1.14	41 754	**−2.78**
NL10	**61 193**	69 958	**−14.32**	61 351	**−0.25**	63 277	**−3.40**
NL12	**120 655**	125 086	**−3.67**	120 531	0.21	116 421	3.05
NL14	**206 274**	230 874	**−11.92**	206 434	**−0.77**	215 665	**−4.55**
NL16	**308 413**	300 744	2.48	288 674	0.04	288 674	5.40

Instance	**SLS**	**ANT − HYP**	Gap	**CPMT**	Gap	**GA**	Gap
NL4	**8 276**	—	—	—	—	—	—
NL6	**23 916**	23 916	**0**	—	—	23 916	**0**
NL8	**40 621**	40 361	0.64	41 928	**−3.21**	41 505	**−2.17**
NL10	**61 193**	65 168	**−6.49**	65 193	**−6.53**	—	—
NL12	**120 655**	123 752	**−2.56**	120 906	**−0.20**	—	—
NL14	**206 274**	225 169	**−9.16**	208 824	**−1.23**	—	—
NL16	**308 413**	321 037	**−4.09**	287 130	6.90	—	—

Table 12. A comparative study for NL instances

Instance	**SLS**	**Tabu Search**	Gap %
CON4	**17**	**17**	**0**
CON4	**43**	48	**−11.62**
CON8	**80**	81	**-1.25**
CON10	**124**	**124**	**0**
CON12	**182**	184	**−1.09**
CON14	**252**	253	**−0.09**
CON16	**336**	342	**−1.72**

Table 13. A comparative study for *CON* instances

5 CONCLUSION

We proposed a search method for constrained optimization. The proposed method handles optimality and feasibility separately. It is applied to the well-known NP-hard traveling tournament problem (TTP). TTP is concerned with finding a tournament schedule that minimizes the total distances traveled by the teams. The TTP has attracted significant interest recently since a favorable TTP schedule can generate large incomes in the budget of managing the league's sport. The proposed approach is a combination of the stochastic local search algorithm (SLS) as an optimization technique and the local search method (VNS/SA) as a search method for feasible configurations. The method is implemented, evaluated on publicly available standard benchmarks and compared with other techniques for TTP. The proposed method provides competitive results and finds solutions of high quality. It matches the best-known solutions on seventeen instances and outperforms some interesting

methods. The advantage of the novel method is that it can reduce mainly the problem complexity since we consider only feasible configurations. However, we can lose the global optima in some situation since we do not explore full search space. We plan to study the impact of exact techniques on the proposed method. It would be nice to study the effectiveness of our method in solving other constrained optimization problems.

REFERENCES

[1] Challenge Traveling Tournament Instances. `http://mat.tepper.cmu.edu/TOURN/`, 2016.

[2] Anagnostopoulos, A.—Michel, L.—Van Hentenryck, P.—Vergados, Y.: A Simulated Annealing Approach to the Traveling Tournament Problem. Journal of Scheduling, Vol. 9, 2006, No. 2, pp. 177–193, doi: 10.1007/s10951-006-7187-8.

[3] Boughaci, D.: Metaheuristic Approaches for the Winner Determination Problem in Combinatorial Auction. In: Yang, X. S. (Ed.): Artificial Intelligence, Evolutionary Computing and Metaheuristics. Springer, Berlin, Heidelberg, Studies in Computational Intelligence, Vol. 427, 2013, pp. 775–791, doi: 10.1007/978-3-642-29694-9_29.

[4] Pérez Cáceres, L.—Riff, M. C.: AISTTP: An Artificial Immune Algorithm to Solve Traveling Tournament Problems. International Journal of Computational Intelligence and Applications, Vol. 11, 2012, No. 1, Art. No. 1250008, doi: 10.1142/s1469026812500083.

[5] Chen, P.-C.—Kendall, G.—Vanden Berghe, G.: An Ant Based Hyper-Heuristic for the Travelling Tournament Problem. 2007 IEEE Symposium on Computational Intelligence in Scheduling (SCIS '07), 2007, pp. 19–26, doi: 10.1109/scis.2007.367665.

[6] Choubey, N. S.: A Novel Encoding Scheme for Traveling Tournament Problem Using Genetic Algorithm. International Journal of Computer Applications (IJCA), Special Issue on Evolutionary Computation for Optimization Techniques (ECOT 2010), Vol. 2, 2010, No. 7, pp. 79–82, doi: 10.5120/1536-139.

[7] Costa, F. N.—Urrutia, S.—Ribeiro, C. C.: An ILS Heuristic for the Traveling Tournament Problem with Predefined Venues. Annals of Operations Research, Vol. 194, 2012, No. 1, pp. 137–150, doi: 10.1007/s10479-010-0719-9.

[8] Moreira de Carvalho, M. A.—Nogueira Lorena, L. A.: New Models for the Mirrored Traveling Tournament Problem. Computers and Industrial Engineering, Vol. 63, 2012, No. 4, pp. 1089–1095, doi: 10.1016/j.cie.2012.08.002.

[9] de Werra, D.: Some Models of Graphs for Scheduling Sports Competitions. Discrete Applied Mathematics, Vol. 21, 1988, No. 1, pp. 47–65, doi: 10.1016/0166-218x(88)90033-9.

[10] Di Gaspero, L.—Schaerf, A.: A Composite-Neighborhood Tabu Search Approach to the Traveling Tournament Problem. Journal of Heuristics, Vol. 13, 2007, No. 2, pp. 189–207, doi: 10.1007/s10732-006-9007-x.

[11] Easton, K.—Nemhauser, G.—Trick, M.: The Traveling Tournament Problem Description and Benchmarks. In: Walsh, T. (Ed.): Principles and Practice of Constraint Programming – CP 2001. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2239, 2001, pp. 580–584, doi: 10.1007/3-540-45578-7_43.

[12] Goerigk, M.—Westphal, S.: A Combined Local Search and Integer Programming Approach to the Traveling Tournament Problem. Annals of Operations Research, Vol. 239, 2016, No. 1, pp. 343–354, doi: 10.1007/s10479-014-1586-6.

[13] Guedes, A. C. B.—Ribeiro, C. C.: A Heuristic for Minimizing Weighted Carry-Over Effects in Round Robin Tournaments. Journal of Scheduling, Vol. 14, 2011, No. 6, pp. 655–667, doi: 10.1007/s10951-011-0244-y.

[14] Hansen, P.—Mladenović, N.: Variable Neighborhood Search: Principles and Applications. European Journal of Operational Research, Vol. 130, 2001, No. 3, pp. 449–467, doi: 10.1016/s0377-2217(00)00100-4.

[15] Hoos, H. H.—Boutilier, C.: Solving Combinatorial Auctions Using Stochastic Local Search. Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI), 2000, pp. 22–29.

[16] Irnich, S.: A New Branch-and-Price Algorithm for the Traveling Tournament Problem. European Journal of Operational Research, Vol. 204, 2010, No. 2, pp. 218–228, doi: 10.1016/j.ejor.2009.10.024.

[17] Kendall, G.—Knust, S.—Ribeiro, C. C.—Urrutia, S.: Scheduling in Sports: An Annotated Bibliography. Computers and Operations Research, Vol. 37, 2010, No. 1, pp. 1–19, doi: 10.1016/j.cor.2009.05.013.

[18] Khelifa, M.—Boughaci, D.: A Variable Neighborhood Search Method for Solving the Traveling Tournaments Problem. Electronic Notes in Discrete Mathematics, Vol. 47, 2015, pp. 157–164, doi: 10.1016/j.endm.2014.11.021.

[19] Khelifa, M.—Boughaci, D.: Hybrid Harmony Search Combined with Variable Neighborhood Search for the Traveling Tournament Problem. In: Nguyen, N. T., Iliadis, L., Manolopoulos, Y., Trawiński, B. (Eds.): Computational Collective Intelligence (ICCCI 2016). Springer, Cham, Lecture Notes in Computer Science, Vol. 9875, 2016, pp. 520–530, doi: 10.1007/978-3-319-45243-2_48.

[20] Kuester, J. L.—Mize, J. H.: Optimization Techniques with Fortran. McGraw-Hill, New York, 1973.

[21] Miyashiro, R.—Matsui, T.: Semidefinite Programming Based Approaches to the Break Minimization Problem. Computers and Operations Research, Vol. 33, 2006, No. 7, pp. 1975–1982, doi: 10.1016/j.cor.2004.09.030.

[22] Mladenović, N.—Hansen, P.: Variable Neighborhood Search. Computers and Operations Research, Vol. 24, 1997, No. 11, pp. 1097–1100, doi: 10.1016/s0305-0548(97)00031-2.

[23] Rasmussen, R. V.—Trick, M. A.: The Timetable Constrained Distance Minimization Problem. Annals of Operations Research, Vol. 171, 2009, No. 1, pp. 45–49, doi: 10.1007/s10479-008-0384-4.

[24] Ribeiro, C. C.: Sports Scheduling: Problems and Applications. International Transactions in Operational Research, Vol. 19, 2012, No. 1-2, pp. 201–226, doi: 10.1111/j.1475-3995.2011.00819.x.

[25] Rossi-Doria, O.—Sampels, M.—Birattari, M. et al.: A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem. In: Burke, E., De Causmaecker, P. (Eds.): Practice and Theory of Automated Timetabling IV (PATAT 2002). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2740, 2003, pp. 329–351, doi: 10.1007/978-3-540-45157-0_22.

[26] Snyman, J. A.: Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer US, Applied Optimization, Vol. 97, 2005.

[27] Thielen, C.—Westphal, S.: Complexity of the Traveling Tournament Problem. Theoretical Computer Science, Vol. 412, 2011, No. 4-5, pp. 345–351, doi: 10.1016/j.tcs.2010.10.001.

[28] Van 't Hof, P.—Post, G.—Briskorn, D.: Constructing Fair Round Robin Tournaments with a Minimum Number of Breaks. Operations Research Letters, Vol. 38, 2010, No. 6, pp. 592–596, doi: 10.1016/j.orl.2010.08.008.

[29] Westphal, S.—Noparlik, K.: A 5.875-Approximation for the Traveling Tournament Problem. Annals of Operations Research, Vol. 218, 2014, No. 1, pp. 347–360, doi: 10.1007/s10479-012-1061-1.

Meriem Khelifa is a Ph.D. student at the University of Science and Technologies Houari Boumediene (USTHB). She received her Master degree in computer science from the University Kasdi Merbah, Ouargla, Algeria. She is an active member of the Laboratory for Research in Artificial Intelligence (LRIA). Her research interests include issues related to the meta-heuristic approaches, optimization problems, and the sports scheduling problem. She is an author of some scientific papers on sport scheduling and traveling tournament problems. Actually, she is working on developing new algorithms based on graph theory and machine learning algorithms to solve large sport scheduling instances at the HERON laboratory (Higher Education Research on Emotional Intelligence and Privacy Protection), UdeM Université de Montréal, Canada.

Dalila Boughaci is Full Professor in computer science at the University of Science and Technology USTHB (Algeria). She got her Ph.D. from the University of Aix-Marseille (France) in 2008 and her "Habilitation" Post-Doctoral diploma from USTHB University in 2009. She earned her Bachelor of Engineering degree in computer science from Algiers University, M.Sc. degree in computer science and her second Ph.D. in programming systems from the University of Sciences and Technology, Beb-Ezzouar, Algiers in 1997, 2001 and 2008, respectively. Her current research interests are in the areas of data mining, deep learning, evolutionary computation, artificial intelligence, meta-heuristics, multi-agent systems, network security, credit scoring and e-commerce. She has published several papers on these research topics in journals and conferences and directed several Ph.D., M.Sc. and B.Sc. students' projects. She has taught parallel computing, machine learning, web service, object oriented programming, algorithmic, software engineering, databases, Java, agents and programming languages at Algiers, and served on several program committees. She is a member of the LRIA Artificial Intelligence Laboratory at the University of Algiers. She is the head of the research team: Optimization, Reasoning and Application of the LRIA Laboratory.

Computing and Informatics, Vol. 37, 2018, 1411–1442, doi: 10.4149/cai_2018_6_1411

USING PROBABILISTIC TEMPORAL LOGIC PCTL AND MODEL CHECKING FOR CONTEXT PREDICTION

Darine Ameyed

Synchromedia Laboratory, Quebec University, École de Technologie Supérieure Montreal, Canada
e-mail: darine.ameyed.1@ens.etsmtl.ca

Moeiz Miraoui

Higher Institute of Applied Science and Technology of Gafsa, University of Gafsa Gafsa, Tunisia
e-mail: moeizmiraoui@gmail.com

Atef Zaguia

College of Computers and Information Technology, Taif University, Hawiyah Taif, Kingdom of Saudi Arabia
e-mail: zaguia.atef@tu.edu.sa

Fehmi Jaafar

Faculty of Management of Concordia University of Edmonton Canada
e-mail: fehmi.jaafar@concordia.ab.ca

Chakib Tadj

MMS Laboratory, Quebec University, École de Technologie Supérieure Montreal, Canada
e-mail: chakib.tadj@etsmtl.ca

Abstract. Context prediction is a promoting research topic with a lot of challenges and opportunities. Indeed, with the constant evolution of context-aware systems, context prediction remains a complex task due to the lack of formal approach. In this paper, we propose a new approach to enhance context prediction using a probabilistic temporal logic and model checking. The probabilistic temporal logic PCTL is used to provide an efficient expressivity and a reasoning based on temporal logic in order to fit with the dynamic and non-deterministic nature of the system's environment. Whereas, the probabilistic model checking is used for automatically verifying that a probabilistic system satisfies a property with a given likelihood. Our new approach allows a formal expressivity of a multidimensional context prediction. Tested on real data our model was able to achieve 78 % of the future activities prediction accuracy.

Keywords: Context prediction, logic, PCTL, pervasive system, context-aware system, stochastic, transition model

Mathematics Subject Classification 2010: 68T01, 68T30, 68T37, 68U35

1 INTRODUCTION

Prediction is a research topic in different fields: meteorology, economy, trends of prices and stocks as well as in computer science and software engineering such as predicting failure in software [1]. Predictive mechanisms help to anticipate actions and to implement the appropriate preventive measures. Ubiquitous computing systems are no exception in this respect; they do actually follow this trend. To be more proactive, ubiquitous systems have to provide service adaptation, according to the dynamic evolution of their context, in order to offer an adequate service fitting the user's needs.

One significant challenge, in particular, is to proactively assess the user's needs in the real world without requiring explicit input. Furthermore, a ubiquitous system must provide the user with services well adapted to the overall context. Indeed, services will be triggered dynamically and without an explicit user intervention in a proactive way. Making use of the context in applications is a current area of research known as "context-awareness" [2, 7]. A sensitive-context application must perceive the context of the users and their environment and adapt its behaviour accordingly. Most of the work on service adaptation in context-awareness is focused on the current context.

In ubiquitous computing several studies and research have been conducted too, under the prediction topic [2, 3, 4, 5]. These works aim to introduce new prediction techniques to increase the dynamic nature and the proactivity of those pervasive systems.

1.1 Problem and Motivation

Predict the future context allows the pervasive system to choose the most effective strategies to achieve its goals and to provide an active and fast adaptation to future situations.

However, the existing approaches face key issues that need to be addressed:

1. provide a multi-dimensional context prediction,
2. support a temporal constraint and identify the expected time of context variations,
3. improve expressiveness and provide a clear semantics.

Current approaches in context prediction only deduce one-dimensional information for the future context (e.g. future location). As a consequence, their expressiveness and effectiveness are limited. Even more so, if the system is unable to recognize the expected time of such context changes and the underlying behavior.

Moreover, these approaches face a common challenge: the lack of formal and general approaches for dealing with context prediction and more specifically, allowing proactivity and service anticipation using context prediction. They assert the lack of a common development framework for context prediction as well as formal representation for the context and a formal approach for the prediction.

Over the past few years, a more general research trend emerged, focusing on context prediction such as the work described in [5, 6], which discussed directions for research on this issue. They pointed out that the work in this area is mostly limited to location information, and a challenge they face is:

1. to consider more general context information,
2. to be able to support a temporal constraint and
3. to provide a logic-based expressive prediction with a clear semantic and formalism.

1.2 Proposition

Pervasive proactive systems need, therefore, the ability to reason with time dependencies and even more complex than that: spatiotemporal dimensions and the overall context. To be able to recognise a future contextual information (e.g., where is the location of the user X in the next 5 minutes?) and to provide an answer and anticipate a service associated with a future context must be possible (e.g., activity X can be executed on location Y in the next Z minute). A system that can include this kind of knowledge provides more flexibility and allows the ability to act in a more efficient manner.

In previous research work, we emphasized on context prediction context in pervasive context-aware systems. We proposed a new definition that supports prediction

in the same multi-dimension reasoning [8]. In another step towards the goal of providing formal prediction approach to context modeling we proposed a logic-based model including a temporal constraint [9]. This paper is, therefore, another step to provide a new spatiotemporal expressive prediction based on a formal semantic of probabilistic temporal logic and stochastic transition model.

1.3 Contribution

The efficient deployment of a context-sensitive prediction, its dynamic and unpredictable evolution, is still limited due to a semantic gap between the data provided by the physical detection devices and the information needed to predict the future behavior of the system and its users. Our proposed approach exceeds the weaknesses identified in the literature [5, 10, 11] by providing: better context expressivity, more efficient prediction based on logical reasoning, stochastic, non-deterministic modeling and below a multidimensional approach, what fitting better the nature of ambient systems.

In this paper, we are formalizing a new approach to express context prediction in context-aware systems. We express context and the transition in a pervasive system with a formal semantic, using a probabilistic temporal logic PCTL (a probabilistic extension of temporal logic). We propose a probabilistic transition model to encode the system's behavior over the time. Combining PCTL with a stochastic model, we can trace, analyze and predict the future context. Thus, we propose to use the model checking verification to verify the future state properties with a quantitative result and return the future state that has the maximum probability.

1.4 Paper's Structure

The paper is organized as follows. First, we give an overview of the available prediction methods (Section 2) with a synthesis and an evaluation. After that, we present our approach (Section 3) starting with a presentation of temporal logic and an explanation of the choice of probabilistic temporal logic. We then present a model detailing each included component. And we finish this section by explaining the prediction process. Before concluding the paper, we present the evaluation of our approach (Sections 3.6, 3.7) and expected future work (Section 4).

2 RELATED WORK

In this section we give an overview of the available research within the context prediction topic, specifically including proactive adaptation for pervasive systems; we analyze, discuss those various works, and later we present an evaluation/synthesis according to a selected set of criteria. As we have discussed and analyzed the prediction research work in a previous survey [11], according to the technical prediction approaches, we tried in this overview to discuss other related work, mostly from recent research in chronological order.

Also, we circumscribed a survey to research proposing generic models to support context prediction. Hence, the chosen works should support generic context information: works specifically devoted to the location prediction were not considered relevant. As discussed in recent surveys [11, 12] the development of generic approaches is a challenge in this research area.

One of the first contributions in context prediction was proposed by Mayrhofer [13]. Mayrhofer proposed architecture and a framework for context prediction that are based on an unsupervised classification, attempting to find context clusters, previously unknown from the input data. These context clusters represented recurring patterns in the input data. This approach modeled the context as a finite sequence of states where a user or a device triggers the change of the current state from one state to another. This modeling helped to predict the next states of the context based on the current state. He suggested a five-step process, taking sets of observations, each recorded at a specific time, as input and providing as output the current context of the user as well as predicting the future states of the context. The proposed stages are sensor data acquisition, feature extraction, classification, labeling, and prediction.

Mayrhofer proposed a prediction module based on the sequence prediction technique. This technique is based on the prediction task of a theoretical computer sequence and can only be applied if the context is broken down into some form of event flow. The context prediction in this work is based only on high-level context, and the framework does not have any mechanism to support an adaptive strategy.

Like Mayrhofer, Sigg et al. [14, 6] provided a formal definition for the context prediction task relevant to the issues raised on the quality of the context and on how to handle the ambiguity of incomplete data. This method is also based on patterns of context the learning algorithm builds to enable the prediction module. The context prediction module is based on an alignment method that attempts to predict the most likely continuation of a time series starting from the suffix of the observed sequence.

Finally, Sigg et al. [6] also offer a continuous learning module to adapt to the change in the environment or user habits. It continuously monitors the recorded time series stored in context history and updates the relevant patterns.

However, we did not find in this work any specific implementation for this learning module. Only its constraints were given, including the interface specified by the context history and language description of the rules, representing patterns. Sigg does not describe any adaptive mechanism for prediction neither considers any specification for context information.

Meiners et al. [15] suggested a context prediction approach called SCP (Structured Context Prediction). This approach is based on two key principles. The first is making use of knowledge of the application domain that developers can integrate when designing the application. This knowledge is described as a prediction model that specifies how the predictions are to be executed and which configures the prediction system. The second principle sets out the application of several prediction methods, which are interchangeable. These methods are proposed to ensure the

accuracy and effectiveness of predictions relevant to a given domain. They can be selected and combined by the application developers. According to [15] the prediction model assigns a method for each variable to predict its value. The method uses as input the values of other variables that are either already predicted by their own methods, or simply measured by sensors. Also, the authors proposed an architecture for a prediction system which can be used as a reusable component by context-aware applications.

In this work, the proposed Contexts Prediction architecture supports an adaptive mechanism for contexts prediction. However, this mechanism is manual, that is, the designer needs to choose at design time the most suitable algorithms for predictions.

Furthermore, the architecture also has a learning component and supports only low-level context data and does not have a formal context representation.

Contextual spaces theory is an approach developed by Andrey Boytsov [3], to best define context-awareness and to deal with sensor problems that create uncertainty and incur a lack of reliability. This theory used spatial metaphors to represent the context as a multidimensional space. It was designed to make context-awareness clearer.

The theory of context space was initially submitted by Padovitz and Zaslavsky [16]. The authors attempted to provide a general model to help thinking about and to describe the context and develop context-aware applications. This work will be later the basis for several researches of Zaslavzky and Boytsov [4, 3, 17]. Boytsov and Zaslavsky presented the CALCHAS system, which offered context prediction and used an extension to the context space theory to provide proactive adaptation.

This approach addressed the context prediction problem in a general sense. In context spaces theory several methods were tested and used for reasoning about the context. The authors judged sequence technique as the most prospective prediction approach.

For adaptation mechanisms, algebraic operations on situations and some logic-based methods were developed for reasoning regarding situations [18].

This works had presented a general framework model, included an adaptation approach based on prediction but did not propose a new formal or a generic prediction method.

In her work on services prediction, Salma Najar offered a mechanism of discovery and prediction guided both by context and user intent [19]. She used semantic similarity techniques. The system is based on the implementation of a matching algorithm, which computes the matching degree between the intention and the current context of the user and the set of semantic services described accordingly. OWL-SIC (OWL-S Intentional & Contextual) is an extension of OWL-S (Web Ontology Language-Semantic, is an ontology, within the OWL-based framework of the Semantic).

The similarity approach required historical data, to select and recommend services that are not always available. In fact, it needs a first phase of a collection to get enough data which will be processed after that. The intentional approach provided by Najar [19] was a user-centered approach but can generate

conflict: for instance a problem of interoperability between services. Indeed, two compatible intentions do not necessarily map to two technically compatible services. This work also proposed a conceptual framework focused on services prediction.

Joao et al. [5] proposed new framework including a prediction-algorithms library. They named the proposed model ORACON. The architecture of this model is based on the Model-View-Controller (MVC) design pattern. It has three layers, two agents, one library of prediction algorithms, External Histories, External Ontologies, and External Applications. ORACON proposed prediction of entities. An entity, in this sense, can be a living being, an object or even a location. Each entity can have many applications, modeled as External Applications, which can interact with the model in order to obtain predictions. This work focused more on the framework; it did not propose a specific prediction approach. There prediction algorithm library contains four prediction approaches: alignment, enhanced alignment, semi-Markov and collaboration [5]. This proposed model was an interesting work which can be enhanced with many extensions to improve the performance, increase the accuracy of classification and optimize the processing time.

Föll et al. [20] proposed a PreCon as a multi-dimensional context predicting method, composed of three parts: a stochastic model to represent context changes, an expressive temporal-logic query language using CSL (continuous stochastic language) and stochastic algorithms to predict the context. The model based on user behavior was presented as an SMC (Semi-Markov Chain).

This work was the unique formal work using the CSL as a query language of the system, and a Semi-Markov Chain. There is also another work that had tried to automate the recognition of activities using the LTL formalism with a model checking [21].

They concluded their work, noting that a probabilistic extension using a PCTL can increase the expressive power of the formal core.

We found this to be the most relevant work, and we based our approach on it, specifically in a model checking verification. We use PCTL formalism and include action in a model to get a more descriptive model.

2.1 Synthesis

Table 1 summarizes a comparison of the related works. As we can see the majority of works do not support formal representation of the context, low and high context level. They focused more on providing a framework including a predictive module, rather than on the prediction module itself. The essential part of a prediction model being the approach used in the prediction process itself.

Ubiquitous environments are highly dynamic, that is, applications can interact with a great number of different and unknown applications all the time [22, 23, 24]. Hence, it is essential to define a formal representation for the context, so that different systems can easily communicate. Thus, specifying a context representation is considered a key feature for model prediction. This is why we choose

	Adaptive strategy	**Context formal presentation**	**Low-high context level**	**Learning capability**	**Prediction technique**	**Framework proposed**
Mayrhofer 2004	no	no	no	yes	sequence prediction approach	yes
Sigg 2008–2010	no	no	yes	yes	trajectory prolongation	yes
Meiners 2010	yes	no	yes	yes	Bayesian	yes
Boytsov 2011	yes	yes	no	yes	sequence predictor the most perspective approach	yes
S. Föll 2014	no	no	yes	yes	temporal query prediction	no
S. Najar 2014	yes	no	no	yes	semantic similarity (discovering)	yes
H. Joa 2016	yes	yes	yes	yes	alignment semi-markov	yes

Table 1. Comparative overview of context prediction research work

a formal context representation based on a logic perspective [9]. Also, we build a model in a temporal logic formalism providing clear formal semantics by using a probabilistic temporal logic (PCTL), and we propose a new probabilistic-labeled transaction model Model-LPTM. One might also conclude that the prediction approaches supported by previous works compute the most probable future context, based on simple uni-dimensional context information. Existing systems do not allow a formal context prediction through temporal-semantics and multidimensional processes.

In this paper, we propose to investigate the application of probabilistic temporal logic as a powerful formal presentation for context prediction. It also proposes a formal prediction approach based on temporal logic in a multidimensional context space and on a new formalism that integrates probability and labeling; which provide a new probabilistic labeled transaction model thus helping effective context-aware prediction.

3 THE PROPOSED APPROACH

3.1 Temporal Logic in the Context Aware System

Time is a fascinating subject. We are moving through time continuously, and in order to survive and manage ourselves, we regularly have to make temporal-logic-based decisions. In daily lives, people are using time-dependent information, e.g. when to go to the dentist? When is a meeting to be held? With the rise of ubiquitous systems (which ideally aim to provide a smart user-focused service; like reminder services, assisted-living services and more), temporal analysis and reasoning appear best-suited to ensure the proper functioning for this kind of system. Temporal logic can also be used as a programming language. The basic paradigm is to review the past and then take action in the future. Abstractly we have an initial state and certain actions that can be performed in a given state if it satisfies a certain set of conditions. Performing an action on a state produces a new state.

We have defined a variant of TL (temporal logic) as a language for the specification of each situation and its related context. In general, TL has been developed and applied as a formalism for reasoning about the ordering and quantitative timing of events [25]. Several formulations have been proposed to satisfy the needs of different contexts. TL may be classified according to the underlying nature of time: linear temporal logic LTL and computational tree logic CTL.

LTL, CTL and CTL* can express qualitative properties of a system. Real systems such as a pervasive system, however, are quite often characterized by non-deterministic behavior and this is because of the human presence. In order to provide efficient services, to be user-centric and more realistic, those systems should be attuned to the unpredictable behavior of humans. Taking probabilities into account, in addition to non-deterministic behavior, would expand this aspect of the system allowing the quantification of unpredictable behavior, if the specification holds with an arbitrary probability value and within a given time limit.

We propose to use PCTL, which had the expressive power of probabilistic temporal logic (it introduces probability to extend CTL which is inadequate in dealing with a real-life system like a ubiquitous computing system) (Figure 1).

3.2 Probabilistic Temporal Logic Specification

Temporal logic extends the traditional modal logic to allow the description of when a formula is true. That is, rather than just "necessity" or "possibility", a formula may be true at the next point in time or at some other point in the future.

Branching time logic, such as Computation Tree Logic (CTL) [26], enables the choice of a path among multiple possible paths in a tree structure describing probable future events. So that, each choice has to mirror the possible set of behaviors starting from the current state. As opposed to linear-time temporal logic, for which, there is only one possible future path, we can express whether a property holds for all possible paths (A formula), or if there exists at least one path for which it is true

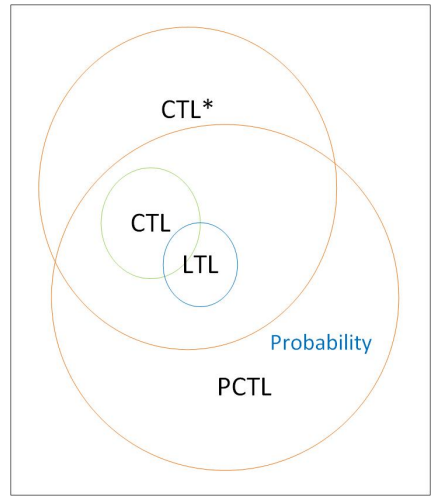


Figure 1. Expressivity CTL vs. LTL vs. CTL* vs. PCTL

(E formula). The values of these formulas are determined with a Kripke structure: a graph with a set of states, transitions between states, and labels indicating which propositions are true within the states.

We will use a probabilistic extension of CTL, Probabilistic Computation Tree Logic (PCTL) [27, 28] as it allows probabilistic state transitions, as well as explicit deadlines for when a formula must hold.

The proposed PCTL syntax is based on the syntax and semantics proposed in [27, 28]. For the sake of clarity, some specific notations, as well as the underlying probabilistic model, have been slightly modified from the original syntax presented by those papers, in order to adapt them to the work context.

In this section, we present the proposed model for the context prediction problem based on the real-world situation and the related features; which represent the contextual information of each situation (e.g. location, time, occupation, ambient information, sound, temperature, etc.).

Figure 2 summarizes the proposed approach. It is based on PCTL formalism, a probabilistic labeled transition model which will be detailed later (Subsection 3.3). The context prediction is based on model checking, which will return the future situation and its probability.

A. Formalism

a. Context

Definition 1. In order to specify this situation-context, let $s = (c_1, c_2, \ldots, C_n) \in S$, s being an n-dimension vector of context information described by a preposition or a combination of prepositions, where each component c_i of s

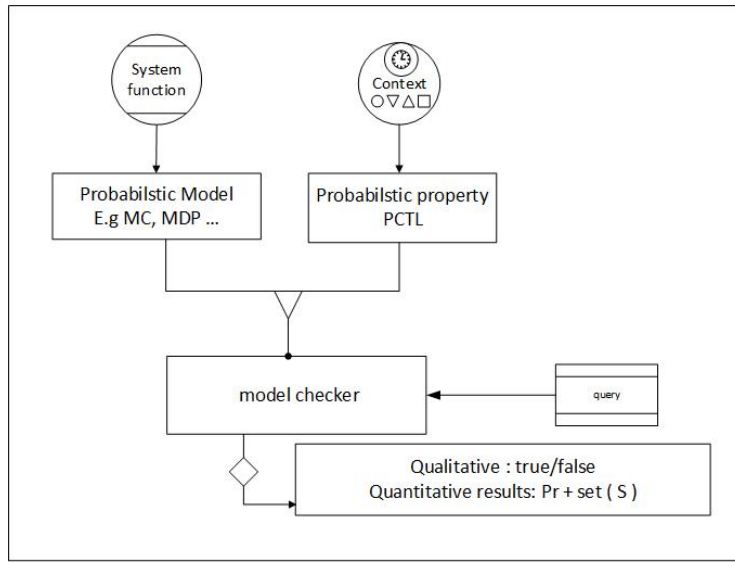


Figure 2. Overview of the proposed approach

is of a specific context type C_i (e.g. $\langle$location$\rangle$, $\langle$occupation$\rangle$, ...). A state s can be multidimensional and expresses composite contextual data describing the features of a specific situation; e.g., $s = ((\text{meeting-room})x(\text{power-point}) \times(\text{occupation} = 5))$ designates the presentation situation on a meeting room.

For each new combination of context information $(c_1, \ldots, c_n)$ that has not been observed before, is detected, a new state s will be inserted into the model and labeled with $(c_1, \ldots, c_n)$. For more details about the context logic-based modeling we refer to previous related work [9].

b. Path and state

The prediction semantics is based on PCTL syntax. For this let $p \in [0, 1]$ be a probability, let $\mathbf{t} \in \mathbb{R}^+$ be a time-bound, and let (C_i, c_i) be a contextual value c_i of type C_i as defined earlier.

Definition 2. Path formulas express the properties and behaviour allocated to paths.

$$\varphi = X^{\leq t}\Phi | \Phi_1 \bigcup^{\leq t} \Phi_2.$$

State formulas express the properties and behaviour allocated to states

$$\Phi := tt | ff | (C_i, c_i) | (A, a) | \neg\Phi | \Phi_1 \vee \Phi_2 | \Phi_1 \wedge \Phi_2 | P \sim p(\varphi)$$

where $c_i \in 2^{AP}$ AP a set of atomic propositions describing situation context (e.g. location: $\langle$meeting room$\rangle$, light: $\langle bright \rangle$, occupation: $\langle 3 \rangle$, application-

running: ⟨power-point⟩), $a \in A$ is a finite set of actions, and $\sim$ is a comparison operator $\sim \in \{\langle, \rangle, \leq, \geq\}$, and p is a probability threshold $p \in [0, 1]$. Path quantifiers as in PCTL are built from one of the temporal modalities: X (next) or U (until) (Table 2). t is a time constraint defining an upper bound on a time interval to describe the duration of a situation, the subsequent transition and when an action will be active.

Quantifiers over Paths	
$A\Phi$ – All	Φ has to hold on all paths starting from the current state.
$E\Phi$ – Exists	There exists at least one path starting from the current state where Φ holds.
Path-Specific Quantifiers	
$G\Phi$ – Globally	Φ has to hold on the entire subsequent path
$F\Phi$ – finally	Φ eventually has to hold
$X\Phi$ – Next	Φ has to hold at the next state
$\Phi U \psi$ – Until	Φ has to hold at least until at some position ψ holds. ψ will be verified in the future

Table 2. Paths quantifiers

Considering Φ a state formula expressed as a pair (C_i, c_i), which describes the type of context and the specific context value in this state (e.g.: location, meeting-room). We leverage these operators to analyze the future context behavior;

- F is the Eventually operator used to verify if a condition ϕ eventually has to hold in any state from s somewhere on a subsequent path in the model.
- G is the Globally operator, and it can be used to check if the condition ϕ holds in every state on all subsequent paths starting in s.
- X is the Next operator: it evaluates a condition ϕ on all immediate successor states to the current state s. It has to hold at the next state (this operator is sometimes noted N instead of X). Since we focus on immediate prediction, we will build a prediction model on this operator in this paper.
- U is the Until operator and expresses that Φ_2 will be verified in the future. And Φ_1 has to hold starting at the current state at least until at some further position Φ_2 holds.

The PCTL state formula $P \sim p(\varphi)$ asserts that, under all schedulers [28], the probability for the event expressed by the path formula φ meets the bound specified by $\sim p$. The probability bounds "$\sim p$" can be understood as quantitative counterparts to the CTL path quantifiers $\exists$ and $\forall$.

B. PCTL Semantic

Definition 3. Let $M = (S, AP, L)$ be a PCTL model, s is a state $\in M$, AP a set of an atomic preposition, L is a labeling function, and ϕ is a PCTL formula. The satisfaction relation is noted as $M, s \vDash \phi$.

Let s be a state, $s \in S$ we can define the satisfaction relation for state formulas as follows:

- M, s true $\forall s \in S$,
- $M, s \vDash c_i \Leftrightarrow c_i \in L(s)$,
- $M, s \vDash \neg\phi \Leftrightarrow M, s \nvDash \phi$,
- $M, s \vDash \phi_1 \wedge \phi_2 \Leftrightarrow M, \vDash \phi_1$ and $s \vDash \phi_2$,
- $M, s \vDash \phi_1 \vee \phi_2 \Leftrightarrow M, \vDash \phi_1$ or $s \vDash \phi_2$,
- $M, s \vDash P \sim p(\varphi) \Leftrightarrow P\{\pi \in Paths(s) | M, \pi \vDash \varphi\} \sim p$.

The satisfaction relation for path formula is defined inductively as follows:

- $M, \pi \models X\Phi \Leftrightarrow \pi = s_0 \rightarrow^{a_0,t_0} s_1 \rightarrow^{a_1,t_1} \ldots s_n \rightarrow^{a_{n-1},t_{n-1}} s_n$ and $M, s_1 \models \Phi$,
- $M, \pi \models \Phi_1 U \Phi_2 \Leftrightarrow \pi = s_0 \rightarrow^{a_0,t_0} s_1 \rightarrow^{a_1,t_1} \ldots s_n \rightarrow^{a_{n-1},t_{n-1}} s_n$ and $\exists k.M$, $s_k \models \Phi_2$ and
- $\forall j < k.M, s_j \models \Phi_2$.

C. Labeled Probabilistic Transition Model: Model-LPTM

A pervasive system follows various behavioral patterns depending on user's behavior. Those patterns cannot be described in a deterministic way. Hence, our choice of a probabilistic non-deterministic model. In the following, we give a description of this model and the proposed approach to predicting the next situation using this formalism.

We represent an LPTM model as a transition system which combines probabilistic choice as in Markov chains with a non-deterministic choice. We define the model with a timed probabilistic transition based on models defined in [27, 28]. The model integrates time and action and will be presented as follows.

Definition 4. Let LPTM be a Kripke (S, A, P, L): a labeled transition probabilistic model defined as follows:

- S: a finite set of states where $s \in S$ and $s_{init} \in S$,
- Act: a finite set of actions where $a \in A$ and $A \subseteq Act$,
- L: $S \rightarrow 2^{AP}$ state labeling function assigning to each state one or several atomic prepositions $\in AP$,
- $P \subseteq S \times A \times \mathbb{R}^+ \times Dist(S)$ is the function assigning a probabilistic transition distribution, such that if $(s, \delta t, a, \rho) \in Dist(S)$ and $\delta t > 0$ after a span time $\triangle t$ in a situation s was spent and a is an active $\in A(s)$ then ρ is a point distribution.

As for probabilistic systems, we can introduce paths for timed probabilistic systems except that transitions are now labeled by a (duration, action, distribution) tuple. Each transition is labeled by a tuple $(\delta t, a, \rho) \in Dist(S)$, where:

- δt is the time span between s_i and s_j (Section 3.6),
- $P(s_i, s_j)$ is the probability assigned to the path transition between s_i and s_j (Section 3.4),
- $a \in A(s_i)$ is an action active between two states s_i and s_j (Section 3.5).

Our contribution using this model consists in considering every $s_i \in S$ as described by a set of context parameters ($c_i \in C_i$) such that $L(s_i) = c_i$ and an action for a transition path with a temporal duration constraint δt.

To avoid transient states, we choose to integrate them as proposals in paths. Thus, the path describes a transient context as an accomplishment action or activity action (see Section 3.5). That can be part of the next state. This makes the modeling more context-aware and proactive.

Using this LPTM, we can formalize the behavior trace and context variation by an infinite state tree like in MDP. The context can be a composite context. The variation of one or several context's element introduces changes on the state. We can describe a pervasive environment according to the user's behaviour with action semantic (Section 3.4), and context variation, at each spatiotemporal interval, we have an active state describing a specific context $s_i \in S$. While the user (e.g.: walking, driving, be, ...) or the environment and the system environment (running process, etc.) act, the context changes and the LPTM moves to the new state $s_j \in S$ expressing the property of new context. This successor state s_j is visited with a probability $p(s_i, s_j)$. Before leaving the current state s_i, the context does not change and stay active for a limited duration of time δt spent in s_i. Example: model (Figure 3).

Explanation: To lead the next situation from the current situation i to the next one j we count:

- $a_{s,n}$ represents an action active for a given state (e.g., a_{01} describes the active action from S_0 to S_1),
- δt_{ij} represents the time span between s_i and s_j (e.g., δt_{01} describes transition duration from S_0 to S_1),
- P_{ij} refers to the transition probability from the situation s_i to the situation s_j such that $\Sigma_j P_{ij} = 1$.

a. Transition Probability

For each transition (S_i, S_j), the transition probability will be:

$$p(s_i, s_j) = P(X_{n+1} = s_j | X_n = s_i) \tag{1}$$

where X_n is the random variable that models the stochastic behavior at the current state and X_{n+1} model the stochastic behavior at the next state.

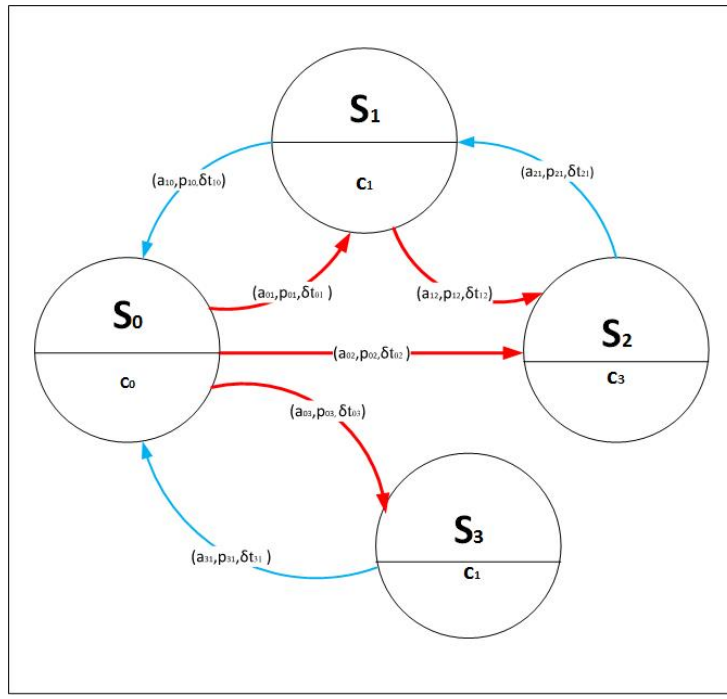


Figure 3. Transition model

Recall that the formulas are defined about a probabilistic structure, as described earlier. While the used structures consist of labeled states and path, they only imply that it is possible to transition from the state at the tail to the state at the head with some non-zero probability.

We express a model as a causal model. In this paper, we assume a dependent relation between current state and the next one. The probabilistic transition

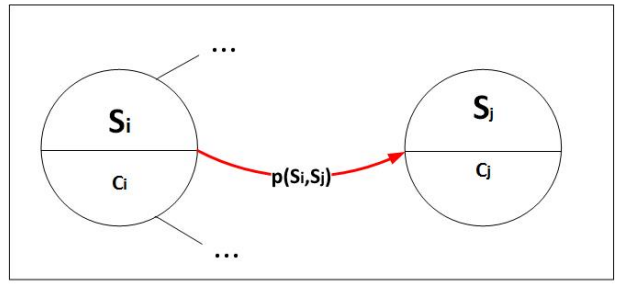


Figure 4. State transition probability

depends only on the current state s_i and s_j is independent of all previous state changes.

The transition probability and the set of prepositions describing contextual feature situations can be estimated and deduced from the history of past trace of state transitions and their linked contextual features.

As in statistic computation, let the transition weight be ω_{ij}, which defines the number of transitions observed from s_i to s_j. The transition probability is calculated as follows:

$$p(s_i, s_j) = P(X_{n+1} = s_j | X_n = s_i) = \frac{\omega_{s_i,s_j}}{\sum_{n \in S} \omega_{s_j,s_n}}. \tag{2}$$

The probability of transition between two states is the ratio of the number of observed state transitions from s_i to s_j to the number of all observed transitions from s_i.

Example: We have a current state s_0 that can lead to any of the immediate next states as in Figure 5 as a distributed probability.

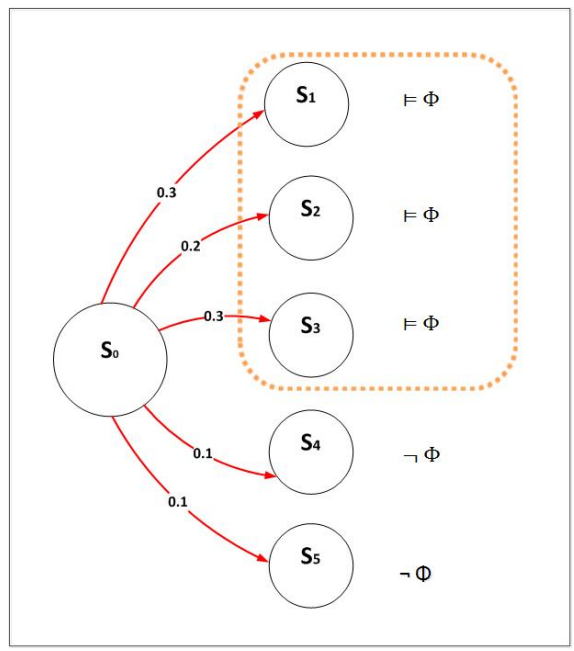


Figure 5. State transition

The probability without any constraint of time or action to lead to any next state when ϕ will be verified as a Next (optimal) in S_1, S_2 and as a Next (all) in S_3 be S_3 or S_1.

b. Action

Observing the system's and user's behavior, we also noted information describing actions which influence a service process and make a change in a situation in the feature context. Based on reference work discussing the linguistics of time and the semantic verbs and time [30, 31], these actions can use the aspectual verbs according to the categories in Table 3.

	Expressivity	Dynamic	Durative	Telic (bound)
Accomplishment	Describing Durative action Ending by a culmination point	Yes	Yes	Yes
Activity	describing durative action	Yes	Yes	No
State	Often durative	No	Yes (temporary state) No (permanent state)	Yes
Achievement	Change of state near punctual duration	Yes	No	Yes

Table 3. The four aspectual categories

In the proposed model we can use accomplishment and activity to describe a transition over a path and a state and achievement in a situation (node).

The computation over the proposed model we use the accomplishment-action on the path because we are reasoning in a dynamic system with a time-bound and we count the durative actions in a bound time during a transition. We can label a graph with state-action and achievement to clearly describe a scenario or an example.

In the proposed model, actions depend on transition and describe a transition over a special path. The set of actions available at $s \in S$ is denoted by $A(s)$. For each action $a \in A(s_i)$, the probabilities can be estimated as other observations from the history of past trace. We count the probability of transitioning from s_i to s_j under the action a, and we denote this probability by $\alpha_a^s i(s_j)$. We refer to [32] for more details about computation in mapping and learning steps.

Example: We have a set $A(s_0) = \{a_1, a_2\}$ and a transition and s_0 can lead to any of the immediate states as in Figure 6.

In this example, the probability next ϕ to occur with any action $a \in A(s_0)$ is

$$\frac{\sum_{s_j \in S \wedge s_j \models \phi} \alpha_a^{s_i}(s_j)}{\sum_{s_j \in S} P(s_j, s_i) \cdot \sum_{s_j \in S \wedge s_j} \alpha_a^{s_i}(s_j)} = 0.45,$$

the optimal next will be the path with a strategy probability ≥ 0.45 in this case that will be the transition (S_0, S_2) under the action a_2.

3.3 Space Time Duration

We will show how we can estimate the time span between s_i and the next s_j. The time was considered in the model as the constraint parameter for states as well as

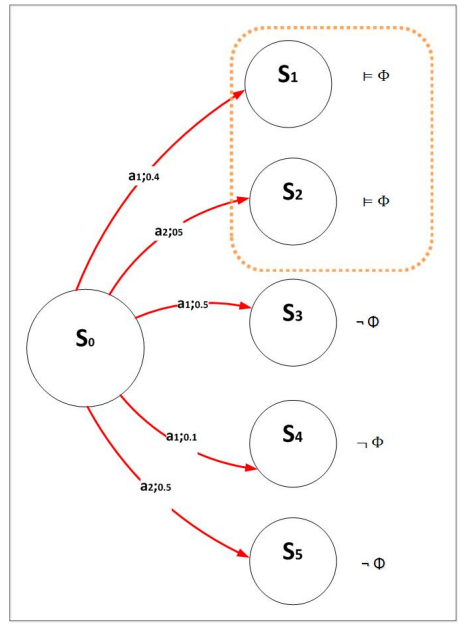


Figure 6. Action transition

transitions (path), as described in a previous contextual definition and model. Every situation has a time interval describing its start time and end time which can be useful as a learning data base [24, 8].

We express the time span as a probability function where μ and α are the mean and standard deviation values, calculated from the time span. In order to limit the computation, we consider in the current work only the observation falling with standard deviation

$$f(\delta t_{ij}, (\mu, \sigma)) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\delta t_{ij}-\mu}{\sigma}\right)^2}. \tag{3}$$

Figure 7 gives an example of transition time span: the typical time span falls in the following range.

We model the time span as a random variable D_n expressing the time spent between s_i and s_j. To figure out D_n, we observe the time periods spent between consecutive states transitions, and we associate an individual distribution to every transition between s_i and s_j. Formally the distribution can be presented as:

$$\Delta_{ij}(\delta t) = P(D_n = \delta t | X_{n+1} = S_j, X_n = S_i). \tag{4}$$

The cumulative distribution is given by $\triangle_{ij}$ which is given as $\int_0^b \triangle_{ij}(\delta t)\,\mathrm{d}\delta t$ and can be computed as the sum of probabilities associated with consecutive intervals up to a desired upper time bound b. The probability of a time span to

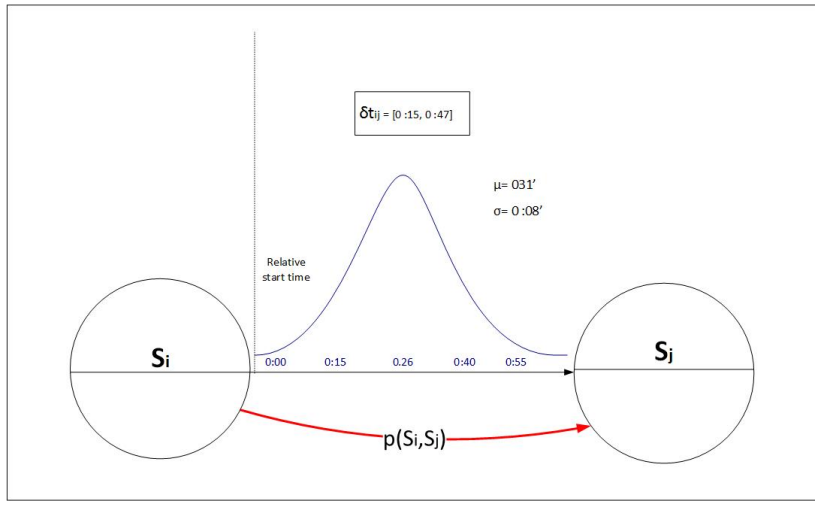


Figure 7. Transition span time

lie within the interval $[a, b]$ can be derived from the cumulative distribution as $\int_0^b \triangle_{ij}(\delta t)\,\mathrm{d}\delta t$.

3.4 Immediate-Context Prediction Processing

The state space can be traversed by going from one state to the next as allowed by transitions among states. The resulting series of visited states (path) models one possible spatiotemporal behavior of context. For context prediction we start at the state $s_i \in S$ occupied in the real world, and we evaluate the possible path starting at s_i and leading to the next state s_j. The state and path follow the PCTL semantic, as explained in Section 3.2.

In the proposed model we can evaluate a satisfaction relation for the path formula as follows:

$$X_{n+1} \leftarrow \mathrm{argmax}_{X_{n+1}} P(X_{n+1} = s_j | X_n = s_i, a \in A(s_i)). \tag{5}$$

The path formula φ is satisfied after $\triangle t$ unit of time elapsed in a situation s and under an action a if and only if the probability $P((s, a, \triangle t) \vDash \varphi)$ satisfies the threshold $\sim p$.

In our case, we need to be able to verify that a given state satisfies the context's state preposition $\phi = (C_i, c_i)$ (as described in Section 3.3). We also need to consider the temporal operator Next $P \sim p[X\phi]$ and define its probability computation.

Using a PCTL, we can investigate the reachability properties using the Next operator, evaluating a condition state formula ϕ, expressed over the contextual information (C_i, c_i), on all immediate successor states s_j of the current situation s_i.

Using this reasoning, the system is able to predict a variety of information about the context (e.g. the next location, the next activity, what time the user finishes work, at what time the next meeting starts, what is the optimal strategy to lead the next situation, etc.). The high threshold probabilities according to a special action describing a transition reduce the number of false prediction prepositions and make the prediction more efficient and more context-aware.

We can derive the prediction based on next operator in PCTL as explained in the next subsection using the verification algorithm in a model checking based on symbolic method [33, 34].

3.5 Computation for PCTL Next Operator

In this paper, we focus on immediate prediction. Thus, we will only use the next operator. In future work, we might extend the proposed approach with the two more temporal operators: (i) Until: $P \sim p[\phi_1 \cup \phi_2]$ and (ii) Bounded Until: $P \sim p[\phi_1 \cup^{\leq k} \phi_2]$ which can be useful for a long-term prediction.

The Next operator restricts the space of satisfaction property of path formula φ to the immediate successor the next state s_j of the current state s_i. We need to determine the Next (optimal) $\phi = P_{max=?}([X\phi])$ which is the maximum probability satisfying Next ϕ.

$$X_{n+1} \leftarrow \text{argmax}_{X_{n+1}} P(X_{n+1} = s_j | X_n = s_i). \tag{6}$$

Or the all Next (all) $\phi = P_{\bowtie\rho}([X\phi])$; here we can find all the policies that satisfy the next state with ϕ property, where:

$$P(X_{n+1} = s_j | X_n = s_i, a \in A(s_i)) \tag{7}$$

$$= P_{\langle a \rangle} \left(X_{\triangle t}^{\leq \delta t_{ij}}(\phi) \right) \tag{8}$$

$$= \frac{\sum_{s_j \in S \wedge s_j \models \phi} P(s_j, s_i). \sum_{s_j \in S \wedge s_j \models \phi} \alpha_a^{s_i}(s_j). \int_{\triangle t}^{\triangle t + \delta t_{ij}} T_{ij}(\delta t)\, \mathrm{d}\delta t}{\sum_{s_j \in S} P(s_j, s_i). \sum_{s_j \in S \wedge s_j} \alpha_a^{s_i}(s_j). \int_{\triangle t}^{\infty} T_{ij}(\delta t)\, \mathrm{d}\delta t}. \tag{9}$$

The optimization function $log(P(\phi|\lambda))$ is proposed to avoid data overflow in the computation of feed forward probability.

The prediction approach is based on the traces contained in the stochastic user model. The traces are used as a search space of possible context changes. Information about the recent sensed context changes (current state's context) is used to condition the prediction on what the optimal Next might be expected in the immediate future. Using a model based on statistical knowledge, the predictions in the proposed approach, work as a scanning process in a stochastic transition system to find the Next verifying the property expressed in the formula. A component diagram of the prediction model can be represented, as shown in Figure 8.

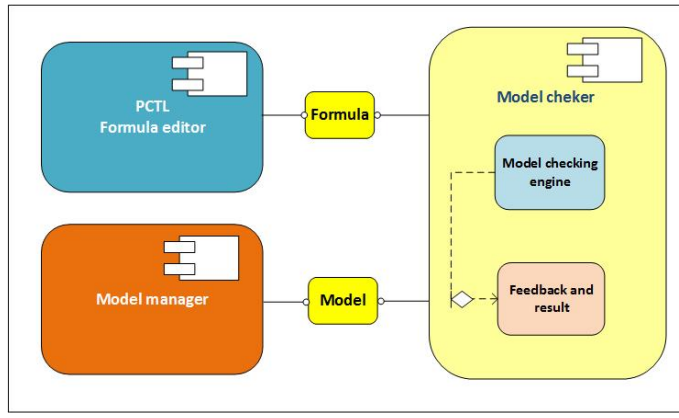


Figure 8. Component diagram of LPTM system

3.6 Use Case and Test

In this section, we present the experimental results for the proposed model. Before getting into the evaluation of the prediction model, we describe the data set we used [35, 36].

We use a real-world context traces from Domus smart home case study. The Domus smart home is one-bedroom apartment mounted inside the University of Sherbrooke. The apartment is equipped with different types of sensors. During the experiments, users have participated to evaluate the early morning routines, which correspond to the basic everyday tasks during the morning. The routine describes morning activities as follow: wake up, toileting, preparing breakfast, having breakfast and other activities. We use this study case to predict the Next activity. The activities we consider in the simulation are as follows: wake up, use toilet, preparing breakfast, having breakfast.

As a simulator tool, we use Petri nets, that means formal models of information flow which support timing specifications and a non-deterministic behavior for more details about tools we refer to [37]. We first model the prediction model as shown in Figure 9.

The model is composed mainly of:

- Generation: this module generates the current context and constraints as a random choice.
- Get related activity: the module gives the activity probability (Section 3.4).
- Get related activity Action: the module determines the action probability (Section 3.5).
- Get related activity time: the module defines the time span probability (Section 3.6).

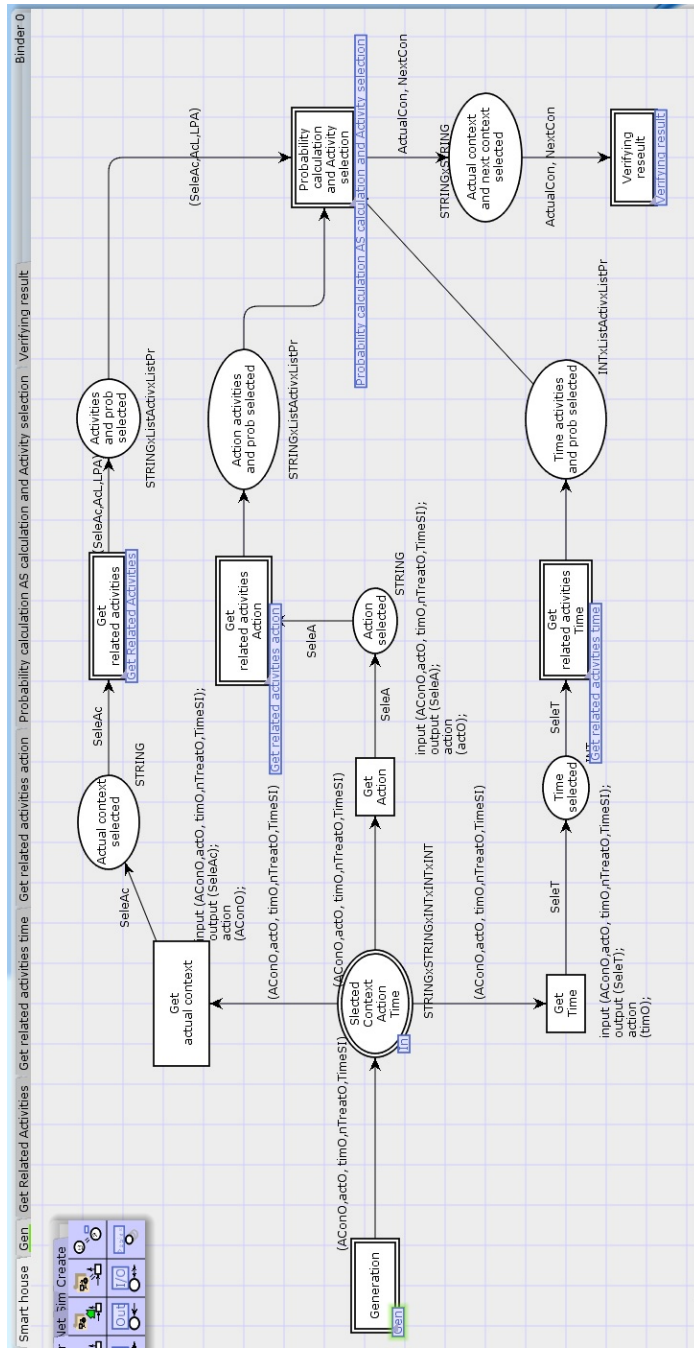


Figure 9. General view of prediction model

- Probability calculation: the module gives the probability of the most probable Next activity (Section 3.7).

The transition between different activities are learned based on the LPTM trace Model as shown in Figure 10.

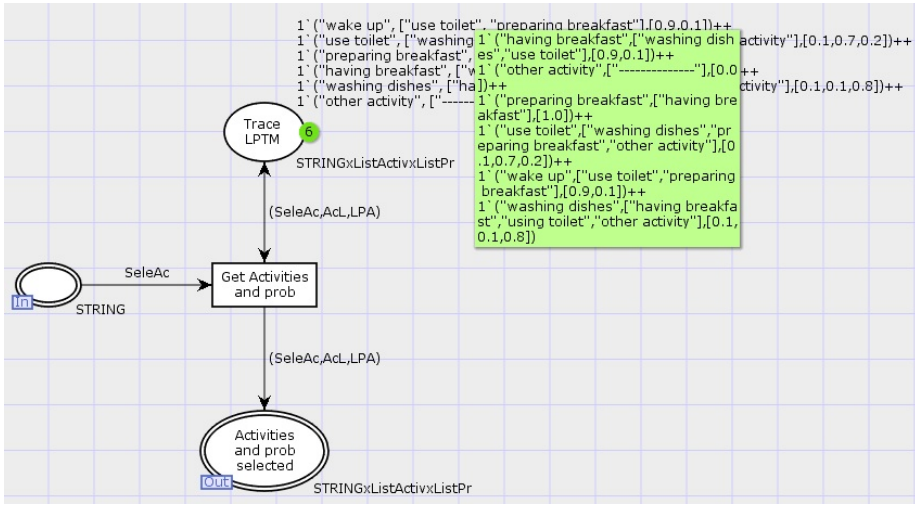


Figure 10. Activities transitions information

To recognize the next activity, we generate a random for a variety of activity (context value) time and action as shown in Figure 10. When an event is detected, this module generates automatically the actual context, the action and the transition time (Figure 10).

As we mentioned before, the Dumas data set that we used for actual context is limited to having breakfast, other activities, preparing breakfast, use toilet, wake up, washing dishes, for action is limited to (close door, open door) and for time is limited to $(5, 10, 15, 20, 25, 30, 35, 40, 45, 50)$ The outputs of this module are:

- The actual context used as input by the transition Get related activity to determine the activity probability.
- The action used as input by the transition Get related activity Action to determine the activity probability.
- The transition time used as input by the transition Get related activity time to determine the time span probability.

The transitions between different activities are learned based on the LPTM trace model, as shown in Figure 11. The input of this module is the actual activity selected randomly by the generator (Figure 10). According to this activity the transition "Get activities and Prob" selects the adequate activities and probabilities

from the place Trace LPTM. As output of this module, we have three parameters: the actual activity presented by the variable SeleAC as string, the list of activities presented by the list ACL and the list of probabilities presented by the list LPA.

For instance, if the actual activity is "Wake up" (SeleAC) then the output of this module will be ("wake up", ["use toilet", "preparing breakfast"], [0.9, 0.1]) the different probabilities in the place "Trace LPTM" are computed from dataset DUMAS. After we get all probabilities, the transition "Probability calculation and Activity selection" (Figure 9) determines the next activity (Section 3.7). Then the role of the transition "verifying result" to test the result generated by the transition "probability calculation and Activity selection" The actual activity and the next activity are the input of this module. We compare the result obtained by the values in the place "DB-Real-Flow Evidence".

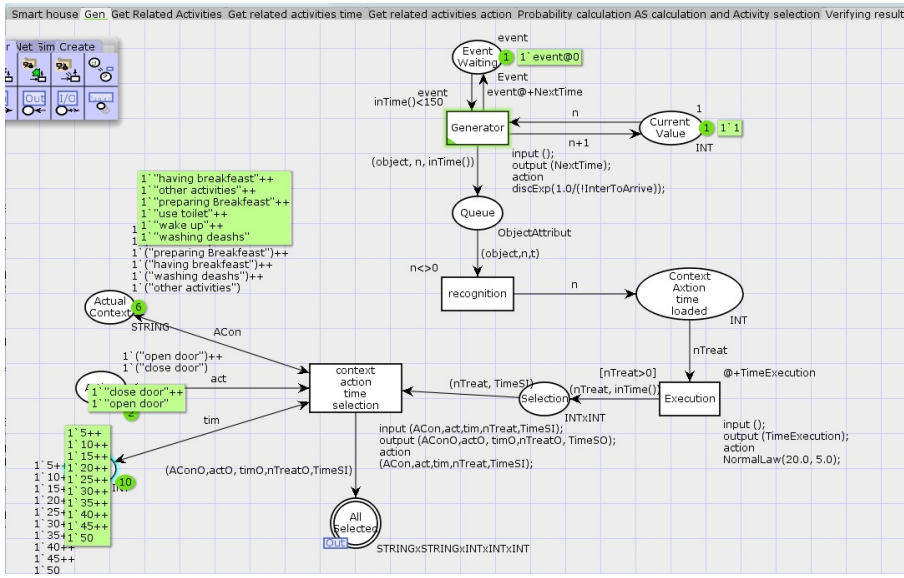


Figure 11. Action-time generators

Finally, after getting the Next activity identified, we evaluate the results based on real flow evidence as shown in Figure 12.

The diagram in Figure 13 resumes the prediction results for each activity. The average of the prediction model was 65 %, we also get 78 % in some activity, as shown in the following diagram.

3.7 Result Discussion

The accuracy criteria can usually be ranging from low/worst performance to high/best performance, depending on the capacity of the approach to be effective in

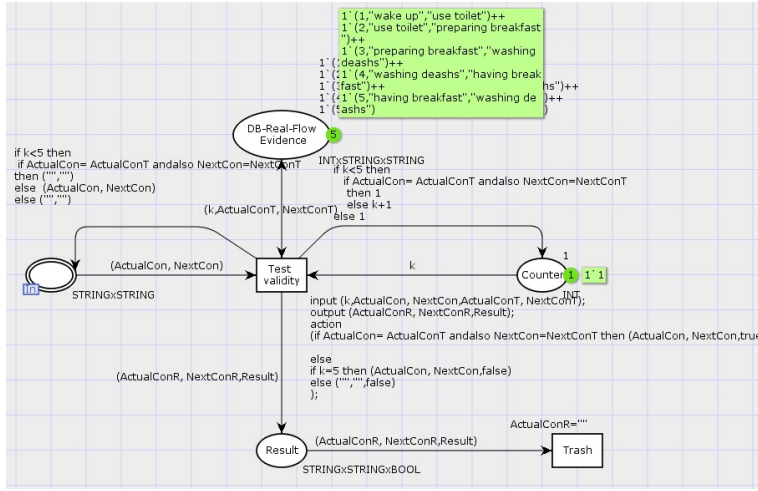


Figure 12. Verifying results Next activity

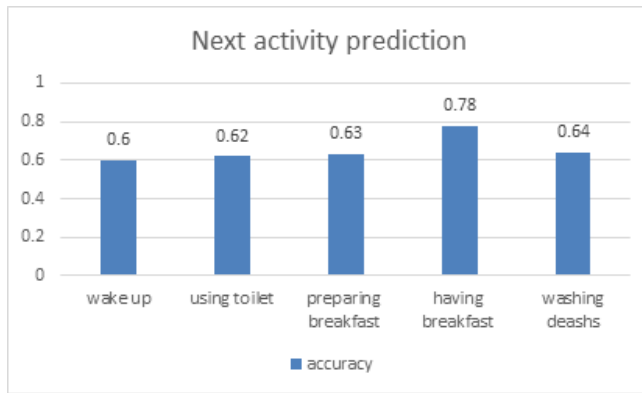


Figure 13. The activities prediction accuracy

a ubiquitous environment. Our model is in the high rang performance comparing to other context prediction model tested in real data. Using Lezi algorithm [38], the authors obtained prediction rate nearing 47 %. Using Markov and Bayes [39], the prediction accuracy achieved was 70 % to 80 %. In Najar's work [19], the system was based on the implementation of matching algorithm the prediction had a result that neared 60 %. Sigg et al. [40] have used ARMA in an analytical test, and we disregarded it for our work because is applicable only for a numerical data set. Da Rosa et al. [5] obtained an average accuracy of 60 % for the alignment method and 72 % for the Semi-Markov approach, and the model does not make a distinction between low or high context level. Föll et al. [41] used CSL and Semi-Markov-

Chain, they achieved 87 %, and they concluded their work noting using PCTL could increase the expressive power of the formal core. Which constitutes an essential and valuable contribution presented in our model including the semantic of action and span time duration as a probability function, to improve the expressivity and obtain better precision for the prediction.

Figure 14 summarizes the comparison of the existing approach and our proposed approach, regarding the different evaluation criteria [11].

Approach		Reasonning criteria						Data criteria		
		Accuracy	Tolerence to messing value	Evolvability	Speed of run time	Observability	Prior knowledge inference	Data loss in preprocessing	Numeric data	No numeric data
Sequence		*	*	*	*	□	✓	✓	X	✓
Markov		**	**	***	**	□	✓	✓	X	✓
Bayesian networks		***	***	****	****	■	✓	✓	✓	✓
Neuronal networks		***	*	**	**	□	X	X	✓	X
Branch method		**	***	***	**	■	✓	✓	X	✓
Trajectory prolongation	Interpolation	**	*	*	****	□	X	X	✓	X
	Approximation		*	*	**	□	✓	X	✓	X
Expert system		**	**	****	***	□	✓	✓	X	✓
Space theory		***	**	***	**	□	✓	✓	✓	✓
Data mining algorithm		***	**	**	**	□	✓	✓	✓	✓
Similarity and Semantic similarity		***	*	**	**	□	✓	✓	✓	✓
LPTM-Model		****	****	****	**	□	✓	X	✓	✓

Legend

Performance	Observability	Confirmation
**** Best performance	■ Black box	No X
* Worst performance	□ White box	Yes ✓

Figure 14. Comparative analysis of approaches

4 CONCLUSION AND FUTURE WORK

The prediction of future context has become a central element in pervasive systems to provide proactive context-awareness adaptation. However, the effective deployment of a context-aware prediction is still limited due to a semantic gap between the data provided by the physical sensing devices and the necessary information to predict future behavior of the system and its users. In this paper, we have demonstrated how formal methods could be adapted to offer a formal ground to reduce this semantic gap and provide improved expressiveness via the PCTL logic. And therefore, verify reachability a next-state in the future. Introducing the constraints of time and action adds logic-based expressiveness and provides a clear tracing and learning model. Thus, increasing the effectiveness of probabilistic measures.

In this paper, we present a new formal approach using probabilistic temporal logic and model checking to provide an immediate prediction. The proposed approach allows a formal expressivity of prediction. This is useful in pervasive computing systems to deal with their inherently heterogeneous nature. The model offers a real-time ability to discover a future context on multidimensional space and can handle a general context in low or high level. Adopting a PCTL as formalism provides better expressivity to describe the nondeterministic nature of human behavior which can provide an efficient prediction and consequently offer adequate proactivity, fitting with the user's needs. In fact, PCTL can be used to specify properties of probabilistic timed automata adding the semantic of action in our model, Thus, we think it will be useful to specify properties of probabilistic timed labelled automata. Regarding the complexity of model checking with probabilistic timed labelled automata, we consider this in a separate future work after more research in this direction.

In future work, we will extend the current research to include the long-term prediction and possibly discuss a generic framework that can support the proposed prediction model to automating proactive adaptation based on predicted context. We will try to investigate more, the issue of semantic in action to be able to provide a more expressive model, inducing cognitive and linguistic support.

REFERENCES

[1] Salfner, F.—Lenk, M.—Malek, M.: A Survey of Online Failure Prediction Methods. ACM Computing Surveys (CSUR), Vol. 42, 2010, No. 3, Art. No. 10, doi: 10.1145/1670679.1670680.

[2] Boytsov, A.: Context Reasoning, Context Prediction and Proactive Adaptation in Pervasive Computing Systems. Thesis, Luleå Tekniska Universitet, 2011.

[3] Boytsov, A.—Zaslavsky, A.: Extending Context Spaces Theory by Proactive Adaptation. In: Balandin, S., Dunaytsev, R., Koucheryavy, Y. (Eds.): Smart Spaces and Next Generation Wired/Wireless Networking (ruSMART 2010, NEW2AN 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6294, 2010, pp. 1–12, doi: 10.1007/978-3-642-14891-0_1.

[4] Boytsov, A.—Zaslavsky, A.—Synnes, K.: Extending Context Spaces Theory by Predicting Run-Time Context. In: Balandin, S., Moltchanov, D., Koucheryavy, Y. (Eds.): Smart Spaces and Next-Generation Wired/Wireless Networking (ruSMART 2009, NEW2AN 2009). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5764, 2009, pp. 8–21, doi: 10.1007/978-3-642-04190-7_2.

[5] da Rosa, J. H.—Barbosa, J. L. V.—Ribeiro, G. D.: ORACON: An Adaptive Model for Context Prediction. Expert Systems with Applications, Vol. 45, 2016, pp. 56–70, doi: 10.1016/j.eswa.2015.09.016.

[6] Sigg, S.—Haseloff, S.—David, K.: An Alignment Approach for Context Prediction Tasks in UbiComp Environments. IEEE Pervasive Computing, Vol. 9, 2010, No. 4, pp. 90–97, doi: 10.1109/mprv.2010.23.

[7] Zaguia, A.—Tadj, C.—Ramdane-Cherif, A.: Context-Based Method Using Bayesian Network in Multimodal Fission System. International Journal of Computational Intelligence Systems, Vol. 8, 2015, No. 6, pp. 1076–1090, doi: 10.21307/ijssis-2017-824.

[8] Ameyed, D.—Miraoui, M.—Tadj, C.: A Spatiotemporal Context Definition for Service Adaptation Prediction in a Pervasive Computing Environment. arXiv preprint arXiv:1505.01071, 2015.

[9] Ameyed, D.—Miraoui, M.—Tadj, C.: Spatiotemporal Context Modelling in Pervasive Context-Aware Computing Environment: A Logic Perspective. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 7, 2016, No. 4, pp. 407–414, doi: 10.14569/ijacsa.2016.070454.

[10] Barbosa, J. L. V.: Ubiquitous Computing: Applications and Research Opportunities. 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2015, doi: 10.1109/iccic.2015.7435625.

[11] Ameyed, D.—Miraoui, M.—Tadj, C.: A Survey of Prediction Approach in Pervasive Computing. International Journal of Scientific and Engineering Research, Vol. 6, 2015, No. 5, pp. 306–316.

[12] David, K.—Kusber, R.—Lau, S. L.—Sigg, S.—Ziebart, B.: 3rd Workshop on Recent Advances in Behavior Prediction and Pro-Active Pervasive Computing. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct), 2014, pp. 415–420, doi: 10.1145/2638728.2641675.

[13] Mayrhofer, R.: Context Prediction Based on Context Histories: Expected Benefits, Issues and Current State-of-the-Art. Cognitive Science Research Paper, University of Sussex CSRP, 2005, Vol. 577, p. 31.

[14] Sigg, S.—Haseloff, S.—David, K.: A Novel Approach to Context Prediction in Ubicomp Environments. 2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications, 2006, doi: 10.1109/pimrc.2006.254051.

[15] Meiners, M.—Zaplata, S.—Lamersdorf, W.: Structured Context Prediction: A Generic Approach. In: Eliassen, F., Kapitza, R. (Eds.): Distributed Applications and Interoperable Systems (DAIS 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6115, 2010, pp. 84–97, doi: 10.1007/978-3-642-13645-0_7.

[16] Padovitz, A.—Loke, S. W.—Zaslavsky, A.: Towards a Theory of Context Spaces. In Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004, doi: 10.1109/percomw.2004.1276902.

[17] Boytsov, A.—Zaslavsky, A.: Context Prediction in Pervasive Computing Systems: Achievements and Challenges. In: Burstein, F., Brézillon, P., Zaslavsky, A. (Eds.): Supporting Real-Time Decision-Making. Springer, Boston, MA, Annals of Information Systems, Vol. 13, 2011, pp. 35–63, doi: 10.1007/978-1-4419-7406-8_3.

[18] Abowd, G. D.—Dey, A. K.—Brown, P. J.—Davies, N.—Smith, M.—Steggles, P.: Towards a Better Understanding of Context and Context-Awareness.

In: Gellersen, H. W. (Ed.): Handheld and Ubiquitous Computing (HUC 1999). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1707, 1999, pp. 304–307, doi: 10.1007/3-540-48157-5_29.

[19] Najar, S.—Kirsch Pinheiro, M.—Souveyet, C.: A New Approach for Service Discovery and Prediction on Pervasive Information System. Procedia Computer Science, Vol. 32, 2014, pp. 421–428, doi: 10.1016/j.procs.2014.05.443.

[20] Föll, S.—Herrmann, K.—Rothermel, K.: PreCon – Expressive Context Prediction Using Stochastic Model Checking. In: Hsu, C. H., Yang, L. T., Ma, J., Zhu, C. (Eds.): Ubiquitous Intelligence and Computing (UIC 2011). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6905, 2011, pp. 350–364, doi: 10.1007/978-3-642-23641-9_29.

[21] Magherini, T.—Fantechi, A.—Nugent, C. D.—Vicario, E.: Using Temporal Logic and Model Checking in Automated Recognition of Human Activities for Ambient-Assisted Living. IEEE Transactions on Human-Machine Systems, Vol. 43, 2013, No. 6, pp. 509–521, doi: 10.1109/tsmc.2013.2283661.

[22] Wagner, A.—Barbosa, J. L. V.—Barbosa, D. N. F.: A Model for Profile Management Applied to Ubiquitous Learning Environments. Expert Systems with Applications, Vol. 41, 2014, No. 4, Part 2, pp. 2023–2034, doi: 10.1016/j.eswa.2013.08.098.

[23] Miraoui, M.: Dynamic Adaptation in Ubiquitous Services: A Conceptual Architecture. In: Ramanathan, R., Raja, K. (Eds.): Handbook of Research on Architectural Trends in Service-Driven Computing, Chapter 7. IGI Global, 2014, pp. 160–180, doi: 10.4018/978-1-4666-6178-3.ch007.

[24] Bohn, J.—Coroamă, V.—Langheinrich, M.—Mattern, F.—Rohs, M.: Social, Economic, and Ethical Implications of Ambient Intelligence and Ubiquitous Computing. In: Weber, W., Rabaey, J. M., Aarts, E. (Eds.): Ambient Intelligence. Springer, Berlin, Heidelberg, 2005, pp. 5–29, doi: 10.1007/3-540-27139-2_2.

[25] Schnoebelen, P.: The Complexity of Temporal Logic Model Checking. Advances in Modal Logic, Vol. 4, 2002, pp. 393–436.

[26] Clarke, E.—Grumberg, O.—Peled, D.: A Model Checking. MIT Press, 1999.

[27] Kwiatkowska, M.—Norman, G.—Sproston, J.—Wang, F.: Symbolic Model Checking for Probabilistic Timed Automata. Information and Computation, Vol. 205, 2007, No. 7, pp. 1027–1077, doi: 10.1016/j.ic.2007.01.004.

[28] Baier, C.—de Alfaro, L.—Forejt, V.—Kwiatkowska, M.: Model Checking Probabilistic Systems. In: Clarke, E., Henzinger, T., Veith, H., Bloem, R. (Eds.): Handbook of Model Checking. Chapter 28. Springer, 2018, pp. 963–1000, doi: 10.1007/978-3-319-10575-8_28.

[29] Wolovick, N.—Johr, S.: A Characterization of Meaningful Schedulers for Continuous-Time Markov Decision Processes. In: Asarin, E., Bouyer, P. (Eds.): Formal Modeling and Analysis of Timed Systems (FORMATS 2006). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4202, 2006, pp. 352–367, doi: 10.1007/11867340_25.

[30] Vendler, Z.: Verbs and Times. The Philosophical Review, Vol. 66, 1957, No. 2, pp. 143–160, doi: 10.2307/2182371.

[31] Ratté, S.—Ratté, S.: Computational Event Structures – Part I. 1994.

[32] Lahijanian, M.—Andersson, S. B.—Belta, C.: Control of Markov Decision Processes from PCTL Specifications. Proceedings of the 2011 American Control Conference, 2011, IEEE, doi: 10.1109/acc.2011.5990952.

[33] Kwiatkowska, M.—Norman, G.—Parker, D.: Stochastic Model Checking. In: Bernardo, M., Hillston, J. (Eds.): Formal Methods for Performance Evaluation (SFM 2007). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4486, pp. 220–270, doi: 10.1007/978-3-540-72522-0_6.

[34] Kwiatkowska, M.—Norman, G.—Sproston, J.: PCTL Model Checking of Symbolic Probabilistic Systems. Technical report CSR-03-2, University of Birmingham, School of Computer Science, 2003.

[35] Kadouche, R.—Pigot, H.—Abdulrazaka, B.—Giroux, S.: Support Vector Machines for Inhabitant Identification in Smart Houses. In: Yu, Z., Liscano, R., Chen, G., Zhang, D., Zhou, X. (Eds.): Ubiquitous Intelligence and Computing (UIC 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6406, 2010, pp. 83–95, doi: 10.1007/978-3-642-16355-5_9.

[36] Chikhaoui, B.—Wang, S.—Pigot, H.: A New Algorithm Based on Sequential Pattern Mining for Person Identification in Ubiquitous Environments. KDD Workshop on Knowledge Discovery from Sensor Data, 2010.

[37] Liu, Y.—Miao, H.-K.—Zeng, H.-W.—Ma, Y.—Liu, P.: Nondeterministic Probabilistic Petri Net – A New Method to Study Qualitative and Quantitative Behaviors of System. Journal of Computer Science and Technology, Vol. 28, 2013, No. 1, pp. 203–216, doi: 10.1007/s11390-013-1323-7.

[38] Gopalratnam, K.—Cook, D. J.: Active LeZi: An Incremental Parsing Algorithm for Sequential Prediction. International Journal on Artificial Intelligence Tools, Vol. 13, 2004, No. 4, pp. 917–929, doi: 10.1142/s0218213004001892.

[39] Kaowthumrong, K.—Lebsack, J.—Han, R.: Automated Selection of the Active Device in Interactive Multi-Device Smart Spaces. Workshop at UbiComp, 2002, Citeseer.

[40] Sigg, S.: Development of a Novel Context Prediction Algorithm and Analysis of Context Prediction Schemes. Kassel University Press GmbH, 2008.

[41] Föll, S.: State-Based Context Prediction in Mobile Systems. Dissertation, University of Stuttgart, 2014.

Darine Ameyed is currently Post-Doctoral Researcher in Synchromedia Laboratory in the École de Téchnologie Supérieure (ÉTS). She received her Ph.D. in engineering from the École de Téchnologie Supérieure (ÉTS), University of Quebec, Montreal, Canada, in 2016. She obtained her M.Sc. in digital art and technology from the University Rennes 2 (UR2), University of Upper Brittany, France, in 2010. She obtained her M.Sc. in multimedia engineering from the University Paris-Est Marne-la-Vallée (UPEM), University of Paris 11, France, in 2008. She received her B.Sc. in software and management from the Institut Supérieur de Gestion (ISG), University of Tunis, Tunisia, in 2005. Her research interests include pervasive and ubiquitous computing, context-aware systems, predictive modelling, activity recognition, human centred computing, cognitive IoT. She has published papers in national and international conferences and journals. She had also several years of industry experience in Canada and Africa in the areas of software engineering, mobile computing, ubiquitous computing, CIoT, and management.

Moeiz Miraoui received his Ph.D. (2009) in computer science from the École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, Canada. He obtained his M.Sc. in computer science from the National Engineering School of Sfax, University of Sfax, Tunisia, in 2003. He received his B.Sc. Ing. in computer science from the National School for Computer Science, University of Tunis, Tunisia, in 1999. He began working as Assistant Professor in computer science at High Institute of Applied Science and Technology, University of Gafsa, Tunisia, in 2009. He has supervised several M.Sc. students and co-supervises Ph.D. projects. His research interests include pervasive and ubiquitous computing, context-aware systems and smart spaces. He has many published papers in national and international conferences and journals, and he is a reviewer and a technical committee member of several journals and conferences.

Atef Zaguia is currently Assistant Professor in College of Computers and Information Technology, Taif University, Kingdom of Saudi Arabia. He received his Ph.D. and M.Sc. degrees in computer science from École de Téchnologie Supérieure (ÉTS), University of Quebec, Montreal, Canada, and his Bachelor degree in computer engineering from Ottawa University. He spent one year at École de Téchnologie Supérieure (ÉTS), University of Quebec, Montreal, Canada, as a postdoctoral fellow. He was working on developing application for newborn cry-based diagnosis system with the integration of interaction context, financial by Bill and Melinda Gates Foundation. His research interests include a multimodal system, pervasive and ubiquitous computing and context-aware systems. He has published papers in national and international conferences and journals. He was in Program Committee for the Tenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies UBICOMM 2016, Venice, Italy.

Fehmi Jaafar is Adjunct Professor in the Faculty of Management of Concordia University of Edmonton, Canada. Previously, he was postdoctoral fellow at Queen's University and Polytechnique Montreal, and Research Scientist in computer security and software engineering at the Research and Development Team of Ubitrak Inc. He received his Ph.D. from the Department of Computers Sciences and Operations Research of the Université de Montréal in Quebec, Canada, 2013. His research interests include the analysis of software quality and evolution, cyber security, and techniques and tools for mining software repositories. He also had several years of industry experience in Canada and Africa in the areas of software engineering, computer security, web and mobile computing, and business process management.

Chakib Tadj serves as Professor at École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, Canada. He received his Ph.D. degree in signal and image processing from ENST Paris, Paris, France, in 1995. His main research interests include signal processing, speech recognition, pervasive computing and multimodal systems.

Computing and Informatics, Vol. 37, 2018, 1443–1463, doi: 10.4149/cai_2018_6_1443

METHOD TO OVERCOME THE AMBIGUITIES IN SHALLOW PARSE AND TRANSFER MACHINE TRANSLATION

Jernej Vičič

University of Primorska, The Andrej Marušič Institute
Muzejski trg 2, 6000 Koper, Slovenia
&
Research Centre of the Slovenian Academy of Sciences and Arts
The Fran Ramovš Institute
e-mail: jernej.vicic@upr.si

Marko Grgurovič

University of Primorska, The Andrej Marušič Institute
Muzejski trg 2, 6000 Koper, Slovenia
e-mail: marko.grgurovic@student.famnit.upr.si

Abstract. The manuscript proposes a new architecture for a Shallow Parsing and Shallow Transfer Rule-Based Machine Translation System. The newly proposed architecture omits the disambiguation module in the starting phases of the translation pipeline and stores all translation candidates generated by the ambiguous process in the morphological analysis phase. The architecture is based on multigraphs. We propose a simplified version of k-best shortest path algorithm for this special case of directed multigraph. The k-best set is ranked using a trigram language model. The empirical evaluation shows that the new architecture produces better translation quality results with constant delay in time.

Keywords: Multigraph, k-shortest paths, shallow parse and transfer RBMT, machine translation

Mathematics Subject Classification 2010: 68T50

1 INTRODUCTION

The manuscript proposes a change in the architecture for a Shallow Parsing and Shallow Transfer Rule-Based Machine Translation System. One of the methods, which guarantees relatively good results for the translation of closely related languages, is the method of a rule-based shallow-parse and shallow-transfer approach which uses a simple architecture, thus relying on (mostly structural) similarity of the language pair. It has a long tradition and it has been successfully used in a number of MT systems, the most notable of which are Apertium [3] and Česílko [11].

Shallow-transfer systems usually use a relatively linear and straightforward architecture where the analysis of a source language is usually limited to the morphemic level. Most such systems still rely on morphological analysis of the source text in the source language. This part of the process is ambiguous. The newly proposed architecture omits the disambiguation module in the starting phases of the translation pipeline and stores all translation candidates generated by the ambiguous process in the morphological analysis phase.

The time and space complexity of the proposed architecture are discussed along with the presentation of the algorithms and data structures. An experimental prototype system as a proof of concept has been constructed on the basis of the Apertium [3] machine translation framework and using language data for the language pair Slovenian-Serbian [14].

Neural machine translation (NMT) has recently replaced the "classical statistical machine translation – SMT" and becomes the dominant research paradigm [1], but there are still reports of RBMT systems outperforming SMT or NMT systems such as [14] and also there are use cases where RBMT systems are more suitable (where the deterministic behaviour of the system is important).

The rest of the manuscript is organised as follows: The domain description is presented in Section 2, the motivation for the research is presented in Section 3. The methodology is presented in Section 4, space and time complexity of the presented data structures and algorithms is presented in Section 5, empirical evaluation and results are presented in Section 6 and conclusions in Section 7.

2 DOMAIN DESCRIPTION

The European Association for Machine Translation (EAMT) [5] describes Machine Translation as any translation task that involves the use of computers (machines). In the scope of this paper we will use the term in a narrower scope: Fully Automatic Machine translation [5] where the task of translating the text from the source language to the target language is done by the computer (machine). More specifically the research focuses on the translation of similar languages. One of the most suitable paradigms for this domain is the Shallow Transfer Rule-Based MT (ST-RBMT).

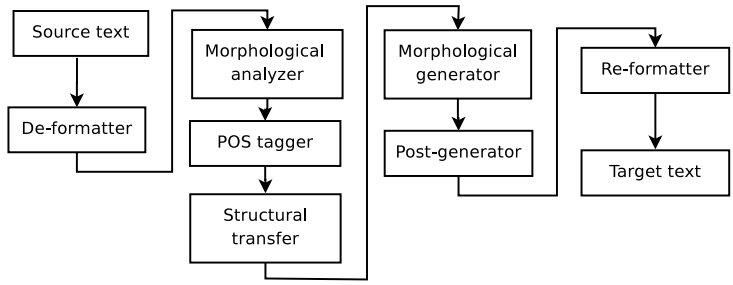


Figure 1. The modules of a typical shallow parsing and shallow transfer translation system. The frameworks for the construction of MT systems [3, 12, 17, 21] follow this design.

2.1 Shallow Parsing and Shallow Transfer Rule-Based MT

Figure 1 shows the architecture of the most known translation systems for related languages Apertium [3] and Česílko [12]. The newest version of Česílko [26] is available under open source license.

The monolingual dictionaries are used in the morphological parsing of the source text by the morphological analyser module and in the generation of the translation text in the target language by the morphological generator module. The Part Of Speech (POS) tagger module is used to disambiguate the ambiguous output of the morphological analyser module. The bilingual dictionary is used for word-by-word translation: in our case the translation is based on lemmata. The shallow transfer rules are used to address local syntactic and morphological rules such as local word agreement and local word reordering. The module using the bilingual dictionary and the shallow transfer rules is the structural transfer module. The remaining modules deal with text formatting which is not the domain of this paper. All methods and materials discussed in this paper were tested on a fully functional machine translation system based on GUAT [24], a translation system for related languages based on Apertium [3], which is a widely used open source toolkit for creating machine translation systems between related languages.

The majority of the translation systems for related languages use the shallow parsing machine translation architecture [25].

2.1.1 Apertium

Apertium is an open-source machine translation platform, initially aimed at related-language pairs but recently expanded to deal with more divergent language pairs (such as English-Catalan). The platform provides a language-independent machine translation engine, tools to manage the linguistic data necessary to build a machine translation system for a given language pair and linguistic data for a growing number of language pairs. All these properties make Apertium a perfect choice in a cost-effective machine translation system development.

2.1.2 GUAT

All methods and materials discussed in this paper were tested on a fully functional machine translation system based on GUAT [14] and [24], a translation system for related languages based on Apertium [3]. The system GUAT was used as the sandbox for the implementation of proposed methods. GUAT is automatically constructed so there is still a room for improvement, mainly through data correction tasks. The basic architecture of the system follows the architecture of Apertium [3] and is presented in Figure 1.

2.2 Multigraph

In mathematics, a graph is a structure amounting to a set of objects in which some pairs of the objects are in some sense "related". The objects correspond to mathematical abstractions called vertices (also called nodes or points) and each of the related pairs of vertices is called an edge (also called an arc or line). A graph G is an ordered pair $G := (V, E)$ with:

- V a set of vertices or nodes,
- E a set of unordered pairs of vertices, called edges or lines.

A multigraph or pseudograph is a graph which is permitted to have multiple parallel edges between nodes, that is, edges that have the same start and end nodes. Thus two vertices may be connected by more than one edge. Formally, a multigraph G is an ordered pair $G := (V, E)$ with:

- V a set of vertices or nodes,
- E a multiset of unordered pairs of vertices, called edges or lines.

In our example we use a subset of the above definition, a directed multigraph, where the edges defined by the pairs of the E multiset are ordered (directed from start to finish). Another addition to the typical definition of a graph for our example was the addition or a starting node.

The new definition of a directed multigraph with a starting node used in the paper is:

- V a set of vertices or nodes,
- E a multiset of ordered pairs of vertices, called edges or lines,
- s a starting node.

Such multigraphs allow compact description of all available translation hypotheses.

2.3 k-Shortest Paths

A problem related to the well-known single-source shortest path problem is the k-shortest paths problem. In the latter we are tasked with finding not only the shortest path to a given vertex, but up to k-shortest paths. There are many variations on this theme: depending on the structure of the graph; whether we are interested in only a specific s–t path or multiple such pairs; or even if the paths ought to be loopless or not. There exist efficient algorithms for computing these paths [6]. A variant of the k-shortest paths algorithm, which we will describe later, was implemented and used to construct a k-best set of translation candidates.

3 MOTIVATION

The shallow-transfer RBMT architecture usually relies on disambiguation after the morphological analysis to cope with the possible multiple outcomes. Figure 1 shows the disambiguation module following the morphological analysis. This process can be done by a set of rules in a form of a Constraint Grammar [15] using Visl[1] as used in [20] or in most cases by using a statistical POS tagger such as [18]. This phase precedes the more or less deterministic transfer phase. This is obviously a huge limitation, especially for the lexical transfer, since in most language pairs there are many words where translation depends upon the syntactic and/or semantic context. If the system contains some (shallow) syntactic parser and/or structural transfer, they also tend to produce ambiguous output relatively often.

The most important reasons for this research are:

- The production of a new POS tagger, especially a good quality tagger, is not a simple task. One of the easiest methods is the training of a stochastic tagger based on HMM algorithm [27]. Some parts of this task can be automatized using unsupervised learning methods or supervised learning methods like [2], but it still involves the selection of a new tag set, the production of a tagged training corpus, testing of the corpus and, at the end, the basic learning process.
- The quality level of the tagging process of today's state-of-the-art POS taggers for highly inflectional languages like Czech and Slovak [10] and Slovenian, Croatian and Serbian [8] is relatively low, comparing to the quality of POS taggers for the analytical languages like the English language, and also comparing to the overall quality of the translation systems for related languages.
- According to the today's most used designs for translation systems for related languages, the shallow transfer translation systems, the disambiguation module follows the source language morphological analysis at the beginning of the translation process. This design is shown in Figure 1. Such a design is adopted by Apertium [3] and Česílko [12]. Errors produced at the early stages of the transla-

[1] `http://visl.sdu.dk/constraint_grammar.html`

tion process usually cause bigger problems than errors introduced at later phases as later phases of the translation rely on the output of the preceding phases.

- Multiple translation candidates allow selection of the best candidates in the final phase when all available data for the translation has been accumulated. The most common translation errors are fluency errors of the target language and not adequacy errors. These errors usually do not interfere with the meaning of the translation but rather with the grammatical correctness of the translation. They are mostly caused by the errors in morphological analysis or morphological syntheses.

The omission of the tagger and introduction of a ranking scheme based on target language statistical model, as suggested in [13], yields better translation results. The introduction of multiple translation candidates generated from all possible morphological ambiguities, as suggested in [13], leads to an exponential growth of the number of possible translation candidates.

The paper [25] proposed a rule-based method for eliminating the impossible translation candidates, thus lowering the number of possible translation candidates. A statistical ranking method was used to select an arbitrary number of best candidates. The rules were automatically constructed.

The method proposed in this paper keeps all possible translation candidates in the starting phases of the translation process, all the candidates are considered and the best candidate (or n-best set of candidates) is selected in the last phase, the ranking phase. The ranking phase uses a standard statistical language model to score possible translation candidates.

4 METHODOLOGY

4.1 Proposed Architecture

The unified data structure would result in the rewrite of almost all modules of the original Apertium system. One of the most appealing features of the Apertium system is the transparency of the translation process. All data is shared in a human-readable text form through simple UNIX pipes resulting in easy error discovery and easy debugging. The proposed data structure would have to be serialized in a human-readable form.

No change was made to the Apertium toolset for the means of the presented experiment. A new module has been added to the architecture, the Multigraph supervisor, which constructs the multigraph data structure and communicates with the Apertium modules through UNIX named pipes. The new architecture is presented in Figure 2. The Multigraph supervisor module constructs the translation candidates by sending parts of the sentences, connecting edges in the data structure, to the appropriate module and saves the result in the same data structure gradually constructing all translation candidates. The data structure is further presented in Section 4.1.1.

The POS tagger module that handles the morphological disambiguation has been omitted from the architecture as all the ambiguities are stored and dealt with in the later modules.

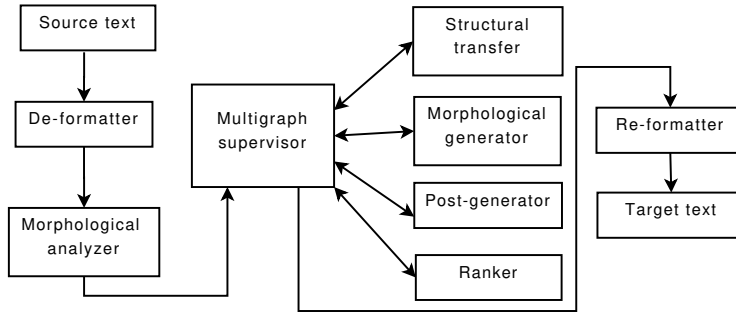


Figure 2. The proposed new architecture of a shallow transfer RBMT system using multigraphs. The Supervisor module uses the original modules in the translation process.

4.1.1 The Data Structures

Two data structures based on multigraphs were used in the Multigraph supervisor module: the construction process and the most distinct properties are presented in the subsections following this section.

The pilot implementation of the presented architecture was done in Java using the Java Universal Network/Graph Framework – JUNG, which is an open-source software library that provides a common and extendible graph/network analysis and visualization framework. JUNG also provides a visualisation framework that makes it easy to construct tools for the interactive exploration of network data.

4.1.2 Morphological Analysis

The morphological analysis produces ambiguous results. There are multiple Morpho-Syntactical descriptors – MSD [7] that can be attributed to one word form; an example of an ambiguously tagged sentence is presented in Figure 3. The MSD tags used in this example are:

- ⟨adv⟩ – adverb,
- ⟨vbser⟩ – auxiliary verb to be,
- ⟨pres⟩ – present tense,
- ⟨p3⟩ – third person,
- ⟨sg⟩ – singular,
- ⟨vblex⟩ – regular verb,
- ⟨f⟩ – female gender,

- ⟨acc⟩ – accusative case,
- ⟨nom⟩ – nominative case,
- ⟨pos⟩ – positive,
- ⟨nt⟩textgreater – neuter gender,
- ⟨n⟩ – noun.

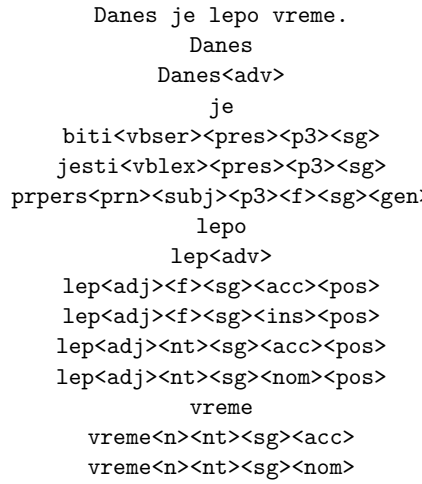


Figure 3. The ambiguously tagged sentence *Danes je lepo vreme*

The introduction of multiple translation candidates generated from all possible morphological ambiguities, as suggested in [13], leads to an exponential growth of the number of possible translation candidates.

The output of the morphological analysis is a set of all possible morphological tags describing each word. Every word with more than one tag can be observed as a set of possible ambiguities. In the case of highly inflectional languages like the pair presented in this paper the number of ambiguous possibilities increases. The set of all possible translation candidates is constructed as the vector product of all ambiguous sets. The number of possible translation candidates grows exponentially with the length of the sentence, the upper limit of the number of possible translation candidates is: $\prod_{i=0}^{|S_{max}|} x_{i_{max}}$, where S_{max} is the longest sentence and $x_{i_{max}}$ is the biggest number of ambiguities for a word. The average number of possible translation candidates is much lower, it is $\bar{x}^{\bar{S}}$, where $\bar{x}$ is the average number of ambiguities and $\bar{S}$ is the average length of a sentence.

The following example shows empirical values for an example source sentence and typical numbers collected from a corpus test-set: the maximal values: $|S| = 40$, $x = 15$, $|TC| = \prod_{i=0}^{40} 15 = 110.573323209e{+}45$ and the average values: $\bar{S}| = 15$, $\bar{x} = 3$, $|TC| = \prod_{i=0}^{15} 3 = 14\,348\,907$.

The data structure that can contain all the information produced by the morphological analysis in a compact form and also enable easy access to all translation candidates is a multigraph, where nodes represent word boundaries and edges represent all possible ambiguous word forms. An example multigraph of the same example sentence from Figure 3 is shown in Figure 4.

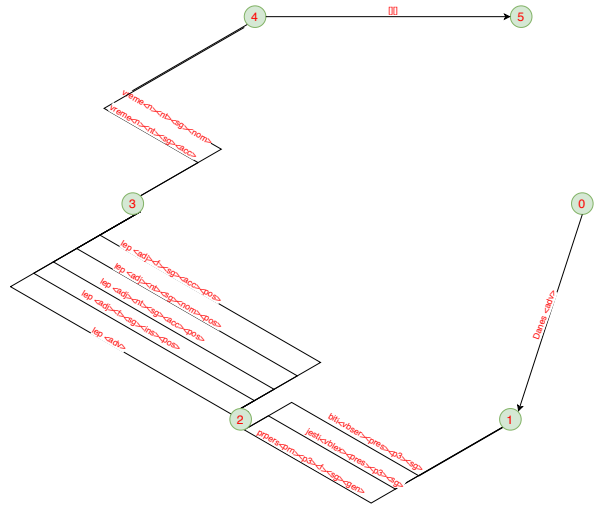


Figure 4. The multiple possibilities of the morphological analysis are stored in the edges of the multigraph. The example multigraph is constructed from the data in Figure 3.

4.1.3 Structural Transfer

The structural transfer rules are usually made in two parts: the search pattern (context) and action. We will concentrate on Apertium style rules although the abstraction would apply to most systems. A search through the Apertium systems[2] showed that the length of 3 elements for the search pattern suffices for more than 98 % of rules. The linguistic explanation is that the rules act in a very limited context.

Although we can safely use the length of the longest context (search pattern) of 3 in the majority of cases, we will abstract the length to an arbitrary length l_R. All possible candidates of the length l_R are constructed starting at the beginning of the multigraph – an example is shown in Figure 4 – and gradually moving to the last node of the multigraph. This technique enables the applications of the rules in the Left to Right Longest Match (LRLM) order which has been proven to be effective by [23].

[2] Apertium project at Sourceforge: `http://sourceforge.net/projects/apertium/`

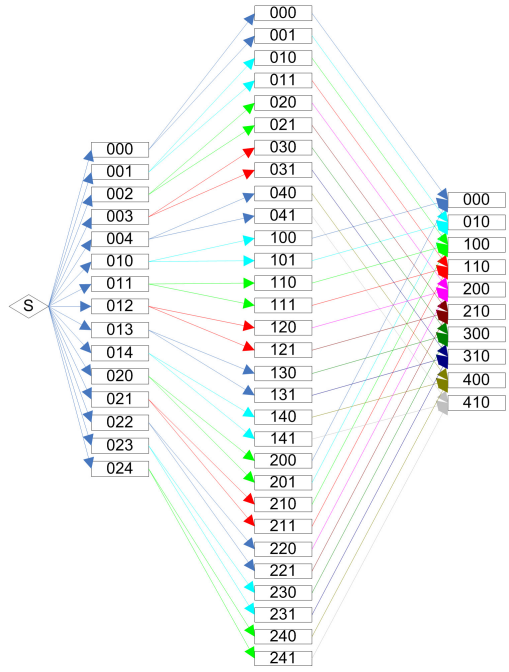


Figure 5. The data structure storing all the data from the Structural transfer module

It can be easily proven that this algorithm constructs an LRLM coverage of all translation candidates. The candidates are sent to the structural transfer module and the result is stored in a new data structure shown in Figure 5. The value for l_R has been set to 3 to simplify the visualization of the data structure.

Figure 5 shows a representation of the complex data structure produced from the morphological output, presented in Figure 3, and stored in the multigraph presented in Figure 4. All the morphological descriptors have been numbered. The paths connecting the morphological descriptors have been constructed using these numbers. Let us observe the examples in Figure 6 where the first two trigrams of the lexical units are represented by the strings *"000"* and *"001"*.

```
Danes je lepo vreme.
Danes<adv> biti<vbser><pres><p3><sg> lep<adv>
000
Danes<adv> biti<vbser><pres><p3><sg> lep<adj><f><sg><acc><pos>
001
```

Figure 6. The first two trigrams represented by the strings *"000"* and *"001"*, respectively

All strings of a certain length are stored in the same column. The trigrams containing the morphological descriptors of three adjacent word forms are stored in nodes: the edges will be used to store the probabilities of the trigrams.

4.1.4 Morphological Generation

The lexical units that were stored as n-grams in the complex data structure are fed to the Morphological generator which generates the linearized text. The data is stored in the same data structure.

4.1.5 Ranking

The ranking process simulates the Statistical N-gram Language Models, in particular the Trigram Language Model. The Trigram Language Model is the most used statistical language model. Probability of a string $P(S)$ is represented by Equation (1), where w_i denotes the i^{th} word of the string S. The probability of a string S is equal to the product of the conditional probabilities of the words constituting the set, with the condition that all the previous words of the string S appear before the i^{th} word.

$$\begin{aligned} P(s) &= P(w_1)P(w_2|w_1)P(w_3|w_1w_2)\dots p(w_l|w1\dots w_{l-1}) = \\ &= \prod_{i}^{l} P(w_i|w_1\dots w_{i-1}). \end{aligned} \tag{1}$$

The process computes the probabilities for all trigrams and stores them in the multigraph data-structure. The probability of the observed trigram is stored in the edge finishing in the observed node.

The problem of ranking the best translation candidate using the trigram language model based on the presented data structures becomes the search for the minimal path in the graph from a starting node to the finishing node. The most known algorithm that can be used is the Dijkstra algorithm [4]. An algorithm that produces k-shortest paths must be used in order to produce an n-best-set, actually k-best-set according to the presented nomenclature, of translation candidates. The graph presented in Figure 5, that presents the final data structure with candidates for the final translation, is an acyclic, directed (multi)graph.

Since the essentially optimal algorithm given in [6] is somewhat involved, we present a simplified method for computing up to k-shortest paths in directed acyclic graphs (DAGs) and provide the straightforward analysis. In DAGs, the question of whether the computed paths are loopless or not is moot: there are no loops. We will denote the out-degree of a vertex $v \in V$ by $\deg^-(v)$ and the cost of an edge $(u, v) \in E$ by $\ell(u, v)$. Furthermore, we use the following convention: $n = |V|$ and $m = |E|$.

The algorithm works by traversing the set of vertices in reverse topological order, which can be obtained in linear time, starting with the target vertex t. Each vertex

keeps a list of up to k distances of shortest paths to t, which is initially empty, except for t which contains a distance of 0. Then, each vertex i loops through its outgoing edges (i, j) and inserts into a priority queue each endpoint j, with a priority equal to the first element in the list of j plus the cost of the edge (i, j). Once this is done, each vertex pops up to k elements from the priority queue and inserts them into its list. After an element is popped, a new one is inserted into the priority queue from the same list as the popped element. Formalizing in pseudocode we get Algorithm 1.

Algorithm 1 Proposed k-shortest path algorithm for DAGs

```
procedure k-SHORTEST-PATHS(V, E, k, t)
    paths := array of n lists
    paths[t] := empty list
    paths[t].append(0)
    for all v ∈ V \ {t} do                    ▷ In reverse topological sorted order
        paths[v] := empty list
        PQ := empty
        pointers := array of deg⁻(v) list pointers
        for all (v, j) ∈ E do
            pointers[j] := paths[j]
            PQ.enqueue(j, pointers[j].data + ℓ(v, j))
        end for
        count := 0
        while count < k ∧ ¬PQ.empty do
            j := PQ.deleteMin()
            paths[v].append(pointers[j].data + ℓ(v, j))
            count := count + 1
            if pointers[j].next ≠ null then
                pointers[j] = pointers[j].next
                PQ.enqueue(j, pointers[j].data + ℓ(v, j))
            end if
        end while
    end for
end procedure
```

The time and space complexity of the presented algorithm is presented in Section 5.1.

5 SPACE AND TIME COMPLEXITY

A few definitions that will alleviate the discussion of the complexity of the presented algorithms and data structures.

- l_R – longest rule pattern length,

- l_S – longest sentence length,
- a – biggest number of ambiguities.

In numbers: setting the longest sentence to 30 words and the biggest number of ambiguities for a word to 36 (from the corpus Multext-east [9] and the GUAT system morphology [22]). The average number of ambiguities is 3. Setting the longest rule to 3 as more than 98 % of all the rules in Apertium systems and all the rules in the GUAT system have 3 or fewer lexical units in the pattern.

The data structure presented in Section 4.1.2 is a multigraph with l_S nodes and a edges between two nodes. The number of edges m is defined as: $m = l_S \cdot a$ and giving the maximum and minimum number of edges: $m_{max} = 30 \cdot 36 = 1\,080$ and $m_{average} = 30 \cdot 3 = 90$, respectively. The average number of edges in our example setting is $m = l_S \cdot a = 30 \cdot 14 = 780$ and the number of nodes is $n = 30$.

The data structure presented in Section 4.1.3 is a multigraph with $l_S - l_R + 1$ sets of nodes, where each set of nodes can have up to a^{l_R} nodes. Each node is connected with up to a edges (valence) to nodes in the adjacent set of nodes.

Equations (2) and (3) present the total number of nodes and edges and the continuation presents the empirical projections of the presented values using maximal and average values from the corpus Multext-east [9], and Equation (3) presents the total number of edges using the same values, as presented in the previous example.

$$\begin{aligned} n &= a^{l_R} \cdot (l_S - l_R + 1), \\ n_{max} &= 36^3 \cdot 28 = 1\,306\,368, \\ n_{average} &= 3^3 \cdot 28 = 252; \end{aligned} \tag{2}$$

$$\begin{aligned} m &= na, \\ m_{max} &= a^4 \cdot 28 = 36^4 \cdot 28 = 47\,029\,248, \\ m_{average} &= a^4 \cdot 28 = 3^4 \cdot 28 = 2\,268. \end{aligned} \tag{3}$$

The worst-case scenario cannot be reached as most of the word positions and POS variations are dependent.

The construction of the graph, basically the morphological analysis, and the execution of the structural transfer process are linear to the number of edges, so the time complexity can be attributed mostly to the largest contributor, the ranking process.

5.1 The Ranking Process

Analysis is as follows. In the first stage, each vertex $i \in V$ goes through its outgoing edges and places the first element found in each of its neighbors' lists into the priority queue. It is not difficult to implement the priority queue insert operation in time $O(1)$, and using adjacency lists the loop through the neighbors can be done in $O(\deg^-(i))$. We can write the cost for the first stage over all vertices as $O\left(\sum_{i=0}^{n} \deg^-(i)\right) = O(m)$ by the handshaking lemma.

In the second stage, each vertex $i \in V$ performs k deleteMin operations, and for each deleteMin operation it also inserts the next element from the list into the priority queue. The deleteMin operation can be performed in $O(\lg \deg^-(i))$, since there are at most $\deg^-(i)$ elements inside the priority queue at any given time. Finding the next element in the list and inserting it into the priority queue can be done in $O(1)$. Writing the cost over all vertices for this stage, we get the values presented in Equation (4):

$$\begin{aligned} O\left(\sum_{i=0}^{n} k \lg \deg^-(i)\right) &= O\left(k \lg \left(\prod_{i=0}^{n} \deg^-(i)\right)\right) \\ &= O\left(k \lg \left(\left(\frac{m}{n}\right)^n\right)\right) \\ &= O\left(nk(\lg m - \lg n)\right). \end{aligned} \tag{4}$$

Together with the first phase, this amounts to $O(m+nk(\lg m-\lg n))$. In general we can bound the number of edges by $m = O(n^2)$ and obtain $O(m + nk \lg n)$. We point out, that our time bound is never better than that of [6], which for DAGs amounts to $O(m+n+k)$. The space bounds of Algorithm 1 are simply $O(m+nk)$, since each vertex keeps a list of length at most k and the $\deg^-(v)$ pointers are simply $O(m)$ over the entire algorithm, by the handshaking lemma.

For our particular problem, we can bound the running time as follows, using the worst-case values we have computed in the previous section. Equation (5) presents the final values.

$$O\left(m + k \lg \left(\left(\frac{m}{n}\right)^n\right)\right) = O\left(m + k \lg \left(\left(\frac{na}{n}\right)^n\right)\right) = O\left(m + nk \lg(a)\right) \tag{5}$$

where a, the number of ambiguities, is a very small number, e.g. 3, as argued in the previous section. Therefore the log term is of no practical significance.

6 EMPIRICAL EVALUATION AND RESULTS

Two main goals were evaluated in this experiment:

- the change in the quality of the final results, the translations,
- the time complexity.

The newly proposed system was compared to two already available translation systems for the same language pair:

- the original off-the-shelf Apertium system with the Slovenian-Serbian translation data, described in [24],
- to the system presented in the experiment [25].

The later system uses a method to restrict the number of translation candidates.

6.1 Translation Quality Comparison

The Human-targeted Translation Edit Distance – HTER [19], which is derived from the edit-distance [16], was used to evaluate the translation quality. The metric counts the number of deletions, insertions and substitutions that need to be performed among the observed sequences, i.e., count the minimal number of edits needed to produce a correct target sentence from an automatically translated sentence. The definition of a correct translation understood in this experiment is a translation that is syntactically correct and expresses the same meaning as the source sentence and is achieved with the least amount of transformation. This procedure shows how much work has to be done to produce a good translation. The metric roughly reflects the complexity of the post-editing task.

Two evaluations were performed:

- The comparison between the system presented in this paper (GUAT Multigraph) and systems from a similar experiment [25] – the evaluation was performed on a small test-set (57 sentences);
- The translation quality evaluation using the same methodology as the former comparison using a bigger test-set (500 sentences).

The first evaluation was done on a relatively small test-set due to the constraints of the systems evaluated in the experiment [25]; one of the systems (GUAT all candidates) used all possible morphological ambiguities for the generation of translation candidates resulting in possibly millions of translation candidates. Such translations lasted hours and even days. The test data for this part of the experiment comprised of the 57 sentences. The sentences were chosen by length (sentences shorter than 15 words). This limitation still enabled a fair comparison of the translation quality of all the systems. The complexity of each sentence was arbitrary, there was no special selection of the sentences using this criteria although shorter sentences are usually simpler in structure. The results of this part of the evaluation are presented in Figure 7.

GUAT original – the reference system, based on Apertium architecture.

GUAT all candidates – a system that kept all translation candidates to the last phase (best translation performance, exponential growth of possible translation candidates).

GUAT rules selection – the system with a method that restricted the number of possible translation candidates in the starting phases of the translation.

GUAT Multigraph – the system with the newly proposed architecture.

The newly proposed system (GUAT Multigraph) mean value of HTER is 0.1712 which is better than the off-the-shelf system (GUAT original) mean value of HTER (0.2213). The paired two sample ttest for means in Table 1 shows that mean values are significantly different. The P value of the ttest is less than alpha (0.05), so

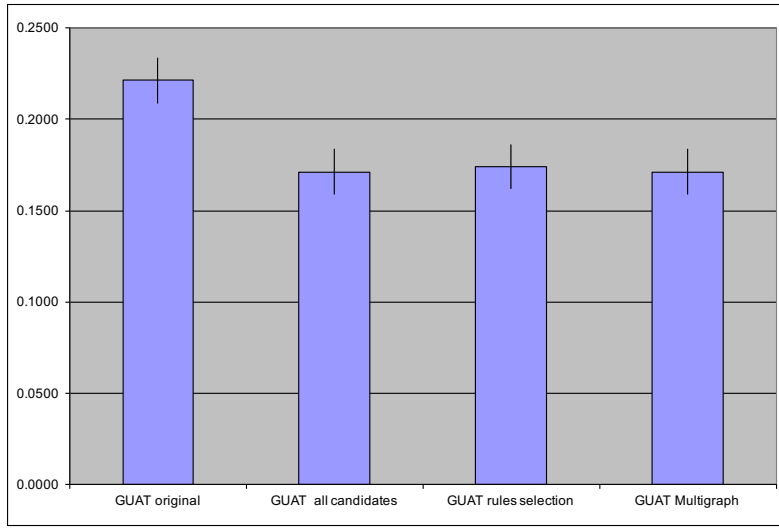


Figure 7. The translation quality evaluation, using HTER metric (less is better). The newly proposed system (GUAT Multigraph) outperforms the original system from [25] and equals the improved systems from the same experiment.

we can reject the null hypothesis (that there is no difference in mean) and we can accept that the mean value of GUAT Multigraph is lower than the mean value of the original system.

System	GUAT original	GUAT Multigraph
Mean	0.221269	0.171212
Variance	0.018006	0.013760
Observations	57	57
Hypothesis	no difference	
$P(T \leq t)$ two-tail	0.00011	

Table 1. The ttest shows that the mean values are significantly different

Table 2 shows that there is no significant difference in the mean values of GUAT rules selection and GUAT Multigraph. The values of the GUAT all candidates and GUAT Multigraph are the same, all translations were equal (both systems prepared and selected the same candidate translations).

The second evaluation was made to support the results of the first evaluation on a bigger test-set comparing the system presented in this paper (GUAT Multigraph) and the off-the-shelf GUAT system, which is based on the original Apertium architecture. The basic methodology was the same as in the first comparison. A new test-set was prepared which was comprised of 500 sentences randomly selected from the MULTEXT-East corpus [9]. Both systems used the same translation data, the

System	GUAT rules sel.	GUAT Multigraph
Mean	0.174157	0.171212
Variance	0.014498	0.013760
Observations	57	57
Hypothesis	no difference	
Alpha	0.05	
$P(T \leq t)$ two-tail	0.160599	

Table 2. The ttest shows that the difference in mean values could be coincidental (we cannot trust the difference)

construction process of the translation data was described in [24] and the data is available at the Sourceforge[3]. The results of this part of the comparison are presented in Figure 8. The system with the newly proposed architecture shows an improvement over the reference system.

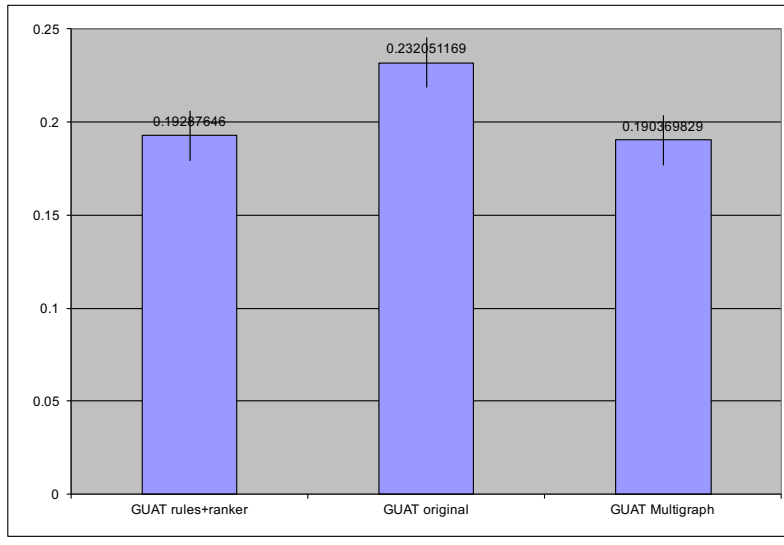


Figure 8. The translation quality evaluation, using HTER metric, of the original GUAT system based on the original Apertium architecture comparing to the newly proposed system. The newly proposed system outperforms the original system.

The ttest presented in Table 3 shows that the mean values are significantly different.

[3] `http://sourceforge.net/projects/apertium/`

System	GUAT original	GUAT Multigraph
Mean	0.232557	0.190251
Variance	0.020446	0.017484
Observations	200	200
Alpha	0.05	
Hypothesis	no difference	
$P(T \leq t)$ two-tail	0.00000000794	

Table 3. The ttest shows that the mean values are significantly different

6.2 Time Complexity

The empirical evaluation of the time complexity was done simultaneously with the evaluation of the translation quality. The test-data is described in Section 6.1, which was comprised of 200 sentences randomly selected from [9] corpus. The tests were performed on a personal computer[4]. Table 4 shows the time complexity comparison between the original GUAT system and the newly proposed system.

System:	GUAT	Multigraph
Nr. of sentences	200	200
Total time (seconds)	377.78	6 453.56
Per translation	1.89	22.27
Ratio	1.00	11.78

Table 4. Empirical evaluation of the time complexity. The newly proposed system is roughly 6 times slower than the original on the selected test sentences.

Description of Table 4:

GUAT – The reference system, based on [24].

Multigraph – the system with the newly proposed architecture.

Total time (seconds) – the total time the system spent to translate all 200 test sentences.

Per translation – the average time spent per translation.

Ratio – The ratio between the time spent by the reference system – GUAT and the described system.

7 CONCLUSIONS

The presented architecture represents a viable solution to the problem of exponential growth of the number of possible translation candidates in a non-disambiguated shallow transfer translation system. The empirical evaluation showed an improvement in the translation quality compared to the original system and also compared

[4] Laptop computer with Intel i3 CPU M330@2.13 GHz and 4 GB of memory

to the values presented in [25], which was an attempt to limit the number of possible translation candidates. The empirical evaluation also showed that the new system performed as well as a system that included all possible translation candidates, which shows that it always selected the best translation candidate.

The empirical evaluation of the time consumption showed that the new system performed roughly 12 times slower than the reference system and the constant factor was present in all test examples showing that the time complexity differed to a constant factor.

The proof-of-the-concept system has been implemented and it proved to be working as expected. A true implementation of the newly proposed architecture with the new module is already in progress. It could be incorporated into the Apertium framework.

REFERENCES

[1] Bojar, O.—Helcl, J.—Kocmi, T.—Libovick, J.—Musil, T.: Results of the WMT17 Neural MT Training Task. Proceedings of the Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, 2017, pp. 525–533, doi: 10.18653/v1/w17-4757.

[2] Brants, T.: TnT – A Statistical Part-of-Speech Tagger. Proceedings of the 6th Applied NLP Conference, Seattle, WA, 2000, pp. 224–231, doi: 10.3115/974147.974178.

[3] Corbí-Bellot, A. M.—Forcada, M. L.—Ortiz-Rojas, S. et al.: An Open-Source Shallow-Transfer Machine Translation Engine for the Romance Languages of Spain. Proceedings of the EAMT Conference, 2005, pp. 79–86.

[4] Dijkstra, E. W.: A Note on Two Problems in Connexion with Graphs. Numerische Mathematik, Vol. 1, 1959, No. 1, pp. 269–271, doi: 10.1007/bf01386390.

[5] EAMT. European Association for Machine Translation, 2018.

[6] Eppstein, D.: Finding the k-Shortest Paths. SIAM Journal on Computing, Vol. 28, 1999, No. 2, pp. 652–673, doi: 10.1137/S0097539795290477.

[7] Erjavec, T.: MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC '04), ELRA, 2004, pp. 1535–1538.

[8] Erjavec, T.: Multilingual Tokenisation, Tagging, and Lemmatisation with Totale. Proceedings of the 9th INTEX/NOOJ Conference, 2006, pp. 33–34.

[9] Erjavec, T.: MULTEXT-East Version 4: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In: Chair, N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (Eds.): Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC '10), Valletta, Malta, ELRA, 2010, pp. 2544–2547.

[10] Hajič, J.: Morphological Tagging: Data vs. Dictionaries. Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference (NAACL 2000), 2000, pp. 94–101.

[11] Hajič, J.—Hric, J.—Kuboň, V.: Česílko – An MT System for Closely Related Languages. Proceedings of ACL 2000, 2000, pp. 7–8, doi: 10.3115/974147.974149.

[12] Hajič, J.—Homola, P.—Kuboň, V.: A Simple Multilingual Machine Translation System. In: Hovy, E., Macklovitch, E. (Eds.): Proceedings of the MT Summit IX, New Orleans, USA, AMTA, 2003, pp. 157–164.

[13] Homola, P.—Kuboň, V.: Improving Machine Translation Between Closely Related Romance Languages. Proceedings of EAMT, 2008, pp. 72–77.

[14] Vičič, J.—Homola, P.—Kuboň, V.: Automated Implementation Process of a Machine Translation System for Related Languages. Computing and Informatics, Vol. 35, 2016, No. 2, pp. 441–469.

[15] Karlsson, F.—Voutilainen, A.—Heikkilä, J.—Anttila, A. (Eds.): Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text. Mouton de Gruyter, Berlin, New York, 1995.

[16] Levenshtein, V. I.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Doklady Akademii Nauk SSSR, Vol. 163, 1965, No. 4, pp. 845–848 (in Russian).

[17] Scannell, K. P.: Machine Translation for Closely Related Language Pairs. Proceedings of the Workshop Strategies for Developing Machine Translation for Minority Languages, Genoa, Italy, 2006, pp. 103–109.

[18] Sheikh, Z. M. A. W.—Sánchez-Martínez, F.: A Trigram Part-of-Speech Tagger for the Apertium Free/Open-Source Machine Translation Platform. Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation, Alicante, 2009, pp. 67–74.

[19] Snover, M.—Dorr, B.—Schwartz, R.—Micciulla, L.—Makhoul, J.: A Study of Translation Edit Rate with Targeted Human Annotation. Proceedings of AMTA Conference, 2006, pp. 223–231.

[20] Tyers, F. M.—Donnelly, K.: Apertium-cy – A Collaboratively-Developed Free {RBMT} System for Welsh to English. The Prague Bulletin of Mathematical Linguistics, Vol. 91, 2009, pp. 57–66, doi: 10.2478/v10108-009-0016-4.

[21] Tyers, F. M.—Wiechetek, L.—Trosterud, T.: Developing Prototypes for Machine Translation Between Two Sámi Languages. Proceedings of 13th Annual Conference of the EAMT, 2009, pp. 120–127.

[22] Vičič, J.: Rapid Development of Data for Shallow Transfer RBMT Translation Systems for Highly Inflective Languages. Proceedings of the Conference on Language Technologies, Institut Jožef Stefan, Ljubljana, 2008, pp. 98–103.

[23] Vičič, J.—Forcada, M. L.: Comparing Greedy and Optimal Coverage Strategies for Shallow-Transfer Machine Translation. Proceedings of the International Conference on Intelligent Information Systems XVI (IIS '08), 2008, pp. 307–316.

[24] Vičič, J.—Homola, P.: Speeding Up the Implementation Process of a Shallow Transfer Machine Translation System. Proceedings of the 14th EAMT Conference, Saint Raphael, France, 2010, pp. 261–268.

[25] Vičič, J.—Homola, P.—Kuboň, V.: A Method to Restrict the Blow-Up of Hypotheses of a Non-Disambiguated Shallow Machine Translation System. Proceedings of the International Conference RANLP, Borovec, Bulgaria, ACL, 2009, pp. 460–464.

[26] Vičič, J.—Kuboň, V.—Homola, P.: Česílko Goes Open-Source. The Prague Bulletin of Mathematical Linguistics, Vol. 107, 2017, pp. 57–66, doi: 10.1515/pralin-2017-0004.

[27] Welch, L. R.: Hidden Markov Models and the Baum-Welch Algorithm. IEEE Information Theory Society Newsletter, Vol. 53, 2003, No. 4, 5 pp.

Jernej Vičič is Assistant Professor at the University of Primorska, Primorska Institute for Natural Sciences and Technology in Koper, Slovenia. His main research interest is in the field of hybrid machine translation for related languages.

Marko Grgurovič is Teaching Assistant at the University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies in Koper, Slovenia.

Computing and Informatics, Vol. 37, 2018, 1464–1484, doi: 10.4149/cai_2018_6_1464

ACP SEMANTICS FOR PETRI NETS

Slavomír Šimoňák, Martin Tomášek

Department of Computers and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovakia
e-mail: {slavomir.simonak, martin.tomasek}@tuke.sk

Abstract. The paper deals with algebraic semantics for Petri nets, based on process algebra ACP. The semantics is defined by assigning a special variable to every place of given Petri net, expressing the process initiated in the place. Algebraic semantics of the Petri net is then defined as a parallel composition of all the variables, where corresponding places hold tokens within the initial marking. Resulting algebraic specification preserves operational behavior of the original net-based specification.

Keywords: Petri nets, process algebra ACP, formal methods, semantics, specification, transformation

Mathematics Subject Classification 2010: 68-Q85

1 INTRODUCTION

In formal methods community a widely accepted assertion states that a single formal method, that will cover all aspects of the system design process in an acceptable way, will never be developed [26]. That is given by the complexity of the process and a variety of systems to be designed and analyzed. There have been attempts to integrate two or more formal methods to cope with that situation. According to [7] it is particularly fruitful to study combinations of several methods with different characteristics and complementary strengths. A similar approach has been applied at the authors' home institution regarding the development of an environment (termed mFDTE) [21] for the design and analysis of discrete systems based on integration of three formal description techniques: Petri nets, process algebra and B-method.

The choice of the techniques mentioned has been driven by their properties that in a complementary way cover different aspects of the system design and analysis. In this work we pay attention to two of them: Petri nets and process algebra.

Petri nets, as a widely accepted formalism for design and analysis of discrete systems [24], have proved their qualities in many situations [15, 33]. Both the states and the actions of the system are clearly described using a Petri net model. So there are analysis techniques [10] available for investigation of properties based on states as well as on the dynamic behavior of a model. Process algebraic specification of a system usually has no explicit representation of states, and is more focused on the description of its dynamic behavior. Techniques available in process algebra are especially useful when comparing the behavioral descriptions of concurrent systems, like a high-level system specification and a more detailed description of its implementation [7]. In addition to the differences mentioned above, the de/composition of specifications in the case of Petri nets is not so natural and fluent as it is in the case of process algebra. Consequently, it can be assumed that Petri nets and process algebra meet the property of complementarity in several aspects.

In the recent years an active research has been performed in the matter of two fundamental models of concurrency – Petri nets and process algebras [5]. In [17] a CCS-like calculus named as Finite-Net Multi-CCS was introduced. The given calculus is provided by a labeled transition system and the P/T net semantics. As a consequence, well-formed Finite-Net Multi-CCS processes are able to represent finite P/T nets [18]. A simple process calculus (Petri calculus) is defined in [27]. Translation from Petri nets with boundaries to the calculus, and also translation of Petri calculus terms to Petri nets is introduced. In such a way, the same expressiveness of both formalisms is shown. Another interesting work [22] compares a range of notions for treatment of concurrency, such as Petri nets, Mazurkiewicz trace languages and Zielonka automata to process algebra and raises several interesting questions.

Further, a selection of classical but very influential works on the topic is given. In [25] relations between nets, terms and formulas are treated. Particularly, the net semantics of terms and the process semantics of nets are defined. In [11] relations between the process algebra PBC (Petri Box Calculus) and a class of P/T nets are studied. Syntax and semantics of PBC terms are carefully selected to allow the definition of a transformation yielding the P/T nets preserving structural operational semantics of the source terms. The transformation allows a composition of P/T nets. A process semantics of elementary nets is defined using concepts of partial algebra in [13]. In the paper, a partial algebra is proposed, as a suitable tool for defining the true-concurrency semantics for arbitrary restrictions of the occurrence rule. In the work [4] authors treat the issue of partial-order algebras and their relations to P/T nets based on the theory of BPA and ACP. Authors of [6] propose an approach to algebraic semantics for hierarchical P/T nets. The PTNA (Place/Transition Net Algebra) is defined within the paper, based on process algebra ACP and an algebraic semantics for P/T nets is given, such that a P/T net and the term representation of the net have the same operational behavior. The

actions of the process algebra correspond to consumption and production of tokens by transitions respectively. The results achieved are further extended to hierarchical P/T nets.

The compositional approach to system design and the possibility of investigating and comparing the system properties by means of algebraic manipulations [27, 17] are two main sources of motivation for defining the algebraic semantics for Petri nets. This work aims to contribute to the topic of integrating Petri nets and process algebra by introducing the algebraic semantics for Petri nets, based on Algebra of Communicating Process (ACP) [2]. Our previous work on integration of Petri nets and process algebra includes the definition of algebraic semantics for Petri nets based on a dedicated process algebra APC (Algebra of Process Components) [30]. While it has several interesting properties, mainly the ability to model Petri nets processes in a simple and natural way, it also has one essential drawback, which hampers its practical utilization. There is currently no available tool supporting the analysis of APC specifications. This motivated us to define the algebraic semantics for Petri nets based on process algebra with a reasonable tool support. The available tool support in this case includes the PSF-Toolkit [14], which is based on the process algebra ACP.

When comparing the proposed solution with the presented existing solutions, several differences can be identified. We use a widely-adopted process algebra without defining its special extensions, we can find them in many existing solutions, which ensures the reasonable tool support out of available tools. Moreover, the existing implementation of our transformation supports the simpler practical utilization. Most of the solutions, except of those defining their own process language, are based on process algebras like CCS and CSP. We have chosen the process algebra ACP, as we believe it is beneficial. ACP is based on an equational style of reasoning, while CCS and CSP are model-based. In the case of ACP, the central point is an equational theory, which may have several semantic interpretations [7, 3].

The paper is organized as follows. Section 1 introduces the topic and the motivation for the work. Basic notions and definitions for the class of Petri nets used are given in Section 2. In Section 3, Algebra of Communicating Process is presented briefly from the syntactic as well as the semantic point of view. Section 4 concentrates on defining the algebraic semantics for given class of Petri nets. The example presented in Section 5 demonstrates the approach introduced within the paper. Section 6 concludes the paper and contains the possible directions for future research.

2 PETRI NETS

In the paper we assume the class of ordinary Petri nets [9] and provide a brief description of the basic notions and notations in the following paragraphs in style of [20].

Definition 1. The Petri net is a 4-tuple $N = (P, T, pre, post)$, where P is a finite set of places, T is a finite set of transitions ($P \cap T = \emptyset$), $pre : P \times T \to \{0, 1\}$ is the preset function and $post : P \times T \to \{0, 1\}$ is the postset function.

The marking of Petri net $N = (P, T, pre, post)$ is a totally defined function $m : P \to \mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers. We use m to describe a configuration of N. We fix some ordering of places ($P = \{p_1, \ldots, p_k\}$), so we can consider m to be a k-dimensional nonnegative integer vector: $\vec{m} \in \mathbb{N}^k$. More formally $\vec{m} = (m(p_1), m(p_2), \ldots, m(p_k))$, where $m(p_i)$ is the value of m in the place p_i, $i = 1, 2, \ldots, k$. Marked net, with the marking m, is denoted by $N_0 = (N, m_0)$ or $N_0 = (P, T, pre, post, m_0)$.

For the sake of simplicity we use the denotation m for both interpretations of the marking, when it does not cause any problems. Some useful notations can be further defined, as the sets of pre/post-conditions for given transition $t \in T$ and the sets of pre/post-transitions for given place $p \in P$, respectively:

$$
\begin{aligned}
&{}^\bullet t = \{p | pre(p, t) \neq 0\} \text{ the set of preconditions of } t,\\
&t^\bullet = \{p | post(p, t) \neq 0\} \text{ the set of postconditions of } t,\\
&p^\bullet = \{t | pre(p, t) \neq 0\},\\
&{}^\bullet p = \{t | post(p, t) \neq 0\}.
\end{aligned}
$$

We say that a transition t is enabled in m (and denote it $m \xrightarrow{t}$), if for every $p \in {}^\bullet t, m(p) \geq pre(p, t)$. The effect of firing t in m is the creation of the new marking m' ($m \xrightarrow{t} m'$) and m' is defined in the following way:

$$m'(p) = m(p) - pre(p, t) + post(p, t), p \in P, t \in T.$$

Denotation $(N, m) \xrightarrow{t} (N, m')$ is alternatively used for expressing a step of computation ($m \xrightarrow{t} m'$) within the Petri net N. The set of reachable markings for given Petri net $N_0 = (P, T, pre, post, m_0)$ is defined by:

$$\mathcal{R}(N_0) = \{m | m_0 \xrightarrow{\sigma} m\}$$

where $\sigma = t_1, t_2, \ldots, t_r$ stands for an admissible firing sequence in N_0. The language of Petri net N_0 can be defined by:

$$\mathcal{L}(N_0) = \{\sigma \in T^* | m_0 \xrightarrow{\sigma} m\}.$$

3 ACP – ALGEBRA OF COMMUNICATING PROCESSES

Algebra of Communicating Processes (ACP) [2] is an algebraic framework for studying concurrent communicating processes. It is based on Milner's Calculus of Communicating Systems (CCS) [23], historically the first complete theory. ACP more

emphasizes the algebraic aspect, it is an equational theory with a number of semantical models. In addition, the ACP uses a more general communication scheme compared to CCS and CSP [3].

3.1 Syntax

The signature of ACP contains a set of constants A, partial communication function γ on A, which is commutative and associative, a special constant δ (deadlock $\delta \notin A$) and operators: $+$ (alternative composition), $\cdot$ (sequential composition), $\|$ (parallel composition), $\lfloor\!\lfloor$ (left merge), $|$ (communication merge).

The axiom system of process algebra ACP can be found in Table 1. We also use axioms for encapsulation (D1–D4) and hiding internal transitions (TI1–TI4). Within the table x, y, z stand for processes, $a, b \in A$ and $\delta, \tau \notin A$ are constants. In [2], where many different process algebras are defined, it can be found with the denotation ACP$^\tau$, but for the sake of simplicity we use the name ACP within the paper.

$x + y = y + x$	A1	$\partial_H(a) = a$ if $a \notin H$	D1
$(x + y) + z = x + (y + z)$	A2	$\partial_H(a) = \delta$ if $a \in H$	D2
$x + x = x$	A3	$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$	D3
$(x + y) \cdot z = xz + yz$	A4	$\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$	D4
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	A5		
$x + \delta = x$	A6	$\tau_I(a) = a$ if $a \notin I$	TI1
$\delta \cdot x = \delta$	A7	$\tau_I(a) = \tau$ if $a \in I$	TI2
		$\tau_I(x + y) = \tau_I(x) + \tau_I(y)$	TI3
$a\|b = \gamma(a, b)$ if γ defined	CF1	$\tau_I(x \cdot y) = \tau_I(x) \cdot \tau_I(y)$	TI4
$a\|b = \delta$ otherwise	CF2		
		$x\tau = x$	B1
$x\|y = x\lfloor\!\lfloor y + y\lfloor\!\lfloor x + x\|y$	CM1	$x(\tau(y + z) + y) = x(y + z)$	B2
$a\lfloor\!\lfloor x = ax$	CM2		
$ax\lfloor\!\lfloor y = a(x\|y)$	CM3		
$(x + y)\lfloor\!\lfloor z = x\lfloor\!\lfloor z + y\lfloor\!\lfloor z$	CM4		
$ax\|b = (a\|b) \cdot x$	CM5		
$a\|bx = (a\|b) \cdot x$	CM6		
$ax\|by = (a\|b) \cdot (x\|y)$	CM7		
$(x + y)\|z = x\|z + y\|z$	CM8		
$x\|(y + z) = x\|y + x\|z$	CM9		

Table 1. Axioms of process algebra ACP

3.2 Semantics

The constants in A are called atomic actions, and are considered indivisible actions (events). The sequential composition of two processes $x \cdot y$ is the process that first

executes x and after finishing it, it starts executing y. The alternative composition of two processes x and y is the process $x + y$, that either executes x or y.

The parallel composition of two processes x and y in a process algebra without communication is expressed as $x \| y$. It is assumed that the atomic actions have no duration in time, and that two actions cannot happen simultaneously [2]. So the actions of x are arbitrarily interleaved with those of y. In order to specify the merge by finite number of equations, the auxiliary operator $\lfloor\!\lfloor$ is introduced. The operator has the same meaning as the merge operator ($\|$) with the restriction, that the first step must come from the left process (i.e. the x in case of $x \lfloor\!\lfloor y$).

In process algebra ACP however, the meaning of parallel composition is a bit more complicated. Considering the merge of two processes $x \| y$, there are three possibilities to proceed in general (CM1). Either the process starts with a first step of x (given by $x \lfloor\!\lfloor y$), or a first step of y ($y \lfloor\!\lfloor x$) or with a communication between the two processes ($x|y$).

The communication merge ($x|y$) represents the merge of two processes, where the first step is a communication between x and y. This type of merge is an extension of communication function on atomic actions ($\gamma : A \times A \to A$). When the communication function is not defined, the communication merge is equal to δ.

If we want to state that some actions cannot happen and should be blocked, the actions are renamed to δ by using the unary encapsulation operator ∂_H. A process $\partial_H(p)$ can execute all the actions of p, except those, which names are in the set H. An important tool in the analysis of systems is hiding, as the names of internal events can be hidden and thus the relationship between externally visible events becomes more clear. The hidden action (τ) cannot be observed directly, or communicate with other actions [16].

a)	$a \xrightarrow{a} \surd$	
b)	$x \xrightarrow{a} \surd \Rightarrow$	$x + y \xrightarrow{a} \surd$ and $y + x \xrightarrow{a} \surd$
		$x \cdot y \xrightarrow{a} y$
		$x \| y \xrightarrow{a} y$, $y \| x \xrightarrow{a} y$ and $x \lfloor\!\lfloor y \xrightarrow{a} y$
c)	$x \xrightarrow{a} x' \Rightarrow$	$x + y \xrightarrow{a} x'$ and $y + x \xrightarrow{a} x'$
		$x \cdot y \xrightarrow{a} x' \cdot y$
		$x \| y \xrightarrow{a} x' \| y$, $y \| x \xrightarrow{a} y \| x'$, $x \lfloor\!\lfloor y \xrightarrow{a} x' \| y$
d)	$x \xrightarrow{a} x'$, $y \xrightarrow{b} y'$, $\gamma(a,b) = c \Rightarrow$	$x \| y \xrightarrow{c} x' \| y'$ and $x \| y \xrightarrow{c} x' \| y'$
	$x \xrightarrow{a} x'$, $y \xrightarrow{b} \surd$, $\gamma(a,b) = c \Rightarrow$	$x \| y \xrightarrow{c} x'$, $x \| y \xrightarrow{c} x'$, $y \| x \xrightarrow{c} x'$, $y \| x \xrightarrow{c} x'$
	$x \xrightarrow{a} \surd$, $y \xrightarrow{b} \surd$, $\gamma(a,b) = c \Rightarrow$	$x \| y \xrightarrow{c} \surd$ and $x \| y \xrightarrow{c} \surd$

Table 2. Transition relations for ACP terms

To assign an operational semantics to process expressions, we determine, which actions the process can perform. The fact, that process represented by the term t can execute the action a and turn to the term s is denoted by: $t \xrightarrow{a} s$ (or alternatively a is enabled in t). The symbol $\surd$ stands for successful termination and thus $t \xrightarrow{a} \surd$

denotes the fact that t can terminate by executing a. The definition of action relations is given in Table 2.

4 ACP SEMANTICS FOR PETRI NETS

In this section the transformation is described in more detail. We start with defining a special variable for every place in a Petri net to be transformed. We name such variable as E-variable, and it will be bound to a term representing all possible computations started in corresponding place of the Petri net N. The value (term) assigned to the particular variable depends on the structure of the net in a vicinity of associated place. So, considering the place p, the variable $E(p)$ will be bound to a term representing all the computations within the Petri net N, which are initiated in the place p.

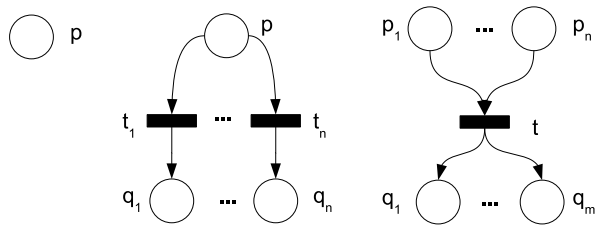


Figure 1. Petri net fragments

The basic configurations are captured in Figure 1. In the case a) a situation is depicted, where no arcs are connected to the place named p. This situation results in the assignment of a term representing no computations to the variable corresponding to such place, i.e. δ (deadlock). The case b) represents the alternative composition (choice). If a token is present in the place p, a choice is to be made, and only one of the transitions $t_1, \ldots, t_n$ can fire. The case c) represents a general composition, where tokens must be present in all the pre-places $(p_1, \ldots, p_n)$ of the transition t to enable it for firing. Otherwise the firing of the transition is not possible. After firing of a corresponding transition, however its post-place(s) are marked and the processes initiated in those places are able to start.

The general composition (the case c) of Figure 1) can be understood as a generalization of three basic kinds of composition: sequential, parallel and synchronization (see Figure 2). The fourth of the basic composition mechanisms being the alternative composition as it was mentioned above. Let n be the number of pre-places and m the number of post-places of the transition t, then:

- if $n = 1 \wedge m = 1$ we obtain the sequential composition (Figure 2 a)),
- if $n = 1 \wedge m > 1$ we obtain the parallel composition (Figure 2 b)),

- if $n > 1 \wedge m = 1$ we obtain the synchronization (Figure 2 c)).

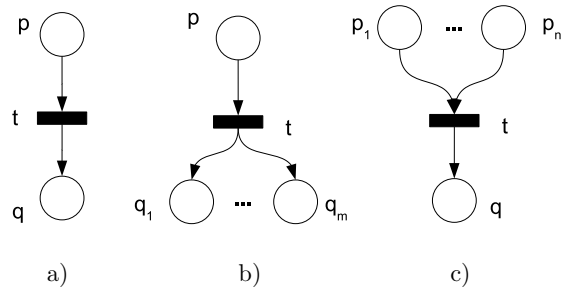


Figure 2. Basic compositions as a special cases of general composition

Now we can proceed to construction of terms representing the possible computations for particular types of places. When constructing the algebraic semantics for a Petri net, the terms will be bound to corresponding E-variables.

Definition 2. According to the structure of a Petri net in the vicinity of a given place, the bounding of terms to corresponding variables for elementary situations depicted in Figures 1, 2 and 3 is defined as follows:

1. deadlock (Figure 1 a)): $E(p) = \delta$,
2. alternative composition (Figure 1 b)): $E(p) = t_1 \cdot E(q_1) + t_2 \cdot E(q_2) + \ldots + t_n \cdot E(q_n)$,
3. sequential composition (Figure 2 a)): $E(p) = t \cdot E(q)$,
4. parallel composition (Figure 2 b)): $E(p) = t \cdot (E(q_1) \parallel \ldots \parallel E(q_m))$,
5. synchronization (Figure 2 c)): $E(p_1) = t_{p1}$, $E(p_2) = t_{p2}$, $\ldots$, $E(p_n) = t_{pn}$,
6. transition without post-place(s) (Figure 3 a)): $E(p) = t$,
7. transition without pre-place(s) (Figures 3 b) and 3 c)): $E(p) = t \cdot (E(p) \| E(q))$.

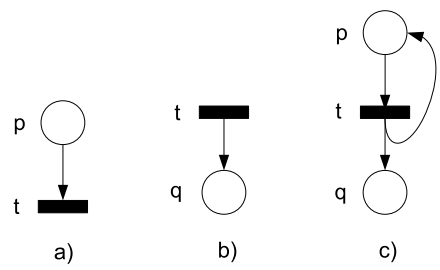


Figure 3. Transitions without input/output

We further give additional notes to the case 5 (synchronization) and the case 7 (transition without pre-place(s)) of the Definition 2. In the case of synchronization [1], the new auxiliary synchronization process is also defined in the following way:

$$Sync_t = (t_{s1} \| t_{s2} \| \dots \| t_{sn}) \cdot t \cdot (E(q) \| Sync_t), \tag{1}$$

ensuring that the synchronization must precede the execution of the action t. To achieve the goal, the following communication functions are also defined: $t_{s1}|t_{p1} = t_{c1}, t_{s2}|t_{p2} = t_{c2}, \dots, t_{sn}|t_{pn} = t_{cn}$. A set of actions to hide (rename to internal action τ), is defined to make the additional synchronization actions invisible: $I = \{t_{c1}, t_{c2}, \dots t_{cn}\}$. A set of actions to encapsulate (rename to δ) is defined too, in order to force the communication of processes to synchronize with the $Sync_t$ process (1) introduced above: $H = \{t_{s1}, t_{p1}, t_{s2}, t_{p2}, \dots, t_{sn}, t_{pn}\}$. The process is then composed (using the parallel composition operator of the ACP) with the rest of the system (2).

Taking into account the case, when a transition without pre-places occurs within the net structure (Figure 3 b)), the following solution is proposed: for every such transition t, a new pre-place is created, which is connected to the transition by two arcs, such that the firing properties of the transition are preserved (Figure 3 c)). Combining the basic principles explored so far, we are able to construct the terms for more complicated net structures.

Definition 3. Let the Petri net be given by the $N = (P, T, pre, post)$, $m \in \mathbb{N}^k$ defines the initial marking and $k = |P|$. Then the algebraic (ACP) semantics for the Petri net N and the marking m is given by the formula:

$$\mathcal{A}(N, m) = \tau_I(\partial_H(E(p_1)^{(i_1)} \| \dots \| E(p_k)^{(i_k)} \| Sync_{t1} \| \dots \| Sync_{ts})). \tag{2}$$

Within the Equation (2), $E(p_i)$ stands for an ACP-term, defined according to a Petri net structure in the vicinity of the place p_i (see Definition 2). The value i_j, given by $i_j = m(p_j)$, represents the marking with respect to the place $p_j, 1 \leq j \leq k$. By the $E(p_j)^{(i)}$ we mean the term $E(p_j) \| \dots \| E(p_j)$, representing a multiple (i-times) parallel composition of a process $E(p_j)$. Note that the $E(p_j)^{(0)} = \delta$. The $Sync_{ti}$ $(1 \leq i \leq s)$ processes are composed to the rest of the system in the case there are s synchronizing transitions within the source (Petri net) specification.

The size of a process $\mathcal{A}(N, m)$, corresponding to Petri net N with the marking m, created by the transformation depends on the number of marked places in the marking m $(E(p_1) \| \dots \| E(p_k))$, as well as on the number of synchronizing transitions within the net N $(Sync_{t1} \| \dots \| Sync_{ts})$. If we suppose the number of tokens in the marking m is $O(k)$, where the $k = |P|$ and the number of synchronizing transitions s is $O(n)$, where $n = |T|$, we can express the number of process variables in the term $\mathcal{A}(N, m)$ by the sum $O(k) + O(n)$. However, the complexity of the term can vary in the case we start to modify it in order to investigate the behavior of the system it describes, as it can be observed in an example introduced within the next section.

Theorem 1. For given Petri net $N = (P, T, pre, post)$, m, m' markings of the net N, ACP-terms $\mathcal{A}(N, m)$ and $\mathcal{A}(N, m')$, representing the algebraic semantics for the net N, and transition $t \in T$ holds:

$$(N, m) \xrightarrow{t} (N, m') \Rightarrow \mathcal{A}(N, m) \xrightarrow{t} \mathcal{A}(N, m').$$

Proof. The proof is constructed by examining all the elementary cases given by the structure of a Petri net from the point of view of its individual places, as described in the Definition 2. If the step in computation of the Petri net N exists $(N, m) \xrightarrow{t} (N, m')$, so the transition t is enabled in the marking m ($\forall p_i \in ({}^\bullet t) : m(p_i) \geq pre(p_i, t)$) and after firing it produces the new marking m' ($\forall p_i \in P : m'(p_i) = m(p_i) - pre(p_i, t) + post(p_i, t)$), a step should also exist in the corresponding algebraic specification $\mathcal{A}(N, m) \xrightarrow{t} \mathcal{A}(N, m')$. As it was defined in the Definition 3, the algebraic semantics for the Petri net N with the marking m is given by:

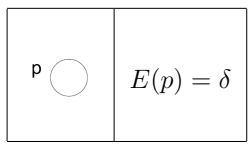
$$\mathcal{A}(N, m) = \tau_I(\partial_H(E(p_1)^{(i_1)} \| \ldots \| E(p_k)^{(i_k)} \| Sync_{t1} \| \ldots \| Sync_{ts})). \tag{3}$$

According to transition relations of ACP (Table 2), a step within the algebraic specification can be performed either by executing an atomic action of a process, or by means of communication of two processes, when the corresponding communication function is defined.

Within the Petri net N, the type of the transition t under consideration can be classified according to size of the set of its pre-places $|{}^\bullet t|$. The case $|{}^\bullet t| = 1$ applies to all the following elementary situations of the Definition 2, except the case 1 (the deadlock) and the case 5 (synchronization, where $|{}^\bullet t| \geq 2$), which will be treated in a slightly different way within the proof. Let us suppose ${}^\bullet t = \{p\}$ (except the cases 1 and 5) and $m(p) \geq pre(p, t)$ (where $m(p)$ is the marking of the place p in the Petri net N), so the transition t is enabled and can fire. Then it also holds that $m'(p_i) = m(p_i) + post(p_i, t) - pre(p_i, t), p_i \in P$ is a new marking of N after firing the transition t. If $E(p)$ represents the corresponding ACP semantics for the process initiated in the place p (Definition 2), then the step (t) is enabled in the $E(p)$ and the $E(p)$ is present in the $\mathcal{A}(N, m)$ (Definition 3).

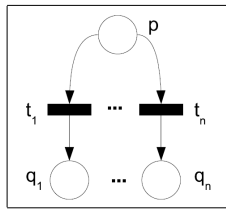
In the following lines we will explore the elementary cases in the order given by the Definition 2.

1. The case represents the situation, where no computation is available, since no transition is connected to the place p.



   The corresponding algebraic term (δ), representing a deadlock is generated.

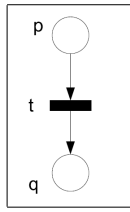
2. The case represents the alternative composition, where the corresponding algebraic term is generated in a way:



$$E(p) = t_1 \cdot E(q_1) + t_2 \cdot E(q_2) + \ldots + t_n \cdot E(q_n)$$

According to transition relations of process algebra ACP (Table 2, case c) the step is possible and one of the alternatives is chosen, given by the transition under consideration ($t \in \{t_1, \ldots, t_n\}$).

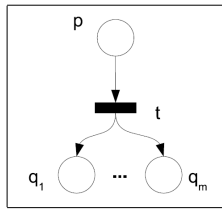
3. The case represents the sequential composition, where the corresponding algebraic term is generated:



$$E(p) = t \cdot E(q)$$

According to the transition relations of process algebra ACP (Table 2, case c), the step t is possible within the corresponding algebraic term.

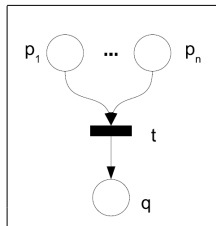
4. The case represents the parallel composition, where the following algebraic term is generated:



$$E(p) = t \cdot (E(q_1) \parallel \ldots \parallel E(q_m))$$

Again, the realization of action t is possible (Table 2, case c) within the corresponding term, regardless the more complicated nature of the process following the action.

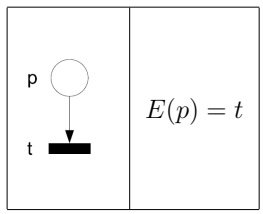
5. In the case, a firing of the transition t with two (or more) preconditions in Petri net N occurs ($|{}^\bullet t| \geq 2$). The situation is depicted in the figure below, which essentially represents the synchronization of processes $p_1, \ldots, p_n$.



$$E(p_1) = t_{p1}, E(p_2) = t_{p2}, \ldots, E(p_n) = t_{pn}$$
$$Sync_t = (t_{s1} \| t_{s2} \| \ldots \| t_{sn}) \cdot t \cdot (E(q) \| Sync_t)$$

If within the marking m of the Petri net N the places $p_1, \ldots, p_n$ contain enough tokens (so the conditions hold: $m(p_1) \geq pre(p_1, t), \ldots, m(p_n) \geq pre(p_n, t)$), the transition t can fire. From the definition of ACP semantics for the Petri net N (Definition 3) and the definition of terms bound to corresponding E-variables (Definition 2, case 5), there is a step available in $\mathcal{A}(N, m)$, too. The step is represented by the communications $(t_{si}|t_{pi} = t_{ci},\ 1 \leq i \leq n)$ of atomic actions $(t_{pi}, 1 \leq i \leq n)$ of synchronizing processes with the corresponding actions $(t_{si}, 1 \leq i \leq n)$ of the $Sync_t$ process. The actions resulting from the communications $(t_{ci}, 1 \leq i \leq n)$ are renamed to internal actions (τ) by means of the operator τ_I and must precede the execution of the action t. Attempts to execute the actions mentioned above (t_{pi} and t_{si}, $1 \leq i \leq n$) without communication are blocked, using the operator ∂_H (Definition 3).

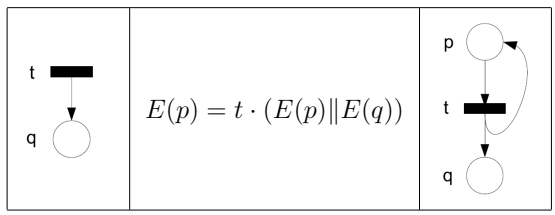
6. Within the case, represented by a transition without post-place(s), the corresponding algebraic term is generated:



$$E(p) = t$$

In this particular case, the step t is possible within the process $E(p)$ in accordance with the transition relations of process algebra ACP (Table 2, case b) and the process terminates after executing the action.

7. The case represents a transition without pre-place(s), where the following algebraic term is generated:



By the substitution introduced within the Definition 2, we have a fragment of Petri net, with the additional place p, which is connected to the transition t by

two arcs, such that the firing properties of the transition are preserved (i.e. if the place p is marked, the transition t can fire arbitrary many times). The step t is possible within the process $E(p)$ in accordance with the transition relations of process algebra ACP (Table 2, case c) and the action t can be executed arbitrary many times.

When the step $\mathcal{A}(N,m) \xrightarrow{t} \mathcal{A}(N,m')$ occurs, the corresponding ACP semantics of the net (N,m') will contain a subterm given by the right hand side of the particular equation for the $E(p)$. The subterm itself represents the particular changes to marking of post-places of the transition t ($t^\bullet$) of the Petri net N. Disappearing of the subterm $E(p)$ on the other hand represents the consumption of the token from the pre-place (p) of the transition t.

We can conclude, that if a step in the Petri net N with the marking m is enabled, so it is enabled also in the corresponding algebraic representation $\mathcal{A}(N,m)$. □

5 AN EXAMPLE

An example is provided within this section to demonstrate the way of using the transformation rules proposed above. The Petri net (depicted in Figure 4) represents a simple system with two processes synchronizing their executions on two common actions. The synchronizing actions are modeled by the transitions t_2 and t_3.

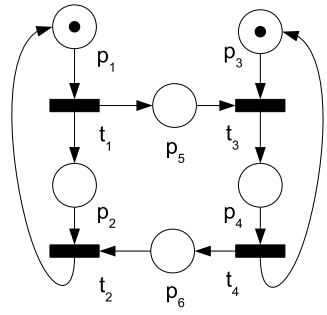


Figure 4. Petri net for two processes

At the beginning we assign the ACP-terms to variables constructed for every place of (source) Petri net N, according to the structure of the net in the neighborhood of the corresponding place.

$$\begin{aligned} &E(p_1) = t_1 \cdot (E(p_5) \| E(p_2)), E(p_2) = t_{2p2}, E(p_3) = t_{3p3}, \\ &E(p_4) = t_4 \cdot (E(p_6) \| E(p_3)), E(p_5) = t_{3p5}, E(p_6) = t_{2p6}. \end{aligned} \tag{4}$$

Next, the synchronizing processes are defined (because of the presence of synchronizing transitions t_2 and t_3):

$$\begin{aligned} Sync_{t2} &= (t_{2s2} \| t_{2s6}) \cdot t_2 \cdot (E(P_1) \| Sync_{t2}), \\ Sync_{t3} &= (t_{3s3} \| t_{3s5}) \cdot t_3 \cdot (E(P_4) \| Sync_{t3}). \end{aligned} \tag{5}$$

We also define four communication functions as we proposed in the Definition 2:

$$t_{2s2}|t_{2p2} = t_{2c2}, t_{3s3}|t_{3p3} = t_{3c3}, t_{2s6}|t_{2p6} = t_{2c6}, t_{3s5}|t_{3p5} = t_{3c5}. \tag{6}$$

A set of actions to encapsulate is defined, in order to force the communication with the $Sync_{t2}$ and the $Sync_{t3}$ processes:

$$H = \{t_{2s2}, t_{2p2}, t_{2s6}, t_{2p6}, t_{3s3}, t_{3p3}, t_{3s5}, t_{3p5}\},$$

as well as a set of actions to abstract away and so hide the internal behavior of the system:

$$I = \{t_{2c2}, t_{2c6}, t_{3c3}, t_{3c5}\}.$$

Since the initial marking of the Petri net N is given as $m_0 = (1, 0, 1, 0, 0, 0)$, only two places (p_1 and p_3) hold tokens. Now initially, only the variables corresponding to these places will be present in an equation describing the algebraic semantics of the Petri net N, together with the processes $Sync_{t2}$ and $Sync_{t3}$, composed by the means of parallel composition. Let the name of the whole composition be Sys and the names of variables $E(p_i)$ be E_i $(1 \leq i \leq 6)$ for the sake of simplicity within this example.

$$\mathcal{A}(N, m_0) = Sys = \tau_I(\partial_H(E_1 \| E_3 \| Sync_{t2} \| Sync_{t3})). \tag{7}$$

According to Equations (4) and (5), we can substitute for the variables present in the specification:

$$Sys = \tau_I(\partial_H(\underline{t_1} \cdot (\underline{E_2} \| \underline{E_5}) \| t_{3p3} \| (t_{2s2} \| t_{2s6}) \cdot t_2 \cdot (E_1 \| Sync_{t2}) \| (t_{3s3} \| t_{3s5}) \cdot t_3 \cdot (E_4 \| Sync_{t3}))).$$

Since the actions $t_{3p3}, t_{2s2}, t_{2s6}, t_{3s3}, t_{3s5}$ are encapsulated by ∂_H, the only available action for execution is the t_1. We also substitute for the E_2 and E_5 process variables and we have (the transition t_1 and the variables E_2 and E_5 are underlined in the equation above for the sake of lucidity):

$$Sys = t_1 \cdot \tau_I(\partial_H(t_{2p2} \| \underline{t_{3p5}} \| \underline{t_{3p3}} \| (t_{2s2} \| t_{2s6}) \cdot t_2 \cdot (E_1 \| Sync_{t2}) \| (\underline{t_{3s3}} \| \underline{t_{3s5}}) \cdot t_3 \cdot (E_4 \| Sync_{t3}))).$$

In this situation all the initial actions of the composed processes are blocked by the ∂_H operation, so the communication (according to definition of communication functions (6)) is the only available action. So we allow to communicate the actions t_{3p5} with t_{3s5} and t_{3p3} with t_{3s3}:

$$Sys = t_1 \cdot \tau_I(\partial_H(t_{2p2} \| (t_{2s2} \| t_{2s6}) \cdot t_2 \cdot (E_1 \| Sync_{t2}) \| (\underline{t_{3c3}} \| \underline{t_{3c5}}) \cdot t_3 \cdot (E_4 \| Sync_{t3}))).$$

After hiding the products of communications (t_{3c3} and t_{3c5}) by renaming them to silent actions τ by means of the abstraction operation (τ_I), we have:

$$Sys = \underline{t_1 \cdot \tau \cdot \tau} \cdot \tau_I(\partial_H(t_{2p2} \| (t_{2s2} \| t_{2s6}) \cdot t_2 \cdot (E_1 \| Sync_{t2}) \| \underline{t_3} \cdot (\underline{E_4} \| Sync_{t3}))).$$

At this point we have the action t_3 which is not affected by the operators τ_I and ∂_H, so it can be placed before them. We also can remove the silent actions (τ) using the axiom B1 (Table 1) and substitute for the variable E_4.

$$Sys = t_1 \cdot t_3 \cdot \tau_I(\partial_H(t_{2p2} \| (t_{2s2} \| t_{2s6}) \cdot t_2 \cdot (E_1 \| Sync_{t2}) \| \underline{t_4} \cdot (\underline{E_6} \| E_3) \| Sync_{t3})).$$

Further, the action t_4 can be moved to the position after the t_3, since it is not affected by the operators τ_I and ∂_H. We also substitute the action t_{2p6} for the variable E_6 within this step.

$$Sys = t_1 \cdot t_3 \cdot t_4 \cdot \tau_I(\partial_H(\underline{t_{2p2}} \| (\underline{t_{2s2}} \| \underline{t_{2s6}}) \cdot t_2 \cdot (E_1 \| Sync_{t2}) \| \underline{t_{2p6}} \| E_3 \| Sync_{t3})).$$

Communications in this case produce the actions t_{2c2} and t_{2c6}.

$$Sys = t_1 \cdot t_3 \cdot t_4 \cdot \tau_I(\partial_H((\underline{t_{2c2}} \| \underline{t_{2c6}}) \cdot t_2 \cdot (E_1 \| Sync_{t2}) \| E_3 \| Sync_{t3})).$$

These actions are not blocked by ∂_H, but are renamed by τ_I to silent actions τ.

$$Sys = t_1 \cdot t_3 \cdot \underline{t_4 \cdot \tau \cdot \tau} \cdot \tau_I(\partial_H(\underline{t_2} \cdot (E_1 \| Sync_{t2}) \| E_3 \| Sync_{t3})).$$

The silent actions can be removed using the axiom B1 (Table 1) and the action t_2 is moved out of scope of the operators τ_I and ∂_H.

$$Sys = t_1 \cdot t_3 \cdot t_4 \cdot t_2 \cdot \tau_I(\partial_H(E_1 \| Sync_{t2} \| E_3 \| Sync_{t3})).$$

After exchanging the order of process variables $Sync_{t2}$ and E_3 we have the equation:

$$Sys = t_1 \cdot t_3 \cdot t_4 \cdot t_2 \cdot \tau_I(\partial_H(E_1 \| E_3 \| Sync_{t2} \| Sync_{t3})).$$

Where the term $\tau_I(\partial_H(E_1 \| E_3 \| Sync_{t2} \| Sync_{t3}))$ is indeed the one from which we started the derivation (7), so we have:

$$Sys = t_1 \cdot t_3 \cdot t_4 \cdot t_2 \cdot Sys. \tag{8}$$

The prefix $t_1 \cdot t_3 \cdot t_4 \cdot t_2$ represents the trace of actions executed by the process Sys before it starts the execution from the initial state again. Thus finally, the operation of the system can be described by the following expression, where the symbol ω expresses the arbitrary number of repetitions.

$$Sys = (t_1 \cdot t_3 \cdot t_4 \cdot t_2)^{\omega}. \tag{9}$$

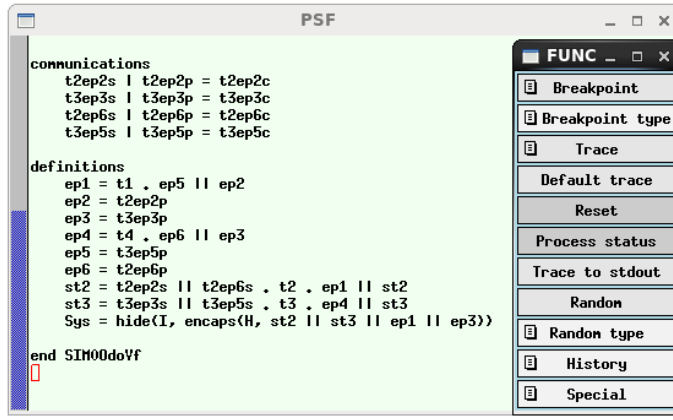


Figure 5. PSF-Toolkit simulation environment

For the purpose of practical validation of the results obtained by means of manual derivation given above, we also performed the simulation using the PSF-Toolkit [14]. The environment of the PSF-Toolkit, together with (part of) the specification is depicted in Figure 5.

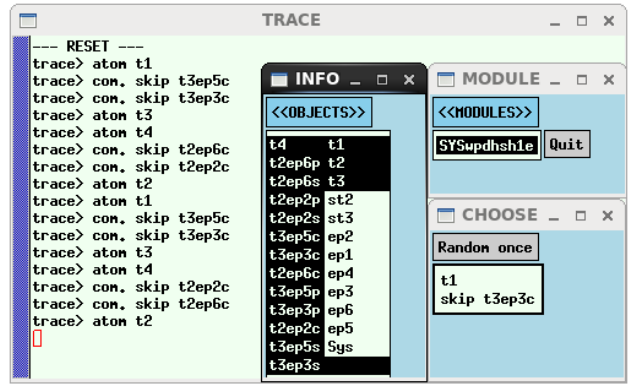


Figure 6. Simulation of the system using the PSF-Toolkit

The names of processes and actions here differ slightly in some cases (like the $st2$ instead of $Sync_{t2}$, or the $t2ep6c$ instead of t_{2c6}), but we hope they are still clear enough. The simulation itself provides the results, which are in-line with the results stated above and the trace of actions (Figure 6) acquired by the simulation of the system corresponds to (9).

Further, we can define a new process (named *ReSys* in Figure 7), corresponding to the external behavior of the system, as it is stated above (8). Finally, we can compare the behavior of the two processes (*Sys* and *ReSys*), using the

```
definitions
  ep1 = t1 . (ep5||ep2)
  ep2 = t2ep2p
  ep3 = t3ep3p
  ep4 = t4 . (ep6||ep3)
  ep5 = t3ep5p
  ep6 = t2ep6p
  st2 = (t2ep2s||t2ep6s) . t2 . (ep1||st2)
  st3 = (t3ep3s||t3ep5s) . t3 . (ep4||st3)
  Sys = hide(I,encaps(H,(st2||st3||ep1||ep3)))
  ReSys = t1 . t3 . t4 . t2 . ReSys
```

Figure 7. Process definitions with the *ReSys* process

equiv tool of the PSF-Toolkit (Figure 8) and find that the processes are bisimilar.

```
[tom@localhost 2018b]$ psf wpd_hsh1er.psf
[tom@localhost 2018b]$ trans SYSwpdhsh1er.til SYSwpdhsh1er.out
[tom@localhost 2018b]$ equiv SYSwpdhsh1er.out

Enter state #1 (none to exit): Sys
Enter state #2 (none to exit): ReSys
Sys and ReSys are branching bisimilar
```

Figure 8. Test of equivalence

6 CONCLUSION

In the paper we presented an approach for constructing an algebraic semantics for Petri nets, based on Algebra of Communicating Processes (ACP). A variable is created for every place of the given Petri net and a term is bound to the variable, which expresses the process initiated in the place. The description of a process representing the computations of Petri net is given by the parallel composition of all the variables associated with the places holding tokens within the initial marking.

The ideas presented in the paper were implemented within the Petri2ACP tool. Similarly, as it was in the case of the Petri2APC tool [31], where the first step is parsing the PNML specification of a Petri net, the PNML Framework [19] was used in this case, too. The resulting ACP specification is produced in a format of the PSF-Toolkit. The tool was implemented using Java platform and is intended to perform a deeper exploration of properties of the transformation, as well as to enable its practical utilization.

As it was demonstrated by the example, a manual investigation of system behavior is possible, but it can be complicated and error-prone task. So having the tool support (e.g. the PSF-Toolkit in this case) available to help with this task is very useful and it was indeed one of the main sources of motivation for this work. One of our previously developed transformations (Petri2APC [30]), which utilizes a dedicated process algebra APC (Algebra of Process Components), provides simpler algebraic specifications indeed, but there is currently no tool support for an-

alyzing such specifications available. The algebraic specifications produced by the Petri2APC transformation are often simpler, since in the APC special constructs are available for expressing the synchronization of processes. So a certain price to pay for the possibility of utilizing powerful tools in the case of ACP is the more complicated modeling of process synchronization based on communication, encapsulation and abstraction.

Although a reverse transformation to the one presented in this work, i.e. the Petri net semantics for ACP terms, is not covered by this paper in detail; we published some results in this respect in the past [28, 29]. An interesting application area of that transformation and the corresponding tool (ACP2Petri) is the design and analysis of communication protocols based on integration of the two mentioned formal methods [32]. It would be very interesting to study the relations between the two transformations in more details.

In the future, it would also be interesting to relate a high-level Petri nets with some variant of data-enriched process algebra [8]. Another possible direction of the future research is connected with the process algebra APC and development of tools supporting the analysis of APC specifications. It would also be useful to enhance the implementation of the transformation tool Petri2ACP in order to include the support for additional tools, like mCRL2 [12].

Acknowledgment

This work was supported by VEGA Grant No. 1/0341/13 Principles and methods of automated abstraction of computer languages and software development based on the semantic enrichment caused by communication.

REFERENCES

[1] Astesiano, E.—Broy, M.—Reggio, G.: Algebraic Specification of Concurrent Systems. In: Astesiano, E., Kreowski, H. J., Krieg-Brückner, B. (Eds.): Algebraic Foundations of Systems Specification. Springer, Berlin, Heidelberg, IFIP State-of-the-Art Reports, 1999, pp. 467–520, doi: 10.1007/978-3-642-59851-7_13.

[2] Baeten, J. C. M.—Weijland, W. P.: Process Algebra. Cambridge University Press, 1990, doi: 10.1017/cbo9780511624193.

[3] Baeten, J. C. M.: A Brief History of Process Algebra. Theoretical Computer Science, Vol. 335, 2005, No. 2–3, pp. 131–146, doi: 10.1016/j.tcs.2004.07.036.

[4] Baeten, J. C. M.—Basten, T.: Partial-Order Process Algebra (and Its Relation to Petri Nets). In: Bergestra, J. A., Ponse, A., Smolka, S. A. (Eds.): Handbook of Process Algebra. Chapter 13. Elsevier Science, Amsterdam, The Netherlands, 2001, pp. 769–872.

[5] Baldan, P.—Bonchi, F.—Gadducci, F.—Monreale, G. V.: Asynchronous Traces and Open Petri Nets. In: Bodei, C., Ferrari, G., Priami, C. (Eds.): Programming Languages with Applications to Biology and Security. Springer, Cham, Lecture

Notes in Computer Science, Vol. 9465, 2015, pp. 86–102, doi: 10.1007/978-3-319-25527-9_8.

[6] Basten, T.—Voorhoeve, M.: An Algebraic Semantics for Hierarchical P/T Nets. Computing Science Report, Ph.D. thesis, Eindhoven University of Technology, 1995, doi: 10.1007/3-540-60029-9_33.

[7] Basten, T.: In Terms of Nets: System Design With Petri Nets and Process Algebra. Ph.D. thesis, Eindhoven University of Technology, 1998, doi: 10.6100/IR516117.

[8] Bergstra, J. A.—Middelburg, C. A.: Data Linkage Algebra, Data Linkage Dynamics, and Priority Rewriting. Fundamenta Informaticae, Vol. 128, 2013, No. 4, pp. 367–412.

[9] Bernardinello, L.—De Cindio, F.: A Survey of Basic Net Models and Modular Net Classes. In: Rozenberg, G. (Ed.): Advances in Petri Nets 1992. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 609, 1992, pp. 304–351, doi: 10.1007/3-540-55610-9_177.

[10] Best, E.—Wimmel, H.: Structure Theory of Petri Nets. In: Jensen, K., van der Aalst, W. M. P., Balbo, G., Koutny, M., Wolf, K. (Eds.): Transactions on Petri Nets and Other Models of Concurrency VII. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 2013, Vol. 7480, pp. 162–224, doi: 10.1007/978-3-642-38143-0_5.

[11] Best, E.—Devillers, R.—Koutny, M.: Petri Nets, Process Algebras and Concurrent Programming Languages. Lectures on Petri Nets II: Applications (ACPN 1996). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1492, 1998, pp. 1–84, doi: 10.1007/3-540-65307-4_46.

[12] Cranen, S.—Groote, J. F.—Keiren, J. J. A.—Stappers, F. P. M.—de Vink, E. P.—Wesselink, J. W.—Willemse, T. A. C.: An Overview of the mCRL2 Toolset and Its Recent Advances. In: Piterman, N., Smolka, S. A. (Eds.): Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2013). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7795, 2013, pp. 199–213, doi: 10.1007/978-3-642-36742-7_15.

[13] Desel, J.—Juhás, G.—Lorenz, R.: Process Semantics of Petri Nets over Partial Algebra. In: Nielsen, M., Simpson, D. (Eds.): Application and Theory of Petri Nets 2000 (ICATPN 2000). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1825, 2000, pp. 146–165, doi: 10.1007/3-540-44988-4_10.

[14] Diertens, B.: Software Engineering with Process Algebra. Ph.D. thesis, University of Amsterdam, 2009.

[15] Ding, Z.—Liu, J.—Wang, J.: An Executable Service Composition Code Automatic Creation Tool Based on Petri Net Model. Computing and Informatics, Vol. 32, 2013, No. 5, pp. 968–986.

[16] Fokkink, W.: Modelling Distributed Systems. Protocol Verification with mCRL. 2nd edition. Springer, 2011.

[17] Gorrieri, R.—Versari, C.: A Process Calculus for Expressing Finite Place/Transition Petri Nets. Proceedings of 17th International Workshop on Expressiveness in Concurrency (EXPRESS'10), Paris, France, August 30, 2010, doi: 10.4204/eptcs.41.6.

[18] Gorrieri, R.: Language Representability of Finite P/T Nets. In: Bodei, C., Ferrari, G., Priami, C. (Eds.): Programming Languages with Applications to Biology and Security. Springer, Cham, Lecture Notes in Computer Science, Vol. 9465, 2015, pp. 262–282, doi: 10.1007/978-3-319-25527-9_17.

[19] Hillah, L. M.—Kordon, F.—Petrucci, L.—Trèves, N.: PNML Framework: An Extendable Reference Implementation of the Petri Net Markup Language. In: Lilius, J., Penczek, W. (Eds.): Applications and Theory of Petri Nets (PETRI NETS 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6128, pp. 318–327, doi: 10.1007/978-3-642-13675-7_20.

[20] Hudák, Š.: Reachability Analysis of Systems Based on Petri Nets. Elfa s.r.o., Košice, 272 pp., 1999.

[21] Hudák, Š.—Zaitsev, D. A.—Korečko, Š.—Šimoňák, S.: MFDTE/PNtool – A Tool for the Rigorous Design, Analysis and Development of Concurrent and Time-Critical Systems. Acta Electrotechnica et Informatica, Vol. 7, 2007, No. 4, pp. 5–12.

[22] Lodaya, K.: A Regular Viewpoint on Processes and Algebra. Acta Cybernetica, Vol. 17, 2006, No. 4, pp. 751–763.

[23] Milner, R.: A Calculus of Communicating Systems. Lecture Notes in Computer Science, Vol. 92, Springer, Berlin, 1980.

[24] Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, Vol. 77, 1989, No. 4, pp. 541–580, doi: 10.1109/5.24143.

[25] Olderog, E. R.: Nets, Terms and Formulas. Cambridge University Press, 1991, doi: 10.1017/cbo9780511526589.

[26] Paige, R. F.: Formal Method Integration via Heterogeneous Notations. Ph.D. dissertation, University of Toronto, 1997.

[27] Sobociński, P.: Representations of Petri Net Interactions. In: Gastin, P., Laroussinie, F. (Eds.): CONCUR 2010 – Concurrency Theory. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6269, 2010, pp. 554–568, doi: 10.1007/978-3-642-15375-4_38.

[28] Šimoňák, S.—Hudák, Š.—Korečko, Š.: ACP2PETRI: A Tool for FDT Integration Support. Proceedings of 8th International Conference (EMES '05), Oradea, Felix Spa, Romania, 2005, pp. 122–127.

[29] Šimoňák, S.: Formal Methods Transformation Optimizations within the ACP2PETRI Tool. Acta Electrotechnica et Informatica, Vol. 6, 2006, No. 1, pp. 75–80.

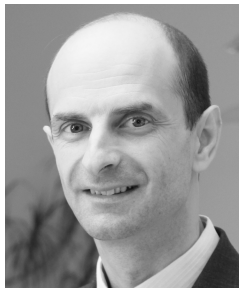
[30] Šimoňák, S.—Hudák, Š.—Korečko, Š.: APC Semantics for Petri Nets. Informatica, Vol. 32, 2008, No. 3, pp. 253–260.

[31] Šimoňák, S.—Hudák, Š.—Korečko, Š.: PETRI2APC: Towards Unifying Petri Nets with Other Formal Methods. Proceedings of the Seventh International Scientific Conference Electronic Computers and Informatics (ECI 2006), Herľany, Slovakia, pp. 140–144.

[32] Šimoňák, S.: Verification of Communication Protocols Based on Formal Methods Integration. Acta Polytechnica Hungarica, Vol. 9, 2012, No. 4, pp. 117–128.

[33] Zhang, X.—Yao, S.: Bank Switching Performance Verification with Object-Oriented Timed Petri Nets. Proceedings of the 2013 IEEE 8th Conference on In-

dustrial Electronics and Applications (ICIEA 2013), Melbourne, Australia, 2013, pp. 1664–1669, doi: 10.1109/iciea.2013.6566636.



Slavomír Šimoňák received his M.Sc. degree in computer science in 1998 and his Ph. D. degree in computer tools and systems in 2004, both from the Technical University of Košice, Slovakia. He is currently Associate Professor at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at the Technical University of Košice, Slovakia. His research interests include formal methods integration and application, communication protocols, algorithms, and data structures.

Martin Tomášek received his M.Sc. degree in computer science in 1998 and his Ph.D. degree in software and information systems in 2005 both at the Faculty of Electrical Engineering and Informatics of the Technical University of Košice, Slovakia. Currently he is Associate Professor at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at the Technical University of Košice, Slovakia. His research interests include distributed systems, component-based systems, and concurrency theory.

Computing and Informatics, Vol. 37, 2018, 1485–1515, doi: 10.4149/cai_2018_6_1485

WEB PERSON NAME DISAMBIGUATION USING SOCIAL LINKS AND ENRICHED PROFILE INFORMATION

Hojjat Emami, Hossein Shirazi

Social Network and Intelligent Systems Laboratory
Department of Information and Communication Technology (ICT)
Malek-Ashtar University of Technology
Tehran, Iran
e-mail: {h_emami, shirazi}@mut.ac.ir

Ahmad Abdollahzadeh Barforoush

Intelligent Systems Laboratory
Computer Engineering and IT Department
Amirkabir University of Technology
Tehran, Iran
e-mail: ahmad@aut.ac.ir

Abstract. In this article, we investigate the problem of cross-document person name disambiguation, which aimed at resolving ambiguities between person names and clustering web documents according to their association to different persons sharing the same name. The majority of previous work often formulated cross-document name disambiguation as a clustering problem. These methods employed various syntactic and semantic features either from the local corpus or distant knowledge bases to compute similarities between entities and group similar entities. However, these approaches show limitations regarding robustness and performance. We propose an unsupervised, graph-based name disambiguation approach to improve the performance and robustness of the state-of-the-art. Our approach exploits both local information extracted from the given corpus, and global information obtained from distant knowledge bases. We show the effectiveness of our approach by testing it on standard WePS datasets. The experimental results are encouraging and show that our proposed method outperforms several baseline methods and also

its counterparts. The experiments show that our approach not only improves the performances, but also increases the robustness of name disambiguation.

Keywords: Web mining, cross-document name disambiguation, social links, profile enrichment, clustering

Mathematics Subject Classification 2010: 97R40, 97R50,68T50, 68U35, 90B40

1 INTRODUCTION

The volume of information available on the web is increasing considerably. This vast volume of information brings new challenges such as lost in hyperspace or information overload [1, 2]. These challenges make it difficult to retrieve useful information about various entities from web pages. Searching for entities, especially people, and their related information is one of the most common activities of Internet users. As personal names are highly ambiguous, personal information extraction and information retrieval systems deal with a fundamental problem, namely name ambiguity problem. The problem of name ambiguity causes the results of a personal name search to be a mix of web pages about different people sharing the same name. This issue emphasizes the necessity of developing high-quality name disambiguation systems to resolve ambiguity between people names and cluster search results according to different people having the same name. Developing such a name disambiguation system can be useful in a wide range of areas including semantic web, information extraction, question answering, machine translation, data fusion, speech recognition, and social network analysis, among others.

In recent years, several research efforts have been conducted towards the name disambiguation in web context. The problem of name disambiguation is usually formulated as a name clustering problem [3, 4, 5, 6]. Clustering-based name disambiguation approaches are well-known due to their superior feasibility and efficiency in dealing with a large amount of data. Clustering-based methods are useful when we do not have a large labeled corpus and there are varying ambiguities in the corpus. A majority of previous work [3, 4, 5, 6, 7, 8, 9, 10, 11] use a combination of various features to compute similarities between entities, and then utilize clustering algorithm to disambiguate entity names. A great deal of research exploits syntactic and semantic local features derived from the given corpus [7, 8, 11]. However, the local information may not be sufficient to resolve ambiguities and the robustness of system will be severely degraded due to

1. the substantial noise and low quality of information extraction (IE) and natural language processing (NLP) systems used for extracting local document-level features, and
2. insufficient information contained in web pages.

To alleviate these problems, authors in [3, 4, 6], besides the local information, have exploited the global information derived from extra corpora or distant knowledge bases. Nevertheless, these solutions do not completely utilize all the semantic information contained in web pages such as entity attributes and social links. In this paper, we attempt to overcome the deficiencies of previous work by proposing a person name disambiguation approach that not only uses local personal attributes and social links stored in given web pages, but also global semantic information about persons embedded in external knowledge bases. Our approach makes a full use of the merits of both attribute-based and social network based methods. Personal attributes and social links are two different sources of information that can complement each other. This leads to more precise person name disambiguation, and confirms the need of a framework for integrating social links and the enriched attributes of a person are needed.

To summarize, our contributions lie in the approaches we propose to solve subtasks of name disambiguation:

- We map all of the information about persons in text documents to an undirected weighted graph. In graph creation process, we propose new methods for each task of *social link extraction*, *profile extraction*, and *profile enrichment*. Specifically, we propose a new method for social links extraction relying on the closeness centrality theory [12]; we propose a profile enrichment method relying on the closeness centrality theory and deep semantic analysis of the text to deal with the problem of data sparseness and to make name disambiguation system more robust.
- We employ BIC-Means algorithm to cluster nodes of graph, and propose a dynamic weight learning method based on a new TF-IDF schema [45] to learn importance coefficient of attributes in computing similarity among entities.
- We perform extensive evaluation of our proposed approach over real, standard WePS datasets [13, 14]. We demonstrate that our method outperforms baseline methods and state-of-the-art counterparts. This justifies that our approach is a promising solution for the problem of person name disambiguation.

Having this short introduction, the rest of this paper is organized as follows. Section 2 is devoted to literature review and presents an overview of the related work. Section 3 introduces the working principle of our name disambiguation approach. Then, in Section 4, the proposed approach is evaluated on benchmark datasets and the results are compared to the baselines and state-of-the-art methods. Finally, Section 5 makes conclusions and discusses some future works.

2 RELATED WORK

Our work addresses the problem of cross-document person name disambiguation in web context. Let $D = \{d_1, d_2, \ldots, d_N\}$ be a collection of web documents referring to a set of persons having the same name, and let $M = \{m_{11}, m_{12}, \ldots, m_{21}, m_{22}, \ldots, \}$,

$m_{ij} \in d_i$ be a set of name observations within collection D, which need to be disambiguated. Name ambiguity can occur within a web document or across documents. In case of within-document ambiguity, name observations in M are from the same document $d_i \in D$. In case of cross-document, the mentions in M are from the entire corpus D. According to the type of ambiguity, name disambiguation systems are categorized into two classes:

1. within-document name disambiguation, which often referred to as within-document co-reference, and
2. cross-document name disambiguation.

Within a document, mentions with the same string typically refer to the same entity in reality, whereas in different documents identical entity mentions may have different meanings [4, 6]. This is important information, which shows that cross-document name disambiguation cannot be solved by applying within-document co-reference resolution to a super document formed by concatenating all documents in the corpus. In this paper, we focus on cross-document name disambiguation. In the following, we briefly present and discuss the most significant research work in the area of cross-document name disambiguation, their limitations, and compare our approach with them. This short discussion highlights the need for developing new and more efficient name disambiguation approaches.

In recent years, many research efforts have been made towards name disambiguation in relational databases and web context. There are several surveys in name disambiguation area, among which we point out Brizan and Tansel [15], Elmagarmid et al. [16], Kopcke and Rahm [17], and Ferreira and Gonçalves [18]. Entity name disambiguation in web is similar to those approaches developed in database domain. However, there are several differences [11]:

1. web documents are often unstructured while database records are structured, and
2. web documents often only contain partial or incomplete information about the entities.

Therefore, most disambiguation methods, which were developed for databases are not directly applicable on web data.

Name disambiguation is often formulated as a clustering problem [3, 4, 5, 6, 11]. Clustering-based name disambiguation approaches are well-known due to their superior efficiency in dealing with a large amount of data. They are useful when we do not have a large labelled corpus, and there are varying ambiguities in a corpus [4]. Clustering methods often include three main steps:

1. feature extraction,
2. similarity computation, and
3. name grouping.

In most of the existing approaches, employed features are either syntactic or semantic. Syntactic features include tokens [7], specific keywords [6, 11], n-gram features, snippet-based features [4], etc. Semantic features include personal attributes [19, 20, 21], hyperlinks [8, 9], named-entities [6, 11], etc. Each web document, which needs to be disambiguated, is represented as a vector of desired features. Similarities among document vectors are then computed using various similarity measures to identify whether they refer to the same entity. Similarity computation forms the basis of name disambiguation in clustering approaches. The quality of similarity computation significantly depends on the type of analyzed input data, similarity measures and features under evaluation. Many existing approaches, such as [19, 20, 21, 22, 23], compute similarity between entities by matching their corresponding feature vectors containing attributes of those entities. However, such methods ignore some important implicit semantic information, such as links between entities. Some other works have harnessed co-occurring entity mentions to compute similarities [8, 24, 25]. These approaches often create a social relationship graph of entity names (especially person names) co-occuring in a document and then partition the graph into sets of groups using graph partitioning algorithms. The idea behind social network-based methods comes from the fact that linked entities might be having the similar characteristics. The main problem of these methods is that they may fail to distinguish entities when a web document does not contain any information about people connections. A few attempts, such as [8, 26], integrate both the entities' links and attributes to resolve ambiguities. The idea behind such models is that the attributes of an entity can complement social network structure, and vice versa. In other words, if one source of information is missing or noisy, the other can make it up. These hybrid models make full use of the merits of both attribute-based and social network based name disambiguation methods. However, these methods do not employ external data sources beyond the given corpus, and use only the information contained in the given corpus being processed. Our approach extends these methods through utilizing both local persons' social links and attributes, and global semantic attributes from distant knowledge bases.

Exploiting external, global features for disambiguation was also studied in previous works [3, 4, 27, 28]. However, the context information including social links among entities and attributes has not been utilized entirely. To alleviate this shortcoming, similar to ours, Dutta and Weikum [6] exploit both the context where entities appear and the information from external knowledge bases for co-referent entities. However, limitations of their approach are that

1. co-occurrence of entities is only considered within web pages,
2. the information contained in web pages is not completely exploited, particularly information expressed in informal-style fragments, and
3. in knowledge enrichment stage, a simple string matching method is utilized for entity matching.

Our approach is robust enough and regardless of the external knowledge features, it uses all of the information contained in the given web pages expressed either in formal-style or informal-style formats. Our approach relies on deep semantic analysis of the text and closeness centrality theory to exploit social links of entities across web pages.

In summary, our work extends previous work by integrating social links with the attributes of the local semantic profile attributes and global attributes from external knowledge bases. Following this way, our approach exploits all of the information about person entities contained in the textual parts of local web pages and external knowledge bases. This leads to more robust name disambiguation approach.

3 OUR PROPOSED APPROACH

In this paper, we formulate the person name disambiguation as a clustering problem. Let $G = \{C_1, C_2, \ldots, C_K\}$ be a set of K clusters, and $C_i \neq \emptyset$, $C_i \cap C_j = \emptyset$, $i \neq j$, $i, j = 1, 2, \ldots, K$, $\bigcup_{j=1}^{K} C_j = D$, where $D = \{d_1, d_2, \ldots, d_N\}$ is the N web pages referring to the different people sharing the same name. The goal of our name disambiguation is to find such G, where objects $\{u_p^i, u_{p+1}^i, \ldots, u_q^i\}$, $(1 \leq p \leq q \leq N)$ in cluster $C_i \in G$ refer to the same entity in reality. Figure 1 shows an overview of our proposed name disambiguation approach. It consists of four main stages as follows:

Pre-processing. Pre-processing takes as input web pages and transforms them to system-desired format using existing pre-processing tools. Pre-processing consists of five main subtasks: *extracting clean text document*, *named-entity tagging*, *intra-document co-reference resolution*, *sentence splitting*, and *sentence type detection*.

Graph creation. This component takes pre-processed text of web documents and extracts person discourse profile; enriches the discourse profile with external semantic features; and extracts social links between entities from the text. It then maps the profile attributes and links into an undirected weighted graph (Attribute-Relationship Graph). This graph is an abstract and structured representation of the information constituents from the web documents and relevant information from external knowledge bases.

Similarity computation. We use a modified random walk model to compute similarities among graph's nodes. The employed similarity measure considers both people's attributes and social links.

Graph clustering. The clustering phase takes as input the attribute-relationship graph and similarity measures, and then groups graph nodes into sets of clusters, where each cluster contains all the nodes referring to a unique person.

In the following, we describe these components in more detail.

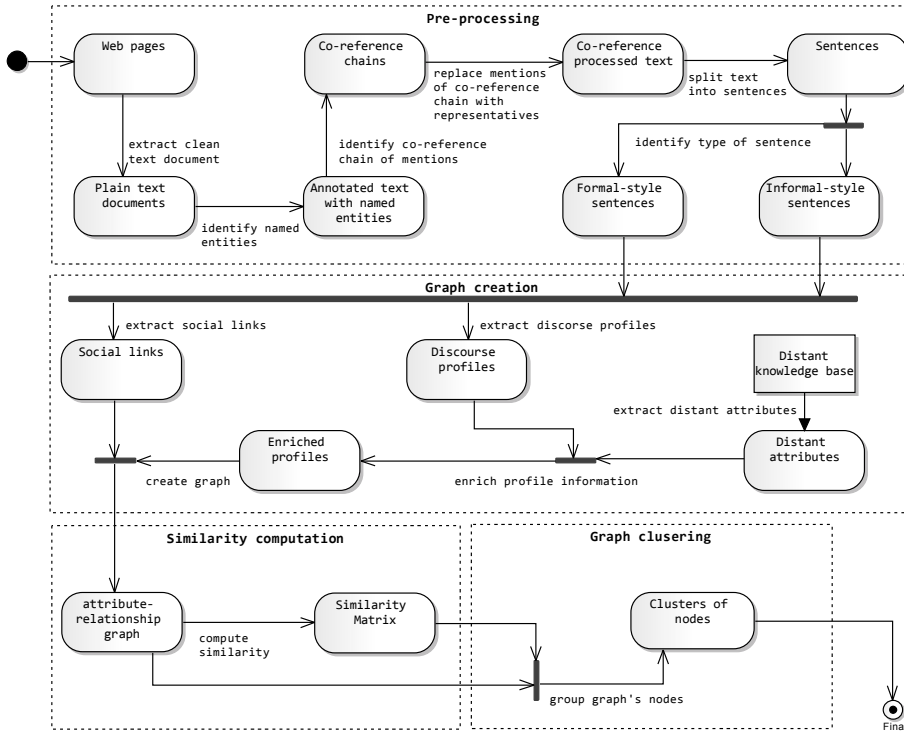


Figure 1. Outline of our name disambiguation approach

3.1 Pre-Processing

In this article, we focus on the textual part of the web pages, because the majority of the information about entities on the web is often expressed in the natural language text. The web pages need to be pre-processed and prepared according to system's desired format. First, for each web page, Jsoup[1] (an HTML parser) is run to cast it into plain text document. Next, for each document the Stanford named-entity tagger [29] is run to tag the text for coarse-grained lexical entity types including person, location, organization, etc. For each identified named-entity, we assign a unique index to distinguish the identity of entity. The annotated text documents are passed to the intra-document co-reference resolution module. We use the state-of-the-art Stanford co-reference resolution system [30] to identify co-reference chains for all the entities mentioned in each document. The mentions in every co-reference chain of interest are then replaced with their corresponding representative mentions. Next, for the co-reference chain of interest within each document, we use the Stanford

[1] Jsoup: Java HTML Parser, `http://jsoup.org/`

CoreNLP toolkit[2] [31] to extract all the sentences from a document. The content on a web page is often expressed in a mixture of different representations: *formal-style* format and *informal-style* format [4, 32]. A formal-style text follows prescribed writing standards, and is prepared for a fairly broad audience [4, 32]. Formal writing needs to be well-structured, clear and unambiguous. Longer and complete sentences are likely to be more prevalent in formal-style text. Complete sentences usually contain a subject, object and one or multiple verbs. On the contrary, informal-style text has few constraints on writing format, mixes various representations, is prepared quickly and intended for a narrow audience [32]. In sentence type identification, we classify each sentence in a web document as one of the two classes, formal-style or informal-style. In Figure 2, we show an excerpt of formal-style and informal-style fragments. We use support vector machines (SVMs) [33] to classify sentences into formal-style or informal-style classes. The main feature for classification is the percentage of capitalized tokens and length of the sentence. The selection of these features comes from the fact that an informal-style sentence mainly is short and contains capitalized tokens. Each of the formal-style and informal-style expression format requires different information extraction method. Identifying the type of sentence expression helps us to overcome the problem of structure variation and choose proper attribute extraction methods (Section 3.2.2) to extract entity-centric information according to data representation format.

The pre-processing tools may produce errors, which propagate to the later stages. However, improving the pre-processing components is beyond the scope of this paper. The remainder of the processing described in the following uses this pre-processed text.

3.2 Graph Creation

In graph creation, the abstract representation we wish to create is an undirected weighted graph G for each pre-processed web document. Formally, we represent this graph as $G = (VS \cup VA, ES \cup EA)$, where VS is the set of structure nodes, VA is the set of attributes nodes, ES is the set of structure edges, and EA is the set of attribute edges. For each entity e, we create a structure node and append it to structure node set VS. Similarly, for each attribute class $a \in A$, we create an attribute node v_a and append it to attribute node set VA. We create a structure edge $es \in ES$ between a pair of structure nodes u and v, if their corresponding entities co-occur in the corpus. The weight of structure edge es between a pair of structure nodes u and v indicates the strength of the relationship between corresponding entities e_i and e_j. We draw an attribute edge $ea \in EA$ between a structure node $u \in VS$ and an attribute node $v_a \in VA$, if the node u corresponding to person e takes a value on attribute class $a \in A$. The weight of attribute edge ea indicates the importance coefficient of the target attribute.

[2] `http://nlp.stanford.edu/software/corenlp.shtml`

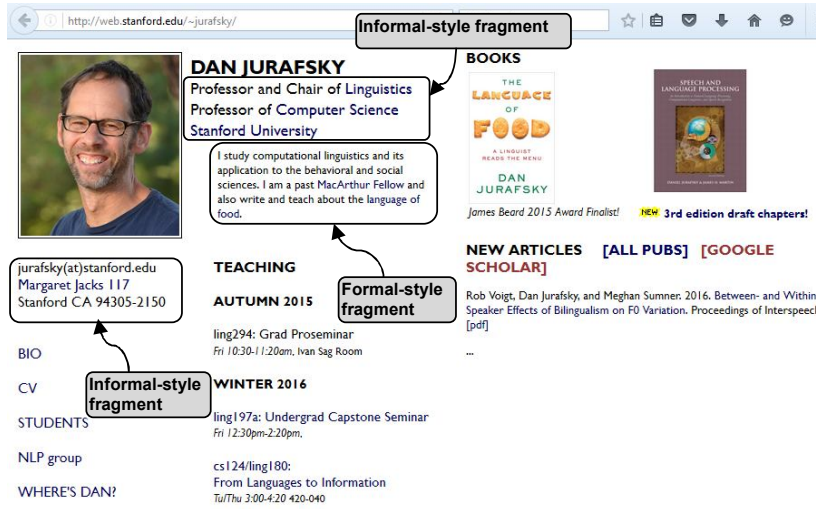


Figure 2. Examples of formal-style and informal-style data expression format on a sample web page

The graph creation consists of three steps:

1. social link extraction,
2. profile extraction, and
3. profile enrichment.

The system first extracts the co-occurring entities in the neighbourhood of each given person. The co-occurring entities form what we call the social relationship network. In the profile extraction stage, the system extracts local attributes associated with every entity in question from given web documents, and forms entity's discourse profile. In the profile enrichment stage, the system then enriches the local discourse profiles with rich global features retrieved from external knowledge bases by considering co-occurring entities and their surrounding context. The linked (co-occurring) entities, global attributes, and local attributes associated with each person are mapped into an undirected graph. In the following, we describe the graph creation stages in more details.

3.2.1 Social Link Extraction

We assume that relationship between entities in the real world is reflected by their closeness in text of the documents they are mentioned in. We assume that two entities are linked if they collocate together in a corpus more frequently. To identify the linked entities with an entity e, we extract the co-occurring entities in the

neighbourhood of entity e. To do this, we first identify sentences in which the focused entity e or its co-referent mentions occur. The sentences of containing these entities contain also other entities that can be co-occurring with other ones as well. Formally, in a document, entities in the neighbourhood of the entity e appear in the following sentences:

$$\begin{aligned} S(e) &= H^n(e), \\ H^1(e) &= \left\{ \bigcup_j s(e_j) \mid \forall \mathrm{e}_j \in M(e) \right\}, \\ H^n(e) &= H^{n-1}(e) \cup \left\{ \bigcup_k H^1(e_k) \mid \exists s_f : (e \in s_f) \wedge (e_k \in s_f) \right\}, \quad \text{for } n \geq 2 \end{aligned} \tag{1}$$

where $S(e)$ is a set of sentences in the neighbourhood of entity e, $H^n(e)$ is a set of n^{th}-hop sentences containing entities co-occurring with target entity e, $M(e)$ is an intra-document co-reference chain of entities with respect to entity e, and s_f is the f^{th} sentence in which both e and e_k co-occur together. Let $N(e) = \{e_1, e_2, \ldots, e_m\}$ be the list of co-occurring entities with target entity e, which appear in $S(e)$. We create a social relationship graph between the entities in $N(e)$. We define the strength of relationship between a pair of linked entities e_i and e_j as the normalized distance-weighted frequency of entities' co-occurrences in the input corpus as follows:

$$w_{ij} = \frac{\theta_{ij}}{\sum_k \sum_l \theta_{kl}} \tag{2}$$

where θ_{ij} is the distance-weighted frequency of entities' co-occurrences in the corpus, which we defined it as follows:

$$\theta_{ij} = \sum_{d \in D} \begin{cases} \sum_{(e_i, e_j) \in d} 1 - \left(\frac{log_2(\chi_{ij})}{2} \right), & \text{if } (\chi_{ij} < \eta), \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where d is a document in the corpus D, e_i and e_j are the i^{th} and j^{th} entities in document d, respectively. χ_{ij} is the position distance between two entities with value 1 if entities e_i and e_j collocate in the same sentence, 2 in neighbouring sentence, and so on. The entities having position distance χ_{ij} above the threshold η are ignored. We empirically set η to 4, which prevents linking far entities. Obviously, the bigger the θ_{ij} is, the bigger the w_{ij} is. w_{ij} is in the range $[0, 1]$. In this paper, we use only the web pages of each ambiguous name in the given corpus as the entity co-occurrence corpus. To compute w_{ij}, one can use large-scale corpora beyond the given corpus as entity co-occurrence corpus. This is considered as one of the future work of this paper. Nonetheless, our experiments show that the given corpus is sufficient for extracting social links.

3.2.2 Profile Extraction

To extract attributes of a certain person and form their discourse profile, we have developed an integrated profile extraction system. Our profile extraction system takes as input the pre-processed text, extracts the person entities with related information and forms their discourse profile. Discourse profile of a person entity contains a set of ⟨attribute, value⟩ pairs. Formally, we define the discourse profile of entity e as follows:

$$P(e) = \{(a, v) \mid a \in A, v \in V(a)\} \tag{4}$$

where A is the vocabulary of attributes that describes characteristics of person entity e, and $V(a)$ represents the set of valid values for attribute $a \in A$. In our implementation, each person entity can take at most sixteen kinds of attributes, which include *affiliation*, *award*, *birth place*, *date of birth*, *other name*, *occupation*, *school*, *major*, *degree*, *mentor*, *nationality*, *relatives*, *phone*, *fax*, *e-mail*, and *website*. Due to limited space, in this paper, we do not give more detailed discussion of our profile extraction system. For more detail refer to our technical report given in (`http://ceit.aut.ac.ir/islab/guest/Emami/PersonProfiling_TechnicalReport.pdf`). Our profiling approach is efficient, and can extract personal attributes from both formal-style and informal-style fragments.

3.2.3 Profile Enrichment

The discourse profile of entities can be applied directly to compute similarities and resolve ambiguity. However, the sparse data contained in discourse profiles may not be sufficient to resolve ambiguities and the system robustness will be degraded due to low quality of profile extraction system. For these reasons, we propose an enrichment method of the persons' profile via global attributes extracted from external knowledge base. Profile enrichment attempts to alleviate the problem of data sparseness and improve the robustness of system. Profile enrichment includes two steps:

1. entity linking, and
2. attribute extraction.

In entity linking step, for an entity mention e, we determine its identity in text to identify the best matching entity in the external knowledge base. In attribute extraction step, we retrieve the global attributes for the target person e from external knowledge base. These attributes are beyond the discourse profile.

Our entity linking system takes as input the target entity mention e and the context $S(e)$ (Equation (1)) around it. It identifies the entity mentions in neighbourhood of entity e and forms a list of co-occurring entity mentions $N(e) = \{e_1, e_2, \ldots, e_m\}$; where each $e_i \in N(e)$ refers to a co-occurring entity mention with target entity mention e appearing in $S(e)$. Entity linking system then extracts the intra-document co-reference chain of the entity e and entities in $N(e)$. To match an entity mention e against an entity from external ontology, our entity linking system creates a phrase query comprising mentions from the co-reference chain of e

and its neighbours $N(e)$. Entity linking system feeds the query to Babelfy [34] to identify the BabelNet synset id of the target entity e, and links it into the corresponding DBPedia URI. Babelfy is a state-of-the-art word sense disambiguation and entity linking system. As a matter of fact, using Babelfy is not mandatory; any disambiguation or entity linking strategy can be used at this stage. However, a knowledge-based unified approach like Babelfy is best suited to our setting. The attribute extraction phase takes as input the DBPedia URI of the target entity e, and retrieves its attribute from DBPedia ontology [36] to enrich discourse profile of entity e. The choice of DBPedia for enrichment is not mandatory, and other knowledge bases such as Freebase and Yago can be used for enrichment purpose. However, DBPedia is best suited to our setting, because it provides high-coverage structured information about entities.

We primarily rely on the Babelfy itself to identify the correct identity of the target entity e. Babelfy may produce some noisy data because in some cases it cannot infer the correct identity of entities. Therefore, to avoid dependency on the output of the Babelfy to infer whether the retrieved external entity t best matches with the target entity e, we rank the candidate entity t by our similarity measure and prune out candidates with low confidence. In similarity computation, we first compare the type tag of the entities e and t. If the entity type tag of entities e and t are not the same, we ignore the external entity t; otherwise we compare the attributes of entity t with local attributes of the target entity e. For this purpose, we compute the normalized similarity between entities t and e based on their attributes:

$$\mathrm{Sim}(e,t) = \frac{1}{|A_{e,t}|} \times \sum_{a \in A_{e,t}} \beta_a \times M_a(e,t) \tag{5}$$

where $M_a(e,t)$ is the similarity of the two entities based on attribute a, β_a is the importance coefficient of attribute a, and $A_{e,t}$ represents all the attributes associated with both entities e and t. $A_{e,t}$ is equal to $A_{e,t} = (A_e \cup A_t)$, where A_e and A_t respectively represent the set of attributes associated with entity e and t. We define a confidence threshold T, such that a candidate entity t having the similarity value $\mathrm{Sim}(e,t)$ below the threshold T is pruned out. If $\mathrm{Sim}(e,t) \geq T$, the system then appends the attributes of entity t to discourse profile of entity e.

In general, each attribute class $a \in A$ may be one of the following types: single-value attribute or multiple-value attribute [37]. Single-value attribute (e.g. *date of birth*) can only take a single value, while multiple-value attribute (e.g. *affiliation*, *occupation*) can take one or more different values. If attribute a is a multiple-value attribute, to compute $M_a(e,t)$ we first compute single-value similarities for all the possible values of a and then aggregate the maximum single-value similarities. Therefore, we define $M_a(e,t)$ as follows:

$$M_a(e,t) = \frac{1}{\min\left(|I_e|, |I_t|\right)} \times \sum_{p \in I_e} \max\left(\delta\left(p, I_t\right)\right) \tag{6}$$

where I_e and I_t represent the item set of attribute a for person e and t, respectively. $\delta(p, I_t)$ is the set of single-value attribute similarities computed between element $p \in I_e$ and all elements in I_t. We define $\delta(p, I_t)$ as follows:

$$\delta(p, I_v) = \{\varphi(p, q) \mid q \in I_t\}. \tag{7}$$

$\varphi(p, q)$ computes the similarity between item p and q using an appropriate standard similarity measure. Personal attributes are heterogeneous; therefore it is not reasonable to use the same similarity measure for different attributes in computing $\varphi(p, q)$. This enforces us to use appropriate similarity measures for any type of the attribute. There are different standard similarity measures, each of which is appropriate for a particular attribute class. In order to determine which similarity measure is appropriate for an attribute class, we adopt the following methodology. Borrowing the idea from [35], first, we adopt five syntactic similarity measures, which include normalized Levenshtein distance (len) [38], Dice's coefficient (dic) [39], Cosine (cos) [39], Jaccard index (jac) [40] and dates' relative similarity (Spd) [48]. The reason to select these measures is that these measures are widely used in literature to calculate similarity of data objects. We then identify the appropriateness of a similarity metric for a particular class of attribute through assessing its importance on name disambiguation. To do this, we use ground truth of training data given in WePS-1 dataset [13] and WePS-2 dataset [14]. We analyse the similarity measures in turn for each person in ground truth of dataset. Each similarity measure was computed for the attributes of each pair of persons that are co-referent, i.e., they are in the same cluster and refer to the same entity in reality. A typical similarity metric is considered to be proper for a particular attribute, if it results in higher normalized similarity value for the people who are co-referent in the ground truth. The simulation results on WePS-1 training dataset are shown in Figure 3, and the results for WePS-2 training dataset are given in Figure 4.

We apply the similarity measures on single-value items of attributes. Each similarity measure has its own strategy to compute similarity value. For example, to compute similarity by Cosine measure, we first transform the single-value items to vectors of occurrences of n-grams (sequences of n characters). In this n-dimensional space, the similarity between two items is the cosine of their respective vectors. In other words, it is computed as $(V_1.V_2) / (|V_1| \times |V_2|)$, where V_1 and V_2 is the vector representation of two comparing items p and q.

For attribute *date of birth*, we first normalize the date values using Stanford SUTime library [41] and then compute the similarity. Borrowing the idea presented in [48], to compare *date of birth* values, we first convert dates into a number of days. We calculate the number of days according to the fix date 01-01-2016. Let d_1 and d_2 be the two day values that are being compared, *Spd*, the dates' relative similarity, is calculated as follows:

$$\mathrm{Spd}(d_1, d_2) = \begin{cases} 1 - \left(\frac{\mathrm{pd}(d_1, d_2)}{pd_{\max}}\right), & \text{if } (\mathrm{pd}(d_1, d_2) < pd_{\max}), \\ 0, & \text{else}, \end{cases} \tag{8}$$

where $pd_{\max}$ $(0 < pd_{\max} < 1)$ is the maximum percentage difference that is tolerated in similarity computation. In our implementations, we empirically set $pd_{\max}$ to 0.2. $\mathrm{pd}(d_1, d_2)$ is the percentage difference, which is defined as follows:

$$\mathrm{pd}(d_1, d_2) = \frac{|d_1 - d_2|}{\max(d_1, d_2)}. \tag{9}$$

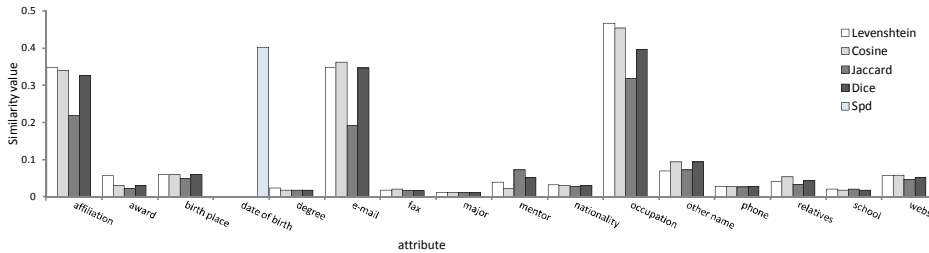


Figure 3. The results reported by different similarity measures for attributes of the co-referent person names in WePS-1 training dataset in terms of $B^3F_{\alpha=0.5}$

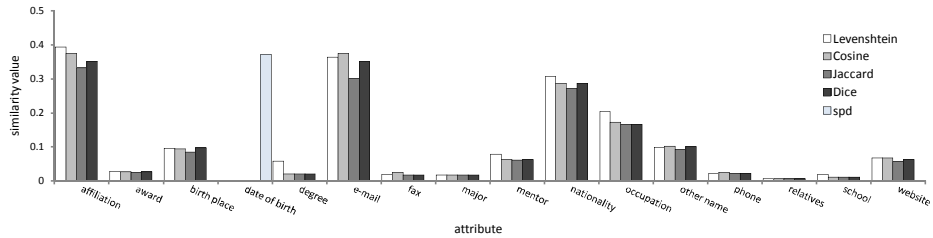


Figure 4. The results reported by different similarity measures for attributes of the co-referent person names in WePS-2 test dataset in terms of $B^3F_{\alpha=0.5}$

Results obtained by our experiments on the given datasets show that the normalized Levenshtein metric is appropriate for the attributes of *affiliation*, *award*, *degree*, *nationality*, *occupation*, and *school*; the Cosine similarity metric for the attributes of *relatives*, *phone*, *fax*, *e-mail*, *website*; and the Jaccard index for the attribute of *mentor*; and Dice coefficient for the attribute of *birth place*. For some attributes one or more similarity measures relatively reported the same results. For the attribute of *other name*, one can use Dice's coefficient or Cosine similarity measure. For the attribute of *major*, it is no matter which similarity measure is used; however, in our implementations, we used the normalized Levenshtein metric for the attribute of *major*. We notice that for attribute *date of birth*, we only use Spd measure.

As shown in Figures 3 and 4, some attributes report higher similarity values rather than others. This means some attributes are more important for name disambiguation. For example, the attribute of *date of birth* is more important than the attribute of *school* for resolving name ambiguity. This issue testifies that we should give different weight for each attribute according to its impact on name disambiguation. To compute β_a, the weight of attribute a, we use its specific similarity measure to compute the average similarity between co-referent persons based on attribute a. The resulting average similarity value is considered as weight of attribute a. To fulfil this aim, we first compute Equation (7) using the specified appropriate similarity measure of the attribute a. We then compute the pairwise similarity between persons using Equation (6) in terms of attribute a. In some cases that we deal with missing data, we consider 1 as the similarity value. Finally, to compute β_a, we calculate the average of pairwise similarity for all co-referent persons:

$$\beta_a = \left(\frac{\sum_{i,j=1}^{R} M_a(e_i, e_j)}{R} \right) / \left(\sum_{k=1}^{m} \max(\beta_{ak}) \right) \tag{10}$$

where R is the number of co-referent persons; e_i and e_j are the i^{th} and j^{th} co-referent entities, respectively. Table 1 shows the appropriate similarity measure and the weight for each attribute on WePS-1 training and WePS-2 training dataset.

		WePS-1 Training	WePS-2 Training
Method	Similarity Metric	β_a	β_a
Affiliation	len	0.164	0.181
Award	len	0.027	0.013
Birth place	dic	0.028	0.044
Date of birth	spd	0.191	0.170
Degree	len	0.011	0.027
Email	cos	0.172	0.172
Fax	cos	0.010	0.011
Major	len	0.006	0.008
Mentor	jac	0.034	0.036
Nationality	len	0.015	0.141
Occupation	len	0.221	0.093
Other name	dic	0.045	0.047
Phone	cos	0.013	0.011
Relatives	cos	0.026	0.003
School	len	0.010	0.009
Web site	cos	0.027	0.031

Table 1. The weight of attributes on WePS-1 training and WePS-2 training datasets

We designed another test to indicate that using appropriate similarity measure for each type of attribute class has great impact on the quality of name disambiguation. To do this, we tested the similarity measures in turn on the dataset, and

computed the similarity matrix. We then use agglomerative clustering algorithm with single linkage merging strategy to group the person names. We compare the results generated by the system with the ground truth included in the WePS training datasets to compute performances. Figure 5 shows the results corresponding to each similarity metric on the WePS-1 training dataset, and Figure 6 shows the results for the WePS-2 training dataset. In Figures 5 and 6, *Fa* refers to B-cubed F-score ($B^3F_{\alpha=0.5}$) [46, 47], and *Fp* refers to purity-based F-score ($F_{p=0.5}$) [46]. Combination similarity measures uses the normalized Levenshtein metric for the attributes of *affiliation*, *award*, *degree*, *nationality*, *occupation*, *school* and *major*, the Cosine similarity metric for the attributes of *relatives*, *phone*, *fax*, *e-mail*, and *website*, and the Jaccard index for attribute *mentor*, Dice coefficient for attribute *other name* and *birth place*, and Spd measure for attribute *date of birth*. As shown in Figures 5 and 6, the correct combination of similarity measures improves the performance of name disambiguation in terms of $B^3F_{\alpha=0.5}$ and $F_{p=0.5}$.

We also designed a test to assess the potential of Babelfy on cross-document name disambiguation. We let Babelfy to disambiguate person names and obtain their BabelNet synset id. We then group persons based on their BabelNet synset id. For this purpose, we use a dump of WePS-1 training and WePS-2 training datasets. The sampled dump contains person names that are taken from Wikipedia. The reason to this choice is that the knowledge base of Babelfy, BabelNet is primarily constructed by linking Wikipedia to WordNet. Figure 7 a) shows the results obtained by the Babelfy system and the attribute-based method (with combination similarity measure) on the sample dump drawn from the WePS-1 training, and Figure 7 b) shows the results obtained for sample dump drawn from the WePS-2 training dataset. In Figure 7, *AV_Combination* shows the attribute-based method equipped with combination similarity measure. Given the results in Figure 7, we conclude that the majority of information for name disambiguation is given in the web pages being processed. This implies that our name disambiguation system does not heavily rely on the Babelfy disambiguation results. However, incorporating the Babelfy and combining it with other name disambiguation methods such as attribute-based method can improve the overall performance of system.

At the end of the graph creation process, we have an undirected weighted graph summarizing all of the information about entities contained in the given web text documents, and the given external knowledge base. The remainder components of the name disambiguation system work with this rich graph instead of web text documents. This enables us to use optimal graph mining algorithms for name disambiguation task. Figure 8 shows an example of mapping people's attributes and social links extracted from the web page given in Figure 2 into a graph. Figure 8 a) shows the discourse profiles extracted by profile extraction system. In Figure 8 a), the local attributes are shown in black colour and external attributes obtained by profile enrichment are shown in blue colour. The profile information and social links are mapped to a graph shown in Figure 8 b). In Figure 8 b), the structure node corresponding to target person *"Dan Jurafsky"* is shown in filled rectangle, the structure nodes for other persons are shown in rectangles, the structure nodes corresponding

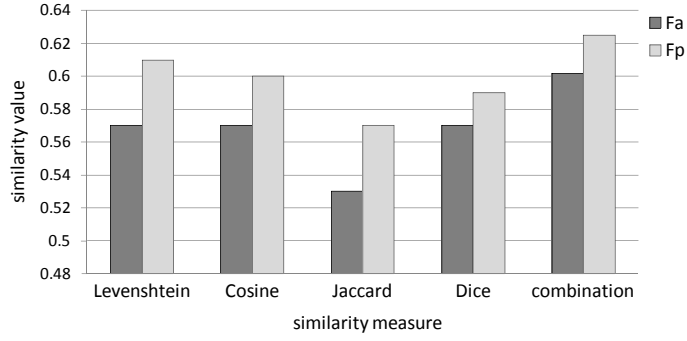


Figure 5. The performance of attribute-based name disambiguation method on WePS-1 training dataset using different similarity measures

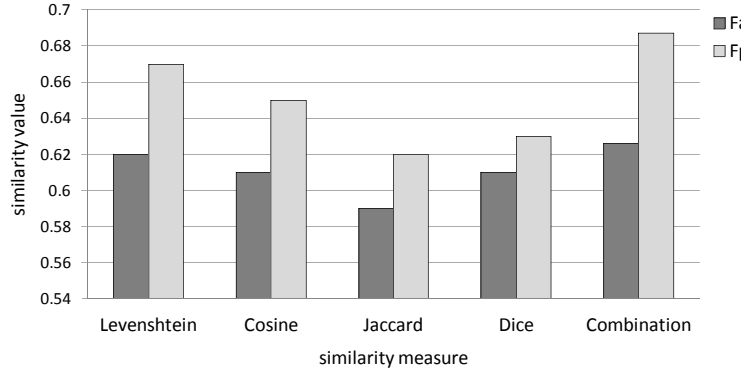
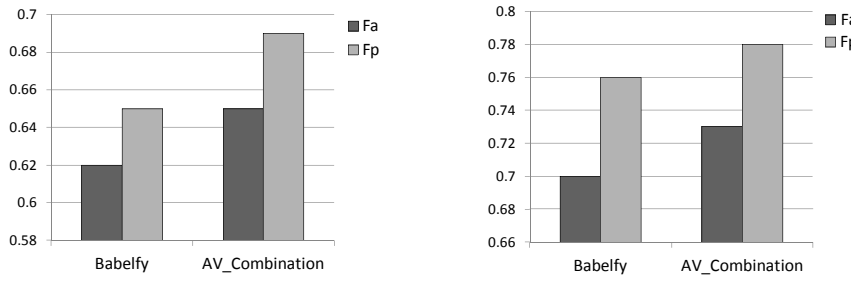


Figure 6. The performance of attribute-based name disambiguation method on WePS-2 training dataset using different similarity measures

to organizations in triangles, attribute nodes in round ellipses, structure edges by solid lines, and attribute edges by dotted lines.

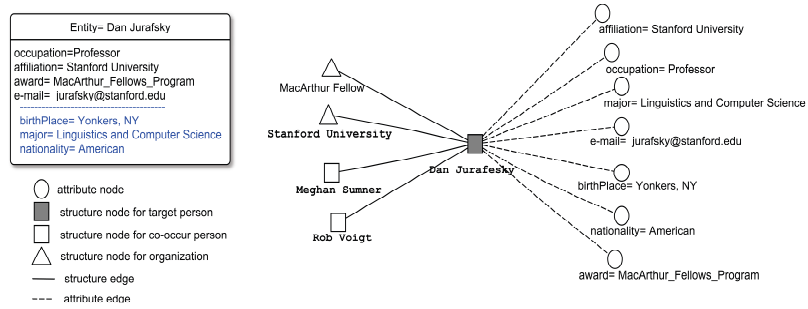
3.3 Similarity Computation

Our similarity metric relies on the node closeness in the graph through both structure and attribute edges. The main idea behind our similarity metric is that "two people are closely related if they share more common attributes and linked through many common entities". Our similarity metric obeys the theory of homophily [42], the principle that people with similar characteristics tend to form a relationship. People in a homophilic relationship share similar characteristics.



a) b)

Figure 7. The performance of Babelfy and AV_Combination method on a dump of Wikipedia person names drawn from a) WePS-1 training dataset and b) WePS-2 training dataset



a) b)

Figure 8. An example of mapping the sample web page given in Figure 2 to attribute-relationship graph; a) discourse profile extracted by our profiling system and enriched with external global attributes; b) attribute-relationship graph. Note that the weights of attribute edges and structure edges are not given here.

Since people's attributes and social links are two different types of information, integrating both of them in a unified similarity metric for name disambiguation is so challenging.

Here, we adapt and modify the recently proposed neighbourhood random walk distance (NRWD) [26] to built our similarity measure. Zhou et al. [26] applied NRWD for graph partitioning and demonstrated that it can improve the performance of clustering. Let τ be a path from node u to node v, whose length is $l(\tau)$ with transition probability $p(\tau)$, L be the longest length that the random walk can proceed, and $\gamma \in (0, 1)$ be the restart probability. The similarity between nodes u

and v using NRWD can be formulated as follows:

$$R(u,v) = \sum_{\substack{\tau : u \to v \\ l(\tau) \leq L}} \gamma(1-\gamma)^{l(\tau)} \times p(\tau) \tag{11}$$

The value of $R(u,v)$ is in the interval $[0,1]$. The transition probability $p(\tau)$ from structure node u to structure node v through structure edge es is computed as follows:

$$p_{u,v}(\tau) = \begin{cases} \frac{ws}{|N(u)|\times ws + |A(u)|\times wa}, & \text{if } (u,v) \in ES, \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

where $N(u)$ represents a set of neighbours of entity u that are connected through structure edges, $A(u)$ represents a set of neighbours of entity u that are connected through attribute edges, ws is the weight of structure edge es, and wa indicates the importance coefficient of attribute edge ea. Similarly, the transition probability from structure node u to attribute node $v_a \in AV$ through an attribute edge ea is calculated as follows:

$$p_{u,v_a}(\tau) = \begin{cases} \frac{wa}{|N(u)|\times ws + |A(u)|\times wa}, & \text{if } (u,v_a) \in EA, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The transition probability from attribute node v_a to structure node u through an attribute edge ea is computed as follows:

$$p_{v_a,u}(\tau) = \begin{cases} \frac{wa}{|N(v_a)|}, & \text{if } (v_a,u) \in EA, \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where $N(v_a)$ is the set of structure nodes that share the attribute node v_a. Since there is no edge between any pair of attribute nodes v_{ai} and v_{aj}, the transition probability between attribute nodes is 0:

$$p_{v_{ai},v_{aj}}(\tau) = 0. \tag{15}$$

3.4 Graph Clustering

Our procedure for clustering takes as input the attribute-relationship graph G, and uses the recently proposed BIC-Means [43], an efficient graph clustering algorithm to partition the graph into sets of clusters. BIC-Means is a bisecting version of the incremental K-means clustering algorithm equipped with Bayesian information criterion (BIC) [44] as a termination criterion. It starts with a single cluster C_1^0 containing all the objects, and bisects this cluster into two sub clusters C_1^1 and C_2^1 by applying the incremental K-means algorithm [43]. C_i^t refers to the i^{th} cluster at the t^{th} level. The algorithm continues by splitting each cluster C_i^t into two sub

clusters C_{i1}^{t+1} and C_{i2}^{t+1}, if the BIC score of sub clusters C_{i1}^{t+1} and C_{i2}^{t+1} is greater than the BIC score of C_i^t. It terminates the divisive procedure when there is no separable leaf cluster according to the BIC score. Our reason to use BIC-Means as clustering algorithm comes from the fact that

1. it is an efficient algorithm, which has low computational cost to hierarchically clustering large-scale data sets,
2. it does not need the number of clusters as input parameter, considering the notion that we do not know the number of clusters previously, and
3. it is robust against the parameter setting.

We run the BIC-Means clustering with the similarity metric given in Equation (8), which measures the closeness between nodes. In clustering process, we use a blocking technique to avoid computational bottlenecks. By this way, we apply clustering algorithm on documents that are about an ambiguous person.

As mentioned before, personal attributes are heterogeneous; therefore each class of attribute has different impact on clustering quality. This enforces us to assign different weights for each particular attribute class. In order to determine the importance coefficient for the attribute classes, we propose a dynamic weight learning approach. Our approach is an extension to the weight learning schema taken by [26]. Our approach for learning the weights of attributes is as follows. Let $Wa^t = \{wa_1^t, wa_2^t, \ldots, wa_m^t\}$ be the attribute weights in the t^{th} iteration of clustering. We initialize $wa_1^0 = wa_2^0 = \ldots = wa_m^0 = 1$ at first iteration of clustering. At each bisection stage of clustering process, we update the value of wa_i^t with a weight increment Δwa_i^t, which indicates the weight update of attribute a_i between the iteration t and $t+1$. The weight of attribute a_i in iteration $t+1$ is calculated as follows:

$$wa_i^{t+1} = \frac{1}{2}(wa_i^t + \Delta wa_i^t). \tag{16}$$

Δaw_i^t is computed by the following formula:

$$\Delta wa_i^t = \frac{m \times \sum_{j=1}^{k} \mathrm{H}\left(a_i, C_j\right)}{\sum_{f=1}^{m} \sum_{j=1}^{k} \mathrm{H}\left(a_f, C_j\right)} \tag{17}$$

where m is the number of attribute classes in attribute vocabulary A, and the scoring function $H(a_i, C_j)$ quantifies the saliency of attribute a_i in cluster C_j. The reasoning behind this weighting schema is as follows: an attribute a_i has great impact on clustering, if a large number of nodes within clusters share the same value on a_i, on the other hand, if nodes within clusters have quite different values on a_i, then it has not a good clustering tendency. To compute $H(a_i, C_j)$, we adopt and modify the term frequency (TF) weighting component of the TF-IDF method developed by [45]. We define $H(a_i, C_j)$ as follows:

$$H(a_i, C_j) = \alpha \times \Phi_1(a_i, C_j) + (1 - \alpha) \times \Phi_2(a_i, C_j) \tag{18}$$

where $\Phi_(a_i, C_j)$ is the relative intra-document TF, $\Phi_2(a_i, C_j)$ is the normalized length regularized TF, and α is a tunable importance coefficient. Without loss of generality, we set $\alpha = 0.5$. $\Phi_1(a_i, C_j)$ controls the distribution of attributes within a document. It computes the importance of attribute a_i by considering its frequency relative to the average frequency of attributes within cluster C_j. We define $\Phi_1(a_i, C_j)$ as follows:

$$\Phi_1(a_i, C_j) = \frac{\log_2(1 + \mathrm{TF}(a_i, C_j))}{\log_2(1 + \mathrm{MTF}(C_j))} \tag{19}$$

where $\mathrm{TF}(a_i, C_j)$ is the frequency of attribute a_i in cluster C_j, and $\mathrm{MTF}(C_j)$ denotes average attribute frequency within cluster C_j. $\Phi_2(a_i, C_j)$ normalizes the attribute frequency by considering the number of objects available in a cluster:

$$\Phi_2(a_i, C_j) = \mathrm{TF}(a_i, C_j) \times \log_2\left(1 + \frac{\mu_G}{|C_j|}\right) \tag{20}$$

where μ_G is the average length of clusters in cluster collection G. The general principle behind scoring function $\Phi_2(a_i, C_j)$ is that if two clusters have different lengths and the same TF values for a given attribute a_i, then the contribution of $\mathrm{TF}(a_i, C_j)$ should be higher for the shorter cluster. Thus, it is necessary to regulate the TF value in accordance with the length of clusters. We define $\mathrm{TF}(a_i, C_j)$ as follows:

$$\mathrm{TF}(a_i, C_j) = \sum_{u \in C_j} X_i(u, c_j) \tag{21}$$

where c_j is the centroid of cluster C_j, and $X_i(u, c_j)$ determines whether node u and v share a same value on attribute a_i.

$$X_i(u, c_j) = \begin{cases} 1, & \text{if } (a_i \in u \,\&\, a_i \in c_j), \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

4 EXPERIMENTS AND RESULTS

In this section, we first describe benchmark datasets and performance metrics, and then give the results obtained by our approach, baseline methods, and the competitive state-of-the-art algorithms.

4.1 Dataset

We used two standard benchmark datasets to validate and compare our approach with other algorithms developed for disambiguation of people names on the web. These datasets include WePS-1 test dataset[3] [13], and WePS-2 test dataset[4] [14]. Each dataset consisted of collections of web pages obtained from the results for

[3] Available for download at `http://nlp.uned.es/weps/weps-1/weps1-data`

[4] Available for download at `http://nlp.uned.es/weps/weps-2/weps2-data`

a person name query to an Internet search engine. For each name the top ranked N web pages (100 for WePS-1 dataset and 150 for WePS-2 dataset) from the search results were included into the dataset. For each person name the ground truth files are also provided by human annotators. These datasets provide a real corpus, which can test a disambiguation system for personal names with varying ambiguity and in different domains. Both WePS-1 test and WePS-2 test datasets consisted of 30 web page collections, each one corresponding to an ambiguous person name. These 30 person names were chosen from three different sources (10 name sets from Wikipedia, 10 from the US Census, and 10 from ACL conference) in order to provide different ambiguity scenarios.

4.2 Performance Measures

Various measures are presented to assess the quality of name disambiguation methods. We conducted evaluations using two types of scoring measures, *B-cubed* scoring measure and the *purity-based* scoring measure. We use three B-cubed scoring measures including B-cubed precision (B^3P), recall (B^3R), and F-score (B^3F_α). A more detailed discussion of these quality metrics is given in [46, 47]. In addition to B-cubed measures, we used three purity-based scoring measures, including purity (Pr), inverse purity (IPr), and harmonic mean of purity and inverse purity (F_p-score). A more detailed discussion of purity-based quality metrics is given in [46].

4.3 Numerical Results and Discussion

In addition to comparing our algorithm to prominent solutions and the state-of-the-art methods in the literature, we also implemented five solutions as our baseline methods. We compare our name disambiguation system with five baseline methods. The baseline algorithms include bag of words model (*BOW* model) [3], social network based method (*SN* model) [25], attribute-based method (*AV* model) [20], *ALL_IN_ONE* [14], and *ONE_IN_ONE* method [14]. The BOW baseline is based on the traditional bag of words models: agglomerative vector space clustering with TF/IDF weighting schema. The BOW method is widely employed as a benchmark in a series of previous works; e.g. in [3, 8, 24]. The SN baseline [25] represents the approach where only the social relationships are employed for name disambiguation. The AV baseline [20] is an attribute-based name disambiguation algorithm, which relies only on the people attributes, and it ignores social relations between people. The ALL_IN_ONE and ONE_IN_ONE baselines are provided by the WePS share-task [14]. In ALL_IN_ONE baseline all documents related to a person are placed in a single cluster. In contrast, in ONE_IN_ONE baseline each document is included in a separate cluster. We notice that we implemented the baseline methods as described in their original paper. We perform all experiments on a 3 GHz, and 4 GB RAM Personal Computer Intel® Pentium® 4. We coded all the mentioned algorithms using JAVA and MATLAB language.

We notice that there are two parameters needed to be configured. These parameters include the knowledge base matching threshold T and the random walk length L. To configure these parameters, we adopt a k-fold cross validation strategy (with $k = 4$). We at first randomly partition the training data into four equal folds. At each iteration, we used one of the folds as test data and the other three folds as training data. At each iteration, we empirically learn the value of parameters L and T on training data that provides the best performances for name disambiguation. The final results we report are averaged over four independent iterations. Setting T to 0.60, the results obtained for different values of L from 2 (2-hop neighbours) to 10 (10-hop neighbours) on training data are given in Table 2. We observe that the algorithm achieves the best result when the value of L is between 4 and 8. The results indicate that the algorithm is not very sensitive to L, thus the exact tuning of the parameter is not an important matter. We set $L = 6$ for the remainder of our experiments. Table 3 shows the results obtained on training datasets with different values of T from 0 (append all external attributes into local discourse profile) to 1 (ignore external attributes) with step 0.2. We observe that the algorithm achieves its best performance when the value of T is between 0.4 and 0.8. The results indicate that the algorithm is not very sensitive to T, thus the exact tuning of parameter T is not an important matter. We set $T = 0.60$ for the remainder of our experiments.

Dataset	L				
	2	4	6	8	10
WePS-1	69.8	**70.7**	70.5	69.5	69.3
WePS-2	74.2	76.5	**77.8**	77.4	76.1

Table 2. The results obtained by our approach for different values of L on training datasets in terms of $B^3F_{\alpha=0.5}$

Dataset	T					
	0	0.2	0.4	0.6	0.8	1
WePS-1	69.5	69.9	70.1	**70.4**	69.7	68.8
WePS-2	75.7	76.2	**76.8**	76.2	75.2	74.6

Table 3. The results obtained by our approach for different values of knowledge base matching threshold T on training datasets in terms of $B^3F_{\alpha=0.5}$

Table 4 shows the results obtained by our name disambiguation methods on WePS-1 test dataset. Table 5 shows the results for WePS-2 test dataset. In order to indicate the effect of each clustering feature, in Table 4 and 5, we begin with the feature of social links and then add features of local discourse profile attributes and external attributes one by one. The results clearly show the effect of profile enrichment and integrating attributes with social relationships. In Table 4 and 5, we notice that the performance is consistently increasing when incorporating more

clustering features. The final feature model (*social links + local attributes + external attributes*) achieves the best performances. In Table 4, the performances on WePS-1 test dataset increase about +1.9 % in terms of $B^3F_{\alpha=0.5}$ and +2.06 % in terms of $F_{p=0.5}$ from the local feature model *social links+local attributes* to final feature model *social links + local attributes + external attributes*. As given in Table 5, the improvement rate from the local feature model to final feature model is about +1.55 % in terms of $B^3F_{\alpha=0.5}$ and +2.33 % in terms of $F_{p=0.5}$ for WePS-2 test dataset. This implies that the majority of information for name disambiguation is given in the web pages being processed. However, incorporating the external attributes improves performances.

Method	B^3P	B^3R	$B^3F_{\alpha=0.5}$	Pr	IPr	$F_{p=0.5}$
Social links	67.3	73.7	68.1	71.5	87.1	80.4
+ Local attributes	74.4	80.5	75.2	78.6	89.3	82.2
+ external attributes	75.2	81.6	77.1	79.1	90.8	84.26

Table 4. Performances of our name disambiguation approaches on WePS-1 test datasets

Method	B^3P	B^3R	$B^3F_{\alpha=0.5}$	Pr	IPr	$F_{p=0.5}$
Social links	66.3	74.5	68.7	68.2	87.4	72.0
+ Local attributes	84.7	80.1	82.5	83.6	89.24	86.3
+ external attributes	86.2	82.9	84.05	85.4	90.8	88.63

Table 5. Performances of our name disambiguation approaches on WePS-2 test datasets

Table 6 shows the best performance obtained from the baselines and our method on WePS-1 test dataset. Table 7 shows the results for WePS-2 test dataset. As shown in Table 6 and 7, our method clearly outperforms the baseline methods for both datasets in terms of both $B^3F_{\alpha=0.5}$ and $F_{p=0.5}$. For WePS-1 dataset, on average our method outperforms BOW, SN, AV, ALL_IN_ONE, and ONE_IN_ONE by +7.8 %, 8.5 %, 16.4 %, 21.1 %, and 45.1 %, respectively, in terms of $B^3F_{\alpha=0.5}$, and +8.76 %, 5.76 %, 21.76 %, 12.86 % and 50.26 %, respectively, in terms of $F_{p=0.5}$. The improvement is also evident for WePS-2 dataset, in which our method obtains +11.55 %, 16.75 %, 22.65 %, 31.05 %, and 50.05 % improvement compared to BOW, SN, AV, ALL_IN_ONE, and ONE_IN_ONE, respectively, in terms of $B^3F_{\alpha=0.5}$, and with respect to $F_{p=0.5}$, the improvements are +11.83 %, 18.13 %, 21.33 %, 21.43 % and 54.63 %, respectively. The ONE_IN_ONE baseline obtained the best result in terms of B^3P and Pr measure on both WePS-1 and WePS-2 dataset. The ALL_IN_ONE baseline outperformed other algorithms in terms of B^3R and IPr measure. The higher B^3P and Pr for ONE_IN_ONE baseline arises from the fact that in WePS-1 and WePS-2 datasets documents are distributed among the clusters. Since in average half of the documents in the dataset belong to one specific person, the ALL_IN_ONE baseline gave better results in terms of IPr and B^3R.

Table 8 summarizes the average performance obtained by our proposed method and four state-of-the-art methods for the benchmark datasets. We compared the

Method	B^3P	B^3R	$B^3F_{\alpha=0.5}$	Pr	IPr	$F_{p=0.5}$
BOW	62.1	75.5	69.3	70.4	85.0	75.5
SN	65.0	73.5	68.6	69.7	89.2	78.5
AV	59.4	68.4	60.7	62.1	71.4	62.5
ALL_IN_ONE	44.0	**100**	56.0	59.0	**100**	71.4
ONE_IN_ONE	**100**	20.0	32.0	**100**	22.0	34.0
Our method	75.2	81.6	**77.1**	79.1	90.8	**84.26**

Table 6. Comparison of results obtained by baselines and our method on WePS-1 test dataset

Method	B^3P	B^3R	$B^3F_{\alpha=0.5}$	Pr	IPr	$F_{p=0.5}$
BOW	66.2	80.5	72.5	78.1	82.0	76.8
SN	64.2	81.1	67.3	64.2	90.7	70.5
AV	62.5	77.7	61.4	62.6	79.4	67.3
ALL_IN_ONE	43.0	**100**	53.0	56.0	**100**	67.2
ONE_IN_ONE	**100**	24.0	34.0	**100**	24.0	34.0
Our method	86.2	82.9	**84.05**	85.4	90.8	**88.63**

Table 7. Comparison of results obtained by baselines and our method on WePS-2 test dataset

results obtained by our method with those reported in Han and Zhao [3] and Chen et al. [4] on WePS-1 and WePS-2 test dataset; Dutta and Weikum [6] and Yerva et al. [11] on WePS-2 test dataset. We notice that the comparison is not precise, because the mentioned algorithms were implemented and tested with different settings on machines with different processing characteristics. In Table 8, Han and Zhao [3] did not report obtained B-cubed scores. As shown in Table 8, our approach performs well on datasets, exceeding or matching the best performance obtained by the state-of-the-art methods in terms of $B^3F_{\alpha=0.5}$.

		Han and Zhao [3]	Yerva et al. [11]	Chen et al. [4]	Dutta and Weikum [6]	Our method
WePS-1	$B^3F_{\alpha=0.5}$	–	–	76	–	77.1
	$F_{p=0.5}$	84	–	90	–	84.26
WePS-2	$B^3F_{\alpha=0.5}$	–	74.7	82	83.48	84.05
	$F_{p=0.5}$	86	78.8	89	–	88.63

Table 8. Comparison of results obtained by state-of-the-art methods and our method on WePS-1 test and WePS-2 test dataset

In this research, we concerned to answer the question "what is the maximum performance that a name disambiguation system can obtain if it uses information found in the web documents (local information) and attributes from external knowledge base (global information)?" The results indicate that integrating global information with local information improves the performance.

The results look promising but far from ideal. This justifies that name disambiguation in web is a demanding problem and more effort is needed in this respect. We analysed the failed cases where the algorithm could not resolve ambiguity. Our manual investigation over failed cases reveals that almost half of the failures were because of the inefficiency of pre-processing subtasks including named-entity tagging, intra-document co-reference resolution, and the erroneous attributes in discourse profiles extracted by profile extraction system, and not because of the inefficiency of our name disambiguation approach. For example, the employed co-reference resolution system identifies incorrect co-reference chains of mentions, which propagate to the later stages. Similarly, named-entity tagger could not correctly identify entities in web documents, which leads to a significant degradation in performance of the name disambiguation system. Pre-processing tools could not perform effectively on web documents because

1. most of the pre-processing tools have trained on news corpora,
2. web documents are quite diverse, noisy and contain partial information about entities.

These problems decrease the performance of name disambiguation system. Therefore, it is needed to develop pre-processing components suitable for web documents. Nonetheless, improving pre-processing tools is beyond the scope of this paper. We have attempted to improve the results by exploiting the context-independent features such as social links.

To summarize, our approach has several advantages. Using people profile attributes for name disambiguation degrades the undesired effect of noisy data and increases the efficiency of name disambiguation. It also decreases time complexity because instead of raw web text, we only compare structured attributes to compute similarities. Our approach could be considered as a language-neutral and domain-independent approach, because instead of raw text, it works on abstract information including entity attributes and social links. It could be extended to a more complex setting and applied to many applications, for example, social network extraction and information integration. In our implementations, we only consider the weighted co-occurrence of entities as social relationships. We simulate implicit social relationships and give meaningful semantics to meaningless co-occurrence relationships by exploiting personal attributes. In general, both insufficient entity-centric attributes and noisy social relations affect the robustness of name disambiguation. The results show that our method is capable to integrate both local and global attributes and social relationships, and provide more information for name disambiguation.

5 CONCLUSIONS

In this paper, we proposed a cross-document person name disambiguation approach in web context. Our approach attempts to use both the local information about persons available in the given web pages and the global information from external

knowledge bases. The local information includes profile attributes and social links, and the global information includes the attributes extracted from external knowledge bases. We evaluated the effectiveness of our proposed approach on WePS-1 and WePS-2 datasets. It achieved 77.1 % B-cubed F-score on the WePS-1 test dataset, and 84.05 % on WePS-2 test dataset. The results indicate that our approach is robust enough and outperformed the state-of-the-art name disambiguation approaches. Although our results seem satisfactory, some points to improve our research have remained. One of the most interesting directions is to consider other clustering features and media such as images, and integrating them with a text for name disambiguation, which can improve the results. As the final results show, our system depends on the effectiveness of three main subtasks including profile extraction, profile enrichment and social link extraction. Therefore another interesting future work is to develop more robust systems for these subtasks, which subsequently can improve the overall quality of the name disambiguation system. As the information about entities on the web changes over time, an interesting future work is to develop a dynamic name disambiguation system using dynamic clustering algorithms. Finally, we plan to develop a generic name disambiguation system in which entities are not limited to persons.

REFERENCES

[1] Barla, M.—Tvarožek, M.—Bieliková, M.: Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-Based Systems. Computing and Informatics, Vol. 28, 2009, No. 4, pp. 399–427.

[2] Barla M.: Towards Social-Based User Modeling and Personalization. Information Sciences and Technologies Bulletin of the ACM Slovakia, ACM Slovakia, Vol. 3, 2011, No. 1, pp. 52–60.

[3] Han X.—Zhao, J.: Structural Semantic Relatedness: A Knowledge-Based Method to Named Entity Disambiguation. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010, pp. 50–59.

[4] Chen, Y.—Mei Lee, S. Y.—Huang, C. R.: A Robust Web Personal Name Information Extraction System. Expert Systems with Applications, Vol. 39, 2012, No. 3, pp. 2690–2699, doi: 10.1016/j.eswa.2011.08.125.

[5] Khabsa, M.—Treeratpituk, P.—Giles, C. L.: Online Person Name Disambiguation with Constraints Categories and Subject Descriptors. Proceedings of the 15th ACM/IEEE-CS on Joint Conference on Digital Libraries (JCDL '15), ACM, 2015, pp. 37–46, doi: 10.1145/2756406.2756915.

[6] Dutta S.—Weikum, G.: Cross-Document Co-Reference Resolution Using Sample-Based Clustering with Knowledge Enrichment. Transactions of the Association for Computational Linguistics, Vol. 3, 2015, pp. 15–28, doi: 10.18653/v1/d15-1101.

[7] Bagga, A.—Baldwin, B.: Entity-Based Cross-Document Coreferencing Using the Vector Space Model. Proceedings of the 36th Annual Meeting of the Association

for Computational Linguistics and 17th International Conference on Computational Linguistics – Volume 1 (ACL '98/COLING '98), 1998, pp. 79–85.

[8] Kalashnikov, D. V.—Chen, Z.—Mehrotra, S.—Nuray-Turan, R.: Web People Search via Connection Analysis. IEEE Transactions on Knowledge and Data Engineering, Vol. 20, 2008, No. 11, pp. 1550–1565, doi: 10.1109/tkde.2008.78.

[9] Gong, J.—Oard, D. W.: Determine the Entity Number in Hierarchical Clustering for Web Personal Name Disambiguation. The 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, 2009, doi: 10.1145/1571941.1572124.

[10] Chen, Z.—Kalashnikov, D. V.—Mehrotra, S.: Exploiting Context Analysis for Combining Multiple Entity Resolution Systems. Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD '09), 2009, pp. 207–218, doi: 10.1145/1559845.1559869.

[11] Yerva, S. R.—Miklós, Z.—Aberer, K.: Quality-Aware Similarity Assessment for Entity Matching in Web Data. Information Systems, Vol. 37, 2012, No. 4, pp. 336–351, doi: 10.1016/j.is.2011.09.007.

[12] Brandes, U.—Erlebach, T. (Eds.): Network Analysis: Methodological Foundations. Springer-Verlag, Berlin, Heidelberg, Theoretical Computer Science and General Issues, Vol. 3418, 2005, doi: 10.1007/b106453.

[13] Artiles, J.—Gonzalo, J.—Sekine, S.: The SemEval-2007 WePS Evaluation: Establishing a Benchmark for the Web People Search Task. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval '07), Prague, Czech Republic, 2007, pp. 64–69, doi: 10.3115/1621474.1621486.

[14] Artiles, J.—Gonzalo, J.—Sekine, S.: Weps 2 Evaluation Campaign: Overview of the Web People Search Clustering Task. The 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, 2009.

[15] Brizan, D. G.—Tansel, A. U.: A Survey of Entity Resolution and Record Linkage Methodologies. Communications of the IIMA, Vol. 6, 2006, No. 3, pp. 41–50.

[16] Elmagarmid, A. K.—Ipeirotis, P. G.—Verykios, V. S.: Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and Data Engineering, Vol. 19, 2007, No. 1, pp. 1–16, doi: 10.1109/tkde.2007.250581.

[17] Köpcke, H.—Rahm, E.: Frameworks for Entity Matching: A Comparison. Data and Knowledge Engineering, Vol. 69, 2010, No. 2, pp. 197–210, doi: 10.1016/j.datak.2009.10.003.

[18] Ferreira, A. A.—Gonçalves, M. A.—Laender, A. H. F.: A Brief Survey of Automatic Methods for Author Name Disambiguation. ACM SIGMOD Record, Vol. 41, 2012, No. 2, pp. 15–26, doi: 10.1145/2350036.2350040.

[19] Mann, G. S.—Yarowsky, D.: Unsupervised Personal Name Disambiguation. Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 – Volume 4 (CONLL '03), 2003, pp. 33–40, doi: 10.3115/1119176.1119181.

[20] Nagy, I.: Person Attribute Extraction from the Textual Parts of Web Pages. Acta Cybernetica, Vol. 20, 2012, No. 3, pp. 419–440, doi: 10.14232/actacyb.20.3.2012.4.

[21] Srinivasan, H.—Chen, J.—Srihari, R.: Cross Document Person Name Disambiguation Using Entity Profiles. Proceedings of the Text Analysis Conference (TAC) Workshop, 2009.

[22] Lan, M.—Zhang, Y. Z.—Lu, Y.—Su, J.—Tan, C. L.: Which Who Are They? People Attribute Extraction and Disambiguation in Web Search Results. The 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, 2009.

[23] Song, F.—Cohen, R.—Lin, S.: Web People Search Based on Locality and Relative Similarity Measures. The 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, 2009.

[24] Bekkerman, R.—McCallum, A.: Disambiguating Web Appearances of People in a Social Network. Proceedings of the 14th International Conference on World Wide Web (WWW '05), 2005, pp. 463–470, doi: 10.1145/1060745.1060813.

[25] Malin, B.: Unsupervised Name Disambiguation via Social Network Similarity. SIAM SDM Workshop on Link Analysis, Counterterrorism and Security, 2005, pp. 93–102.

[26] Zhou, Y.—Cheng, H.—Yu, J. X.: Graph Clustering Based on Structural/Attribute Similarities. Proceedings of the VLDB Endowment, Vol. 2, 2009, No. 1, pp. 718–729, doi: 10.14778/1687627.1687709.

[27] Han, X.—Zhao, J.: Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge. Proceedings of the 18th ACM Conference on Information and knowledge management (CIKM '09), 2009, pp. 215–224, doi: 10.1145/1645953.1645983.

[28] Li, Y.—Wang, C.—Han, F.—Han, J.—Roth, D.—Yan, X.: Mining Evidences for Named Entity Disambiguation. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 1070–1078, doi: 10.1145/2487575.2487681.

[29] Finkel, J. R.—Grenager, T.—Manning, C.: Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005), 2005, pp. 363–370, doi: 10.3115/1219840.1219885.

[30] Lee, H.—Chang, A.—Peirsman, Y.—Chambers, N.—Surdeanu, M.—Jurafsky, D.: Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules. Computational Linguistics, Vol. 39, 2013, No. 4, pp. 885–916, doi: 10.1162/coli_a_00152.

[31] Manning, C. D.—Surdeanu, M.—Bauer, J.—Finkel, J.—Bethard, S. J.—McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60, doi: 10.3115/v1/p14-5010.

[32] Minkov, E.—Wang, R. C.—Cohen, W. W.: Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text. Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP '05), 2005, pp. 443–450, doi: 10.3115/1220575.1220631.

[33] Cortes, C.—Vapnik, V.: Support-Vector Networks. Machine Learning, Vol. 20, 1995, No. 3, pp. 273–297, doi: 10.1007/bf00994018.

[34] Moro, A.—Raganato, A.—Navigli, R.: Entity Linking Meets Word Sense Disambiguation: A Unified Approach. Transactions of the Association for Computational Linguistics, Vol. 2, 2014, pp. 231–244, doi: 10.1145/1459352.1459355.

[35] Mazhari, S.—Fakhrahmad, S. M.—Sadeghbeygi, H.: A User-Profile-Based Friendship Recommendation Solution in Social Networks. Journal of Information Science, Vol. 41, 2015, No. 3, pp. 284–295, doi: 10.1177/0165551515569651.

[36] Auer, S.—Bizer, C.—Kobilarov, G.—Lehmann, J.—Cyganiak, R.—Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Aberer, K. et al. (Eds.): The Semantic Web (ISWC 2007, ASWC 2007). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4825, 2007, pp. 722–735, doi: 10.1007/978-3-540-76298-0_52.

[37] Akcora, C. G.—Carminati, B.—Ferrari, E.: User Similarities on Social Networks. Social Network Analysis and Mining, Vol. 3, 2013, No. 3, pp. 475–495, doi: 10.1007/s13278-012-0090-8.

[38] Yujian, L.—Bo, L.: A Normalized Levenshtein Distance Metric. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, 2007, No. 6, pp. 1091–1095, doi: 10.1109/tpami.2007.1078.

[39] Liu, C.: Discriminant Analysis and Similarity Measure. Pattern Recognition, Vol. 47, 2014, No. 1, pp. 359–367, doi: 10.1016/j.patcog.2013.06.023.

[40] Koutrika, G.—Bercovitz, B.—Garcia-Molina, H.: FlexRecs: Expressing and Combining Flexible Recommendations. Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD '09), 2009, pp. 745–757, doi: 10.1145/1559845.1559923.

[41] Chang, A. X.—Manning, C. D.: SUTime: A Library for Recognizing and Normalizing Time Expressions. Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC '12), 2012, pp. 3735–3740.

[42] Easley, D.—Kleinberg, J.: Networks, Crowds, and Markets: Reasoning about a Highly Connected World. Cambridge University Press, 2010, doi: 10.1017/cbo9780511761942.

[43] Hourdakis, N.—Argyriou, M.—Petrakis, E. G. M.—Milios, E. E.: Hierarchical Clustering in Medical Document Collections: The BIC-Means Method. Journal of Digital Information Management, Vol. 8, 2010, No. 2, pp. 71–77.

[44] Schwarz, G. E.: Estimating the Dimension of a Model. The Annals of Statistics, Vol. 6, 1978, No. 2, pp. 461–464, doi: 10.1214/aos/1176344136.

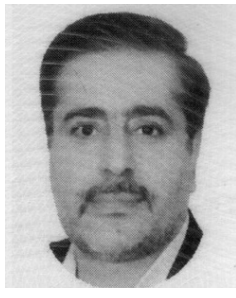
[45] Paik, J. H.: A Novel TF-IDF Weighting Scheme for Effective Ranking. Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '13), 2013, pp. 343–352, doi: 10.1145/2484028.2484070.

[46] Amigó, E.—Gonzalo, J.—Artiles, J.—Verdejo, F.: A Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints. Information Retrieval, Vol. 12, 2009, No. 4, pp. 461–486.

[47] Cai, J.—Strube, M.: Evaluation Metrics For End-to-End Coreference Resolution Systems. Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2010), 2010, pp. 28–36.

[48] Christen, P.: Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer, Heidelberg, New York, Dordrecht, London, 2012.

Hojjat Emami is Assistant Professor at the Computer Engineering Department, University of Bonab. He received his B.Sc. degree in software engineering and M.Sc. degree in artificial intelligence from University of Tabriz. He received his Ph.D. degree in artificial intelligence under the supervision of Prof. H. Shirazi and Prof. A. A. Barforoush. His interest research areas are: data mining, machine learning, evolutionary computation, multi-agent systems, and social network analysis.



Hossein Shirazi received his B.Sc. from Mashhad University, Iran. He received his M.Sc. and Ph.D. in artificial intelligence from the University of New South Wales, Australia. He is currently Associate Professor at the Malek-Ashtar University of Technology, Iran.

Ahmad Abdollahzadeh Barforoush is Professor in Computer Engineering and IT Department of Amir Kabir University of Technology, Iran. He is the author of books entitled "Introduction to Distributed Artificial Intelligence" and "Software Quality Assurance Methodology". His research areas are: data quality, artificial intelligence, agent-based systems, automated negotiation, expert systems, natural language processing, decision support systems, business intelligence, data mining, data warehouse, and software engineering.

Computing and Informatics, Vol. 37, 2018, 1516–1535, doi: 10.4149/cai_2018_6_1516

APPLICATION OF ADVANCED INFORMATION AND COMMUNICATION TECHNOLOGIES IN A LOCAL FLOOD WARNING SYSTEM

Ivan Mrnčo, Peter Blštak, Peter Hudec, Matej Kochan

ITKON, spol. s r. o.
Dohnányho 2
917 02 Trnava, Slovakia
e-mail: {ivan.mrnco, peter.blstak, peter.hudec, matej.kochan}@itkon.sk

Tomáš Gibala

DHI SLOVAKIA, s.r.o.
Hattalova 12
831 03 Bratislava, Slovakia
e-mail: tg@dhigroup.com

Ondrej Habala

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: ondrej.habala@savba.sk

Abstract. This paper deals with the practical application of a local flood warning system. The system is built on the mathematical model of a selected area. The rainfall-runoff processes are simulated in real-time. The warning system is designed as an on-line, real-time data inputs-processing system so that it can provide a timely warning. The warning system is based on a mathematical model and it uses modern information and communication technology tools. For the system to work properly, it is absolutely necessary to adhere to a real mathematical model, and therefore

a calibration on real historical data and direct measurements is required. This article describes the tasks of data collection, of building the mathematical model of the rainfall-runoff process, and the monitoring system design. The composed algorithm is able, based on the measured input data and the modeled situation, send a notification message to the monitoring centre and warn respective civil protection authorities via SMS messages.

Keywords: Flood warning system, mathematical modeling, hydro-informatics, data mining, rainfall-runoff processes, MIKE 21, MIKE URBAN

Mathematics Subject Classification 2010: 65-C20

1 INTRODUCTION

Nowadays Information and Communication Technologies (ICT) permeate most domains of science, research and technology, from the hydraulic domain [17] (which is the target of this article), to meteorology [13], hydrology [16] and to the related field like crisis management [15], to such remote domains as psychology [14]. These technologies offer different approaches to computations, some based on the knowledge of physical principles resulting in traditional sets of differential equations [11], others based on statistical approaches [20]. All are enabled by advances in computer hardware power and affordability [12] and novel methods of its use and sharing [10, 21].

Modern hydraulic modeling extensively uses ICT tools for data acquisition, preparation, processing, presentation [18] and visualization [6]. The presented local flood warning system is built on a mathematical model of the target area, where it in real time simulates rainfall-runoff physical processes. Since rainfall-runoff modeling is a complex scientific field, using several sophisticated modeling tools, it was necessary to analyze, in cooperation with our academic partners [19], these tools not only from the point of view of their scientific merits, but also from the point of view of their usability in a warning system. Not-measured catchment areas are the basic surface building blocks of the warning system. Their extent does not allow to solve this task in a direct assignment to public authorities – SHMI[1] or SWME[2] – as they usually represent one small catchment area containing one small stream. The warning system has been designed as working on-line, that is on real-time data inputs, so that the warning can be issued immediately or as soon as possible. The warning system's core is a computer model. For it to work correctly, it is necessary that the model is as realistic as possible, so it must be calibrated on historical sets of

[1] Slovak Hydrometeorological Institute, `http://www.shmu.sk/en/?page=1`

[2] Slovak Water Management Enterprise, `https://www.svp.sk/en/uvodna-stranka-en/`

hydrological parameters coupled with direct measurements in the target catchment areas.

As the target an area south of Trenčín and north of Trenčianska Turná has been selected, shown in Figure 1. The area contains the shopping mall Laugaricio Trenčín and is in the catchment area of the *Lavičkový potok* stream.

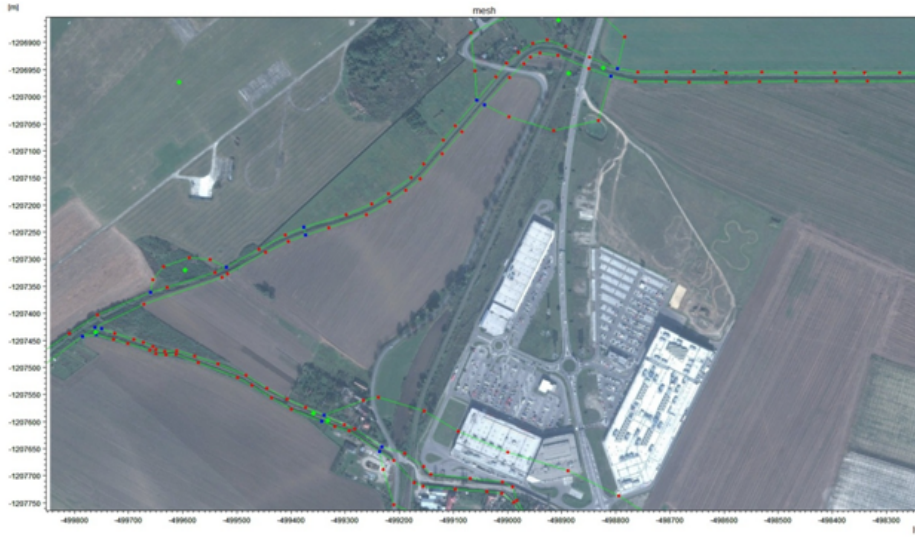


Figure 1. The target area of our model is marked as a green polygon

The area is influenced by the sewer system of Trenčín since the stream is the recipient of a relieve drain of the sewer system (Figure 2), which significantly increases its flow during intense rainfall [1].

2 FLOOD MODELING IN EUROPEAN UNION

To put the work described in this article in the context of other work regarding flood management in the European Union, we will present several previous projects done by DHI or in cooperation with DHI. They were effected mainly as part of the effort envisaged by the so-called EU Flood Directive [23]. This directive, among other things, requires also flood-mapping which means creation of flood risk maps for several flow volume values for a mapped river or stream. These flood maps are key inputs to planning flood protection, flood risk management, urban planning as well as crisis management. In Slovakia, several flood mapping initiatives have already started, with various results depending on the requirements of their respective customers. We will summarise them below.

Figure 2. The extent of the model of sewer and drainage system of the city Trenčín

2.1 Banská Bystrica Flood Maps

- Modeled and mapped flow: Q5, Q10, Q20, Q50, Q100, Q1000
- Hydraulic modeling: 2D model with flexible computational mesh (MIKE 21 FM)
- Modeled detail: buildings and streets
- Outputs of hydrodynamical modeling: flood extents, water depth, flow velocity
- Map scale: 1:5 000
- Flood damage assessment: no

2.2 Levice Flood Maps

- Modeled and mapped flow: Q5, Q10, Q20, Q50, Q100, Q1000
- Hydraulic modeling: 2D model with flexible computational mesh (MIKE 21 FM)
- Modeled detail: buildings and streets

- Outputs of hydrodynamical modeling: flood extents, water depth, flow velocity
- Map scale: 1:7 500
- Flood damage assessment: no

2.3 Košice and Prešov Flood Maps

- Modeled and mapped flow: Q1, Q5, Q10, Q20, Q50, Q100
- Hydraulic modeling: 2D model with flexible computational mesh (MIKE 21 FM)
- Modeled detail: buildings and streets
- Outputs of hydrodynamical modeling: flood extents, water depth, flow velocity for grid 5×5 m
- Map scale: 1:10 000
- Flood damage assessment: no

2.4 Morava and Vlára Flood Maps

- Created in the international project CEFRAME [22]
- Modeled and mapped flow: Q50, Q100, Q500
- Hydraulic modeling: 2D model with flexible computational mesh (MIKE 21 FM)
- Modeled detail: less detailed, however important topological features are present: channels, dikes, road and railroad embankments
- Outputs of hydrodynamical modeling: flood extents, water depth, flow velocity for grid 5×5 m
- Map scale: Morava 1:50 000, Vlára 1:25 000
- Flood damage assessment: yes

3 DATA COLLECTION

A modern ICT system for transfer of measured data in water management uses a telemetry system, with a passive collection point receiving data from active measurement stations. The system of active measurement stations sends the measured data to a central server at certain intervals. Between those intervals the measurement station does not need to be connected to the telemetry system and may be in a standby mode, saving its stored power and extending its availability. The passive collection point is the database server. The data collected in the database from the measurement stations is subsequently processed and used in the modeling of the target area. Advances in measurement technology allow us to follow selected target attributes online and in offline mode, to store them and use them in the operation, maintenance, creation, reporting and also training of the mathematical prediction model. The data are processed by hydro-informatics systems.

The collection of data for our local flood warning system has been done on two automated rain gauges (telemetric probe HYDRO LOGGER H1 and a SR03 rain gauge), on one flow meter (M4016-G3 unit and a KDO probe) and on one water stage indicator (M4016 unit and the ultrasound probe US3200). Testing the data transfer has been executed using these modes:

- metallic symmetric mode,
- optical symmetric mode,
- wireless symmetric mode in a licenced frequency band 3.5 GHz,
- wireless symmetric mode in a free frequency band 5.8 GHz,
- GSM/GPRS/EDGE mode.

Testing proved GSM/GPRS/EDGE devices as the most successful because of their excellent signal coverage, high mobility of the measurement devices, independence on external power or on additional software and hardware.

	Metallic	**Optical**	**3.5 GHz**	**5.8 GHz**	**GSM**
External power	Yes	Yes	Yes	Yes	No
Additional sw	Yes	Yes	Yes	Yes	No
Additional hw	Yes	Yes	Yes	Yes	No
Connectivity	Needs conn. line		Needs direct visibility		Yes
Mobility	No	No	No	No	Yes
Mass deployment	Unsuitable				**Suitable**

Table 1. Comparison of different connection types of sensors

4 DEPLOYED INFORMATION INFRASTRUCTURE

A cloud composed of capable servers has been used for the data connection and evaluation tasks. This infrastructure hosts also the server which receives data from the deployed measurement stations. The data are first preprocessed and then stored in a database for later processing and archiving.

Based on the demands of the local flood warning system following servers were instantiated in the cloud:

- control and monitoring interface – used to monitor the cloud infrastructure, deployed measurement stations and component services of the warning system,
- data server for data collection and preprocessing,
- web server for measured data presentation in the form of graphs and tables,
- user interface server including an artificial intelligence component evaluating the probability of flood based on the measured data and underlying mathematical model of the target area.

A basic schema of the infrastructure used in the presented flood prediction system is shown in Figure 3.

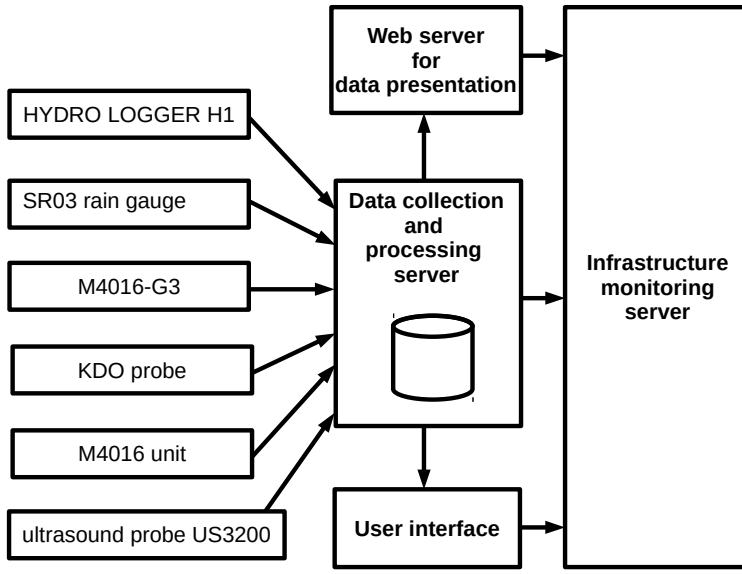


Figure 3. Schema of the deployed information infrastructure used to capture data, store them, present them and use them for flood modeling and prediction

5 MATHEMATICAL MODEL

To facilitate the creation of a mathematical model of the target area it was necessary to acquire and verify the set of input data. It contains a relatively large amount of information and their processing and verification is especially time-consuming [7].

The hydraulic mathematical model had to be then calibrated using the data from the measurement campaign, which had to be executed in advance. Missing data points, not acquired during the campaign (for example certain precipitation amounts) had to be added to the set by data mining.

Data mining is the process of non-trivial extraction of implicit, not known in advance, and potentially useful information from large data sets. It is therefore a process of discovery of hidden relations among the data and of patterns which can be then used for modeling and prediction.

There are also data mining applications in hydrology and meteorology, for example experiments realised during the FP7 project ADMIRE [2, 3], which for us simplify the data mining process.

6 CREATION OF THE HYDROLOGICAL SIMULATION MODEL

The hydrological model to simulate the rainfall-runoff process in the target watershed area has been created using the MIKE URBAN (MOUSE) level A module [9]. We have posited that there is 30 % of balast water in the sewage from Trenčín. The sewage flow has been computed using data on water consumption in the city. The balast water has been added to the computation for a total of 160 litres per person a day, including the added balast water.

The hydrodynamic model has been created using the simulation tool MOUSE, used to compute slow-changing, continuous flow in sewer systems and drainage canals.

As the basis of the creation of the sewer system model, the following data were used:

- sewer topology,
- geometrical data of the sewer pipes and other objects,
- pipe connections information,
- other data.

While creating the model a partial internal schematization was made, meaning the model of the sewer system contains all drains, objects and sewer segments, but smaller-dimension pipes do not include all drains.

For the composition of a two-dimensional hydrodynamical model and of the simulation itself the tool MIKE 21 FM has been used. MIKE 21 FM is a 2D mathematical model of continuous flow with a flexible computation mesh created by DHI. It is a complex simulation environment for 2D modeling of flow with a free surface. It is based on the solution of two-dimensional governing RANS (Reynolds averaged Navier-Stokes) equations integrated along the depth dimension [8]. The numerical solution is based in a discretization by division into non-overlapping components (triangles or quadrangles). The dependent variables of the system are represented as constant for the whole component and are tied to its center.

During the actualization and refinement of the model a computation mesh composed of triangles and quadrangles has been created, covering the complete target area. The mesh has been designed so as to sufficiently approximate the terrain details of the area. Mesh density is variable and chosen to allow for both sufficient model accuracy and numerical stability of the simulation. The mesh is composed of 211017 vertices and 309465 components. Part of it is shown for illustration in Figure 4.

Each vertex of the mesh in the 2D model has been described by the corresponding terrain height from a DTM model of the area. Based on geodetic measurements the model topography has been updated. This has lead to a model topography which we call *bathymetry* as it represents the bed of the potentially flooded area. It is partially shown in Figure 5.

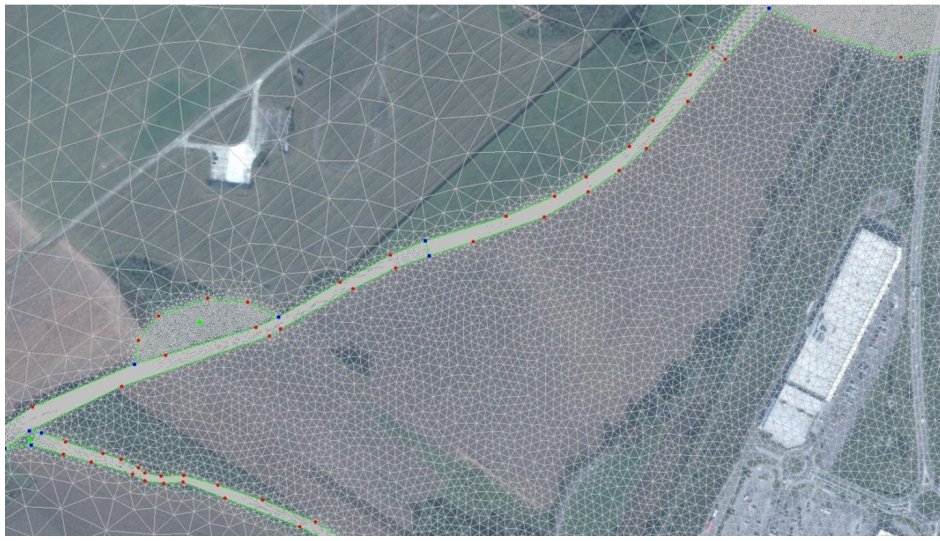


Figure 4. Partial computation mesh of the created two-dimensional mathematical model

Each component of the computation mesh of the model has a defined value of friction resistance in the form of Manning's roughness coefficient based on the type of terrain. The values have been selected based on the study of the ortophotomap, fotodocumentation and on an on-site inspection.

7 CALIBRATION OF THE HYDRODYNAMIC MODEL

The composition and following calibration of simulation models are highly specialized tasks with the goal to prepare a simulation environment for the following experiments with a variety of states of the model. Most of the tools for solving the surface flow tasks are based on a more or less exact description of the physical-chemical-biological processes in the rainfall-runoff action. This description is also determined by selection of the initial and boundary conditions of the simulation – the selection of parameters describing the initial state of the system (description of the watershed area, abundance of impermeable surfaces, infiltration capacity of the permeable surfaces and similar physical parameters). Some of these parameters are very difficult to measure. To obtain the correct values of these various coeficients and parameters of the governing equations or empirical equations a *model calibration* is used.

To calibrate a model, it is necessary to obtain a set of input and output parameters for known initial conditions (for example measured rainfall and the matching flow in the sewer system). Calibration assumes not only detailed knowledge of the mathematical model and its governing equations, but also the correct selection of a monitoring strategy, which will allow us to acquire sufficient empirical knowledge

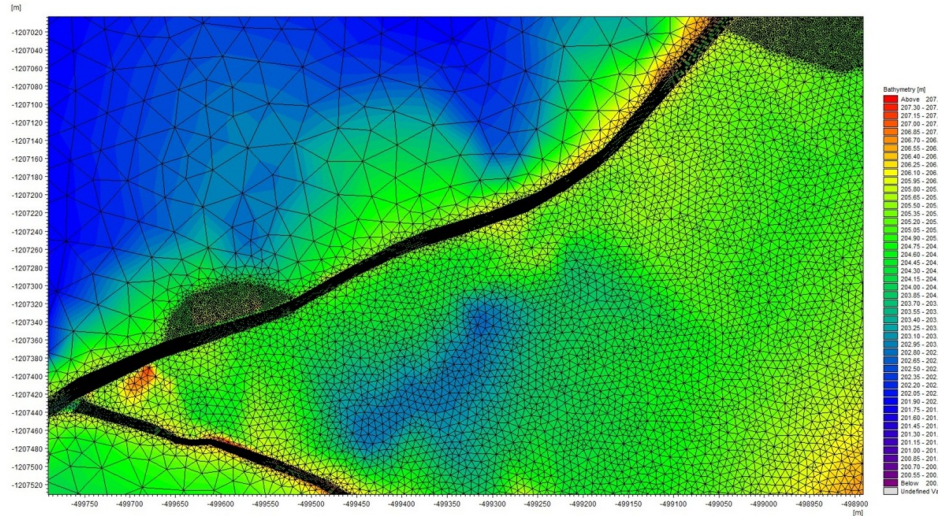


Figure 5. Partial bathymetry of the target area

of the parameters of the system in various meteorological conditions. It is the comparison of real, measured input parameters (causes) of the modeled system in various representative conditions and real, measured output parameters of the system (effects), which allow us to evaluate and tune the initial and boundary conditions of the model and the coefficients of the governing equations in situations when these cannot be reliably measured. Verification of the model is comparable to the process of calibration. The goal of verification is to evaluate the quality of the model on an independent series of input and output parameters. The process of calibration and verification of the simulation model is schematically shown in Figure 6 and in more detail described in [5].

It is obvious that the calibration process, additionally to the mentioned inputs, requires also extensive effort and resources from the user of the warning system. Selection of optimal parameters, which will provide the model with sufficient inputs for its calibration, while at the time will save as much time and resources as possible is one of the most important steps while solving the problem of urban drainage using modern simulation tools.

Despite being time-consuming and often costly, the process of calibration is an unavoidable part of the application of simulation models to most physical phenomena.

In order to simplify the calibration and subsequent verification of the mathematical model, a sensitivity analysis of the model may be an advantage. A sensitivity analysis of the input parameters (the influence of fraction of impermeable soil in the modeled area on the maximum flow, for example) can significantly speed up the calibration and verification process, even without considering simplification of the

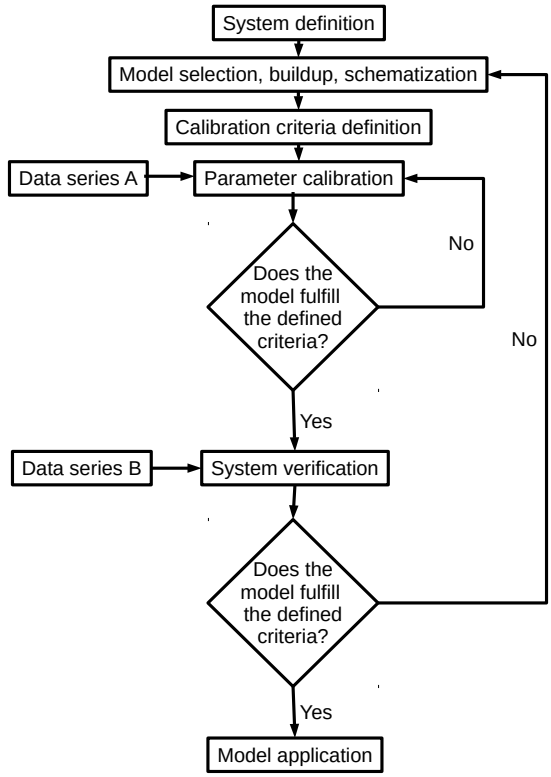


Figure 6. Schematic representation of the steps taken during the calibration and verification of the model

decision making on the monitoring strategy for the purpose of the calibration and verification of the model.

The schematization of the model is the initial step of the whole process, it defines the simplification of the whole system so that it is solvable, while not omitting any important element and allowing to achieve the stated goals for which the system is being created. After schematization, two tasks are being executed in parallel – physical measurements of the watershed and the composition and calibration of the simulation tools, based on the chosen technology.

Regarding schematization it is important to mention that some watershed areas have their parameters altered because of industrial drainage systems connecting into them. These are not included in the model, but are considered as point inflows (the Naza and Uni-Mier compounds in Trenčín, for example), or the drainage coeficient of the watershed may be increased.

8 MONITORING OF INFRASTRUCTURE AND SERVICES

Monitoring of the flood warning system can be divided into several groups:

- monitoring the availability of the measurement probes,
- monitoring the availability of network connectivity,
- monitoring the servers,
- monitoring the outputs being stored in the database,
- notification production and delivery.

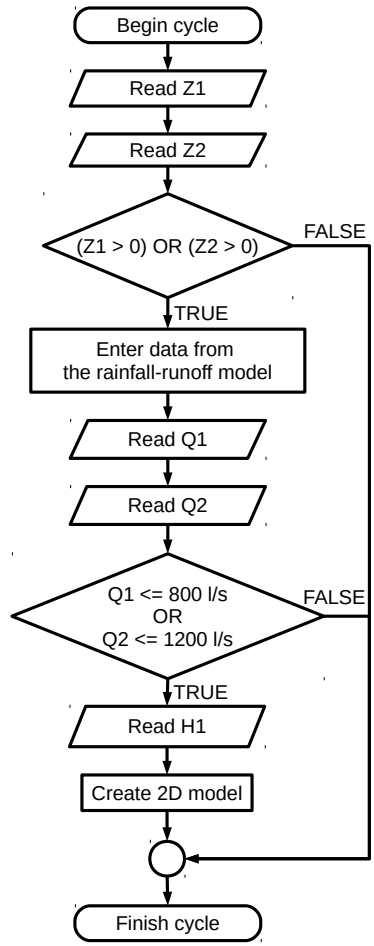


Figure 7. Algoritm for assessment of the flood possibility

For the purpose of the monitoring and control of services and hardware during the experimental work on the local flood warning system it was necessary to correctly configure the Nagios monitoring system [4]. Nagios' working principle is sufficiently uncomplicated. It performs checks of the services and hardware of the system using defined commands (plug-ins). If a check ends in a non-standard state, a predefined person will be notified about this problem.

Data which have been collected during the experimental setup were acquired in an actual stream, the *Lavičkový potok* in Trenčín. Data from the measurement probes were sent in selected time intervals to two data servers, called *sql1* and *sql2*, processed and stored in a database. The measured data on the state of the physical parameters of the stream were later used for monitoring measurement suites and processed using a neural net, which serves as the mathematical and hydraulic basis for the boundary conditions of the 2D model used to determine the flood extent, based on the rainfall-runoff model. For this purpose the data model shown in Figure 8 has been devised.

After receiving and processing data from the precipitation measurement suites (*TN_Z1* and *TN_Z2*), the flood possibility is determined using an algorithm based in the 2D rainfall-runoff model, shown in Figure 7.

The algorithm for sending notifications (Figure 9) to responsible persons is executed every time after a map of flooding is created and the extent of the flooded area is ascertained. This algorithm ensures that the respective persons are notified either by e-mail or an SMS message.

Display and presentation of outputs and results from the local flood warning system must be as simple as possible, so that the presentation is not time consuming or overly complicated. To fulfill this condition, a single-purpose web portal has been created (see Figure 10), which allows quick lookup of information about flooding by defining just three parameters. The parameters are:

- Total precipitation
  - less than 25 mm
  - from 25 to 50 mm
  - from 50 to 100 mm
  - more than 100 mm
- Rainfall duration
  - less than 2 hours
  - from 2 to 6 hours
  - more than 6 hours
- Watershed saturation
  - dry watershed
  - partially wet watershed
  - fully saturated watershed

After selecting the values of these three parameters, a flood extent map is shown, available in two different modes of quality (middle and high).

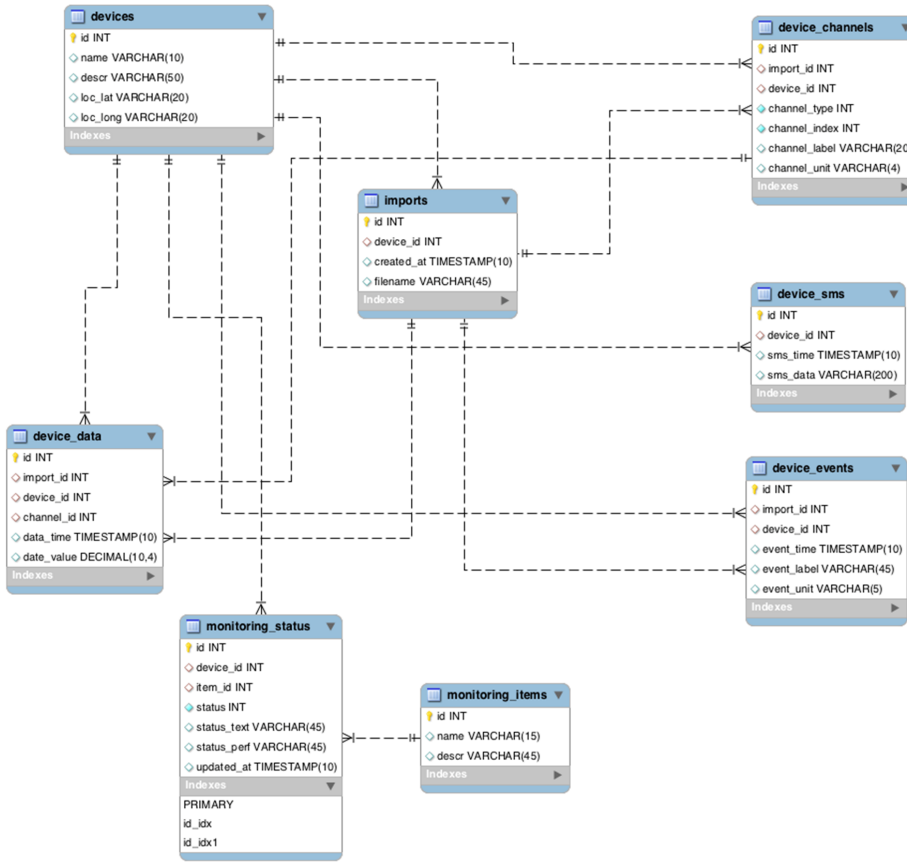


Figure 8. Data model of the database storing data of the experimental flood warning system

9 CONCLUSIONS

Results of our research show, that capabilities of current simulation models from the point of view of the underlying equations are sufficient to allow us to steer the effort to increase the effectivity and shorten the duration of the computation cycle – by way of parallelisation of the software implementation of the mathematical model, or by using modern high-performance ICT technologies in a combination with modeling tools.

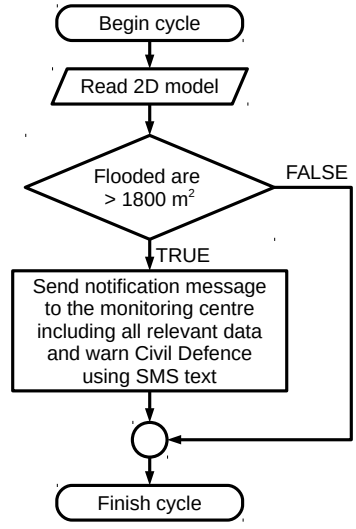


Figure 9. Notification decision algorithm of the local flood warning system

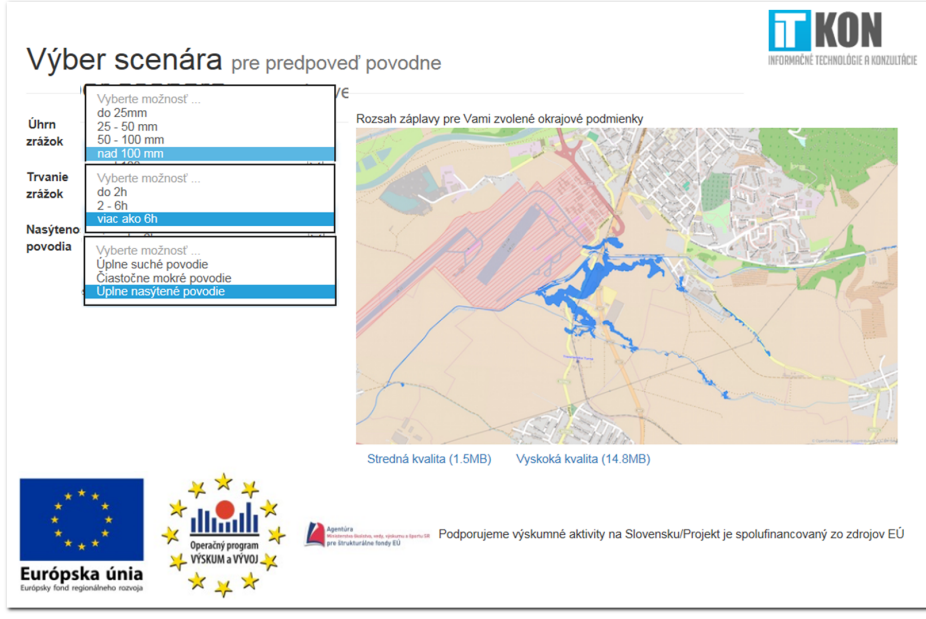


Figure 10. Portal for presentation of the flood extent, with the selection of parameters

The main result of our research is well tested and explored methodology to combine the state-of-the-art ICT technologies and mathematical models for hydraulics and create a viable local flood warning system which can be used not only in the target area selected for our experiment, but as well in any other locality with available terrain data.

Presenting the user with detailed results of the mathematical modeling is counterproductive, as that does not give a clear view of possible threats he/she is facing and thus unable to react quickly. It would diminish the effort put into creating a fast-reacting, on-line flood warning system. Research of viable user interfaces and their simplification show that it is important to present the results of the flood warning system as simply as possible, with the aim to streamline our decision making, and classify the results into several categories (for example various threat levels the modeled flood is presenting). Then the user can use the already prepared directly applicable response scenarios.

Results of the simulation of the rainfall-runoff process and its transformation in the target area for extreme rainfalls show, that the flood will be present in parts of the inundation area. We have prepared visualizations created by the 2D hydraulic model showing water depth map (Figure 11) as well as the inundation area without and with the flooded area (Figure 12). The results of the drainage system simulation show that it is able to perform its function even in the case of extreme rainfall transfering the runoff into the *Lavičkový potok* stream causing a downstream flooding.

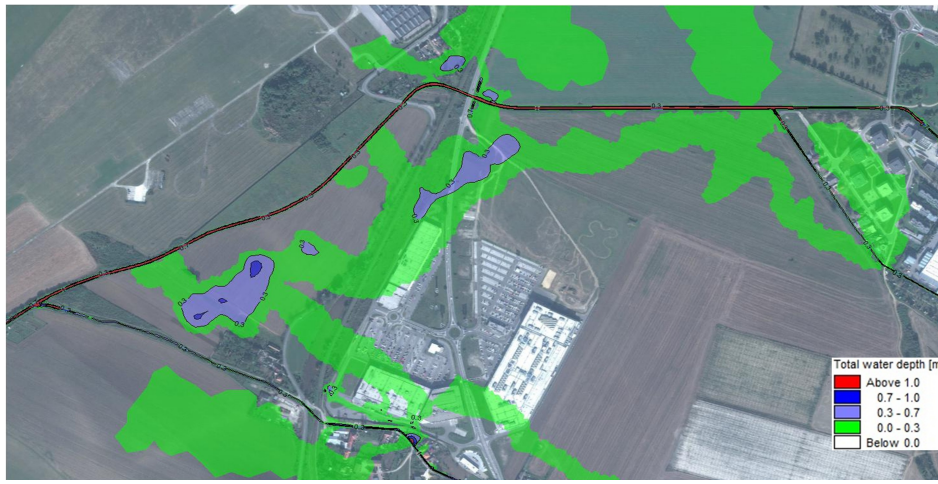


Figure 11. Presentation of the flood extent, with the selection of parameters

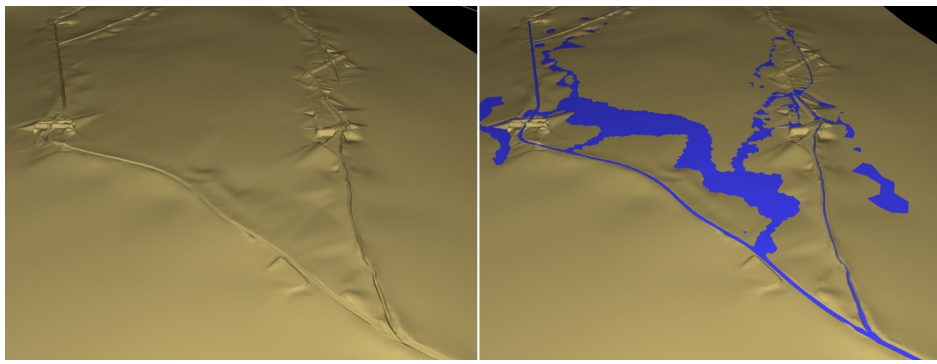


Figure 12. Presentation of the flood extent

Acknowledgements

This article is the result of the implementation of the project: Competency Centre for Knowledge Technologies Applied in Innovation of Production Systems in Industry and Services, ITMS: 26220220155, supported by the Research and Development Operational Programme funded by the ERDF.

REFERENCES

[1] Mrnčo, I.—Poláček, M.—Blšták, P.—Hudec, P.—Gibala, T.—Brtko, P.: Wastewater Drainage as a Source of Flood Threats. In: Bodík, I. et al. (Eds.): Odpadové vody 2012, Conference Proceedings, NOI Bratislava, 2012 (in Slovak). ISBN: 978-80-970896-2-7.

[2] Hluchý, L.—Habala, O.—Tran, V. D.—Krammer, P.—Šimo, B.: Using ADMIRE Framework and Language for Data Mining and Integration in Environmental Application Scenarios. 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2011), 2011, pp. 2437–2441, doi: 10.1109/FSKD.2011.6020018.

[3] Hluchý, L.—Habala, O.—Tran, D. V.—Šimo, B.: Environmental Risk Management. In: Atkinson, M. et al. (Eds): The Data Bonanza: Improving Knowledge Discovery in Science, Engineering, and Business. Chapter 15. New Jersey, USA, John Wiley and Sons, Inc., 2013, pp. 301–326. ISBN: 978-1-118-39864-7, doi: 10.1002/9781118540343.ch15.

[4] Josephsen, D.: Building a Monitoring Infrastructure with Nagios. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2007. ISBN: 0132236931.

[5] Mrnčo, I.—Blšták, P.—Poláček, M.: Structure of Data in Hydroinformatics Systems and Their Interconnection. In: Dubová, V. et al. (Eds.): Inovatívne informačno-komunikačné technológie vo vodnom hospodárstve, Conference Proceedings, Bratislava, 2015 (in Slovak). ISBN: 978-80-89535-14-9.

[6] Abbott, M. B.: Hydroinformatics: Information Technology and the Aquatic Environment. The University of California, Avebury Technical, 1991. ISBN: 978-1856288323.

[7] Johnson, L. E.: Geographic Information Systems in Water Resources Engineering. CRC Press, 2008. ISBN: 978-1420069136, doi: 10.1201/9781420069143.

[8] MIKE 21 Flow Model FM, Hydrodynamic Module. User Manual. DHI, August 2014.

[9] MIKE URBAN, Collection System. Modelling of Storm Water Drainage Networks and Sewer Collection System. User Manual. DHI, August 2014.

[10] Bobák, M.—Hluchý, L.—Tran, D. V.: Application Performance Optimization in Multicloud Environment. Computing and Informatics, Vol. 35, 2016, No. 6, pp. 1359–1385. ISSN: 1335-9150.

[11] Hluchý, L.—Froehlich, D.—Tran, D. V.—Astaloš, J.—Dobrucký, M.—Nguyen, G.: Parallel Numerical Solution for Flood Modeling Systems. In: Wyrzykowski, R., Dongarra, J., Paprzycki, M., Waśniewski, J. (Eds.): Parallel Processing and Applied Mathematics (PPAM 2001). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2328, 2001, pp. 485–492, doi: 10.1007/3-540-48086-2_53.

[12] Dobrucký, M.—Astaloš, J.: ÚI SAV: SIVVP Klaster. HPC Focus: časopis Výpočtového strediska Slovenskej akadémie vied, Vol. 2, 2015, pp. 31–37 (in Slovak). Available online: `http://vs.sav.sk/magazine/issues/magazine_201501.pdf`.

[13] Šipková, V.—Hluchý, L.—Dobrucký, M.—Bartok, J.—Nguyen, B. M.: Manufacturing of Weather Forecasting Simulations on High Performance Infrastructures. Proceeding IEEE International Conference on e-Science (e-Science): ECW 2016 Environmental Computing Workshop, 2016, pp. 432–439. ISBN: 978-1-5090-4273-9. ISSN: 2325-372X, doi: 10.1109/escience.2016.7870932.

[14] Kvassay, M.—Krammer, P.—Hluchý, L.—Schneider, B.: Causal Analysis of an Agent-Based Model of Human Behaviour. Complexity, Vol. 2017, 2017, Article ID 8381954, doi: 10.1155/2017/8381954.

[15] Balogh, Z.—Gatial, E.—Hluchý, L.: Poll Sourcing for Crisis Response. Proceedings of the International ISCRAM Conference (ISCRAM), 2016, Vol. 9. ISBN: 978-84-608-7984-8. ISSN: 2411-3387.

[16] Pajorová, E.—Hluchý, L.: Water Management Simulations, Computing and Virtual Reality of Research Results. International Journal of Advances in Electronics and Computer Science, Vol. 4, 2017, No. 10, pp. 39–41. ISSN: 2393-2835.

[17] Nguyen, G.—Bobák, M.—Hluchý, L.: Geospatial Preprocessing for Situational Assessment Through Hydraulic Simulations. 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2017), IEEE, 2017, pp. 2402–2408. ISBN: 978-1-5386-2164-6, doi: 10.1109/fskd.2017.8393154.

[18] Tran, D. V.—Šipková, V.—Hluchý, L.—Nguyen, G.—Dobrucký, M.—Astaloš, J.: Problem Solving Environment for Water-Supply Systems. 15th International Conference on Electronics, Informatics and Communications (ICEIC 2016), Danang, Vietnam, IEEE, 2016, pp. 119–122. ISBN: 978-146738016-4, doi: 10.1109/elinfocom.2016.7562923.

[19] Nguyen, G.—Šipková, V.—Krammer, P.—Hluchý, L.—Dobrucký, M.—Astaloš, J.: Center for Risk Research in Urban Water-Supply Systems. 19th IEEE International Conference on Intelligent Engineering Systems (INES 2015), Slovak University of Technology in Bratislava, Bratislava, 2015, pp. 39–43. ISBN: 978-1-4673-7938-0, doi: 10.1109/ines.2015.7329735.

[20] Krammer, P.—Hluchý, L.: Optimized Computing of Parameters for Functional Regression in Data Mining. 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012), IEEE, 2012, pp. 1603–1609. ISBN: 978-1-4673-0023-0, doi: 10.1109/fskd.2012.6234314.

[21] Hluchý, L.—Nguyen, G.—Astaloš, J.—Tran, D. V.—Šipková, V.—Nguyen, B. M.: Effective Computation Resilience in High Performance and Distributed Environments. Computing and Informatics, Vol. 35, 2016, No. 6, pp. 1386–1415. ISSN: 1335-9150.

[22] CE-FRAME: Central European Flood Risk Assessment and Management in CENTROPE. European Regional Development Fund project, 2010–2013, https://www.ceframe.eu/.

[23] Directive 2007/60/EC of the European Parliament and of the Council of 23 October, 2007 on the Assessment and Management of Flood Risks (Text with EEA Relevance). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32007L0060.

Ivan Mrnčo is an experienced specialist for urban water management in relation to collection systems, waste water treatment plants, water distribution infrastructure and flood modeling. He has over 13 years' experience in water management. He was involved as a technical expert for the project of hydraulic evaluation of water supply and sewage systems in Bulgaria. He has over 5 years' experience as Project Leader in wastewater master planning, water distribution infrastructure and flood forecasting systems with connection to Slovak water legislation, environmental impact and urban development, and in other projects.

Peter Blštak is a skilled and experienced programming engineer and expert for data collection, data mining and data processing. In the research and development project he was responsible for the continuous data collection design, data mining and data processing of measured data in database systems, real time processing of the data, extrapolation and application of the result data in a mathematical model.

Peter Hudec is a skilled and experienced programming engineer and specialist for processing and postprocessing of database outputs, and mathematical models, cloud computing and server applications of cloud solutions. He participated in various research and development projects on application of theoretical mathematical models to sofware design in cloud solution, where his work was focused to processing up to presentation level. He was also responsible for the overall security and control system and the security of data transport.

Matej Kochan is skilled and experienced specialist for IT and Telco solutions. He has experience with wide deployment of wired and wireless communication solutions. In research and development he was involved in desing innovating solutions designs, testing and practical application and realization of wired and wirelles technology with the aim to NGN wirelless networks, GSM/3G/4G/LTE, IoT and 5G.

Tomáš Gibala is Senior Specialist at DHI SLOVAKIA, ltd. He works mainly on mathematical modeling and distributed systems, applying them into solving domain-specific problems in water management, meteorology and hydrology. He has participated in several national research projects, including development of new modeling software products. He is the author of more than 30 publications in his research field.

Ondrej Habala is Researcher at the Institute of Informatics of the Slovak Academy of Sciences. He works mainly on distributed computing system, applying them into solving domain-specific problems in meteorology and hydrology. He has participated in more than 10 national and international research projects, including EU FP5, FP6, FP7 and H2020 projects. He is the author of more than 80 publications in his research field.