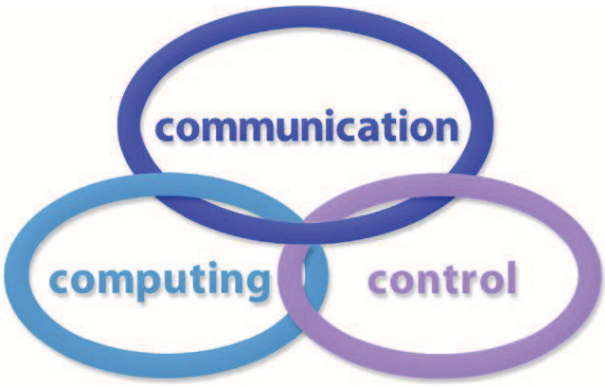
INTERNATIONAL JOURNAL

of

COMPUTERS COMMUNICATIONS & CONTROL

ISSN 1841-9836



A Bimonthly Journal
With Emphasis on the Integration of Three Technologies

Year: 2017 Volume: 12 Issue: 1 Month: February

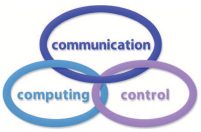
This journal is a member of, and subscribes to the principles of, the Committee on Publication Ethics (COPE).

http://univagora.ro/jour/index.php/ijccc/

CCC Publications

BRIEF DESCRIPTION OF JOURNAL

Publication Name: International Journal of Computers Communications & Control.
Acronym: IJCCC; **Starting year of IJCCC:** 2006.
ISO: Int. J. Comput. Commun. Control; **JCR Abbrev:** INT J COMPUT COMMUN.
International Standard Serial Number: ISSN 1841-9836.
Publisher: CCC Publications - Agora University of Oradea.
Publication frequency: Bimonthly: Issue 1 (February); Issue 2 (April); Issue 3 (June); Issue 4 (August); Issue 5 (October); Issue 6 (December).
Founders of IJCCC: Ioan DZITAC, Florin Gheorghe FILIP and Misu-Jan MANOLESCU.
Logo:



Indexing/Coverage:

- Since 2006, Vol. 1 (S), IJCCC is covered by Thomson Reuters and is indexed in ISI Web of Science/Knowledge: Science Citation Index Expanded.

  2016 Journal Citation Reports® Science Edition (Thomson Reuters, 2016):
  Subject Category: (1) Automation & Control Systems: Q4(2009,2011,2012,2013,2014,2015), Q3(2010); (2) Computer Science, Information Systems: Q4(2009,2010,2011,2012,2015), Q3(2013,2014).

  Impact Factor/3 years in JCR: 0.373(2009), 0.650 (2010), 0.438(2011); 0.441(2012), 0.694(2013), 0.746(2014), 0.627(2015).
  Impact Factor/5 years in JCR: 0.436(2012), 0.622(2013), 0.739(2014), 0.635(2015).

- Since 2008 IJCCC is indexed by Scopus (SNIP2014= 1.029):
  Subject Category: (1) Computational Theory and Mathematics: Q4(2009,2010,2012,2015), Q3(2011,2013,2014); (2) Computer Networks and Communications: Q4(2009), Q3(2010, 2012, 2013, 2015), Q2(2011, 2014); (3) Computer Science Applications: Q4(2009), Q3(2010, 2011, 2012, 2013, 2014, 2015).

  SJR: 0.178(2009), 0.339(2010), 0.369(2011), 0.292(2012), 0.378(2013), 0.420(2014), 0.319(2015).

- Since 2007, 2(1), IJCCC is indexed in EBSCO.

Focus & Scope: International Journal of Computers Communications & Control is directed to the international communities of scientific researchers in computer and control from the universities, research units and industry.

To differentiate from other similar journals, the editorial policy of IJCCC encourages the submission of original scientific papers that focus on the integration of the 3 "C" (Computing, Communication, Control).

In particular the following topics are expected to be addressed by authors: (1) Integrated solutions in computer-based control and communications; (2) Computational intelligence methods (with particular emphasis on fuzzy logic-based methods, ANN, evolutionary computing, collective/swarm intelligence); (3) Advanced decision support systems (with particular emphasis on the usage of combined solvers and/or web technologies).

Editorial Address:
Agora University/ R&D Agora Ltd. / S.C. Cercetare Dezvoltare Agora S.R.L.
Piata Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526
Tel./ Fax: +40 359101032
E-mail: ijccc@univagora.ro, rd.agora@univagora.ro, ccc.journal@gmail.com
Journal website: http://univagora.ro/jour/index.php/ijccc/

IJCCC EDITORIAL BOARD MEMBERS

Gheorghe Păun
Institute of Math. of Romanian Academy
Bucharest, PO Box 1-764, Romania
gpaun@us.es

Mario de J. Pérez Jiménez
Dept. of CS and Artificial Intelligence
University of Seville, Sevilla,
Avda. Reina Mercedes s/n, 41012, Spain
marper@us.es

Dana Petcu
Computer Science Department
Western University of Timisoara
V.Parvan 4, 300223 Timisoara, Romania
petcu@info.uvt.ro

Radu Popescu-Zeletin
Fraunhofer Institute for Open
Communication Systems
Technical University Berlin, Germany
rpz@cs.tu-berlin.de

Imre J. Rudas
Óbuda University
Budapest, Hungary
rudas@bmf.hu

Yong Shi
School of Management
Chinese Academy of Sciences
Beijing 100190, China &
University of Nebraska at Omaha
Omaha, NE 68182, U.S.A.
yshi@gucas.ac.cn, yshi@unomaha.edu

Athanasios D. Styliadis
University of Kavala
Institute of Technology
65404 Kavala, Greece
styliadis@teikav.edu.gr

Gheorghe Tecuci
Learning Agents Center
George Mason University
U.S.A.
University Drive 4440, Fairfax VA
tecuci@gmu.edu

Horia-Nicolai Teodorescu
Faculty of Electronics and
Telecommunications
Technical University "Gh. Asachi" Iasi
Iasi, Bd. Carol I 11, 700506, Romania
hteodor@etc.tuiasi.ro

Dan Tufiş
Research Institute for Artificial Intelligence
of the Romanian Academy
Bucharest, "13 Septembrie" 13, 050711, Romania
tufis@racai.ro

Lotfi A. Zadeh
Director,
Berkeley Initiative in Soft Computing (BISC)
Computer Science Division
University of California Berkeley,
Berkeley, CA 94720-1776
U.S.A.
zadeh@eecs.berkeley.edu

DATA FOR SUBSCRIBERS
Supplier: Cercetare Dezvoltare Agora Srl (Research & Development Agora Ltd.)
Fiscal code: 24747462
Headquarter: Oradea, Piata Tineretului Nr.8, Bihor, Romania, Zip code 410526
Bank: BANCA COMERCIALA FEROVIARA S.A. ORADEA
Bank address: P-ta Unirii Nr. 8, Oradea, Bihor, România
IBAN Account for EURO: RO50BFER248000014038EU01
SWIFT CODE (eq.BIC): BFER

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, Volume 12, Issue 1, February, 2017.

Contents

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(1):7-25, February 2017.

Improved Timing Attacks against the Secret Permutation in the McEliece PKC

D. Bucerzan, P.L. Cayrel, V. Dragoi, T. Richmond

Dominic Bucerzan*
Aurel Vlaicu University of Arad
Department of Mathematics and Computer Science
Romania, 310330 Arad, Elena Dragoi, 2
Corresponding author:dominic@bbcomputer.ro

Pierre-Louis Cayrel
Laboratoire Hubert Curien, UMR CNRS 5516,
Université de Lyon, Saint-Etienne, France
pierre.louis.cayrel@univ-st-etienne.fr

Vlad Dragoi
Laboratoire LITIS - EA 4108
Université de Rouen - UFR Sciences et Techniques,
76800 Saint Etienne du Rouvray, France
vlad.dragoi1@univ-rouen.fr

Tania Richmond
Laboratoire IMATH, EA 2134,
Avenue de l'Université , BP 20132,
83957 La Garde Cedex, France
tania.richmond@univ-tln.fr

Abstract: In this paper, we detail two side-channel attacks against the McEliece public-key cryptosystem. They are exploiting timing differences on the Patterson decoding algorithm in order to reveal one part of the secret key: the support permutation. The first one is improving two existing timing attacks and uses the correlation between two different steps of the decoding algorithm. This improvement can be deployed on all error-vectors with Hamming weight smaller than a quarter of the minimum distance of the code. The second attack targets the evaluation of the error locator polynomial and succeeds on several different decoding algorithms. We also give an appropriate countermeasure.
Keywords: communication systems, theory of error correcting codes, code-based cryptography, McEliece PKC, side-channel attacks, timing attack, extended Euclidean algorithm.

1 Introduction

In the history of cryptography public key schemes are quite recent. In the classic era both encryption and decryption algorithms are symmetric, that is why having access to the keys gives a total control on both encryption and decryption steps. These types of schemes are called symmetric cryptosystem and are still widely used.

Nonetheless several security properties can hardly be achieved with the use of symmetric cryptography. That is the main reason public key cryptography appeared. The first public key cryptosystem was invented by Diffie and Hellman [5]. But despite the fact that public key schemes are rather new, they are widely spread in practice. Moreover there are few constructions to be used in practice and they are all based on number theory problems, more exactly the hardness of factoring and discrete logarithm problem. But this is rather concerning since there is little theoretical support indicating that these problems are indeed hard. One of the main threats for these schemes is the arrival of the quantum computer. Peter Shor has shown that both computation of discrete logarithm and factoring problem can be done in polynomial time on a quantum machine [13].

In reaction to this thread, several solutions have been proposed, such as hash-based cryptography, code-based cryptography, lattice-based cryptography, and multivariate cryptography. The new facts concerning the post-quantum cryptography are well discussed in [2]. Code-based cryptosystems were introduced in 1978 by Robert J. McEliece [8]. But many variants were attacked and partially or totally broken. Up to now, none of the proposed variants seemed as

strong and secure as the original McEliece public-key cryptosystem (PKC) using Goppa codes. Structural attacks managed to reveal the secret key and totally break variants that used the generalized Reed-Solomon codes [15] or QC-LDPC codes [10] and many other variants.

As the Goppa codes still resist to structural attacks, they present a real interest in our approach. So we focus our attention on the cryptanalysis of the McEliece PKC using Goppa codes. More exactly on the side-channel attacks using time differences between two executions of the same task. The interest of timing attacks is both practical and theoretical: we avoid unsecured implementations and discover new attacks succeeding in a polynomial time. The main purpose of these type of attacks is to reveal a part of the secret key and a breaking point of an algorithm. The authors of such exploits usually end up by giving the necessary countermeasures and the secure variant of the algorithms.

In the case of the McEliece PKC using Goppa codes, most of the timing attacks were discovered since 2008. Falko Strenzke's articles mention several weak points mostly situated in the decoding algorithm [14, 16, 18, 19]. Some of these can be repaired by an intelligent and cautious way of the programming manner where countermeasures were proposed in [1, 3, 19]. All of the mentioned attacks were realised on a McEliece PKC implementation using the Patterson algorithm (cf. Fig. 1) for decoding Goppa codes. The number of error corrections in the Patterson algorithm is bounded: up to t errors can be corrected, where t is the degree of the Goppa polynomial.

Our contribution is to reveal a new timing attack against the error-locator polynomial (ELP) evaluation and to improve two existing attacks. In our new version of the two existing attacks combined, we detail how the relation between the two attacks is crucial in order to avoid eventual errors. The attacks are executed on the extended Euclidean algorithm (EEA) and exploit the number of iterations. As the authors mentioned, the initial attacks are limited and may not allow the total break of the permutation. This limit is situated in the number of equations detected by their attack. We will use a new relation between the number of iterations in the two steps in order to expand the system and to fully determine the secret permutation. We will also give a single countermeasure, which is efficient to all types of attacks exploiting the EEA in this particular manner.

The second contribution is in giving a new timing attack against the ELP evaluation. The importance of this new attack is that it operates on the polynomial evaluation, applied in several decoding algorithms as the Patterson algorithm, Berlekamp-Massey algorithm or any general decoder for alternant codes. We will show that this attacks succeeds on several variants of the polynomial evaluation.

2 Background

For all the necessary background on coding theory we address the reader to any book in this field, for example [7]. Nevertheless we give here the details concerning binary Goppa codes.

2.1 Goppa codes

Definitions and Properties

We will focus exclusively on binary Goppa codes in this paper, but it is easy to generalize our results to q-ary codes:
-Goppa polynomial: $g(x)$ is a polynomial over $\mathbb{F}_{2^m}[x]$ with $\deg(g) = t$.
-Goppa support: $\mathcal{L} = \{\alpha_0, \alpha_1, .., \alpha_{n-1}\}$ subset of $\mathbb{F}_{2^m}$ s.t. $g(\alpha_i) \neq 0$.
The syndrome polynomial associated to $c \in \mathbb{F}_2^n$: $\mathcal{S}_c(x) = \sum_{i=1}^{n} \frac{c_i}{x+\alpha_i}$.

Definition 1 (Binary Goppa code)**.** Given $g(x), \mathcal{L}$ and $\mathcal{S}_c(x)$ the binary Goppa code is defined as:

$$\Gamma(\mathcal{L}, g) = \{c \in \mathbb{F}_2^n \mid \mathcal{S}_c(x) \equiv 0 \mod g(x)\}.$$

Among the most important properties that a Goppa code satisfies we recall the followings:

Proposition 2. *A Goppa code $\Gamma(\mathcal{L}, g)$ is a linear code over $\mathbb{F}_2$. Its length is given by $n = |\mathcal{L}|$, its dimension is $k \geq n - mt$, where $t = \deg(g)$ and its minimun distance $d \geq t + 1$.*

The syndrome polynomial $\mathcal{S}_c(x)$ satisfies the following property:

$$\mathcal{S}_c(x) = \tfrac{\omega(x)}{\sigma(x)} \mod g(x),$$

where $\sigma(x) = \prod_{i=1}^{t}(x + a_i)$ is called the error locator polynomial (ELP) and the elements $\forall i \in \{1, \ldots, t\} a_i \in \mathcal{L}$ are the error positions.

Irreducible binary Goppa codes are defined by an irreducible Goppa polynomial g and admit the maximum length $n = 2^m$. We use this type of codes in the rest of the paper and adopt the following notations:

- For the permutation of the support elements:
  $\Pi(\mathcal{L}) = \mathcal{L}' = (\Pi(0), \Pi(1), \ldots, \Pi(\alpha_i), \ldots, \Pi(\alpha_{n-2}))$. where Π is an element of the symmetric group.
- Let $P(x)$ be a monic polynomial of degree t over $\mathbb{F}_{2^m}$ with t roots denoted a_i:

  $$P(x) = x^t + S_{t-1}^t x^{t-1} + S_{t-2}^t x^{t-2} + \ldots + S_2^t x^2 + S_1^t x + S_0^t,$$

  where the coefficients $S_i \in \mathbb{F}_2^m$ are the elementary symmetric functions:

  $S_{t-1}^t = \sum_{i=1}^{t} a_i$, $S_{t-2}^t = \sum_{\substack{i=1,j=1 \\ i \neq j}}^{t} a_i a_j$, $\ldots$,

  $S_1^t = \sum_{j=1}^{t} \prod_{\substack{i=1 \\ i \neq j}}^{t} a_i$ and $S_0^t = \prod_{i=1}^{t} a_i$.

Alternant decoders

For the (irreducible) binary Goppa codes, we can use (at least) three different decoding algorithms: 1. the extended Euclidean algorithm (EEA); 2. the Berlekamp-Massey algorithm; 3. the Patterson algorithm.

Extended Euclidean Algorithm (EEA). The first decoding algorithm can correct up to $\frac{t}{2}$ errors. We can increase the error-correction capabcity and correct up to t errors when the syndrome associated to g^2 is used. Unfortunately, the corresponding parity check matrix has two times more rows and the construction is more complex.

The Berlekamp-Massey algorithm. Similarly as the EEA, the Berlekamp-Massey algorithm has to use g^2 in order to decode t errors. The advantage of this algorithm is that it isn't vulnerable to several existing timing attacks and it allows a fast and constant-time computation. Some advantages are listed in [3].

The Patterson algorithm. The Patterson algorithm offers another solution for the syndrome decoding. The decryption described in [12] permits to correct up to t errors by using the syndrome associated to g but not to g^2.

2.2 The McEliece Cryptosystem

The McEliece PKC [8] is composed by the three following algorithms.

*Key generation:*The first step is to generate the support (the set of $n = 2^m$ elements) and the Goppa polynomial g of degree t. Then, the parity check matrix can be built and brought to a systematic form: $[\mathcal{I}_{n-k}|R]$ in order to recover a generator matrix G of the Goppa code. We randomly choose a non-singular $k \times k$ matrix S and a $n \times n$ permutation matrix Π, and compute the public $k \times n$ generator matrix $\mathcal{G} = SG\Pi$. The key generation procedure outputs $\mathsf{sk} = (\Gamma(\mathcal{L}, g), S, \Pi)$ and $\mathsf{pk} = (n, t, \mathcal{G})$.

Message encryption:

- *Inputs:* message $\mathsf{m} \in \mathbb{F}_2^k$, public key $\mathsf{pk} = (n, t, R^T)$.
- *Output:* ciphertext $\mathsf{z} \in \mathbb{F}_2^n$.

1. Randomly choose an n-bit error-vector with weight $\mathsf{wt}(e) = t$;
2. Encode $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$;
3. Return z.

Message decryption:

- *Inputs:* ciphertext $\mathsf{z} \in \mathbb{F}_2^n$, secret key $\mathsf{sk} = (\Gamma(\mathcal{L}, g), S, \Pi)$.
- *Output:* message $\mathsf{m} \in \mathbb{F}_2^k$.

1. Compute $\mathsf{z}' = \mathsf{z}\Pi^{-1}$;
2. Find $\mathsf{m}' = \mathsf{m}S$ from $\mathsf{z}' \oplus e$ using $\mathcal{D}ecode(\mathsf{z}')$ with the secret code;
3. Compute $\mathsf{m} = \mathsf{m}'S^{-1}$;
4. Return m.

$\mathcal{D}ecode(.)$ is an alternant decoder (presented in the previous subsection).

Existing side-channel attacks There are several papers on side-channel attacks against the McEliece PKC and a quick review must be done in order to clear up the reader's understanding. Most of the attacks target the Patterson decoding algorithm and exploit several weaknesses.

Table 1: Patterson algorithm: existing timing attacks and countermeasures

Step	Ref.	Countermeasure
❶ $\mathsf{z}' = \mathsf{z}\Pi^{-1}$		
❷ $\mathcal{S}_{\mathsf{z}'}(x) = \mathcal{H}'\mathsf{z}'(x^{t-1}, \ldots, x^2, x, 1)^T$		
❸ $\mathcal{S}_{\mathsf{z}'}(x)^{-1} \mod g(x)$ *via EEA*	[18]	control flow
❹ $\tau(x) = \sqrt{x + \mathcal{S}_{\mathsf{z}'}(x)^{-1}}$		
❺ $b(x)\tau(x) \equiv a(x) \mod g(x)$ $\deg(a) \leq \lfloor \frac{t}{2} \rfloor$; $\deg(b) \leq \lfloor \frac{t-1}{2} \rfloor$ *via EEA*	[14, 16]	in EEA make sure $\deg(r_i) = \deg(r_{i-1}) - 1$ and $\deg(\tau) = t - 1$
❻ $\sigma(x) = a^2(x) + xb^2(x)$		
❼ $e = (\sigma(\alpha_0), \sigma(\alpha_1), \ldots, \sigma(\alpha_{n-1})) \oplus (1, \ldots, 1)$	[1, 17, 19]	the non-support or make sure $\deg(\sigma) = t$
❽ $e' = e\Pi$		
❾ $\mathsf{z} = \mathsf{z}' \oplus e'$		

There are mainly two types of attacks classified by their goal:
1. Attacks recovering the secret message m [1, 17, 19]; 2. Attacks recovering (fully or partially) the secret key sk [14, 16–18].

The attacks on steps ❸ and ❺ are able to determine some relations on the support elements by counting the number of iterations in the EEA. We improve it in Section 3.

The attack on step ❼ reveals error positions using timing differences in the ELP evaluation. The attacker is able to find the error-vector with a certain non negligible probability. The basic idea is that two different polynomials, with some different degrees, are not evaluated in the same time. So the timing difference gives some information on the error-vector. We improve this attack in Section 4.

In the rest of this paper, we assume that an attacker chooses a weight $0 < r < t$ for the error-vector e and we use the following notations: $\deg(g) = t$ and $\mathsf{wt}(e) = r$.

In the next Sections we will detail the complexity analysis of these attacks as well as adapted parameter values.

3 Timing attack against double using of the EEA

Goal: The attacker's goal is to recover the secret permutation Π.

Identification of a leakage: The leakage is identified at steps ❸ and ❺ of the Patterson algorithm. This type of attack was already published in [16, 18]. The two steps using the EEA are considered as independent parts. In this section, we propose to show the relation existing between both steps and thus attack them. In fact, the main problem of previous attacks is the limited number of cases that can be exploited. They just can be applied on $\mathsf{wt}(e) \in \{2, 4\}$ as shown in [16] or $\mathsf{wt}(e) \in \{2, 4, 6\}$ as presented in [18].

The problem comes from a simple fact: the number of iterations is given by two conditions. One of the condition is that all of the quotients in the EEA must be polynomials of degree equal to one. So when this condition is not fulfilled the number of iterations could not be any longer controlled by the attacker. We will use $N_{❸}$ and $N_{❺}$ as notations for the number of iterations in the 3rd step (respectively 5th step) of the Patterson algorithm.

Motivations of our attack: We will show that using the relation between both steps will allow us to fully control the number of iterations. The other contribution is in finding the relation between both steps and in using it for building a larger set of equations. We will show that we are able to extend the limited equation number of the system up to $\mathsf{wt}(e) = \frac{\deg(g)}{2}$.

The main interest is that instead of finding only equations involving the permutation of 2, 4 or maybe 6 elements, we can extend it as much as necessary in order to discover the secret permutation.

In terms of complexity, instead of enumerating all possible permutations, i.e. $n!$ permutations, we reduce the complexity to the following expression:

$$\sum_{i=3}^{p} \binom{n}{i}, \quad \text{where } p \leq \frac{\deg(g)}{2} - 1$$

Hence for small values of p, we have:

$$\sum_{i=3}^{p} \binom{n}{i} \leq \binom{n}{p} \times (p-2) \leq \left(\frac{en}{p}\right)^{p} \times (p-2)$$

(where $e = 2.718281828\ldots$ is the basis of the natural logarithms).

In order to make clear the difference between the naive attack and our attack, we propose to give a lower bound for the complexity of the naive attack:

$$\left(\frac{n}{e}\right)^n \leq n!$$

Finally:

$$\sum_{i=3}^{p} \binom{n}{i} \leq \left(\frac{en}{p}\right)^p \times (p-2) << \left(\frac{n}{e}\right)^n \leq n!$$

So the naive attack would be exponential in the length of the code as a timing attack would only have a complexity exponential in the maximum of the error-vector's weight needed for the attack (often extremely small in comparison with the code's length).

Scenario: The attacker proceeds in the three following steps:

1. He chooses a random message m and computes $c = \mathsf{m}\mathcal{G}$;
2. He randomly chooses an error-vector e of small weight $\mathsf{wt}(e) < t$ (t is the correction capacity of the code) and computes $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$;
3. He sends z to an oracle ($\mathcal{O}$), which outputs the message m and the number of iterations in steps ❸ and ❺ of the Patterson algorithm from Fig. 1.

Main idea: For $\mathsf{wt}(e) = 2p$ with $p \in \mathbb{N}$, the attacker will find equations having the following form: $\sum_{i=1}^{\mathsf{wt}(e)} \Pi(\alpha_i) = 0$. He will be able to build this type of equations with $0 < \mathsf{wt}(e) < \frac{\deg(g)}{2}$. We will denote by $N_❸$ the number of iterations in the ❸ step.

Conditions: The general assumption is that the attacker knows the public key pk, the order of all elements in the support $\mathcal{L}$ ($\mathcal{L}$ is supposed to be public, for example in the lexicographic order) and has access to an oracle $\mathcal{O}$. These assumptions are the same as in previously mentionned works. We improve the attack in the same context. The oracle $\mathcal{O}$ is also able to give some extra informations: the timing for the whole particular algorithm or just one step. We assume that the attacker can violate the procedure by adding $\mathsf{wt}(e) < t$ errors. The attacker is able to choose the number and the positions of errors.

3.1 Step ❸ in the Patterson algorithm

It was shown in [18] that the syndrome inversion leaks some information. The attack is based on the number of iterations used in the EEA, in order to compute the inverse of the syndrome polynomial $S(x)$ modulo the Goppa polynomial $g(x)$. It uses the following properties:

$$N_❸ \leq \deg(\sigma) + \deg(\sigma'), \text{ for } \mathsf{wt}(e) < \frac{\deg(g)}{2}.$$

We will not detail here all conditions, as they are well explained in [18], but we only give some important facts in order to make things clearer and to prepare the attack. Let us consider the ELP

$$\sigma(x) = x^r + S^r_{r-1}x^{r-1} + S^r_{r-2}x^{r-2} + \cdots + S^r_2 x^2 + S^r_1 x + S^r_0,$$

with $r \equiv 0 \mod 2$. Then $\sigma'(x) = S^r_{r-1}x^{r-2} + \cdots + S^r_3 x^2 + S^r_1$.

In this case the maximum number of iterations is given by the coefficient $S^r_{r-1} = \sum_{i=1}^{r} a_i$. So if $S^r_{r-1} \neq 0$, we obtain $N_{❸} = 2r - 2$ and all quotients have a degree equal to 1. If $S^r_{r-1} = 0$, $S^r_{r-3} \neq 0$, then $N_{❸} = 2r - 4$ and all quotients have a degree equal to 1.

3.2 Step ❺ in the Patterson algorithm

Locate the leakage: Two observations have to be done in order to understand and to locate the leakage point. The first one is about the number of iterations. It was proven (in [14]) that this number is (with a high probability):

$$N_{❺} = \sum_{i=1}^{N_{❺}} \deg(q_i) = \deg(b).$$

In the following paragraph, we will give some relations between $\tau(x)$, $b(x)$, $a(x)$ and $\sigma(x)$ (given in steps ❹, ❺ and ❻ in Fig. 1). We will prove some new relations. The new relations between these polynomials allow us to build the attack in such a manner that previous ambiguous cases were eliminated. These relations are crucial for better understanding of the entire decryption algorithm as they influence each step of the process and each particular form of the involved polynomials.

There are some useful properties that are going to be used in our approach:

Proposition 3. *1. If* $r \equiv 0 \mod 2$, *then* $\deg(a) = \frac{r}{2}$ *see [14].*

2. If $r \equiv 1 \mod 2$, *then* $\deg(b) = \frac{r-1}{2}$ *see [14].*

3. If $\deg(\tau) \leq \lfloor \frac{r}{2} \rfloor$, *then* $\deg(a) = \deg(\tau) + \deg(b)$.

4. If $\deg(\tau) \leq \lfloor \frac{r}{2} \rfloor$ *and* $\deg(\tau) \neq 0$, *then* $\mathsf{wt}(e) \equiv 0 \mod 2$.

Fact:

- *When* $\mathsf{wt}(e)$ *is odd:* For an error-vector with Hamming weight $\mathsf{wt}(e) = 2k + 1$, with $k \leq p - 1$, we have the following relations:
  $\deg(b) = k$, $\deg(a) \leq k$ and $\deg(\tau) \geq 2p - k$.
- *When* $\mathsf{wt}(e)$ *is even:* For an error-vector with Hamming weight $\mathsf{wt}(e) = 2k$, with $k \leq p$, we have the following relations:
  $\deg(a) = k$, $\deg(b) \leq k - 1$ and $\deg(b) = 0 \Leftrightarrow \deg(\tau) = k$.

3.3 Number of iterations

We saw that the number of iterations in the EEA equals $\deg(b)$ so we will focus on the form of the polynomial $b(x)$. More exactly, in the case when r is even. Let:

$$\sigma(x) = x^{2p} + S^{2p}_{2p-1}x^{2p-1} + S^{2p}_{2p-2}x^{2p-2} + S^{2p}_{2p-3}x^{2p-3} + \ldots + S^{2p}_{2}x^2 + S^{2p}_{1}x + S^{2p}_{0}.$$

We separate odd powers from even ones and get:

$$\begin{aligned}
\sigma(x) &= (x^{2p} + S^{2p}_{2p-2}x^{2p-2} + \ldots + S^{2p}_{2}x^2 + S^{2p}_{0}) \\
&\quad +(S^{2p}_{2p-1}x^{2p-1} + S^{2p}_{2p-3}x^{2p-3} + \ldots + S^{2p}_{1}x) \\
\sigma(x) &= (x^p + \sqrt{S^{2p}_{2p-2}}x^{p-1} + \ldots + \sqrt{S^{2p}_{2}}x + \sqrt{S^{2p}_{0}})^2 \\
&\quad +x(\underbrace{\sqrt{S^{2p}_{2p-1}}x^{p-1} + \ldots + \sqrt{S^{2p}_{1}}}_{b(x)})^2
\end{aligned}$$

So $\deg(b)$ is given by the coefficients S^{2p}_{2i-1} with $i \in \{1, 2, \ldots, p\}$. Therefore the number of iterations could be given by the same coefficients under an extra condition: all of the quotients have a degree equal to one. So we can distinguish $p-1$ possible cases depending on the coefficients, if the degree of the coefficients is equal to 1 in each iteration. Therefore:

$$\begin{cases} N_{\mathbf{5}} = p-1 & \text{if } S^{2p}_{2p-1} \neq 0 \\ N_{\mathbf{5}} = p-2 & \text{if } S^{2p}_{2p-1} = 0 \text{ and } S^{2p}_{2p-3} \neq 0 \\ N_{\mathbf{5}} = p-3 & \text{if } S^{2p}_{2p-1} = 0\ S^{2p}_{2p-3} = 0 \text{ and } S^{2p}_{2p-5} \neq 0 \\ \quad \vdots & \end{cases}$$

In all cases, the same assumption is made: the degree of the quotient equals 1 in each iteration. It means that we might have the number of iterations without any condition on the coefficients.

3.4 Attack against the pair $(N_{\mathbf{3}}, N_{\mathbf{5}})$

How it works. In this paragraph, we will explain how our attack works. We start by presenting the general relation for the pair $(N_{\mathbf{3}}, N_{\mathbf{5}})$. Using 3.1 and 3.2 we get the following property:

Proposition 4. *Let* $\mathsf{wt}(e) = 2p < t/2$.

$$(N_{\mathbf{3}}, N_{\mathbf{5}}) = (4p-4, p-2) \Rightarrow \sum_{i=1}^{2p} a_i = 0$$

with probability $\mathcal{P}_{success}$.

We will give a snapshot of each step in the attack and give some information on the success probability.

1. Find the position of $\Pi(0)$ (see [14]).
2. Set $\mathsf{wt}(e) = 4$:
   Fix $\Pi(0) \in \{\text{error-vector}\}$ and find $\sum_{i=1}^{3} a_i$ with $a_i \neq 0$.
   Fix $\Pi(0) \notin \{\text{error-vector}\}$ and find $\sum_{i=1}^{4} a_i$ with $a_i \neq 0$.
3. Set $\mathsf{wt}(e) = 6$:
   Fix $\Pi(0) \in \{\text{error-vector}\}$ and find $\sum_{i=1}^{5} a_i$ with $a_i \neq 0$.
   Fix $\Pi(0) \notin \{\text{error-vector}\}$ and find $\sum_{i=1}^{6} a_i$ with $a_i \neq 0$.
4. ...
5. Set $\mathsf{wt}(e) = \lfloor \frac{t}{2} \rfloor$:
   Fix $\Pi(0) \in \{\text{error-vector}\}$ and find $\sum_{i=1}^{\mathsf{wt}(e)-1} a_i$ with $a_i \neq 0$.
   Fix $\Pi(0) \notin \{\text{error-vector}\}$ and find $\sum_{i=1}^{\mathsf{wt}(e)} a_i$ with $a_i \neq 0$.

In Appendix 1, a toy example is presented for a better comprehension.

3.5 Success probability

The success probability $\mathcal{P}_{success}$ is described by the following event: *{All the quotients have a degree=1}*. If we consider all elements of our support as uniformly distributed variables and the independence of each step inside the EEA, under the initial assumptions we have:

$$\begin{aligned}\mathcal{P}_{success} &= \mathcal{P}(\{N_{❸} = 4p-4\} \cap \{N_{❺} = p-2\}) \\ &= \mathcal{P}(\{N_{❸} = 4p-4\})\mathcal{P}(\{N_{❺} = p-2\}) \\ &= (1-\tfrac{1}{n})^{N_{❸}+N_{❺}}.\end{aligned}$$

Experimental results show that for $n = 2048$ and $\mathsf{wt}(e) = 4$, in order to find equations of the following form: $\Pi(\alpha_1) + \Pi(\alpha_2) + \Pi(\alpha_3) + \Pi(\alpha_4) = 0$ the probability equals 0.998. It means that less that 0.2% of the cases are not exploitable among all possible cases under the condition: $N_{❸} = 4$ and $N_{❺} = 0$. In other words, each time this combination is revealed, the probability of having a good equation for our attack equals 0.998 for the given parameters.

3.6 Experimental work

In order to validate the relations that we presented in the previous paragraph for $(N_{❸}, N_{❺})$, we used a *Pari/GP* implementation of the McEliece cryptosystem (the code will be publicly available). We computed a keypair, then encoded and decoded a given message multiple times, by checking the value of the couple $(N_{❸}, N_{❺})$, searching for the valid combinations described above. We also used different values for m, the extension degree of the finite field. We ran the algorithm until we got the specific combination about hundred times. Then, we obtained an average value for the necessary iterations required to get the searched combination. The results are presented in the following table:

Table 2: Number of necessary iterations to get the combination for error-vectors of Hamming weight 4, 6 and 8

Combination:			
$N_{❸}$	4	8	12
$N_{❺}$	0	1	2
Number of iterations for $m = 7$	127	138	142
Number of iterations for $m = 8$	235	270	273

It means that for $m = 7$, we need to send to the oracle in average 127 different ciphertexts, in order to get the wanted relation $(\Pi(\alpha_1) + \Pi(\alpha_2) + \Pi(\alpha_3) + \Pi(\alpha_4) = 0)$. In the case of the previous equation, the density of such configurations equals in average

$$\frac{1}{127} \times \mathcal{P}_{success} = \frac{1}{127} \times \left(1 - \frac{127}{128}\right)^4.$$

Knowing one relation allows us, by fixing one of the positions, to reduce the number of ciphertexts that has to be sent to the oracle. It means that wanted relations are revealed more often as we progress in the attack. It also gives the first intuition on the structure of the permutation (see Appendix 1).

Attack implementation In order to practically test our attack, we used the same software implementation. In order to reveal timings close to real values, we repeated the attack for more than 10^6 times. We presented the obtained results are in the following table:

Table 3: Timings (in sec.) for decryption in the case of $n = 2^{11}$ and $t = 16$

$\mathsf{wt}(e)$	Timings for expected attack equation	Timings for random type equation
4	30141892×10^{-6}	304856×10^{-4}
6	3072799×10^{-5}	310234×10^{-4}
8	31597171×10^{-6}	32242382×10^{-6}
10	3285724×10^{-6}	3345847×10^{-5}

Remarks: We didn't give the timings for the odd values as they are constant and independant from the linear combinations between the permutations of the error positions. From Figure 3, we observe that there's a slight difference between the attack on this type combinations and on randomly distributed combinations. As we mentioned before for the random combinations those with the maximum number of iterations are more likely to appear (the case when all coefficients are different from zero). So in this case, we have the timing difference required for our attack to succeed. In Section 3.7, we will explain how the patch will work not only on this type of attacks but even on other types as the bit-flipping attacks.

3.7 Countermeasures

We have seen that it is possible to attack a system by knowing how many times the EEA is repeated. The number of iterations can go from 0 to $t-1$ in the syndrome inversion and from 0 to $t/2$ in the ELP determination. In order to avoid a correlation-finding from the number of iterations, we propose to introduce extra iterations into the EEA. The number of extra iterations should be chosen between 0 and a value that we call *extra*. The *extra* value is either $t/2$ or $t-1$, for the syndrome inversion ❸ or the ELP determination ❺, respectively. The variable i contains the number of iterations realized in the first part of the secured EEA. We chose to use integer values in the extra EEA steps, in order to avoid divisions by zero that may occur if we keep the previous terms. The point is to keep computing things that are as computationally expensive as the original EEA, so that an attacker can't make the difference between true steps and extra steps. We present the proposed secured modified EEA:

Input: $f(x)$, $g(x)$, d_{break} and t.
Output: $a(x)$ and $b(x)$ s.t. $a(x) \equiv b(x)f(x) \mod g(x)$

1. $d \leftarrow d_{break}$
2. $[b_{-1}, b_0] \leftarrow [0, 1]$
3. $[r_{-1}, r_0] \leftarrow [g(x), f(x)]$
4. $i \leftarrow 0$
5. While $\deg(r_i) > d$ do

$$\begin{array}{rcl} i & \leftarrow & i+1 \\ r_{i-2}(x) & = & r_{i-1}(x)q_i(x) + r_i(x) \\ b_i(x) & \leftarrow & b_{i-2}(x) + q_i(x)b_{i-1}(x) \end{array}$$

   end while

6. $a(x) \leftarrow r_i(x)$
7. $b(x) \leftarrow b_i(x)$

8. $extra = f(t, d_{break})$
9. While $i < extra$ do

$$\begin{array}{rcl} i & \leftarrow & i+1 \\ r_{i-2}(x) & = & 3q_i(x)+5 \\ b_i(x) & \leftarrow & 5+6q_i(x) \end{array}$$

end while

The new security parameters: We recall the fact that normal security parameters do not take in consideration timing attacks. Usually security parameters are given under the assumption of possible naive attacks or known structural attacks as ISD [9] For example for the McEliece PKC the usual parameters are:

$$\begin{array}{c} \text{100-bit security } n = 2048,\ t = 50 \\ \text{128-bit security } n = 2960,\ t = 56 \\ \text{256-bit security } n = 6624,\ t = 115 \end{array}$$

For the first parameters a timing attack with $p = 6$ would reveal a complexity less than 2^{61} elementary operations, that is way lower than the original security level proposals. So for timing attacks larger parameters have to be taken in consideration in order to maintain the same level of security. For example in order to same a 100 bit security lever against this type of timing attacks one should propose $n = 131072$.

The usual solution is not to increase the values of the parameters but to propose secure variant of the algorithm, variant that is not vulnerable to the specified attack. Our proposal is less faster that the original algorithm, it operates $(t-1) \times O(1)$ for the syndrome inversion and $\frac{t}{2} \times O(1)$ for the key equation (where $O(1)$ is the usual complexity for a division).

Meanwhile it is secure against timing attacks described below. The proof is very simple and it based on the fact that this particular type of timing attacks are based on the number of iterations is the EE Algorithm. Since our algorithm performs the same number of iterations no matter what relations are hidden between the polynomial coefficients it can't reveal any of such secret relations.

Once the countermeasure was applied, we ran the same attack and got the following timings for selected Hamming weights (the average timings are presented for more than 10^7 simulations in Fig. 4).

Table 4: Timings (in sec.) for decryption in the case of $n = 2^{11}$ and $t = 10$

$\mathsf{wt}(e)$	Timings for attack type equations	Timings for a random type combination
6	57.99	57.90
7	57.89	
8	57.94	58.03
9	58.33	
10	57.81	57.89

Remark: We observe that the protected implementation is impossible to attack (using the same techniques). We stress that the proposed countermeasure is also efficient in the case when an attacker wants to use previous techniques, like in [14, 16, 18].

4 Timing attack against the ELP evaluation

Goal: The attacker's goal is to find the secret permutation Π.

Identification of a leakage: A leakage is identified at step ❼ of the Patterson algorithm: the ELP evaluation. We recall that the ELP is denoted σ in Subsection 2.1. The attack is based on the fact that the form of the polynomial differs from the element to decode. We will prove that the algorithm's complexity is strongly related to the coefficients of $\sigma(x)$. We will then perform a timing attack on the ELP evaluation and control the values of the coefficients of $\sigma(x)$.

Motivations of our attack: One of the main motivations of our attack is that it can operate on all existing implementations of a general alternant decoder. It operates on the ELP evaluation, step that has to be computed in any decoding algorithm.

We will give two basic algorithms for the ELP evaluation with some improvements and show that even with the published improvements our attack succeeds. We will choose the polynomial evaluation from right to left (the naive algorithm) and from left to right (the Ruffini-Horner scheme). Imagine that our polynomial has a degree equal to an integer t. The first algorithm computes the result within $3t-1$ operations (t additions and $2t-1$ multiplications). As for the second one it computes the result within $2t$ operations (t additions and t multiplications). It was proven by V. Pan in 1966 [11] that the Ruffini-Horner's scheme [6] is optimal in terms of complexity.

The main idea of the improvement is to use the fact that some support elements have particular properties (like 0 and 1). Knowing the fact that one coefficient equals zero fasten up the algorithm as operations like multiplication or sum have fix values if they take zero as one of the input element. The same thing happens within the multiplication by one. So we will exploit these properties in order to improve our implementation. Each time a coefficient equals one or zero it will be store in a special table used afterwards for multiplication or addition. The case where a coefficient equals zero is rare and its probability has been studied in [4].

Nevertheless, each time there's a coefficient equal to zero we will no longer multiply it by the corresponding element as the multiplication equals zero. So we will use the predefined tables to get rid of the useless operations. We will proceed exactly the same way when the multiplication of an element has to be done when a coefficient equals to one. So each time we have one coefficient equal to zero, using our predefined tables, we get rid of two operations (one addition and one multiplication).

Scenario: The attack scenario is the same as in the previous attack except for the last step. In fact, the attacker gets the running time for the ELP evaluation in this section (step ❼ in Figure 1).

Idea: For $\mathsf{wt}(e)=2$, the attacker will find the positions of $\Pi(0)$ and $\Pi(1)$ the permutation of zero and one. After enough iterations, he will fix those two positions and repeat this attack with $\mathsf{wt}(e)=3$, he will then find the secret permutation Π (using exhaustive search for the remaining positions).

Conditions: The assumptions are the same as in the previous attack excepted that the attacker does not know the order of the elements in the support $\mathcal{L}$.

4.1 Success probability

As we said, in this attack we will only consider polynomials with a degree lower than three. For the case $r = 3$ we will give the full table of probabilities. We will start with the following general problem:

Problem: Let $P(x)$ be a monic polynomial of degree r with r distinct roots over $\mathbb{F}_{2^m}$. What is the probability that all its coefficients are different from zero?

This problem was treated in [4] and the results show that the probability can be bounded. For the classical parameters of the McEliece PKC, i.e. $n = 2048$ and $t \leq 50$, the authors obtain:

$$\mathcal{P} \geq 0.95$$

Proposition 5. *Let $P(x)$ be a monic polynomial of degree 3 with three distinct roots over $\mathbb{F}_{2^m}$ and $m \mod 2 = 1$.*

The probability $\mathcal{P}_3$ that all its coefficients are different from zero satisfies:

$$\mathcal{P}_3 = 1 - \frac{5}{2^m}.$$

4.2 Finding the permutation of the support elements zero and one

1. Consider the error-vectors e_i with $\mathsf{wt}(e_i) = 1$.

   In this case, the error locator polynomial has the following form:

   $$\sigma(x) = x + a_i, \text{ with } a_i \in \mathcal{L} = \{0, 1, \alpha, \ldots, \alpha^{n-2}\}.$$

   If $a_i \neq 0$, there is one addition $(+)$ in the $\sigma(x)$ evaluation.

2. Consider the error-vectors e_i with $\mathsf{wt}(e_i) = 2$.

   In this case, the error locator polynomial has the following form:

   $$\sigma(x) = x^2 + S_1^2 x + S_0^2, \text{ with } S_1^2 = a_i + a_j \text{ and } S_0^2 = a_i a_j.$$

   We distinguish two possible cases:

   (a) $\sigma(x) = x^2 + S_1^2 x + S_0^2$ if $a_i a_j \neq 0$

   (b) $\sigma(x) = x^2 + S_1^2 x$ if $a_i a_j = 0$

   The case (b) leads to a computation of the polynomial evaluation with one extra addition $(+)$ and the timings reveal all the couples $(\alpha_i, 0)$. We can assume now that the position of $\Pi(0)$ is known.

3. We fix this position and we seek for the position of $\Pi(1)$. Since the polynomial $\sigma(x) = x^2 + S_1^2 x$, the fastest evaluation is obtained for the couple $(\Pi(0), \Pi(1))$ as there is only one addition $(+)$ and one square computation.

4.3 Attack scenario when $r = 3$

We will consider error-vectors with Hamming weight that equals 3. The corresponding $\sigma(x)$ polynomial has always one of the eight following representations:

1. $\sigma(x) = x^3 + S_2^3 x^2 + S_1^3 x + S_0^3$ if $S_1^3 S_2^3 S_0^3 \neq 0$
2. $\sigma(x) = x^3 + S_2^3 x^2 + S_1^3 x$ if $S_0^3 = 0$ and $S_2^3 S_1^3 \neq 0$
3. $\sigma(x) = x^3 + S_2^3 x^2 + S_0^3$ if $S_1^3 = 0$ and $S_2^3 S_0^3 \neq 0$
4. $\sigma(x) = x^3 + S_1^3 x + S_0^3$ if $S_2^3 = 0$ and $S_1^3 S_0^3 \neq 0$
5. $\sigma(x) = x^3 + S_2^3 x^2$ if $S_2^3 \neq 0$ and $S_1^3 = 0$ and $S_0^3 = 0$
6. $\sigma(x) = x^3 + S_1^3 x$ if $S_1^3 \neq 0$ and $S_2^3 = 0$ and $S_0^3 = 0$
7. $\sigma(x) = x^3 + S_0^3$ if $S_0^3 \neq 0$ and $S_2^3 = 0$ and $S_1^3 = 0$
8. $\sigma(x) = x^3$ if $S_0^3 = 0$ and $S_2^3 = 0$ and $S_1^3 = 0$

Straightforward we deduce the following cases:

$$
\begin{array}{ll}
(a).\ \sigma(x) = x^3 + S_2^3 x^2 + S_1^3 x + S_0^3 & \text{if } S_1^3 S_2^3 S_0^3 \neq 0 \text{ and } \mathcal{P} = \frac{n-5}{n} \\
(b).\ \sigma(x) = x^3 + S_2^3 x^2 + S_1^3 x & \text{if } S_0^3 = 0 \text{ and } S_1^3 S_2^3 \neq 0 \text{ and } \mathcal{P} = \frac{3}{n} \\
(c).\ \sigma(x) = x^3 + S_2^3 x^2 + S_0^3 & \text{if } S_1^3 = 0 \text{ and } S_1^3 S_0^3 \neq 0 \text{ and } \mathcal{P} = \frac{1}{n} \\
(d).\ \sigma(x) = x^3 + S_1^3 x + S_0^3 & \text{if } S_2^3 = 0 \text{ and } S_1^3 S_0^3 \neq 0 \text{ and } \mathcal{P} = \frac{1}{n}
\end{array}
$$

Several cases can be eliminated by considering the fact that we accomplished the first step and we know the position of $\Pi(0)$. If we consider all the error-vectors where $a_i \neq 0 \quad \forall i \in \{1, 2, \ldots, n-1\}$ (i.e. 0 is not a root of $P(x)$), we reduce the possibilities for $\sigma(x)$. The new form of the system is the following:

$$
\begin{cases}
\sigma(x) = x^3 + S_2^3 x^2 + S_1^3 x + S_0^3 & \text{if } S_1^3 S_2^3 S_0^3 \neq 0 \\
\sigma(x) = x^3 + S_2^3 x^2 + S_0^3 & \text{if } S_1^3 = 0 \text{ and } S_2^3 S_0^3 \neq 0 \\
\sigma(x) = x^3 + S_1^3 x + S_0^3 & \text{if } S_2^3 = 0 \text{ and } S_0^3 S_1^3 \neq 0
\end{cases}
$$

In all cases, x^3 must be computed so we will not consider this part in the timing differences. In the structure that computes the polynomial evaluation the fastest is the last one. But this case is performed only when $S_2^3 = 0$.

4.4 Finding the positions of two elements such that $\Pi(\alpha_j)\Pi(\alpha_k) = 1$

In order to increase the number of equations in our system, we exploit the fact that $(\mathbb{F}_{2^m})^*$ is cyclic.

Recall: we know the positions of $\Pi(0), \Pi(1)$ and $\Pi(\alpha_1) + \Pi(\alpha_2) + \Pi(\alpha_3) = 0$. Without loss of generality, we choose to fix "$\Pi(0)$" on the first position and choose two other positions such that the sum is different from 1.

We are able to do that because we know the position of "$\Pi(1)$" and the couples (α_1, α_2) such that $1 + \alpha_1 + \alpha_2 = 0$. We get two new positions b_1 and b_2 such that $b_1 + b_2 \neq 1$. The error locator polynomial is: $\sigma(x) = x^3 + S_2^3 x^2 + S_1^3 x$.

For $b_1 b_2 = 1$ we get $\sigma(x) = x^3 + S_2^3 x^2 + x$. This form is the fastest to be computed as there is one less multiplication compare to the other case.

4.5 System resolution

Number of equations:

We will give the number of linear and quadratic equations obtained by the attacker. Finding the positions of $\Pi(0)$ and $\Pi(1)$ reduces the search set to $(n-2)$ elements.

- The first set of linear equations:

$$\text{Equation type (1): } \Pi(\alpha_j)\Pi(\alpha_k) = 1 \Rightarrow \sharp eq. = \tfrac{n-2}{2}$$

The last equation is determined by all the other ones because for the last couple only one possible solution remains available. For instance, if the attacker finds $(\frac{n-2}{2} - 1)$ different equations the last equation can be directly determined.

- The second set of linear equations:

$$\text{Equation type (2): } \Pi(\alpha_j) + \Pi(\alpha_k) = 1 \Rightarrow \sharp eq. = \tfrac{n-2}{2}.$$

As the first set, the last one can be determined by all others. This comes from the fact that for the three positions, we fixed the position of $\Pi(1)$ as the first one. So we have $(n-2)$ possibilities on the second position. But there are two repetitions for each $(\Pi(1), \Pi(\alpha_j), \Pi(\alpha_k))$-vector.

- The third set of quadratic equations:

$$\text{Equation type (3): } \Pi(\alpha_i) + \Pi(\alpha_j) + \Pi(\alpha_k) = 0 \Rightarrow \sharp eq. = \tfrac{(n-2)(n-4)}{6}$$

The total number of equations for $\Pi(\alpha_i) + \Pi(\alpha_j) + \Pi(\alpha_k) = 0$ including the second set equals $(n-1)(n-2)$ as the third position is fixed and the two others are free and different. Here, the number of repetitions equals six. So we obtain $\left(\frac{(n-2)(n-1)}{6} - \frac{n-2}{2}\right)$ equations.
To illustrate how the attack works a toy example is given in Appendix 2.

Conclusion

In this article, we focused our attention on the cryptanalysis of the McEliece PKC with the binary Goppa codes. We showed the existing weak points in the Patterson decoding algorithm and determined the relations between the number of iterations in two different steps of the algorithm and the secret permutation. Since those relations were the main connection idea between the two extended Euclidean algorithms, we set up a timing attack based on this fact. The advantage of this attack is that it increased the probability of success by avoiding ambiguous cases, undetectable in previous attacks. The other advantage is that it allows higher expansion of the number of equations determined by the attacker in order to find the secret permutation.

The second important contribution of our article is a new attack that can be performed on several different decoding algorithms. It reveals that even intelligent variants of some polynomial evaluation algorithms might leak information and need to be patched or replaced. The ideas discovered in the attacks might be reused in any further implementations using the algorithms mentioned before. So secure variants must be used in order to avoid any leakage point.

Bibliography

[1] Roberto Avanzi, Simon Hoerder, Dan Page, Mike Tunstall (2010), Side-channel attacks on the McEliece and Niederreiter public-key cryptosystems, *Cryptology ePrint Arch.*, Report 2010/479, 2010.

[2] Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen (eds.) (2009), *Post-Quantum Cryptography*, Springer, 2009.

[3] Daniel J. Bernstein, Tung Chou, Peter Schwabe (2013), McBits: fast constant-time code-based cryptography, https://binary.cr.yp.to/mcbits-20130616.pdf, 1-26.

[4] Vlad Dragoi, Pierre-Louis Cayrel, Brice Colombier, Tania Richmond (2013), Polynomial structures in code-based cryptography, *Indocrypt 2013*, LNCS2850: 286-296.

[5] Whitfield Diffie, Martin Hellman (1976), New directions in cryptography, *IEEE Trans. Inform. Theory*, 22(6):644–654.

[6] W.G. Horner (1819), A new method of solving numerical equations of all orders by continuous approximation, *Phil. Trans. R. Soc. Lond.*, 109:308–335.

[7] Florence J. MacWilliams, Neil J. A. Sloane (1986), *The Theory of Error-Correcting Codes*, North–Holland, Amsterdam, 5th ed., 1986.

[8] Robert J. McEliece (1978), A public-key cryptosystem based on algebraic coding theory, *Jet Propulsion Laboratory DSN Progress*, Report 42-44, 114–116.

[9] Robert Niebuhr et al. (2010), On lower bounds for information set decoding over fq, In *C. Cid, J.-C. Faugere, (eds.), Proc. of the Second Intl. Conf. on Symbolic Computation and Cryptography, SCC 2010*, 143–157.

[10] Ayoub Otmani, Jean-Pierre Tillich, Leonard Dallot (2008), Cryptanalysis of a McEliece cryptosystem based on quasi-cyclic LDPC codes, *Proc. of First Intl. Conf. on Symbolic Computation and Cryptography (SCC 2008)*, 69–81.

[11] Victor Y. Pan (1966), On Methods of Computing the Values of Polynomials, *UspeKhi Mathematicheskikh Nauk*, 21:103–134.

[12] Nicholas J. Patterson (1975), The algebraic decoding of goppa codes, *IEEE Transactions on Information Theory*, 21(2): 203–207.

[13] Peter W. Shor (1997), Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Journal on Computing*, 26(5):1484-1509.

[14] Abdulhadi Shoufan et al. (2010), A Timing Attack against Patterson Algorithm in the McEliece PKC, *ICISC 2009*, LNCS 5984: 161–175.

[15] V.M. Sidelnikov and S.O. Shestakov (1992), On the insecurity of cryptosystems based on generalized Reed-Solomon codes, *Discrete Math. Appl.*, 2(4):439–444.

[16] Falko Strenzke (2010), A Timing Attack against the Secret Permutation in the McEliece PKC, In *N. Sendrier (ed.), Post-Quantum Cryptography, Third intl. workshop*, LNCS6061: 95–107.

[17] Falko Strenzke (2010), Fast and secure root-finding for code-based cryptosystems, *Cryptology ePrint Arch.*, Report 2011/672, 2011.

[18] Falko Strenzke (2011), Timing attacks against the syndrome inversion in code-based cryptosystems, *Cryptology ePrint Arch.*, Report 2011/683, 2011.

[19] Falko Strenzke et al. (2008), Side channels in the McEliece PKC, In *J. Buchmann and J. Ding (eds.), Post-Quantum Cryptography, Second intl. workshop*, LNCS5299: 216–229.

Appendix

1 Toy example for the EEA attack

Consider $\mathbb{F}_{2^4}[x] = \frac{\mathbb{F}_2[x]}{x^4+x+1}$. The generator matrix $\mathcal{G}$ of the Goppa code and the support $\mathcal{L} = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{14}\}$ are public. Let $\mathsf{m} \in \mathbb{F}_2^k$ be the message and $\mathcal{O}$ the decoding oracle. We notice that if $\mathcal{L}$ is public, one can find $G(x)$ such that $\mathcal{L} = \frac{\mathbb{F}_2[x]}{G(x)}$. The other way is equaly true: if $G(x)$ is public then one can easily find $\mathcal{L}$. Suppose that the secret permutation is:

$$\Pi(\mathcal{L}) = \mathcal{L}' = \{\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{14}, 0, 1\} = \{\ell_i \,|i \in (1 \ldots 16)\}$$

- 1^{st} *step:*
  - The attacker asks $\mathcal{O}$ to decode all the $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$ with $\mathsf{wt}(e) = 1$.
  - ⋆ $N_{❸}$ and $N_{❺}$ reveals the position of $\Pi(0)$: ℓ_{15}.
  - ◦ This is mainly due to: $\sigma(x) = x$ we have $\tau(x) = 0$ and $S^{-1}(x) = x$
- 2^{nd} *step:*
  - The attacker asks $\mathcal{O}$ to decode all the $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$ with $\mathsf{wt}(e) = 4$ (the positions $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3})$ are the three non-zero positions of e and $\ell_{i_4} = \ell_{15}$).
  - ⋆ The couple $\begin{pmatrix} N_{❸} \\ N_{❺} \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$ reveals all $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3})$ such that $\ell_{i_1} + \ell_{i_2} + \ell_{i_3} = 0$.
    Here $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3}) \in \{(\ell_1, \ell_4, \ell_{16}), (\ell_3, \ell_{14}, \ell_{16}), \ldots\}$.
  - ◦ $\deg(\sigma) = 4$ and $\deg(\omega) = \begin{cases} 2 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} \neq 0 \\ 0 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} = 0 \end{cases}$
  - ◦ $\deg(b) = \begin{cases} 1 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} \neq 0 \\ 0 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} = 0 \end{cases}$
- 3^{rd} *step:*
  - The attacker asks $\mathcal{O}$ to decode all the $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$ with $\mathsf{wt}(e) = 4$ (the positions $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3}, \ell_{i_4})$ are the four non-zero positions of e).
  - ⋆ The couple $\begin{pmatrix} N_{❸} \\ N_{❺} \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$ reveals all $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3}, \ell_{i_4})$ such that $\ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \ell_{i_4} = 0$.
    Here $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3}, \ell_{i_4}) \in \{(\ell_1, \ell_2, \ell_{10}, \ell_{16}), (\ell_2, \ell_3, \ell_{13}, \ell_{16}), \ldots\}$.
  - ◦ $\deg(\sigma) = 4$ and $\deg(\omega) = \begin{cases} 2 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \ell_{i_4} \neq 0 \\ 0 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \ell_{i_4} = 0 \end{cases}$
  - ◦ $\deg(b) = \begin{cases} 1 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \ell_{i_4} \neq 0 \\ 0 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \ell_{i_4} = 0 \end{cases}$
- 4^{th} *step:*
  - The attacker asks $\mathcal{O}$ to decode all the $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$ with $\mathsf{wt}(e) = 6$ (the positions $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3}, \ell_{i_4}, \ell_{i_5})$ are the five non-zero positions of e and $\ell_{i_6} = \ell_{15}$).
  - ⋆ The couple $\begin{pmatrix} N_{❸} \\ N_{❺} \end{pmatrix} = \begin{pmatrix} 8 \\ 1 \end{pmatrix}$ reveals all $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3}, \ell_{i_4}, \ell_{i_5})$ such that $\ell_{i_1} + \ell_{i_2} + \cdots + \ell_{i_5} = 0$.
    Here $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3}, \ell_{i_4}, \ell_{i_5}) \in \{(\ell_1, \ell_2, \ell_3, \ell_{12}, \ell_{16}), (\ell_3, \ell_4, \ell_8, \ell_{12}, \ell_{16}), \ldots\}$.

$\circ$ $\deg(\sigma) = 4$ and $\deg(\omega) = \begin{cases} 4 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \ell_{i_4} + \ell_{i_5} \neq 0 \\ 2 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \ell_{i_4} + \ell_{i_5} = 0 \end{cases}$

$\circ$ $\deg(b) = \begin{cases} 2 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \ell_{i_4} + \ell_{i_5} \neq 0 \\ 1 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \ell_{i_4} + \ell_{i_5} = 0 \end{cases}$

- 5^{th} *step:*
  - The attacker asks $\mathcal{O}$ to decode all the $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$ with $\mathsf{wt}(e) = 6$ (the positions $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3}, \ell_{i_4}, \ell_{i_5}, \ell_{i_6})$ are the six non-zero positions of e).
  - $\star$ The couple $\binom{N_{\textcircled{8}}}{N_{\textcircled{5}}} = \binom{8}{1}$ reveals all $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3}, \ldots, \ell_{i_6})$ such that $\ell_{i_1} + \ell_{i_2} + \cdots + \ell_{i_6} = 0$.
    Here $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3}, \ldots, \ell_{i_6}) \in \{(\ell_1, \ell_2, \ell_3, \ell_4, \ell_6, \ell_{16}), \ldots\}$.
  - $\circ$ $\deg(\sigma) = 4$ and $\deg(\omega) = \begin{cases} 4 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \cdots + \ell_{i_6} \neq 0 \\ 2 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \cdots + \ell_{i_6} = 0 \end{cases}$
  - $\circ$ $\deg(b) = \begin{cases} 2 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \cdots + \ell_{i_6} \neq 0 \\ 1 & \text{if } \ell_{i_1} + \ell_{i_2} + \ell_{i_3} + \cdots + \ell_{i_6} = 0 \end{cases}$
- $\ldots$
- *Last step:* The attacker has to solve the following system of quadratic equations in order to find the secret permutation:

$$\begin{cases} \ell_{15} = \Pi(0)\ ; & 1^{st}\ step \\ \ell_1 + \ell_4 + \ell_{16} = \ell_3 + \ell_{14} + \ell_{16} = \cdots = 0 & 2^{nd}\ step \\ \ell_1 + \ell_2 + \ell_{10} + \ell_{16} = \ell_2 + \ell_3 + \ell_{13} + \ell_{16} = \cdots = 0 & 3^{rd}\ step \\ \ell_1 + \ell_2 + \ell_3 + \ell_{12} + \ell_{16} = \ell_3 + \ell_4 + \ell_8 + \ell_{12} + \ell_{16} = \cdots = 0 & 4^{th}\ step \\ \ell_1 + \ell_2 + \ell_3 + \ell_4 + \ell_6 + \ell_{16} = \cdots = 0 & 5^{th}\ step \\ \ldots \end{cases}$$

Solving the system will allow to fully determine the secret permutation

$$\Pi(\mathcal{L}) = \mathcal{L}' = \{\alpha, \alpha^2, \alpha^3, \alpha^4, \ldots, 0, 1\}.$$

2 Toy example for the ELP evaluation attack

Consider $\mathbb{F}_{2^3}[x] = \frac{\mathbb{F}_2[x]}{x^3+x+1}$. $\mathcal{G}$ and the support $\mathcal{L} = \{0, 1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6\}$ are public, $\mathsf{m} \in \mathbb{F}_2^k$ and $\mathcal{O}$ is the decoding oracle. We notice that if $\mathcal{L}$ is public one can find $G(x)$ such that $\mathcal{L} = \frac{\mathbb{F}_2[x]}{G(x)}$. The other way is equaly true: if $G(x)$ is public then one can easily discover $\mathcal{L}$. Suppose that the secret permutation is:

$$\Pi(\mathcal{L}) = \mathcal{L}' = \{\alpha, \alpha^3, 1, \alpha^4, \alpha^5, 0, \alpha^2, \alpha^6\} = \{\ell_i \mid i \in \{1, \ldots, 8\}\}$$

- 1^{st} *step:*
  - The attacker asks $\mathcal{O}$ to decode all the $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$ with $\mathsf{wt}(e) = 2$ (the positions (ℓ_j, ℓ_k) are the two non-zero positions of e).

  - ⋆ The faster step ❼ reveals the position of $\Pi(0)$: ℓ_6.
  - The attacker asks $\mathcal{O}$ to decode all the $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$ with $\mathsf{wt}(e) = 2$ (the positions (ℓ_6, ℓ_k) are the two non-zero positions of e).
  - ⋆ The faster step ❼ reveals the position of $\Pi(1)$: ℓ_3.

- 2^{nd} *step:*
  - The attacker asks $\mathcal{O}$ to decode all the $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$ with $\mathsf{wt}(e) = 3$ (the positions (ℓ_3, ℓ_j, ℓ_k) are the three non-zero positions of e).
  - ⋆ The faster step ❼ reveals all the couples (ℓ_j, ℓ_k) such that $\ell_3 + \ell_j + \ell_k = 0$. Here $(\ell_j, \ell_k) \in \{(\ell_1, \ell_2), (\ell_4, \ell_5), (\ell_7, \ell_8)\}$.

- 3^{rd} *step:*
  - The attacker asks $\mathcal{O}$ to decode all the $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$ with $\mathsf{wt}(e) = 3$ (the positions (ℓ_6, ℓ_j, ℓ_k) are the three non-zero positions of e).
  - ⋆ The faster step ❼ reveals all the couples (ℓ_j, ℓ_k) such that $\ell_j \ell_k = 1$. Here $(\ell_j, \ell_k) \in \{(\ell_1, \ell_8), (\ell_2, \ell_4), (\ell_5, \ell_7)\}$.

- 4^{th} *step:*
  - The attacker asks $\mathcal{O}$ to decode all the $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$ with $\mathsf{wt}(e) = 3$ (the positions (ℓ_i, ℓ_j, ℓ_k) are the three non-zero positions of e).
  - ⋆ The faster step ❼ reveals all the triplets (ℓ_i, ℓ_j, ℓ_k) such that $\ell_i + \ell_j + \ell_k = 0$. Here $(\ell_i, \ell_j, \ell_k) \in \{(\ell_1, \ell_4, \ell_7), (\ell_1, \ell_5, \ell_8), (\ell_2, \ell_4, \ell_8), (\ell_2, \ell_5, \ell_7)\}$.
  - The attacker has to solve the following system of quadratic equations in order to find the secret permutation:

$$\begin{cases} \ell_6 = \Pi(0)\,; \ell_3 = \Pi(1) & 1^{st}\ step \\ \ell_1 + \ell_2 = \ell_4 + \ell_5 = \ell_7 + \ell_8 = 1 & 2^{nd}\ step \\ \ell_1\ell_8 = \ell_2\ell_4 = \ell_5\ell_7 = 1 & 3^{rd}\ step \\ \ell_1 + \ell_4 + \ell_7 = \ell_1 + \ell_5 + \ell_8 = 0 & 4^{th}\ step \\ \ell_2 + \ell_4 + \ell_8 = \ell_2 + \ell_5 + \ell_7 = 0 & 4^{th}\ step \end{cases}$$

Solving the system will allow to fully determine the secret permutation $\Pi(\mathcal{L}) = \{\alpha, \alpha^3, 1, \alpha^4, \alpha^5, 0, \alpha^2, \alpha^6\}$.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(1):26-40, February 2017.

A Similarity Measure-based Optimization Model for Group Decision Making with Multiplicative and Fuzzy Preference Relations

X.R. Chao, G. Kou, Y. Peng

Xiangrui Chao, Yi Peng*
School of Management and Economics
University of Electronic Science and Technology of China
Chengdu, China, 610054
*Corresponding author: pengyi@uestc.edu.cn

Gang Kou
School of Business Administration
Southwestern University of Finance and Economics
Chengdu, China, 610074
⋆ Authors are alphabetically ordered and contribute equally to this work.

Abstract: Group decision making (GDM) problem based on different preference relations aims to obtain a collective opinion based on various preference structures provided by a group of decision makers (DMs) or experts, those who have varying backgrounds and interests in real world. The decision process in proposed question includes three steps: integrating varying preference structures, reaching consensus opinion, selecting the best alternative. Two major approaches: preference transformation and optimization methods have been developed to deal with the issue in first step. However, the transformation processes causes information lose and existing optimization methods are so computationally complex that it is not easy to be used by management practice. This study proposes a new consistency-based method to integrate multiplicative and fuzzy preference relations, which is based on a cosine similarity measure to derive a collective priority vector. The basic idea is that a collective priority vector should be as similar per column as possible to a pairwise comparative matrix (PCM) in order to assure the group preference has highest consistency for each decision makers. The model is computationally simple, because it can be solved using a Lagrangian approach and obtain a collective priority vector following four simple steps. The proposed method can further used to derive priority vector of fuzzy AHP. Using three illustrative examples, the effectiveness and simpleness of the proposed model is demonstrated by comparison with other methods. The results show that the proposed model achieves the largest cosine values in all three examples, indicating the solution is the nearest theoretical perfectly consistent opinion for each decision makers.
Keywords:group decision making; multiplicative preference relations; fuzzy preference relations; similarity measure; optimization model.

1 Introduction

Group decision making (GDM) aims to obtain a solution alternative(s) to a given question based on the opinions provided by a set of experts. When comparing alternatives in real word problem, experts may use any of the following preference structures: preference orderings, utility functions, multiplicative preference relations, and fuzzy preference relations (Herrera et al.[1], Herrera-Viedma et al.[2]). The process is composed of the three parts. 1) Integrating the different preference structures provided by varying decision makers [1,2,3,4,5,6,7,8,9]. 2) Reaching consensus based on modified the preference information when the collective opinion cannot be

accepted by non-cooperative decision makers [10,11,12,13,14]. 3) Selecting the best alternative from the collective opinion.

In first step of previous question, two main groups of techniques have been developed for integrate different preference relations in GDM problem. The first class of techniques (Chiclana et al.[3,5]; Delgado et al.[6]; Herrera et al.[1]; Herrera-Viedma[2]) transforms different preference structures into uniform formats, after which a selection operator is implemented to rank alternatives on the basis of a fuzzy majority. The second class of techniques (Fan et al.[7]; Ma et al.[8]; Xu et al.[9]) uses optimization models to directly obtain a collective priority vector instead of using transform functions to obtain uniform preference information. The limitations of these two classes of approaches are: 1) the transformation functions may cause information loss during the conversions, and 2) existing optimization methods are so complex to obtain solution that it is not easy to be used to management practice.

The goal of this study is to propose a cosine similarity measure-based optimization method for deriving a collective priority vector in decision making problem with multiplicative preference relations and fuzzy preference relations. The basic idea of the model is that the collective priority vector should be as similar per column as possible to a pairwise comparative matrix (PCM) from each decision maker so that the solution is the nearest theoretical perfectly consistent opinion for each decision makers. The proposed model can be implemented in four steps with simple computation for integrating multiplicative preference relations and fuzzy preference relations. Moreover, the proposed model can also be used to derive priority vector from PCM in fuzzy AHP. Furthermore, the model can be extended to other preference structures by transforming other preference formats into a mixture of multiplicative preference relations and fuzzy preference relations, and used the consensus reaching methods by iterative modification based on our proposed priority vector obtaining method to improve consensus degree (such as Chen et al.[15];Wu and Xu[16]; Dong et al[17]; Xu et al[18,19], Yu et al.[20], etc.).

The remainder of this study is organized as follows. Section 2 reviews related work and provides some preliminaries. Section 3 describes the proposed cosine similarity measure-based optimization model. Section 4 uses three numeric examples to illustrate the implementation of the proposed model and compare with some well-known methods. Section 5 concludes the study.

2 Related work and preliminaries

2.1 Related work

The group decision making problem with different preference structures, which was also called multi-person decision making in their papers, has a long research history and been defined as a decision-making process by different DMs or experts with different decision information presented in terms of utility functions, preference orderings, or preference relations with multiplicative or fuzzy formats (Arrow [21]; Sen[22]; Kickert et al.[23]; Saaty[24]; Kacprzyk and Fedrizzi [25]; Chiclana et al.[3,5]; Herrera et al.[1], etc.). A utility function comprises real numbers for alternatives to represent the utility evaluations provided by DMs. A preference ordering is a ranking of alternatives in best-to-worst order, with no additional information. A preference relation is based on pairwise comparison. It is a simple process in which experts provide ratios that compare one alternative to another. For a multiplicative preference relation, the ratio can be a real number in the set {1/9, 1/8, ..., 1/2, 1, 2,...,8 ,9} . For a fuzzy preference relation, the ratio is provided as a fuzzy number.

An important issue in this problem is the derivation of a collective opinion on the basis of different preference formats provided by each decision maker (Chiclana et al.[3,5]; Delgado et al.[6]; Herrera et al.[1]; Herrera-Viedma[2]). Various approaches have been developed to solve

above problems in following steps: (1) convert different preference information into uniform structures, (2) aggregate uniform preference structures using selection operators, and (3) rank the alternatives and make a decision.

Delgado et al. [6] developed a fusion operator for integrating numerical and linguistic information using transformation functions between the numerical and linguistic values. Chiclana et al. [3] proposed a general model for integrating different preference formats using fuzzy preference relations. They aggregate uniform preference information using an ordered weighted averaging (OWA) geometric operator with a fuzzy quantifier. Chiclana et al.[5] discussed the properties of different preference formats and multiplicative preference relations and proposed transformation functions that can convert varying preference information into uniform representations and then used an OWA geometric operator to rank alternatives. The operator, which includes two degree choices, is a multi-criteria decision-making method (MCDM). The transformation functions proposed in Chiclana et al.[26] have been used in many studies as a benchmark for comparing different methods. Herrera et al.[1] defined a transformation function between the multiplicative and fuzzy preference relations. They also adopted fuzzy preference relations as a uniform preference format. The OWA geometric operator was also used to rank alternatives. Chiclana et al. [26] developed a fuzzy MCDM model to derive a collective opinion when preferences were presented using preference orderings, utility functions, and fuzzy preference relations and studied the internal consistency of the model. Herrera-Viedma et al. [2] proposed a consensus model for varying preference information in GDM questions. Their model can automatically obtain a consensus. Xu [27] studied different interval preference relations and proposed a model for deriving overall weights in multi-attribute decision making.

Another class of methods for obtaining a collective opinion is optimization-based. One strength of this kind of approach is that it can avoid information loss during the conversion of preference relations because there is no need to transform different preference formats into a uniform structure. Fan et al. [7] constructed a goal-programming model to reduce differences in collective opinions and each decision maker's PCMs, thereby demonstrating that the collective opinion must be close to perfectly consistent. Ma et al. [8] proposed an optimization model for integrating four different preference formats without unifying them into one structure. They analyzed perfectly consistent conditions in four preference relations and constructed an objective function on the basis of the idea that the derived priority vector has the least distance from each decision maker's preference. Xu et al.[9] introduced a nonlinear programming method for handling decision makers coming from different areas and providing varying preference information, including utility values, preference orderings, and multiplicative, incomplete multiplicative, fuzzy, and incomplete fuzzy preference relations. Wang et al. [28] proposed a chi-square model to derive priority methods wherein the goal is to develop an optimal priority vector as close to each decision maker's opinion as possible.

2.2 Preliminaries

The GDM problem with different preference formats can be divided into two stages; first aggregating the varying preference information and determining a collective group preference and then selecting a ranking of alternatives using a previous collective group preference.

Assume that $\Omega = \{A_1, A_2, ..., A_n\}$ is a finite set containing n alternatives and $\Pi = \{DM_1, DM_2, ..., DM_n\}$ is a finite set including K decision makers.$\Lambda = \{\sigma_1, \sigma_2, ..., \sigma_n\}$ is a finite set containing the degrees of importance of decision makers. In general, $\sum_{i=1}^{n} \sigma_i = 1$. with each σ_i being prespecified by decision makers. The collective group preference $w = \{w_1, w_2, ..., w_n\}$ indicates the degree of importance and best-to-worst selection order ranking the alternatives. The preference relations can be represented in different formats with orderings, utility values, and multiplicative

and fuzzy preference relations. Herein, we discuss the aggregation of multiplicative and fuzzy preference relations, and other preferences can be transformed into these two types of preference relations (Chiclana et al.[26]).

Multiplicative preference relations are the most widely used in decision-making situations. A multiplicative preference relation is essentially a binary relation $f = \Omega \times \Omega \to \{1/9, 1/8, \cdots, 8, 9\}$ represented as a PCM matrix $(a_{ij})_{n\times n}$, i.e. $f(A_i, A_j) \mapsto a_{ij}$.The entry $a_{ij}$of the PCM denotes the relative degree of importance of alternative A_i with respect toA_j. A_i and $A_j$are equally important if $a_{ij} = 1$, and a higher a_{ij} value indicates a greater weight for the alternative.When $a_{ij} = 9$, A_i is preferred absolutely to A_j, and A_i is inferior to every A_j if $a_{ij} = 1/9$.It is evident that the PCM of a multiplicative preference relation is positive reciprocal, i.e. $a_{ij} \times a_{ji} = 1$ for arbitrary $i, j \in \{1, 2, \cdots, n\}$. A PCM is called perfectly consistent if $a_{ij}a_{jk} = a_{ik}$for arbitrary $i, j, k \in \{1, 2, \cdots, n\}$.In practice, the PCM is not always perfectly consistent. Saaty [24][29] proposed the consistency ratio index, which must be less than 0.1 when consistency can be accepted. Aguarón et al. [30] and Escobar et al. [31] proposed a geometric consistency index for AHP. Many other methods for measuring consistency have also been proposed Ergu et al. [32]; Lin et al.[33,34]; Lin and Kou[35].etc). If consistency is present, there are many derivation methods for priority vectors, including the eigenvector method (Saaty [36]), weighted least squares method (Chu, Kalaba, and Spingarn [37]), logarithmic least squares method (Crawford and Williams [38]), a heuristic approach (Lin et al.[39]), and the cosine maximization method (Kou and Lin[40], etc). Moreover, Gomez-Ruiz [41] proposed a estimation method using neural network.

Fuzzy preference relations have been extensively studied (Tanino [42]; Chiclana et al.[3]; Herrera-Viedma et al.[43]) and have many practical applications in the decision-making process owing to the difficulty of comparing two alternatives using a crisp real number. A fuzzy preference relation is a binary fuzzy membership function $g : \Omega \times \Omega \to [0, 1]$, i.e., $g(A_i, A_j) \mapsto p_{ij}$, where p_{ij} is the fuzzy degree of importance of A_i with respect to A_j. The fuzzy pairwise comparative matrix $(p_{ij})_{n\times n}$ is constructed as a real number from 0 to 1. if $p_{ij} = 0.5$, A_i has the same preference as those of A_j, $p_{ij} = 1$ and $p_{ji} = 0$ indicates that A_i is unanimously preferred to A_j. $p_{ij} \in (0.5, 1)$ and $p_{ji} \in (0, 0.5)$ indicate that A_i is preferred over A_j. There is an additive reciprocal relation in a fuzzy pairwise comparative matrix $(p_{ij})_{n\times n}$, i.e., $p_{ij} + p_{ji} = 1$ for arbitrary $i, j \in \{1, 2, \cdots, n\}$. The definition of perfectly consistent for a fuzzy pair-wise comparative matrix is different from that for a multiplicative preference relation: $(p_{ij})_{n\times n}$ is considered to be perfectly consistent, if $p_{ij}p_{jk}p_{ki} = p_{ji}p_{kj}p_{ik}$ for arbitrary $i, j, k \in \{1, 2, \cdots, n\}$,

Herrera-Viedma et al.[43]summarized eight consistency properties of fuzzy preference relations and proposed a method for testing and improving the consistency of a fuzzy preference relation. Xu et al. [18] studied the ordinal consistency of a fuzzy preference relation in terms of three cycles in a directed graph. Zhang et al.[44] discussed consistency issues in group decision making with fuzzy preference relations. Mikhailov L. [45] introduced linear fuzzy preference programming for deriving a priority vector for a fuzzy preference relation. Xu and Da [46] proposed a least-deviation method for obtaining a priority vector of a fuzzy preference relation. Wang et al.[28] used a chi-square method for deriving a priority vector on the basis of fuzzy and multiplicative preference relations. Other methods for priority derivation of fuzzy AHP can be found in Kacprzyk[47], Roubens[48], and Chiclana et al.[3].

A similarity index in mathematics is used to measure the similarity of two vectors. Many similarity measurement indices, including similarity coefficient and distance function methods, have been introduced and employed in many domains, such as machine learning and decision science. The most widely used similarity measure is the cosine similarity index. For two vectors $\vec{r_i} = \{r_{i1}, r_{i2}, \cdots, r_{in}\}$ and $\vec{r_j} = \{r_{j1}, r_{j2}, \cdots, r_{jn}\}$ the cosine similarity measure is a binary

function defined as

$$\langle \vec{r_i}, \vec{r_j} \rangle = \frac{\vec{r_i} \cdot \vec{r_j}}{\|\vec{r_i}\| \|\vec{r_j}\|} = \frac{\sum_{k=1}^{n} r_{ik} r_{jk}}{\sqrt{\sum_{k=1}^{n} r_{ik}^2} \sqrt{\sum_{k=1}^{n} r_{jk}^2}} \in [0,1] \tag{1}$$

Kou and Lin [40] used this similarity measure for AHP and constructed a cosine maximization model to derive a priority vector in AHP. The idea is that the derived priority vector is to be most similar to each column of a PCM according to the cosine similarity measure.

Inspired by the work of Kou and Lin [40], this study extends the similarity measure method to group decision-making by applying the cosine similarity measure to the GDM problem with multiplicative and fuzzy preference relations.

3 Cosine similarity measure maximization model for GDM problem with multiplicative and fuzzy preference relations

Assume there are two classes of preference structures: multiplicative preference relations and fuzzy preference relations. let $(a_{ij}^{(k)})_{n\times n}$, $k = 1, 2, \cdots, k_m$ and $(p_{ij}^{(k)})_{n\times n}$, $k = k_{m+1}, k_{m+2}, \cdots, K$ be multiplicative and fuzzy preference relations, respectively.

Existing results show that the PCM is perfectly consistent when it satisfies the following conditions:

$$a_{ij}^{(k)} = \frac{w_i^k}{w_j^k}, i = 1, 2, \cdots, n; j = 1, 2, \cdots, n. \tag{2}$$

$$p_{ij}^{(k)} = \frac{w_i^k}{w_i^k + w_j^k}, i = 1, 2, \cdots, n; j = 1, 2, \cdots, n. \tag{3}$$

Let $\vec{a_j}^{(k)} = (a_{ij}), a_{2j}, \cdots, a_{nj})^T$ be the column vector of $(a_{ij}^{(k)})$ and $\vec{w} = (w_1, w_2, \cdots, w_n)^T$. Kou and Lin [40] proved that the existing similarity relation between each column of a PCM and a derived priority vector-particularly, the cosine similarity measure-is equal to one if and only if the PCM is perfectly consistent.

The higher the degree of consensus, the closer the cosine similarity measure is to 1. That is

$$\langle \vec{a_j}^{(k)}, \vec{w_j} \rangle = \frac{\sum_{k=1}^{n} a_{kj} w_k}{\sqrt{\sum_{k=1}^{n} a_{kj}^2} \sqrt{\sum_{k=1}^{n} w_k^2}} = \frac{\sum_{k=1}^{n} \frac{w_k}{w_j} w_k}{\sqrt{\sum_{k=1}^{n} \frac{w_k}{w_j}^2} \sqrt{\sum_{k=1}^{n} w_k^2}} = \frac{\sum_{k=1}^{n} w_k^2}{\sqrt{\sum_{k=1}^{n} w_k^2} \sqrt{\sum_{k=1}^{n} w_k^2}} = 1 \tag{4}$$

For fuzzy preference relations, we can show that there exists a similarity relation between a derived priority vector and each column of the PCM after the transformation:

$$b_{ij}^{(k)} = \frac{p_{ij}^{(k)}}{1 - p_{ij}^{(k)}} \tag{5}$$

Let $\vec{b_j} = (b_{1j}, b_{2j}, \cdots, b_{nj})^T, j = 1, 2, \cdots, n$. When the PCM is perfectly consistent, i.e., $b_{ij} = \frac{w_i}{w_j}$, the following relation can be obtained:

$$\langle \vec{b_j}^{(k)}, \vec{w_j} \rangle = \frac{\sum_{k=1}^{n} b_{kj} w_k}{\sqrt{\sum_{k=1}^{n} b_{kj}^2} \sqrt{\sum_{k=1}^{n} w_k^2}} = \frac{\sum_{k=1}^{n} \frac{p_{kj}}{1-p_{kj}} w_k}{\sqrt{\sum_{k=1}^{n} \frac{p_{kj}}{1-p_{kj}}^2} \sqrt{\sum_{k=1}^{n} w_k^2}} = \frac{\sum_{k=1}^{n} \frac{w_k}{w_j} w_k}{\sqrt{\sum_{k=1}^{n} \frac{w_k}{w_j}^2} \sqrt{\sum_{k=1}^{n} w_k^2}} = 1 \tag{6}$$

Therefore, it is evident that the cosine similarity measure is equal to 1 if the multiplicative and fuzzy preference relations of decision makers are perfectly consistent. If an entry in a PCM is 1 or 0 when a preference relation is fuzzy, one alternative is unanimously preferred over the others. In this case,$b_{ij}$is 0 or does not exist. Now, b_{ij} is defined as the limit of p_{ij}close to one. If we assume, without loss of generality, that $p_{kj} = 1$, then

$$\langle \vec{b_j}^{(k)}, \vec{w_j} \rangle = \frac{b_{1j}w_1 + b_{ij}w_2 + \cdots + b_{kj}w_k + \cdots + b_{nj}w_n}{\sqrt{b_{1j}^2 + b_{2j}^2 + \cdots + b_{kj}^2 + \cdots + b_{nj}^2}\sqrt{\sum_i w_i^2}} = \frac{\frac{b_{1j}}{b_{kj}}w_1 + \cdots + w_k + \cdots + \frac{b_{nj}}{b_{kj}}w_n}{\sqrt{\frac{b_{1j}^2}{b_{kj}^2} + \cdots + 1 + \cdots + \frac{b_{nj}^2}{b_{kj}^2}}\sqrt{\sum_i w_i^2}} \tag{7}$$

Since $b_{ij} = \lim_{x\to 1^-} \frac{p_{ij}}{1-P_{ij}} \to +\infty$, the cosine similarity measure yields

$$\lim_{b_{kj}\to+\infty} \langle \vec{b_j}^{(k)}, \vec{w_j} \rangle = \lim_{b_{kj}\to+\infty} \frac{\frac{b_{1j}}{b_{kj}}w_1 + \cdots + w_k + \cdots + \frac{b_{nj}}{b_{kj}}w_n}{\sqrt{\frac{b_{1j}^2}{b_{kj}^2} + \cdots + 1 + \cdots + \frac{b_{nj}^2}{b_{kj}^2}}\sqrt{\sum_i w_i^2}} = \frac{w_k}{\sqrt{\sum_i w_i^2}} \tag{8}$$

Letting $\lim_{b_{kj}\to+\infty} \langle \vec{b_j}^{(k)}, \vec{w_j} \rangle = 1$, we obtain $w_k = 1$ and $w_i = 0$, $i = 1, 2, \cdots, k-1, k+1, \cdots, n$, which is nonsensical in terms of economic and management theories and practices. This shows that the PCM is not perfectly consistent in this case. Therefore, in practice, we assume that $p_{ij}^{(k)} \approx 1$(e.g.$p_{ij}^k = 0.9999$) and $p_{ij}^{(k)} \approx 0$(e.g.$p_{ij}^k = 0.0001$)when DMs apply the cosine similarity measure to GDM problem. Including one in each column and row of a fuzzy PCM directly yields a preference ordering from the fuzzy PCM.

In the GDM problem, the derived collective priority vector should be most largely consistent for each decision makers. Therefore, the group preference should have the highest similarity measure between the derived collective priority and each column of decision makers' PCMs. Inspired by this idea, we construct a cosine similarity measure maximization optimization model as follows to optimize the maximization similarity measure:

$$\begin{aligned} &\max\ C = \sum_{k=1}^{K}\sum_{j=1}^{n} \sigma_k C_j^{(k)} = \sum_{k=1}^{k_m}\sum_{j=1}^{n} \sigma_k \frac{\vec{w}\vec{a_j}^{(k)}}{\|\vec{w}\|\|\vec{a_j}^{(k)}\|} + \sum_{k=k_{m+1}}^{K}\sum_{j=1}^{n} \sigma_k \frac{\vec{w}\vec{b_j}^{(k)}}{\|\vec{w}\|\|\vec{b_j}^{(k)}\|} \\ &s.t.\quad \begin{cases} \sum_{i=1}^{n} w_i = 1 \\ w_i > 0, \qquad i = 1, 2, \cdots, n \end{cases} \end{aligned} \tag{9}$$

Where C is the total similarity measure and $C^{(k)}$ is the similarity measure between the collective priority vector and the PCM of the kth decision maker.

To simplify the computing process, we normalize the vectors and denote $\vec{w}'$, $\vec{a_j}'^{(k)}$ and $\vec{b_j}'^{(k)}$ as follows:

$$\vec{w}' = (w_1', w_2', \cdots, w_n')^T = \frac{\vec{w}}{\|\vec{w}\|} = (\frac{w_1}{\|\vec{w}\|}, \frac{w_2}{\|\vec{w}\|}, \cdots, \frac{w_n}{\|\vec{w}\|})^T \tag{10}$$

$$\vec{a_j}'^{(k)} = (a_{1j}'^{(k)}, a_{2j}'^{(k)}, \cdots, a_{nj}'^{(k)})^T = \frac{\vec{a_j}^{(k)}}{\|\vec{a_j}^{(k)}\|} = (\frac{a_{1j}^{(k)}}{\|\vec{a_j}^{(k)}\|}, \frac{a_{2j}^{(k)}}{\|\vec{a_j}^{(k)}\|}, \cdots, \frac{a_{nj}^{(k)}}{\|\vec{a_j}^{(k)}\|})^T \tag{11}$$

$$\vec{b_j}'^{(k)} = (b_{1j}'^{(k)}, b_{2j}'^{(k)}, \cdots, b_{nj}'^{(k)})^T = \frac{\vec{b_j}^{(k)}}{\|\vec{b_j}^{(k)}\|} = (\frac{b_{1j}^{(k)}}{\|\vec{b_j}^{(k)}\|}, \frac{b_{2j}^{(k)}}{\|\vec{b_j}^{(k)}\|}, \cdots, \frac{b_{nj}^{(k)}}{\|\vec{b_j}^{(k)}\|})^T \tag{12}$$

This transforms the optimization model into the following model:

$$\begin{aligned} \max\ & C = \sum_{k=1}^{k_m}\sum_{j=1}^{n}\sigma_k\vec{w}'\vec{a_j}'^{(k)} + \sum_{k=k_{m+1}}^{K}\sum_{j=1}^{n}\sigma_k\vec{w}'\vec{b_j}'^{(k)} \\ s.t.\ & \begin{cases} \sum_{i=1}^{n} w_i'^2 = 1 \\ w_i\prime > 0, \qquad i = 1, 2, \cdots, n \end{cases} \end{aligned} \tag{13}$$

For the sake of simplicity, the above optimal model naturally can be denoted by vector in space analytic geometry. The objective function can be rewritten as follows:

$$C = \vec{w}' \cdot (\sum_{k=1}^{k_m}\sum_{j=1}^{n}\sigma_k\vec{a_j}'^{(k)} + \sum_{k=k_{m+1}}^{K}\sum_{j=1}^{n}\sigma_k\vec{b_j}'^{(k)}) \tag{14}$$

Denoting the $\vec{v} = (v_1, v_2, \cdots, v_n)^T = (\sum_{k=1}^{k_m}\sum_{j=1}^{n}\sigma_k\vec{a_j}'^{(k)} + \sum_{k=k_{m+1}}^{K}\sum_{j=1}^{n}\sigma_k\vec{b_j}'^{(k)})^T$. It is evident that the $\vec{v}$ must exist and be uniqueness. Therefore, the objective function is

$$\max\ C = \vec{w}' \cdot \vec{v} \tag{15}$$

Since the $\vec{w}'$ and $\vec{v}$ are normalized vectors, the C essentially is cosine similarity value of two vectors. Therefore, we can draw the conclusion that the maximum value of (15) must be 1. In this time, the solution $\vec{w}'$ must be $\vec{v}$ or $-\vec{v}$ and the latter is rounded off in real life questions.

From the constraint condition $\sum_{i=1}^{n} w_i'^2 = 1$, it follows that $\vec{w}'$ should be normalize and the solution is (16).

$$\vec{w}' = \frac{\vec{v}}{\|\vec{v}\|} \tag{16}$$

Now, it is evident that following condition exists by (9) and (13):

$$\sum_{i=1}^{n} w_i = \sum_{i=1}^{n} w_i'\|\vec{w}\| = \frac{\sum_{i=1}^{n} v_i}{\|\vec{v}\|}\|\vec{w}\| = 1 \tag{17}$$

This means that

$$\|\vec{w}\| = \frac{\|\vec{v}\|}{\sum_{i=1}^{n} v_i} \tag{18}$$

We can obtain the unique solution

$$\vec{w} = \vec{w}'\|\vec{w}\| = \frac{\vec{v}}{\|\vec{v}\|}\|\vec{w}\| = \frac{\vec{v}}{\|\vec{v}\|}\frac{\|\vec{v}\|}{\sum_{i=1}^{n} v_i} = \frac{\vec{v}}{\sum_{i=1}^{n} v_i} \tag{19}$$

Remark1: It is evident that total cosine value $C = nK$ is optimal if the decision makers' PCMs are all perfectly consistent. In this case, the priority vector has highest cosine value to each column vectors of PCMs. For single PCM, the cosine value $C = n$ which is order of the PCM if the priority vector has perfectly consistency, owing to the cosine value is 1 between each column vector of the PCM and priority vector in this case.

In order to use the calculation process, the proposed model can be implemented using the following algorithm in four steps with simple mathematical operations.

Algorithm:
Step 1: Transform fuzzy PCMs (p_{ij}), $k = k_{m+1}, k_{m+2}, \cdots, K$ into (b_{ij}), $k = k_{m+1}, k_{m+2}, \cdots, K$ by (5);
Step 2: Calculate $\vec{a_j}'^{(k)}$, $k = 1, 2, \cdots, k_m$ and $\vec{b_j}'^{(k)}$,$k = k_{m+1}, k_{m+2}, \cdots, K$ by (11) and (12);
Step 3: Calculate transformed weights coefficients vector $\vec{w}'$ by (16);
Step 4: Calculate the collective priority vector $\vec{w}$ by (19)
Remark2:In step 2, a column including 1 in the fuzzy PCM is not considered if $b_{ij} \leq 10^4$, in the calculation. Letting $p_{hl}^k = 1$, without loss of generality, $\sigma_l \langle \vec{w}, \vec{b_l}^{(k)} \rangle$ in optimization model (13) cannot be considered, and in this case, the transformed weights coefficient vector $\vec{v}$ is

$$\vec{v} = (\sum_{k=1}^{K} \sum_{j=1}^{n} \sigma_k \vec{a_j}'^{(k)} + \sum_{k=k_{m+1}}^{K} \sum_{j=1, j\neq l}^{n} \sigma_k \vec{b_j}'(k))^T \tag{20}$$

Since the column including $b_{hl} = 1$, approximate to $(0, 0, \cdots, 1_h, \cdots, 0)^T$ and $\lim p_{kl} \to 1 b_{kl} = 0$, $k \neq h$ cannot provide any effective preference information. In this case, this column will be removed to avoid a large deviation in the priority vector.

4 Illustrative examples

This section describes the implementation of the proposed model and presents three numerical examples to compare the proposed model with some existing methods in same problems.

Example 1 Consider the following multiplicative and fuzzy preference structures, which comes from Chiclana et al.[5]; Fan et al.[7]; Wang et al.[28]:

An investment company wishes to obtain the best option for investing a sum of money among four alternatives Car Company, Food Company, Computer Company, and Arms Company. The decision makers come from four consultancy departments of the investment company, and they have the same degrees of importance. Each department provides its preference information, with DM_1 and DM_2 providing the following multiplicative preference relations:

$$DM_1 = \begin{pmatrix} 1 & 1/7 & 1/3 & 1/5 \\ 7 & 1 & 3 & 2 \\ 3 & 1/3 & 1 & 1/2 \\ 5 & 1/2 & 2 & 1 \end{pmatrix} \text{ and } DM_2 = \begin{pmatrix} 1 & 3 & 1/4 & 5 \\ 1/3 & 1 & 2 & 1/3 \\ 4 & 1/2 & 1 & 2 \\ 1/5 & 3 & 1/2 & 1 \end{pmatrix}$$

DM_3 and DM_4 express their preference information in terms of fuzzy preference relations as

$$DM_3 = \begin{pmatrix} 0.5 & 0.1 & 0.6 & 0.7 \\ 0.9 & 0.5 & 0.8 & 0.4 \\ 0.4 & 0.2 & 0.5 & 0.9 \\ 0.3 & 0.6 & 0.1 & 0.5 \end{pmatrix} \text{ and } DM_4 = \begin{pmatrix} 0.5 & 0.5 & 0.7 & 0.1 \\ 0.5 & 0.5 & 0.8 & 0.6 \\ 0.3 & 0.2 & 0.5 & 0.8 \\ 0 & 0.4 & 0.2 & 0.5 \end{pmatrix}$$

It is evident that $p_{14}^{(4)} = 1$ and $\lim_{p_{14}^{(4)} \to +\infty} b_{14}^{(4)} = 1$. In this case,$b_{24}^{(4)} = 0.15 \times 10^{-9} \leq 10^{-4}$,$b_{34}^{(4)} = 0.40 \times 10^{-9} \leq 10^{-4}$ and $b_{44}^{(4)} = 0.1 \times 10^{-9} \leq 10^{-4}$; moreover, we can calculate $\vec{v}$ by (20).

The collective priority vector by the cosine similarity measure optimization model is found to be $(w_1, w_2.w_3, w_4)^T = (0.2075, 0.3882, 0.2124, 0.1919)^T$, with the ranking of alternatives $A_2 \succ A_3 \succ A_1 \succ A_4$, which is the same as the results provided by Chiclana et al. [5]. Chiclana et al. [26] select the best alternative using the weighted average operator on the basis of uniform preference formats using the transformation function

$$p_{ij}^k = \frac{1}{2}(1 + \log_9 a_{ij}^k), i = 1, 2, \cdots, n; j = 1, 2, \cdots, n \tag{21}$$

Then, they aggregate uniform fuzzy information into a collective opinion using the OWA operator ϕ_Q as

$$p_{ij}^{*} = \phi_Q(p_{ij}^1, p_{ij}^2, \cdots, p_{ij}^n) = \sum_{h=1}^{n} v_h d_{ij}^h, i = 1, 2, \cdots, n; j = 1, 2, \cdots, n \tag{22}$$

Where $d_{ij}^h =$ is the hth largest value among the collection of $p_{ij}^1, p_{ij}^2, \cdot, p_{ij}^n$ and $v_1, v_2, \cdots, v_n$is a weight vector with $v_h = Q(\frac{h}{n}) - Q(\frac{h-1}{n})$, $h = 1, 2, \cdots, n$. $Q(\cdot)$is a fuzzy quantifier with the membership function

$$Q(r) = \begin{cases} 0 & r < \gamma, \\ \frac{r-\gamma}{\mu-\gamma} & \gamma < r < \mu \\ 1 & r < \mu. \end{cases} \tag{23}$$

In practice, the fuzzy quantifier (γ, μ) is always selected as (0.3,0.8) ,(0,0.5) and (0.5,1), representing "most", "at least half", and "as much as possible, respectively. The OWA operator ϕ_Q is used to rank the alternatives or select the most desirable alternative(s) through the same calculation as that in the previous process.

Fan et al. [7] established a goal-programming model for GDM problem with two preference information based on the idea that the priority vector must be closest to each column of the PCMs provided by the DMs. Their model does not need to transform different preference formats into a uniform structure. It is constructed as

$$\begin{aligned} &\max \sum_{k=1}^{k_m}\sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_k |w_i - a_{ij}^{(k)} w_j| + \sum_{k=k_{m+1}}^{K}\sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_k |w_i - p_{ij}^{(k)}(w_i + w_j)| \\ &s.t. \quad \begin{cases} \sum_{i=1}^{n} w_i = 1 \\ w_i > 0, \qquad i = 1, 2, \cdots, n \end{cases} \end{aligned} \tag{24}$$

Wang et al. [28] constructed a chi-square optimization model on the basis of the same idea as that of Fan et al. [7]. They differentiate the distance using chi-square optimization

$$\begin{aligned} &\max \sum_{k=1}^{k_m}\sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_k \left[\frac{(a_{ij}^{(k)} - w_i/w_j)^2}{w_i/w_j}\right] + \sum_{k=k_{m+1}}^{K}\sum_{i=1}^{n}\sum_{j=1}^{n} \left[\frac{(a_{ij}^{(k)} - w_i/(w_i + w_j))^2}{w_i/(w_i + w_j)}\right] \\ &s.t. \quad \begin{cases} \sum_{i=1}^{n} w_i = 1 \\ w_i > 0, \qquad i = 1, 2, \cdots, n \end{cases} \end{aligned} \tag{25}$$

Table 1 shows the results as compared with these three existing methods. The model proposed in this paper achieves the greatest cosine similarity measure. There is a small difference in the ranking of alternatives. All four methods provide the same best and second-best alternatives, but they disagree on the last two alternatives.

Example 2 Consider single fuzzy PCM to derive priority vector, which is used in Xu and Da[46]; Wang et al.[28]:

This example was used by Xu and Da [46] and Wang et al. [28] to derive a priority vector in a fuzzy preference relation. The following matrix is the fuzzy PCM in their example.

$$DM = \begin{pmatrix} 0.5 & 0.7 & 0.6 & 0.8 \\ 0.3 & 0.5 & 0.4 & 0.6 \\ 0.4 & 0.6 & 0.5 & 0.7 \\ 0.2 & 0.4 & 0.3 & 0.5 \end{pmatrix}$$

Table 1: Comparative results with existing methods for GDM with multiplicative and fuzzy preference relations

Approaches	w_1	w_2	w_3	w_4	Ranking of alternatives	Cosine value(C)
Chiclana et al. [5]	0.2452	0.2972	0.2487	0.2092	$A_2 \succ A_3 \succ A_1 \succ A_4$	12.6642
Fan et al. [7]	0.1280	0.4301	0.2515	0.1903	$A_2 \succ A_3 \succ A_4 \succ A_1$	12.4711
Wang et al.[28]	0.1697	0.3376	0.2741	0.2184	$A_2 \succ A_3 \succ A_4 \succ A_1$	12.5692
Our model	0.2075	0.3882	0.2124	0.1919	$A_2 \succ A_3 \succ A_1 \succ A_4$	**12.8284**

Xu and Da[46] constructed a least-deviation method for priority vector derivation in fuzzy AHP. A transformation function and rank transitivity property were used in the iteration algorithm.

$$\begin{aligned} &\min \sum_{i=1}^{n}\sum_{j=1}^{n}\left[9^{2p_{ij}-1}(w_i/w_j)+9^{2p_{ij}-1}(w_j/w_i)-2\right] \\ &s.t. \quad \begin{cases} \sum_{i=1}^{n} w_i = 1 \\ w_i > 0, \qquad i = 1,2,\cdots,n \end{cases} \end{aligned} \tag{26}$$

Table 2 summarizes the results of the two above mentioned methods and the proposed model. The rankings of alternatives by these three methods are the same. The cosine similarity measure value generated by our model is slightly greater than those by the other approaches.

Table 2: Comparative results with existing methods for GDM with multiplicative and fuzzy preference relations

Approaches	w_1	w_2	w_3	w_4	Ranking of alternatives	Cosine value(C)
Xu and Da[46]	0.4297	0.1784	0.2769	0.1150	$A_1 \succ A_3 \succ A_2 \succ A_4$	3.998186
Wang et al. [28]	0.4284	0.1286	0.2755	0.1159	$A_1 \succ A_3 \succ A_2 \succ A_4$	3.998209
Our model	0.4300	0.1800	0.2479	0.1151	$A_1 \succ A_3 \succ A_2 \succ A_4$	**3.998231**

Example 3 Consider four different preference formats by transformation functions, which is investigated by Ma et al. [8] and Xu et al.[9]:

This example includes four different preference formats, i.e., preference ordering, utility values, multiplicative preference relations, and fuzzy preference relations.

$$DM_1 = \begin{pmatrix} 1 & 1/7 & 1/3 & 1/5 \\ 7 & 1 & 3 & 2 \\ 3 & 1/3 & 1 & 1/2 \\ 5 & 1/2 & 2 & 1 \end{pmatrix} \text{ and } DM_2 = \begin{pmatrix} 1 & 3 & 1/4 & 5 \\ 1/3 & 1 & 2 & 1/3 \\ 4 & 1/2 & 1 & 2 \\ 1/5 & 3 & 1/2 & 1 \end{pmatrix}$$

DM_3 and DM_4 express their preference information in terms of fuzzy preference relations as

$$DM_3 = \begin{pmatrix} 0.5 & 0.1 & 0.6 & 0.7 \\ 0.9 & 0.5 & 0.8 & 0.4 \\ 0.4 & 0.2 & 0.5 & 0.9 \\ 0.3 & 0.6 & 0.1 & 0.5 \end{pmatrix} \text{ and } DM_4 = \begin{pmatrix} 0.5 & 0.5 & 0.7 & 0.1 \\ 0.5 & 0.5 & 0.8 & 0.6 \\ 0.3 & 0.2 & 0.5 & 0.8 \\ 0 & 0.4 & 0.2 & 0.5 \end{pmatrix}$$

$DM_5 = \{u_i | i = 1,2,3,4\} = \{3,1,4,2\}$, $DM_6 = \{u_i | i = 1,2,3,4\} = \{2,3,1,4\}$

$DM_7 = \{0_i | i = 1,2,3,4\} = \{0.5, 0.7, 1.0, 0.1\}$ and $DM_8 = \{o_i | i = 1,2,3,4\} = \{0.7, 0.9, 0.6, 0.3\}$.

where DM_1 and DM_2 provide multiplicative preference relations, DM_3 and DM_4 express their preference information in terms of fuzzy preference relations, DM_5 and DM_6 utilize preference orderings, and DM_7 and DM_8 use utility values.

We can convert preference orderings and utility values to fuzzy preference relations using the following transformation functions (Chiclana et al.[5]; Herrera et al.[1]; Ma et al.[8]):

$$p_{ij} = 0.5(1 + \frac{o_j}{n-1} - \frac{o_i}{n-1}) \tag{27}$$

$$p_{ij} = \begin{cases} \frac{u_i^2}{u_i^2+u_j^2} & (u_i, u_j) \neq (0,0), \\ 0.5 & (u_i, u_j) = (0,0) \end{cases} \tag{28}$$

By (27) and (28), the multiplicative and fuzzy preference relations can be obtained as follows:

$$\overline{DM_5} = \begin{pmatrix} 0.5 & 0.167 & 0.667 & 0.333 \\ 0.833 & 0.5 & 0.1 & 0.667 \\ 0.333 & 0 & 0.5 & 0.167 \\ 0.667 & 0.333 & 0.833 & 0.5 \end{pmatrix}, \overline{DM_6} = \begin{pmatrix} 0.5 & 0.667 & 0.333 & 0.833 \\ 0.333 & 0.5 & 0.167 & 0.667 \\ 0.667 & 0.833 & 0.5 & 1 \\ 0.167 & 0.833 & 0 & 0.5 \end{pmatrix}$$

$$\overline{DM_7} = \begin{pmatrix} 0.5 & 0.338 & 0.200 & 0.962 \\ 0.833 & 0.5 & 0.329 & 0.980 \\ 0.800 & 0.671 & 0.5 & 0.990 \\ 0.038 & 0.020 & 0.010 & 0.5 \end{pmatrix} \text{ and } \overline{DM_8} = \begin{pmatrix} 0.5 & 0.377 & 0.576 & 0.845 \\ 0.623 & 0.5 & 0.692 & 0.900 \\ 0.424 & 0.308 & 0.5 & 0.800 \\ 0.155 & 0.100 & 0 & 0.5 \end{pmatrix}$$

Then, we can derive a priority vector $(w_1, w_2.w_3.w_4)^T = (0.2006, 0.3811, 0.2886, 0.1297)^T$ using the cosine similarity measure optimization model, with the ranking of alternatives $A_2 \succ A_3 \succ A_1 \succ A_4$.

From Table 3, we observe that the four methods generate the same ranking (Chiclana et al.[3]; Ma et al.[8]; Xu et al.[9]), and that our model again achieves the highest cosine value, indicating that the similarity is most similar to each column of the PCMs of the decision makers.

Table 3: Comparative results with existing methods for four different preference formats in GDM

Approaches	w_1	w_2	w_3	w_4	Ranking of alternatives	Cosine value(C)
Chiclana et al. [3]	0.5651	0.7826	0.6619	0.4973	$A_2 \succ A_3 \succ A_1 \succ A_4$	-
Ma et al. [8]	0.2210	0.3426	0.2755	0.1159	$A_2 \succ A_3 \succ A_4 \succ A_1$	26.0116
Xu et al.[9]	0.2210	0.3426	0.2827	0.1537	$A_2 \succ A_3 \succ A_4 \succ A_1$	25.9763
Our model	0.2006	0.3811	0.2886	0.1297	$A_2 \succ A_3 \succ A_1 \succ A_4$	**26.0279**

Furthermore, though the models developed by Ma et al.[8] and Xu et al.[9] are optimization-based, which are similar to the model proposed in this study, their computational complexities are higher than our model. The former obtain results by transforming preference into certain formats which are suitable to calculate solution of programming, the latter need to solve nonlinear programming using genetic algorithm. Their models are presented as follows:

Ma et al. [8] proposed an optimization model for different formats. They optimize the minimum distance between the priority vector and a perfectly consistent PCM. Their model is given by

$$\begin{aligned} \min & \sum_{k=1}^{m_1}\sum_{i=1}^{n}\sum_{j=1}^{n}(w_i u_j^{(k)} - w_j u_i^{(k)})^2 + \sum_{k=m_1+1}^{m_2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[w_i(n-o_j^{(k)}) - w_j(n-o_i^{(k)})\right]^2 \\ & + \sum_{k=m_2+1}^{m_3}\sum_{i=1}^{n}\sum_{j=1}^{n}(w_i - w_j a_{ij}^{(k)})^2 + \sum_{k=m_3+1}^{m_4}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[w_i - (w_i+w_j)p_{ij}^{(k)}\right]^2 \\ s.t. & \begin{cases} \sum_{i=1}^{n} w_i = 1 \\ w_i > 0, \qquad i = 1, 2, \cdots, n \end{cases} \end{aligned} \tag{29}$$

Xu et al.[9] constructed a nonlinear optimization model to handle a GDM problem with missing values in the PCM. Their model is given as

$$\begin{aligned}
&\min \Big[\sum_{k=1}^{m_1}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|w_i - u_i^{(k)}/\sum_{j=1}^{n}u_j^{(k)}\right|^p + \sum_{k=m_1+1}^{m_2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|w_i - \frac{n-o_i^{(k)}}{n-1}/\sum_{j=1}^{n}\frac{n-o_j^{(k)}}{n-1}\right|^p \\
&+ \sum_{m_2+1}^{m_3}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|a_{ij}^{(k)} - w_i/w_j\right|^p + \sum_{m_3+1}^{m_4}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|p_{ij}^{(k)} - w_i/(w_i+w_j)\right|^p\Big]^{1/p} \qquad (30)\\
&s.t. \quad \begin{cases} \sum_{i=1}^{n} w_i = 1 \\ w_i > 0, \qquad i = 1, 2, \cdots, n \end{cases}
\end{aligned}$$

Both models (Ma et al.[8] and Xu et al. [9]) needed to solve complex optimization models. Particularly, model (30) uses a genetic algorithm to obtain a solution and involves high computational complexity. However, the model proposed herein can be solved using a Lagrangian approach and can be directly used to obtain a collective priority vector by following four simple steps.

5 Discussion and Conclusion

How to aggregate different preference formats is an important question in group decision making problem because it's natural for DMs with different backgrounds to represent their preferences using different formats. In this paper, an optimization model was developed on the basis of the cosine similarity measure to deal with multiplicative preference relations and fuzzy preference relations. The basic idea of the model is that the collective priority vector should be as similar per column as possible to a pairwise comparative matrix (PCM) from each decision maker in order to guarantee the priority vector is nearest to perfectly consistency for each decision makers. Compared with existing optimization-based methods, the proposed model is computationally simple, because it can be solved using a Lagrangian approach and obtain a collective priority vector by following four simple steps. The proposed method can also be used to derive priority vector of fuzzy AHP and provided to reach consensus by mans of existing iterative modification methods.

Three previously published examples were used to compare the proposed model with some existing approaches. The results show that the rankings generated by different approaches are similar and that the proposed model achieves the greatest cosine values in all three examples, indicating that the proposed model achieves the nearest theoretical perfectly consistent opinion for each decision makers. Furthermore, since the proposed model can be adapted to GDM problems with various preferences formats using transformation functions, it provides an efficient and simple way to handle different preference structures.

Acknowledgment

This research was supported in part by grants from the National Natural Science Foundation of China (#71222108, #71325001 and #71471149) and Major project of the National Social Science Foundation of China (# 15ZDB153).

Bibliography

[1] F. Herrera, E. Herrera-Viedma, F. Chiclana (2001), Multiperson decision making based on multiplicative preference relations, *European Journal of Operational Research*, 129(2):372-385.

[2] E. Herrera-Viedma, F. Herrera, F. Chiclana (2002), A consensus model for multiperson decision making with different preference structures, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 32(3):394-402.

[3] F. Chiclana, F. Herrera, E. Herrera-Viedma (1998), Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations, *Fuzzy Sets and Systems*, 97:33-48.

[4] M.P. Brady, S.T. Wu (2010), The aggregation of preferences in groups: Identity, responsibility, and polarization, *Journal of Economic Psychology*, 31(6):950-963.

[5] F. Chiclana, F. Herrera, E. Herrera-Viedma (2001), Integrating multiplicative preference relations in a multipurpose decision making model based on fuzzy preference relations, *Fuzzy Sets and Systems*, 122(2):277-291.

[6] M. Delgado, F. Herrera, E. Herrera-Viedma, L. Marffnez (1998), Combining numerical and linguistic information in group decision making, *Information Sciences*, 107:177-194.

[7] Z.P. Fan, J. Ma, Y.P. Jiang, Y.H. Sun, L. Ma (2006), A goal programming approach to group decision making based on multiplicative preference and fuzzy preference relations, *European Journal of Operational Research* , 174:311-321.

[8] J. Ma, Z. P. Fan, Y. P. Jiang, J. Y. Mao (2006), An optimization approach to multiperson decision making based on different formats of preference information, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 36(5):876-889.

[9] Z.S. Xu, X.Q. Cai, S. S. Liu (2011), Nonlinear programming model integrating different preference structures,*IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 41(1):169-177.

[10] Y.C. Dong, H.J. Zhang, E. Herrera-Viedma (2016), Integrating experts' weights generated dynamically into the consensus reaching process and its applications in managing non-cooperative behaviors, *Decision Support Systems*, 84:1-15.

[11] X.H. Xu, Z.J. Du, X.H Chen (2015), Consensus model for multi-criteria large-group emergency decision making considering non-cooperative behaviors and minority opinions. *Decision Support Systems*, 79:150-160.

[12] Y.C. Dong, H.J. Zhang (2014), Multiperson decision making with different preference representation structures: A direct consensus framework and its properties, *Knowledge-Based Systems*, 58:45-57.

[13] Y.C. Dong, N. Luo, H.M. Liang, Consensus building in multiperson decision making with heterogeneous preference representation structures: A perspective based on prospect theory, *Applied Soft Computing*, (2015)35:898-910.

[14] I. Palomares, L.Martínez, F. Herrera (2014), A Consensus Model to Detect and Manage Noncooperative Behaviors in Large-Scale Group Decision Making, *IEEE Transactions on Fuzzy Systems*, 22:516-530.

[15] X. Chen, H. Zhang, Y. Dong (2015), The fusion process with heterogeneous preference structures in group decision-making: A survey,*Information Fusion*, 24:72-83.

[16] Z.B. Wu, J.P. Xu (2012), A consistency and consensus based decision support model for group decision-making with multiplicative preference relations, *Decision Support Systems*, 52:757-767.

[17] Y. C. Dong, G. Q. Zhang, W. C. Hong, Y. F. Xu (2010), Consensus models for AHP group decision-making under row geometric mean prioritization method, *Decision Support Systems*, 49:281-289.

[18] Y.J. Xu, R. Patnayakuni, H.M. Wang (2013), The ordinal consistency of a fuzzy preference relation, *Information Sciences*, 224:152-164.

[19] Y.J. Xu, K.W. Li, H.M. Wang (2013), Distance-based consensus models for fuzzy and multiplicative preference relations, *Information Sciences*, 253:56-73.

[20] L.A. Yu, K.K. Lai(2011), A distance-based group decision-making methodology for multi-person multi-criteria emergency decision support, *Decision Support Systems*, 51(2):307-315.

[21] K.J. Arrow (1963), *Social Choice and Individual Values*, New York: Wiley, 1963.

[22] A.K. Sen (1970), *Collective Choice and Social Welfare*, HoldenDay, San Francisco, CA, 1970.

[23] W.J.M. Kickert (1978) , *Fuzzy Theories on Decision-Making*, Dordrecht:Nijho, 1978.

[24] T. L. Saaty (1980), *The Analytic Hierarchy Process*, New York, NY, USA: McGraw-Hill, 1980.

[25] J. Kacprzyk, M. Fedrizzi (1990), *Multiperson Decision-Making Models Using Fuzzy Sets and Possibility Theory*, Dordrecht: Kluwer Academic Publishers, 1990.

[26] F. Chiclana, F. Herrera, E. Herrera-Viedma (2002), A note on the interval consistency of various preference representations, *Fuzzy Sets and Systems*, 131:75-78.

[27] Z.S. Xu (2007), Multiple-attribute group decision making with different formats of preference information on attributes, *IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics*, 37(6):1500-1511.

[28] Y.M. Wang, Z.P. Fan, Z.S. Hua (2007), A chi-square method for obtaining a priority vector from multiplicative and fuzzy preference relations, *European Journal of Operational Research* , 182:356-366.

[29] T. L. Saaty (1986), Axiomatic foundation of the analytic hierarchy process, *Management Sciences*, 32(7):841-855.

[30] J. Aguarón, J. M. Moreno-Jiménez (2003), The geometric consistency index approximated thresholds, *European Journal of Operational Research* , 147(1):137-145

[31] M.T. Escobar, J. Aguar on, J.M. Moreno-Jiménez (2004), A note on AHP group consistency for the row geometric, *European Journal of Operational Research* , 153:318-322.

[32] D. Ergu, G. Kou, Y. Peng, Y. Shi (2011), A simple method to improve the consistency ratio of the pair-wise comparison matrix in ANP, *European Journal of Operational Research* , 213:246-259.

[33] C. S. Lin, G. Kou, D. Ergu (2013), A statistical approach to measure the consistency level of the pairwise comparison matrix, *European Journal of Operational Research* , 65(9):1380-1386.

[34] C. S. Lin, G. Kou, D. Ergu (2013), An improved statistical approach for consistency test in AHP, *Annals of Operations Research* 211(1):289-299.

[35] C. S. Lin, G. Kou (2015), Bayesian revision of the individual pair-wise comparison matrices under consensus in AHP-GDM, *Applied Soft Computing* 35:802-811.

[36] T. L. Saaty (1977), A scaling method for priorities in a hierarchical structure, *Journal of Mathematical Psychology*, 15(3):234-281.

[37] A. T. W. Chu, R. E. Kalaba, K. Spingarn (1979), A comparison of two methods for determining the weights of belonging to fuzzy sets, *Journal of Optimization Theory and Applications*, 27:531-538.

[38] G. Crawford, C. Williams (1985), A note on the analysis of subjective judgment matrices, *Journal of Mathematical Psychology*, 29:387-405.

[39] C. S. Lin, G. Kou, D. Ergu (2013), A heuristic approach for deriving the priority vector in AHP, *Applied Mathematical Modelling* , 37:5828-5836.

[40] G. Kou, C.S. Lin (2014), A cosine maximization method for the priority vector derivation in AHP, *European Journal of Operational Research* , 235(1):225-232.

[41] J.A. Gomez-Ruiz, M. Karanik, J. I. Peláez (2010), Estimation of missing judgments in AHP pairwise matrices using a neural network-based model, *Applied Mathematics and Computation*, 216:2959-2975.

[42] T. Tanino (1984), Fuzzy preference orderings in group decision making, *Fuzzy Sets and Systems*, 12:117-131.

[43] E. Herrera-Viedma, F. Herrera, F. Chiclana, M. Luque (2004), Some issues on consistency of fuzzy preference relations, *European Journal of Operational Research* , 154:98-109.

[44] G.Q. Zhang, Y.C. Dong, Y.F. Xu (2012), Linear optimization modeling of consistency issues in group decision making based on fuzzy preference relations, *Expert Systems with Applications*, 39:2415-2420.

[45] L. Mikhailov (2003), Deriving priorities from fuzzy pairwise comparison judgements,*Fuzzy Sets and Systems*,134:365-385.

[46] Z.S. Xu, Q.L. Da (2005), A least deviation method to obtain a priority vector of a fuzzy preference relation,*European Journal of Operational Research*, 64:206-216.

[47] J. Kacprzyk (1986), Group decision making with a fuzzy linguistic majority, *Fuzzy Sets and Systems*, 18:105-118.

[48] M. Roubens (1989), Some properties of choice functions based on valued binary relatios,*European Journal of Operational Research* ,40:309-321.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(1):41-52, February 2017.

Lagrangian Formulation for Energy-efficient Warehouse Design

I. Derpich, J. Sepulveda

Ivan Derpich*, Juan Sepulveda
Departamento de Ingeniería Industrial
Universidad de Santiago de Chile, USACH
Ave. Ecuador 3769, Estacion Central, Santiago, Chile
(ivan.derpich,juan.sepulveda)@usach.cl
*Corresponding author:ivan.derpich@usach.cl

Abstract: Energy consumption in modern warehouses is today an important issue which has not received much attention in the scientific community. In this paper it is addressed the problem of warehouse design considering the energy costs incurred by vehicles and equipment in a fully or partially automated facility. Closed-form solutions are obtained by a formulating the Lagrangian of an operational cost function with equality constraints. The contribution of the paper is to develop formulas for reduced energy consumption and pollution, both relevant aspects in sustainable engineering systems. An example applied to a distributor of MRO items is presented. In this version the energy cost is integrated into the formula, modifying the method presented in [1].[a]
Keywords: Energy optimization, integrated facility design, green supply chains.

1 Introduction

This paper is an extension of [1], which considers separately the cost of travel in the plane and the cost of energy for moving the items along the vertical axis. In this present paper both costs are integrated and formulas are obtained for designing facilities in an automated facility. The design of a warehouse or a distribution center (DC) has become a fundamental decision for the optimization of a supply chain. Even though there exist several articles in the academic literature in which efficient layout models have been studied, it has not been formulated a simple mathematical model in order to obtain the basic variables for the construction of a DC in three dimensions (3D).

In warehousing facilities the movement of goods is performed by various types of equipment such as AGV (Automated Guided Vehicle), fork lifters, stackers, vertical reach trucks, conveyors or by automated robotics equipment of Cartesian movement adjacent to the storage racks, as in AS/R (Automated Storage/Retrieval) systems. In the literature there exist works with formulas to design shelves in two and three dimensions buy they do not adequately consider the problem of movement in the Z-axis, thereby giving inefficient results with very high shelves that ignore energy consumption and sub-optimize costs. This occurs because the known approaches consider the movement only on the X-Y plane, and such horizontal movement as expensive as in height, which in general does not hold in actual facilities. The problem is exacerbated for heavy materials due to greater energy waste. In this paper the above problem is solved by a cost model which considers in addition the cost of movement in height. With the model and derived formulas, more efficient shelves design with less energy consumption is generated. The formulas developed are optimal with respect to travel distances and they are obtained from solving a nonlinear optimization problem with linear constraints through a Lagrange transformation.

The formulas found in the literature mainly address the problem in two dimensions; for instance, Bassan et al. (1980) [2], consider a rectangular warehouse and racks which are parallel or perpendicular to the wall. They discuss also the optimal location of the warehouse door, and the optimal design when the storage area is divided into different zones. All of their work is developed on the horizontal plane; they fix a priori a given number of vertical racks and that value is used as a parameter.

Some formulas for three dimensions that have been developed, consider implicit equations whose resolution must be made with an iterative method, this makes it difficult to use, as for example in Onut et al. (2008) [3]. In this paper it is considered that supplies are received from several suppliers and dispatched to many customers; consequently the study is complemented with the design of a multi -level warehouse (according to product flow) by using an ABC methodology. Baker and Canessa (2009) [4] state that a structured approach for the design of a storage system in the company is not currently available and they make a compilation of different approaches to finalize with the proposition of a new design methodology. Another classic work is Gu et al. (2010) [5] in which a detailed revision of the warehouse design research is presented by considering practical case studies and computational support tools. In addition, they present a framework for a systematic classification. Gu et al. (2010) [5] also conclude that even though there are a large number of papers focusing on details of the different systems, there are few addressing the decision making concerning storage systems.

The work by Rouwenhorst el al. (2000) [6] presents a framework of references and classification of problems regarding design and control of storage facilities. To the review of the literature on storage systems is added the need for studies on isolated design rather than sub-problems. A central idea in [6] is dividing the functions of planning and design of warehouses at three levels: strategic, tactical and operational. At the strategic level, the number of warehouses, size and location of each, the equipment for handling materials, the functional areas, the process flow and warehouse layout, are determined. At the tactical level, manpower needed to operate the warehouse, the location of the products in the functional areas, replacement and order picking policies, are determined.

At the operational level, the concern relates to the routing of the products, batch determination, daily and weekly staff allocation and control tasks. In this framework of analysis it can be seen that the design of the warehouse, and in particular the number of shelves, is a strategic decision. Moreover, the determination of the sizes of product areas according to categories A, B and C is a tactical decision strongly involved with the decisions of the operational level. Thus the determination of the optimal design of a warehouse or distribution center is essential to minimize the transfer of products and therefore to minimize energy expenditure. In a relatively recent work, Heragu et al. (2005) [7] propose a mathematical model of mixed integer linear programming to decide which section to assign each product (dispatch, reservation or cross-docking), and also decide what process flow to associate, in addition to other decisions of operational and tactical level .

A recent article by Zhou et al. in 2016 [8], addresses the question of how to design facilities, in a finite horizon taking into account various requirements of demand with respect to size, price, location, security, among others. They study the impact of re-design and methods to modify the design of the facility and present a mixed integer programming model and solve it by column-generation and branch and bound algorithms.

Roodbergen et al. 2006 [9], they published a work on picking area layout in a warehouse, so that the average travel distance is minimized. They provide formulas that can be used to calculate the average length that an order runs under different routing policies. The optimal layout is determined using a model of nonlinear programming. The optimal number of aisles depends on the discipline of picking and the size of the picking list.

Another approach that is used is dynamic storage allocation (DSAP or dynamic storage assignment problem). In the paper by Li et al. [10] is presented an integrated mechanism for the purpose of optimization which is based on the ABC classification and the mutual affinity of the products. The mechanism is developed by using a data mining technique and the authors show that the DSAP is a NP-hard problem. The results show that it is an advantageous approach to the classic ABC method, but requires the formulation of a complex problem and its resolution through a simulation of a metaheuristic method, which is not easy. In the paper already cited, Zhou et al. [8] present a new approach called self-managed warehouses is shown, whose use has rapidly grown in the world and are a universal trend. The problem in such a warehouse is the need for re-doing the layout frequently, so it is necessary to have direct ways to reconfigure the space continuously.

In this work, the developed mathematical formulas are explicit and easy to use for the design of a warehouse or DC; it is assumed that a rectangular space is available, with a single main entrance gate in front in the center of the longest wall and an exit door located similarly at the rear wall. The formulas give the number of double racks to be used, the number of pallet spaces in each frame and its height also measured in pallets. The objective function is to optimize the total cost of movement of goods in the DC. In the first chapter a review of the literature on the use of mathematical methods in the design of distribution centers is presented. We conclude that not available explicit formulas for the design , so there are models applied to special cases .In the second chapter the formulas proposed in the paper has been much development, a rectangular supposed cellar with a single door in the center and a cost function nonlinear type is used with linear constraints , which are solved using Lagrange system to obtain the sought formulas.

In the third chapter an experiment with a test case taken from the literature, one of the few papers found with similar studies is presented . The results are comparable , obtaining wineries lower height, which also reduces the total cost of transportation of the articles.Finally in chapter four conclusions and future work are presented . This is related to the development of formulas that can be used where space is restricted.

2 Distribution center configuration

In the literature different configurations for the layout of a distribution center reportedl. Francis (1967) [11] studied architecture design theoretically and proved that the best configuration is an area that both measurements are the same , that is a perfect square. However in most cases, a rectangular area , is assumed. in this paper we will assume the same, however eventually develop formulas to tell whether the area is square or rectangular.

2.1 Assumptions

Some assumptions to be made in developing the layout of the distribution center are:

- Goods are stored in double racks, except for the rack adjacent the wall, which has only one side.
- The shape of the store is rectangular (see Figure 1).
- There should be wide aisles between the racks, and along the walls the width of these aisles should be the same.
- Goods enter through a door located at one of the walls of the store and leave through the same door.

The first study of Bassan et al. (1980) [2] conducted on the design of a distribution center provided two possible distributions of the shelves. In this paper, only the distribution shown in Figure 1 is analyzed, where the shelves are located in the sense of the narrower direction (v).

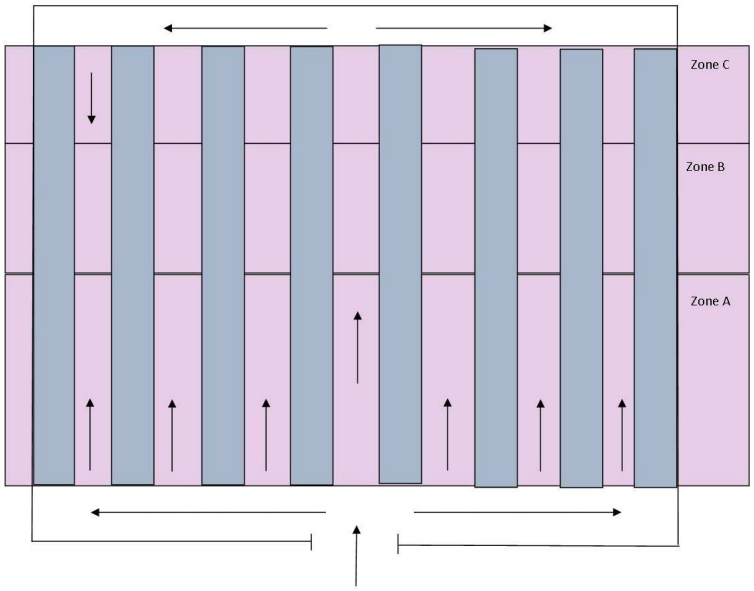


Figure 1: Rectangular distribution center

The notation used in this study is as follows:

- N = total storage capacity in storage spaces (slots).
- a = width of an aisle, where it is assumed that all the aisles have the same width.
- w = width of double shelf.
- W= average weight of a unit storage in a container.
- L = length of a storage unit in the shelf.
- m = total number of storage spaces along a shelf.
- h = number of vertical storage spaces.
- n = number of double shelves.
- u = length of the store.
- v = width of the warehouse.
- d = annual movement (demand) of the warehouse, in storage units. It is assumed that a unit of stored article occupies a unit of space (in items / year).
- C_h =material handling cost of moving a unit of article per unit of length in the x-y surface (in US$/m).
- C_e = energy cost (in US$/unit of energy).
- T_v = average travel distance on the v-direction in meter.

- T_u = average travel distance on the u-direction in meter.
- T_h = average travel distance in height in meter.
- v_e = velocity of moving a unit of weight W in a distance in height (mt/sec)

2.2 Cost function

This paper considers only the cost of moving and handling of goods inside the warehouse. For this, we use Figure 1 as a reference and hence the measures considered. First the length and width of the distribution center will be determined, which are determined by:

Width

$$u = n(w + a) \tag{1}$$

Length

$$v = 2a + mL \tag{2}$$

Then the area of the distribution center is as follows:

$$uv = n(w + a)(2a + mL) \tag{3}$$

In order to calculate the cost of moving goods, the average travel distance of each stored unit is required. It will be assumed that the doors are located at the centers of the walls and that all goods are equally likely to be used. To determine the cost function of the DC, we must first set the probabilities of carrying a good in the horizontal plane and in height. In the horizontal plane, two axes of movement are defined: one for lateral movement (u direction) and one towards the bottom of the warehouse (v direction), as shown in Figure 1. Thus, let T_v , T_u and T_h be the distances in each direction. As the probability of taking a good to the left or right from the door of the CD store is the same, the average travel distance on the horizontal axis is (4):

$$T_u = \frac{u}{4} = \frac{n(w + a)}{4} \tag{4}$$

For the distance on direction v, the length of an average trip depends on the category of the item, that is, A, B or C. Let P_A ,P_B and P_C be the probability that the item to be moved belongs to category A, B and C, respectively. In addition, let m_A, m_B and m_C be the corresponding number of reserved spaces in the direction v for products in categories A, B, C, respectively. With this formulation the average length of a trip in direction v is given by:

$$T_v = a + P_a\frac{m_A L}{2} + P_B(m_A L + \frac{m_B L}{2}) + P_C(m_A L + m_B L + \frac{m_C L}{2}) \tag{5}$$

By property of probabilities, the following holds:

$$P_A + P_B + P_C = 1 \tag{6}$$

By factorizing in function of L and arranging in (5) the equation (7) is obtained

$$T_v = a + L[P_a\frac{m_A}{2} + P_B(m_A + \frac{m_B}{2}) + P_C(m_A + m_B + \frac{m_C}{2})] \tag{7}$$

The term $P_A\frac{m_A}{2}$ corresponds to the average number of slots to be traveled for storing a product of type A; the term $P_B(m_A + \frac{m_B}{2})$ corresponds to the average number of slots to be traveled for storing a product of type B, while the term $P_C(m_A + m_B + \frac{m_C}{2})$ corresponds to the average number of slots to be traveled for storing a product of type C.

The total capacity of category A, B and C within the DC to store goods can be expressed as follows:

$$N_A = 2m_A nh \tag{8}$$

$$N_B = 2m_B nh \tag{9}$$

$$N_C = 2m_C nh \tag{10}$$

Now, let's get the cost of consumption of energy expended in the movement of goods in height h. Considering an AGV to move the articles in height, we have the following expression for power consumption.

$$T_h = WHt \tag{11}$$

Where:

- W is the average mass of conveyed articles.
- t is the average time spent on moving the load a unit distance h.t es el tiempo medio gastado en mover la carga una unidad de distancia h.

By common expression of the average speed $v = \frac{Lh}{t}$ then $t = \frac{Lh}{v_e}$. Let E be the consumption of energy:

$$E = C_e W\frac{hLh}{v_e} \tag{12}$$

In (12) C_e is the cost of energy expended in moving a load of W kilos in height at a constant speed v_e, taking a time Lh. If other equipment is used to move materials in height, as it could be drones or an automated robot, expression (12) will change.

For an AGV equipment, the cost equation for a DC with the ABC method is (13):

$$C = 4dC_h[a+L(P_A\frac{m_A}{2}+P_b(m_A+\frac{m_B}{2})+P_C(m_A+m_B+\frac{m_C}{2}))+\frac{n(w+a)}{4}]+2dC_e[\frac{WLh^2}{v_e}] \tag{13}$$

The model can be expressed as in (14):

$$C = 4dC_h[a+L(P_A\frac{m_A}{2}+P_b(m_A+\frac{m_B}{2})+P_C(m_A+m_B+\frac{m_C}{2})+\frac{n(w+a)}{4}]+2dC_e[\frac{WLh^2}{v_e}] \quad (14)$$

subject to

$$N_A = 2m_A nh \quad (15)$$

$$N_B = 2m_B nh \quad (16)$$

$$N_C = 2m_C nh \quad (17)$$

By transforming the system by converting to the Lagrangian function see the reference Luenberger (1989) [12] , we have:

$$C = 4dC_h[a + L(P_A\frac{m_A}{2} + P_b(m_A + \frac{m_B}{2}) + P_C(m_A + m_B + \frac{m_C}{2}) + \frac{n(w+a)}{4}] + 2dC_e[\frac{WLh^2}{v_e}] + \quad (18)$$

$$\lambda_A[N_A - 2m_A nh] + \lambda_B[N_B - 2m_B nh] + \lambda_C[N_C - 2m_C nh] \quad (19)$$

By partial differentiation on m_A , m_B , m_C , n, h, λ_A , λ_B y λ_C we obtain.

$$\frac{\delta L_g}{\delta m_A} = 4C_h dL(\frac{P_A}{2} + P_B + P_C) - 2\lambda_A nh = 0 \quad (20)$$

$$\frac{\delta L_g}{\delta m_B} = 4C_h dL(\frac{P_B}{2} + P_C) - 2\lambda_B nh = 0 \quad (21)$$

$$\frac{\delta L_g}{\delta m_C} = 4C_h dL(\frac{P_C}{2}) - 2\lambda_C nh = 0 \quad (22)$$

$$\frac{\delta L_g}{\delta n} = 4C_h d\frac{(w+a)}{4} - 2\lambda_A m_A h - 2\lambda_B m_B h - 2\lambda_C m_C h = 0 \quad (23)$$

$$\frac{\delta L_g}{\delta h} = \frac{4dC_e WLh}{v_e} - 2\lambda_A m_A n - 2\lambda_B m_B n - 2\lambda_C m_C n = 0 \quad (24)$$

$$\frac{\delta L_g}{\lambda_A} = N_A - 2m_A nh = 0 \quad (25)$$

$$\frac{\delta L_g}{\lambda_B} = N_B - 2m_B nh = 0 \quad (26)$$

$$\frac{\delta L_g}{\lambda_C} = N_C - 2m_C nh = 0 \quad (27)$$

lete K_1 and K_2 be respectivelly:

$$K_1 = \frac{P_A}{2} + P_B + P_C \quad (28)$$

$$K_2 = \frac{P_B}{2} + P_C \quad (29)$$

As λ_A, λ_B and λ_C represent the shadow price of the corresponding resource, that is , for A, B or C, respectively, values of λ_A, λ_B and λ_C are obtained as shown in (25), (26), (27):

$$\lambda_A = \frac{2C_h dLK_1}{nh} \tag{30}$$

$$\lambda_B = \frac{2C_h dLK_2}{nh} \tag{31}$$

$$\lambda_C = \frac{C_h dLP_C}{nh} \tag{32}$$

Finally we obtain:

$$n = \sqrt[5]{\frac{16C_h L^3 Exp_1^2 C_e W}{(w+a)^3 C_h v_e}} \tag{33}$$

$$h = \sqrt[5]{\frac{(w+a)C_h^2 Exp_1 v_e^2}{8LC_e^2 W^2}} \tag{34}$$

$$m_A = \frac{N_A}{2}\sqrt[5]{\frac{(w+a)^2 C_e W}{2L^2 Exp_1^3 C_h v_e}} \tag{35}$$

$$m_B = \frac{N_B}{2}\sqrt[5]{\frac{(w+a)^2 C_e W}{2L^2 Exp_1^3 C_h v_e}} \tag{36}$$

$$m_C = N_C\sqrt[5]{\frac{(w+a)^2 C_e W}{2L^2 Exp_1^3 C_h v_e}} \tag{37}$$

where :

$$Exp_1 = K_1 N_A + K_2 N_B + \frac{P_C N_C}{2} \tag{38}$$

Theorem 1. *For the values of the Lagrangian multipliers, it holds that:*

$$\lambda_A > \lambda_B > \lambda_C \tag{39}$$

Proof: The only different term between the expressions of λ_A and λ_B is K_1 and K_2. Let us recall the corresponding expressions:

$$K_1 = \frac{P_A}{2} + P_B + P_C \tag{40}$$

$$K_2 = \frac{P_B}{2} + P_C \tag{41}$$

therefore if

$$P_A > 0 \quad \textit{and} \quad P_B > 0 \quad \textit{then} \quad K_1 > K_2 \quad \textit{and} \quad \lambda_A > \lambda_B. \tag{42}$$

On the other hand, it can be seen that if

$$P_B > 0 \qquad then \qquad \lambda_B > \lambda_C. \tag{43}$$

□

This theorem has the following implication, it shows that the shadow price of the space resource intended for items type A are greater than those of items type B and in turn both are greater than than type C's. The shadow price of a resource is a measure of its value or scarcity, in this case it represents how much it is saved when an extra slot unit is allocated to each of the types of dedicated storage areas A, B or C.

3 Experimenting with a test case

In order to validate the developed formulas, various calculations of warehouse distribution were performed using the data of (Onut et al, 2008) [2] for MRO (maintenance, repair, and operational) items showing ABC categories. Note that this is the only reference found in the literature with formulas for the case in three dimensions. In that document, the developed formulas are implicit and the problem was solved by the method of particle swarm optimization (PSO). The store used was programmed to handlefive families of products, including personal cleaning, home cleaning, chemical raw materials, electrical spare parts, and ceramics objects including 10, 14, 62, 11 and 7 types of articles, respectively. Before applying the formulas developed in this paper, a procedure is applied for determining which group according to the ABC method is assigned to a given product. Therefore, the goods in Class A are closer to the door, while class B at a distance of midrange and class C at a greater distance. After completing this process, 14 elements in class A, 24 class B elements and 12 elements of the class C were found. The probabilities of membership in each class were taken as 0.6, 0.3 and 0.1 respectively. The total flow of the warehouse is 120,000 pallets of products per year and the capacity of the warehouse is 6,000 pallets. The capacity of the warehouse is divided into size classes 3,000 (N_A), 2000 (N_B) and 1000 (N_C). With respect to the dimensions of the warehouse, there are available racks with loading by both sides of a total width (w) of 2.2 mt., a length of 0.9 mt. wide and 1 mt. height. The width of the aisles is 2 mt. and the width of doors 4 mt. The cost factors considered are two. First we considered the cost of use the AGV moving in the horizontal plane C_h, it is obtained per meter transported. Second we considered the energy cost generated by the movement in 3D of the AGV (C_e, it is obtained per meter transported per kilo and per the spended time in seconds. With this information, the summary data are:

- L=0.9 meters (length of storage unit or slot)
- N=6000 (total storage capacity in slots)
- a=2.0 meters (aisle width)
- d=120,000 (anual demand for pallets)
- C_h=US$ 1.13*10-3 (material handling cost in US$/mt)
- C_e= US$ 7.91*10-6 (energy cost US$/mt-Kg-seg)
- N_A=3,000 (number of pallets in category A)
- N_B=2,000 (number of palletes in category B)

Table 1: Results obtained using the developed formulas

Values	n	h	m_A	m_B	m_C	m
Obtained	15.06	5.01	19.89	13.26	6.63	39.77
Rounded	15	5	20	13	7	40

Table 2: Shadow prices for each type of space A,B, C.

λ_A	λ_B	λ_C
7.18	2.56	0.51

- N_C=1,000 (number of pallets in category C)
- P_A=0.6 (probability of product is category A)
- P_B=0.3 (probability of product is category B)
- P_C=0.1 (probability of product is category C)
- W=20 (mass,Kg)
- w=2.2 (meters)
- v_e=0.2 (meters/second)

With these values the dimensions that give the formulas for the number of shelves (n), number of containers in height (h), number of shelves type A (m_A), number of shelves type B (m_B) and number shelves type C (m_C) are shown in Table 1 below:

It is noted that the number of double shelves is 15 which gives 30 racks in total, with a number 40 slots, which added to the space occupied by the aisles gives an area of 2,520 square meters. The shadow price values are shown in Table 2 below

It is appreciated that the space dedicated items type A is more expensive than type B, and type C; and that the space dedicated to type B items is more expensive than the articles dedicated to type C. Thus each space unit type A contributes a value of 7.18 US$/slot-year to the cost of moving materials in a year, while each unit type B space contributes 2.56 US$/slot-year while the C type contributes 0.51 US$/slot-year. Thus, if there was additional space for warehousing this should go to space for items type A first and then items type B and C. The total average cost of warehousing separated by movement in the horizontal plane on the one hand and vertical movement for another, it is shown in Table 3.

Table 3: Average annual costs in US$

Cost of movement in the horizontal plane	21,971
Cost of movement in the vertical axis	5,222
Total Cost	27,192

This is the total cost corresponding to the storage and picking operations for the total annual demand, this gives an average cost of 4.53 US$/slot and it is consistent with the shadow prices obtained.

Conclusions and future works

In this paper new approach to the design of distribution centers was presented; particularly for a rectangular design with the heightwise movement. Explicit formulas to calculate the number of pallets, the number of double shelves, the number of unit spaces for each shelf and number of unit spaces in height were developed. The formulas obtained are easy to use. These have been obtained by a Lagrangian function obtained from a quadratic minimization problem with equality constraints. In the literature there are no explicit formulas in three dimensions by which this work is a valuable contribution to the design of distribution centers in three dimensions. Future works developing closed-form solutions as to those found in this paper should be continued, but conforming to a given area, since this is a typical situation in the company, in which the available surface poses constraints to the warehouse design problem.

Acknowledgment

The authors are very grateful to DICYT (Scientific and Technological Research Office), Project Number 061317DC and the Industrial Engineering (IE) Department, both of the University of Santiago of Chile for their support in this work. Also to IE graduates Ren'e Ibacache and Nicole Muńoz who helped in the model implementation.

Bibliography

[1] I. Derpich and J. Sepulveda (2016), A Model for Storage Facility Design with Energy Costs, *Computers Communications and Control (ICCCC), 2016 6th International Conference on*, IEEE Xplore, e-ISSN 978-1-5090-1735-5, DOI: 10.1109/ICCCC.2016.7496753, 147 - 150.

[2] Y. Bassan, Y. Roll and M.J. Rosenblatt (1980), Internal Layout Design of a Warehouse, *IIE Transactions*, 12(4): 317-322.

[3] S. Onut, U.R.Tuzkaya, D. Bilgehan (2008), A particle swarm optimization algorithm for the multiple level warehouse layout design problem, *Computers & Industrial Engineering*, 54: 783-799.

[4] P. Baker and M. Canessa (2009), Warehouse design: A structured approach , *European Journal of Operational Research*, 193: 425-436.

[5] J. Gu, M.Goetschalckx and L.F. McGinnis (2010), Research on warehouse design and performance evaluation: A comprehensive review, *European Journal of Operational Research*, 203:539-549.

[6] B. Rouwenhorst, B. , B. Reuterb, V. Stockrahmb, G.J. van Houtumc, R.J. Mantela and W.H.M. Zijmc (2000), Warehouse design and control: Framework and literature review, *European Journal of Operational Research* , 122(3): 515-533.

[7] S.S. Heragu, L. DU, R.J. Mantel and P.C. Schuur (2005), Mathematical model for warehouse design and product allocation *International Journal of Production Research*, 43(2):327-338.

[8] S. Zhou, Y. Gong and R. De Koster(2016), Designing self-storage warehouses with customer choice, *International Journal of Production Research*, 54(10): 3080-3104.

[9] K.J. Roodbergen and F.A. Vis (2006), A model for warehouse layout, *IIE Transactions*, 38:799-811.

[10] J. Li, M. Moghaddam and S.Y. Nof (2016), Dynamic storage assignment with product affinity and ABC classification - a case study, *Int J Adv Manuf Technol*, 84:2179-2194.

[11] R.L. Francis (1967), On some problems of rectangular warehouse design and layout, *Journal of Industrial Engineering*, 18: 595-604.

[12] D. G. Luenberger (1989), *Linear and non linear programming, Chapter 10*, Addison Wesley Iberoamericana, 299-305.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(1):53-60, February 2017.

Automated 2D Segmentation of Prostate in T2-weighted MRI Scans

J. Jucevičius, P. Treigys, J. Bernatavičienė, R. Briedienė,
I. Naruševičiūtė, G. Dzemyda, V. Medvedev

Justinas Jucevičius*, Povilas Treigys, Jolita Bernatavičienė,
Gintautas Dzemyda, Viktor Medvedev
Vilnius University, Institute of Mathematics and Informatics
Lithuania, 08663 Vilnius, Akademijos str. 4
justinas.jucevicius@mii.vu.lt, povilas.treigys@mii.vu.lt, jolita.bernataviciene@mii.vu.lt,
gintautas.dzemyda@mii.vu.lt, viktor.medvedev@mii.vu.lt
*Corresponding author: justinas.jucevicius@mii.vu.lt

Rūta Briedienė, Ieva Naruševičiūtė
Vilnius University, National Cancer Institute
Lithuania, 08660 Vilnius, Santariškių str. 1
ruta.briediene@nvi.lt, ieva.naruseviciute@gmail.com

Abstract: The prostate cancer is the second most frequent tumor amongst men. Statistics shows that biopsy reveals only 70-80% clinical cancer cases. Multiparametric magnetic resonance imaging (MRI) technique comes to play and is used to help to determine the location to perform a biopsy. With the aim to automating the biopsy localization, prostate segmentation has to be performed in magnetic resonance images. Computer image analysis methods play the key role here. The problem of automated prostate magnetic resonance (MR) image segmentation is burdened by the fact that MRI signal intensity is not standardized: field of view and image appearance is for a large part determined by acquisition protocol, field strength, coil profile and scanner type. Authors overview the most recent Prostate MR image segmentation challenge results and provide insights on T2-weighted MRI scan images automated prostate segmentation problem by comparing the best obtained automatic segmentation algorithms and applying them to 2D prostate segmentation case. The most important benefit of this research will have medical doctors involved in the management of the cancer.
Keywords: computer image processing, 2D prostate segmentation, magnetic resonance imaging (MRI), T2-weighted scan.

1 Introduction

Various data mining methods find application in medicine and health care. Large studies are presented in [9], [22], there are a lot of recent applications, eg. [7], [8], [25], [27]. This paper deals with medical image analysis, particularly with magnetic resonance images and prostate cancer. World Cancer Research Fund International states that prostate cancer is the second most frequent tumor among men and fourth most common among both genders. Lithuanian cancer registry data from 2012 shows that prostate cancer prevalence reaches 34% amongst men aging 55 to 74 years. The mortality from the prostate cancer is the second most common after the lung cancer amongst men and the third most common among both genders after lung and stomach cancers [23]. According to European Association of Urology guidelines, from 10 to 12 core biopsy is recommended in case of prostate-specific antigen level elevation and/or suspicious digital rectal examination findings [1]. Contemporary, random systematic prostate biopsy strategy includes failure to detect clinically significant cancer. Undersampling in up to 30% of cases

with clinically significant tumors being missed on initial biopsy. This diagnostic uncertainty can lead to repeat biopsy, delayed detection of significant disease and disease overtreatment [20]. Despite inaccuracy, biopsy remains the main way that can unambiguously detect prostate cancer if performed on the right location.

The latest recommendations in prostate cancer care include multiparametric magnetic resonance imaging (mpMRI) as the tool for prostate cancer diagnosis, characterization, staging as well as risk stratification among men, who need active surveillance. Today's computer-aided detection programs are associated with European Society of Urogenital recommendations for prostate evaluation by mpMRI are the attractive subject for research that incorporates the development of new image analysis [4], [16], [26] and data mining algorithms. Usually prostate localization and segmentation in magnetic resonance images is done by hand; however, it takes a lot of time and can be inaccurate. This causes the need for software to aid in automated prostate segmentation [13] in a standardized manner. Thus the main objective of this study was to overview the current situation of the field and to adopt today's best methods developed to a procedure named Prostate Template Biopsy [30].

2 Research Motivation and Experiment Setup

The problem of automated prostate MR image segmentation is burdened by the fact that most researchers cannot compare the effectiveness of different algorithms due to either troublesome implementation without the help of the original author or algorithm being closed source. What further aggravates the problem is that MRI signal intensity is not standardized and image appearance is for a large part determined by acquisition protocol, field strength, coil profile, and scanner type [18].

It is challenging task to identify and segment objects within images due to high object and background variability. In computer vision image segmentation can be described as procedure of finding group of image pixels that shares the same feature and describe homogeneous image region. Analysis may take into account object texture, intensities shape and etc [3]. However, the problem arises when investigative object is compound of several regions, object edges are blurred or object's shape is varying. Here the segmentation techniques are applied that makes further processing easier: to each group of pixels that describes the region the unique region label is assigned.

Researchers [10] have put a big effort to summarize prostate segmentation methods. Study reveals four different groups of segmentation algorithms that fall into:

- Contour and shape based methods exploit contour and shape information to accomplish segmentation task.
- Region-based methods analyses predominant prostate intensity distribution in different modalities.
- Supervised and unsupervised classification methods aim at obtaining a partition of the feature space into a set of labels for different regions. For this task classifier and/or clustering techniques are used.
- Hybrid methods combine a priori boundary, shape, region, and feature information of the prostate gland. Methods of this group are robust to noise and produce superior results in presence of shape and texture variations.

Several successful Grand Challenges in Medical Imaging have been organized in recent years to deal with similar issues in the fields of coronary image analysis, retinal image analysis, liver

segmentation on computed tomography (CT) scan, lung registration on CT scan, brain segmentation on MRI and prostate segmentation on MRI. Prostate MR Image Segmentation challenge (PROMISE12) [18] was designed to allow comparison of segmentation algorithms on the basis of robustness and performance by providing hundred T2-weighted MRI scans gathered from four different institutions. T2-weighted MR images were used because they contain most anatomical detail and most current researchers focus on them for segmentation. With the aim to evaluate segmentation results the PROMISE12 challenge introduce those widely used metrics:

- dice similarity coefficient (DSC) [15],
- absolute relative volume difference [12],
- average boundary distance [12],
- 95% Hausdorff distance (HD) [5].

The score for each metric was mapped to a relative value between 0 and 100. The scores for all metrics were then averaged to obtain a score per case, and the average of score over all cases was calculated and used to rank algorithms. Table 1 presents prostate detection algorithms and depicts obtained results of the challenge. Here, A stands for an automatic method that requires no user interaction; S - semi-automatic method that requires some initial user interaction; I - interactive method that requires full user interaction from the beginning of segmentation until the end.

Table 1: Prostate detection algorithms [2]

Team name	**Score**	**Type**	**The idea of the algorithm**
Imorphics	84.36	A	Apply active appearance models on images with increasing resolution by refining results
ScrAutoProstate	83.49	A	Intensity normalization, marginal space learning, boundary refinement
CBA	80.66	I	Smart paint algorithm segments prostate by sweeping the mouse cursor in the object or background
SBIA	78.34	A	Multi-atlas based segmentation, zooming into vicinity
Grisles	77.56	S	Detect location of the prostate, use probabilistic active shape model for boundary detection
Robarts	77.32	S	Contour evolution with the integration of the generic star shapes prior
ICProstateSeg	76.06	A	Multi-atlas based segmentation using local appearance-specific atlases
Utwente	75.23	S	Active appearance models based segmentation
Cimalab	74.68	A	Atlas-based segmentation that selects the most similar templates using multi-scale SURF analysis and applies linear combination
DIAG	73.30	A	Multi-atlas based segmentation using selective and iterative method for performance level estimation algorithm for merging atlas labels
ETHZ	72.38	A	Graph cut based segmentation using image features, context information, semantic knowledge
UBUdG	70.44	S	Random decision-based forest for classification and the propagation of region for segmentation
Rutgers	65.97	A	Segmentation based on active appearance models

Two best fully automated algorithms from the PROMISE12 challenge named Imorphics [28], [11] and ScrAutoProstate [2] having scores of 84.36 and 83.49 respectively were chosen for further analysis. The pipelines of the Imorphic and ScrAutoProstate methods are presented in Figure 1.

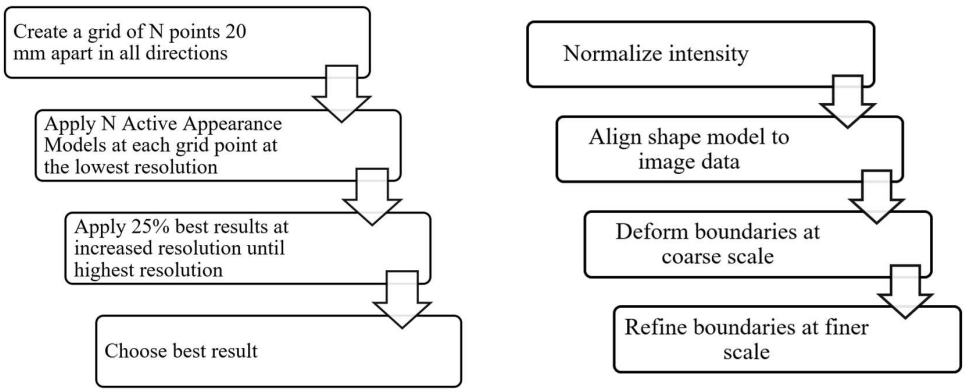


Figure 1: Pipelines of the Imorphics (on the left side) and ScrAutoProstate (on the right side) prostate segmentation methods

Both algorithms were adapted to be suitable for prostate segmentation in 2D space instead of the 3D space originally used in the challenge. All adaptations were performed by choosing the corresponding algorithm designed for 2D space without any modifications. Prostate segmentation in 2D space was chosen because of the biopsy procedure named Prostate Template Biopsy [30]:

- It has an accuracy of 95%.
- It is becoming more and more prevalent.
- It uses only two slices out of 10-40 used in 3D space, thus reduces complexity and computational time needed for segmentation.

The Imorphics method [11], [28] belongs to a group of hybrid prostate segmentation methods. The method converts voxel-based segmentation to surface by using the marching-cube algorithm [19]. Segmentation allows indicating whether surface belongs to the prostate gland or not. Next, to construct statistical prostate appearance model, so called mean image, a set of possible deformations are introduced and registered together [6]. This step allows finding the minimum information needed to code the mean reference image with the deformations that map mean image to each example image. Finally, to obtain features of the image authors use active appearance model that control shape and texture. The model computes the closest match of the prostate gland shape in sample image using the least squares sum of residuals. For the efficiency Jacobian matrix describing the average change in residuals with respect to changes in model parameters on a training set is pre-computed and for the initial estimate of the model, authors introduce grid of starting search points across the image, typically 20 mm apart in all directions.

At the initial stage the ScrAutoProstate method [2] applies region based brightness and contrast Poison editing technique [21]. Then, with the view to constructing statistical shape model, training segmentation mask (represented as a mesh) is constructed the same way as it was presented in the Imorphics method by using the marching cube technique. Then, orientation and scale variations in those statistical shapes are removed after application of Procrustes analysis [14]. The remaining shape variability is finally represented with a point distribution model, and the strongest shape models are extracted through principal component analysis. While testing algorithm with the unseen image, initial segmentation is obtained by applying teh Marginal Space Learning [29] algorithm that computes unknown pose and shape coefficients. The previous step aligns shape model to the image and gives good initial segmentation. At the final step, mesh surface is refined by using non-rigid, hierarchical boundary deformation [17]. Introduced refinement iteratively displaces mesh vertices along the mesh surface normal.

3 Experiments and Results

In this research, we have used images from 50 cases provided by the PROMISE12 [18] challenge. Initial three-dimensional MRI data set used by the challenge was split into separate images representing every slice. The number of slices per case varied from 15 to 54. Images that did not contain prostate were removed. Remaining images were split into two groups representing prostate apex and base parts. From each group, middle image was chosen to represent the apex and base part of the prostate respectively having 100 images in total. All images were gathered from four different institutions and varied in resolution:

- 256 x 256 pixels;
- 320 x 320 pixels;
- 512 x 512 pixels.

To test the performance of the 2D prostate segmentation algorithms we have used leave-one-out cross-validation [24], where each image is segmented using a model, built from the training set with this image removed, i.e. image is "excluded" from the set of images and used for validation. The result was then compared against the reference segmentation. Results, presented in Table 2, show that transition from 3D to 2D space prostate gland segmentation can be accomplished with the minimum loss of accuracy. DSC measure when compared Imorphic and ScrAutoProstate algorithms decreased by 0.04 and 0.01 points respectively. It results that 2D segmentation is successful and opens the possibility to apply Prostate Template Biopsy procedure that unifies biopsy of the prostate gland procedure.

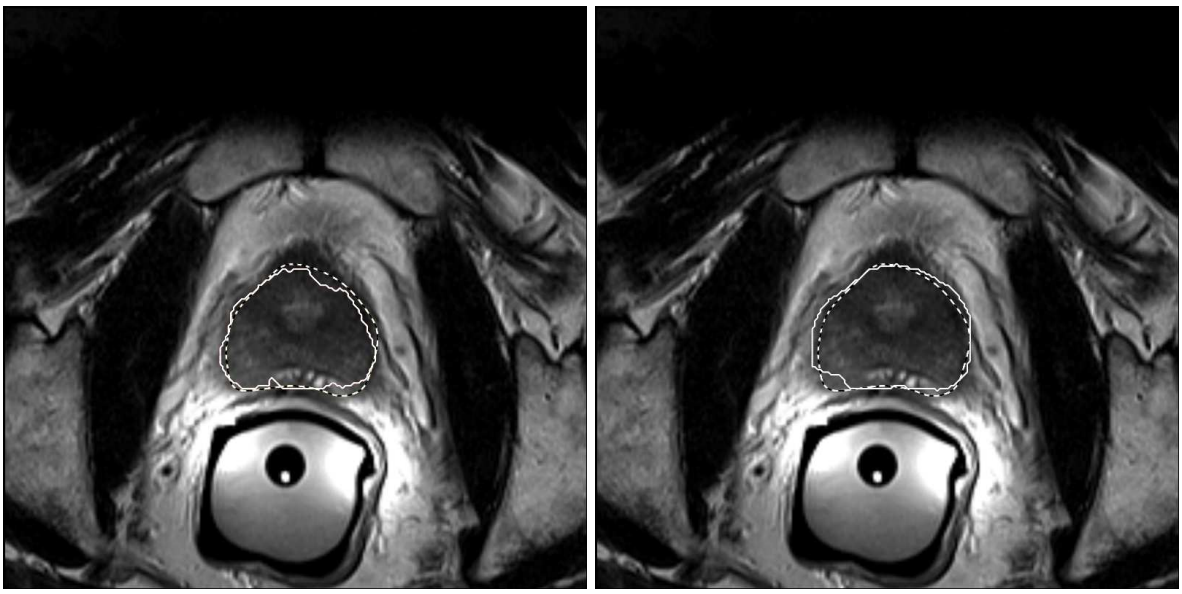


Figure 2: Segmentation results in 2D space of both Imorphics and ScrAutoProstate methods on the left and right respectively. In dashed line - reference segmentation, in white line - automated segmentation.

DSC and 95% Hausdorff distance statistics were selected to match those reported in original algorithms for comparison. The DSC measures the amount of overlap between the reference segmentation and the automated segmentation. DSC can range from zero to one, where zero represents no overlap and one corresponds to identical segmentations. The directed HD identifies

Table 2: Comparison of prostate segmentation results between 2D and 3D space by analyzing T2-weighted scan data

Algorithm/Measures	**Imorphics**		**ScrAutoProstate**	
	2D	**3D**	**2D**	**3D**
DSC	0.84	0.88	0.83	0.82
HD	6.26	4.17	9.73	not specified
Execution Time (s)	51.11	480	0.24	2.30

the point on the reference segmentation that is the farthest from any point on the model segmentation and measures the distance from this point to the nearest point on the model segmentation. The 95% HD is less sensitive to outliers than HD since it considers the point representing the 95^{th} percentile of the distances instead of the farthest. Figure 2 shows sample segmentation results. Visually, both algorithms produces quite similar segmentation results. Metrics measures are as follows: Imorphics DSC and HD values were 0.93 and 4.12; ScrAutoProstate DSC and HD values were 0.93 and 6.99.

Conclusions

3D prostate gland segmentation cannot be directly adapted to today's best methods developed to a procedure named Prostate Template Biopsy. That leads to an investigation whether 3D prostate gland segmentation can be transferred to 2D segmentation while keeping the same segmentation accuracy with the possibility to speed up algorithms execution time. 2D images representing slices are starting point for the analysis and further conclusions.

The investigation presented in this paper has shown that there is a minor loss in algorithms accuracy when moving prostate gland segmentation from 3D space to 2D space. Dice similarity coefficient when compared Imorphics and ScrAutoProstate algorithms have changed by 0.04 and 0.01 points, respectively. However, Imorphics performed slightly better at the cost of execution time.

As expected, both algorithms improved execution time by almost 10 times in 2D in comparison to 3D. Despite both algorithms perform quite well, the development is necessary for practical usage where automated prostate segmentation in MRI is needed. The most important benefit of this research will have medical doctors involved in the management of the cancer: radiologists, urologists, histopathologists, radiotherapists, oncologists.

Bibliography

[1] Archip, N., Clatz, O., Whalen, S., Kacher, D., Fedorov, A., et al. (2007), Non-rigid alignment of preoperative MRI, fMRI, and DT-MRI with intra-operative MRI for enhanced visualization and navigation in image-guided neurosurgery, *NeuroImage*, DOI: 10.1016/j.neuroimage.2006.11.060

[2] Birkbeck, N., Zhang, J., Requardt, M., Kiefer, B., Gall, P., Kevin Zhou, S., (2012), Region-specific hierarchical segmentation of MR prostate using discriminative learning, *Proc. Med. Image Comput. Comput.-Assisted Intervention Conf. Prostate Segment. Challenge*, 4-11.

[3] Borenstein, E., Malik, J. (2006, June). Shape guided object segmentation. *In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 1: 969-976.

[4] Buteikienė, D., Paunksnis, A., Barzdžiukas, V., Bernatavičienė, J., Marcinkevičius, V., Treigys, P. (2012); Assessment of the optic nerve disc and excavation parameters of interactive and automated parameterization methods. *Informatica*, 23(3): 335-355.

[5] Chandra, S., Dowling, J., Shen, K., Raniga, P., Pluim, J., Greer, P., Salvado, O., Fripp, J. (2012); Patient specific prostate segmentation in 3-D magnetic resonance images. *IEEE Trans. Med. Imaging*, 31: 1955-1964.

[6] Cootes, T., Petrovi, C., Schestowitz, R., Taylor, C. (2005); Groupwise construction of appearance models using piece-wise affine deformations. *16th British Machine Vision Conference*, 2:879-888.

[7] Costin, H., Bejinariu, S., & Costin, D. (2016); Biomedical Image Registration by means of Bacterial Foraging Paradigm. *International Journal of Computers Communications & Control*, 11(3): 331-347.

[8] Dua, S., & Acharya, R. (Eds.). (2016); *Data Mining in Biomedical Imaging, Signaling, and Systems*, CRC Press, 2016.

[9] Ghavami, P.K. (2014). *Clinical Intelligence the Big Data Analytics Revolution in Healthcare: A Framework for Clinical and Business Intelligence* Createspace Independent Pub., 2014.

[10] Ghose, S., Oliver, A., Marti, R., Llado, X., Vilanova, J., et al. (2012); *A Survey of Prostate Segmentation Methodologies in Ultrasound, Magnetic Resonance and Computed Tomography Images*, Computer Methods and Programs in Biomedicine, Elsevier, hal-00695557, 2012.

[11] Graham, V., Gwenael, G., Mike, B. (2012); Fully Automatic Segmentation of the Prostate using Active Appearance Models, PROMISE12 challenge website.

[12] Heimann, T. et al. (2009); Comparison and Evaluation of Methods for Liver Segmentation From CT Datasets, *IEEE Transactions on Medical Imaging*, 28(8): 1251-1265, DOI: 10.1109/TMI.2009.2013851.

[13] Hemanth, D. J., Anitha, J., & Balas, V. E. (2015); Performance Improved Modified Fuzzy C-Means Algorithm for Image Segmentation Applications. *Informatica*, 26(4): 635-648.

[14] Kendall, D. (1989); A Survey of the Statistical Theory of Shape. *Statistical Science*, 4(2): 87-99.

[15] Klein, S., van der Heide,I., Lips, M., van Vulpen, M., Staring, M., Pluim, J. (2008); Automatic segmentation of the prostate in 3D MR images by atlas matching using localized mutual information; *Med. Phys*, 35(4):1407-1417.

[16] Lekas, R.et al. (2008); Monitoring changes in heart tissue temperature and evaluation of graft function after coronary artery bypass grafting surgery. *Medicina*, 45(3): 221-225.

[17] Ling, H., Zhou, S.K., Zheng, Y., Georgescu, B., Suehling, M., Comaniciu, D. (2008); Hierarchical, learning-based automatic liver segmentation. CVPR, *IEEE Computer Society*, 1-8.

[18] Litjens, G., Toth, R., van de Ven, W., Hoeks, C., Kerkstra, S., et al. (2014); Evaluation of prostate segmentation algorithms for MRI: The PROMISE12 challenge, *Medical Image Analysis*, 18(2):359-373.

[19] Lorensen, W., Cline, H. (1987); Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4): 163-169.

[20] Mottet, N., Bellmunt, J., Briers, E., Bergh, R., Bolla, M., Casteren, N., e. al. (2015); *Guidelines on prostate cancer*. European Association of Urology, 2015.

[21] Perez, P., Gangnet, M., Blake, A. (2003); Poisson image editing, *ACM Trans.* Graph. 22(3): 313-318.

[22] Reddy, C. K., & Aggarwal, C. C. (Eds.). (2015); *Healthcare data analytics*, CRC Press, 2015.

[23] Smailytė, G., Aleknavičienė, B. (2015); *Vėžys Lietuvoje 2012 m. Nacionalinio vėžio instituto vėžio kontrolės ir profilaktikos centras*, 2015.

[24] Sylvain, A., Alain, C. (2010); A survey of cross-validation procedures for model selection. *Statist. Surv*, 4: 40–79.

[25] Termenon, M., Grana, M., Savio, A., Akusok, A., Miche, Y., Bjork, K. M., & Lendasse, A. (2016); Brain MRI morphological patterns extraction tool based on Extreme Learning Machine and majority vote classification. *Neurocomputing*, 174: 344-351.

[26] Treigys, P., Šaltenis, V., Dzemyda, G., Barzdžiukas, V., & Paunksnis, A. (2008); Automated optic nerve disc parameterization, *Informatica*, 19(3): 403-420.

[27] Trigui, R., Miteran, J., Sellami, L., Walker, P., & Hamida, A. B. (2016); A classification approach to prostate cancer localization in 3T multi-parametric MRI, *Advanced Technologies for Signal and Image Processing (ATSIP), 2016 2nd International Conference on*, IEEE, 113-118.

[28] Vincent, G., Guillard, G., Bowes, M. (2012); *Fully automatic segmentation of the prostate using active appearance models*, MICCAI Grand Challenge: Prostate MR Image Segmentation, 2012.

[29] Zheng, Y., Barbu, A., Georgescu, B., Scheuering, M., Comaniciu, D. (2008). Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features. *IEEE Trans. Med. Imag.* 27(11): 1668-1681.

[30] Prostate Template Biopsy. Essexurology.co.uk. Retrieved 2016 July 30, 2016. from *http://www.essexurology.co.uk/prostate_ template_ biopsy.php*.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(1):61-75, February 2017.

Big Data on Decision Making in Energetic Management of Copper Mining

C. Lagos, R. Carrasco, G. Fuertes, S. Gutiérrez, I. Soto, M. Vargas

Carolina Lagos*
Facultad de Administración y Economía,
Universidad de Santiago de Chile
Av. Libertador Bernardo O'Higgins 3363,
Santiago, Chile
*Corresponding author: carolina.lagos@usach.cl

Raúl Carrasco
1. Departamento de Ingeniería Eléctrica,
Universidad de Santiago de Chile
Av. Ecuador 3519, Santiago, Chile
2. Departamento de Matemáticas y Ciencias de la Computación, Universidad de Santiago de Chile
Las Sophoras 173, Santiago, Chile
raul.carrasco.a@usach.cl

Guillermo Fuertes
Departamento de Ingeniería Industrial,
Universidad de Santiago de Chile
Av. Ecuador 3769, Santiago, Chile
guillermo.fuertes@usach.cl

Sabastián Gutiérrez
1. Facultad de Ciencias Económicas y Administrativas,
Universidad Central de Chile
Lord Cochrane 417, Santiago, Chile
2. Facultad de Ingeniería, Universidad Andres Bello
Antonio Varas 840, Providencia, Santiago, Chile
sebastian.gutierrez@ucentral.cl

Ismael Soto
Departamento de Ingeniería Eléctrica,
Universidad de Santiago de Chile
Av. Ecuador 3519, Santiago, Chile
ismael.soto@usach.cl

Manuel Vargas
Facultad de Ingenieria, Universidad Andres Bello,
Santiago, Chile
manuel.vargas@unab.cl

Abstract: It is proposed an analysis of the related variables with the energetic consumption in the process of concentrate of copper; specifically ball mills and SAG. The methodology considers the analysis of great volumes of data, which allows to identify the variables of interest (tonnage, temperature and power) to reach to an improvement plan in the energetic efficiency. The correct processing of the great volumen of data, previous imputation to the null data, not informed and out of range, coming from the milling process of copper, a decision support systems integrated, it allows to obtain clear and on line information for the decision making. As results it is establish that exist correlation between the energetic consumption of the Ball and SAG Mills, regarding the East, West temperature and winding. Nevertheless, it is not observed correlation between the energetic consumption of the Ball Mills and the SAG Mills, regarding to the tonnages of feed of SAG Mill. In consequence, From the experimental design, a similarity of behavior between two groups of different mills was determined in lines process . In addition, it was determined that there is a difference in energy consumption between the mills of the same group. This approach modifies the method presented in [1].[a]
Keywords: Copper mining, energetic efficiency, big data, process management.

[a]Reprinted (partial) and extended, with permission based on License Number 3962080057504 © [2016] IEEE, from "Computers Communications and Control (ICCCC), 2016 6th International Conference on".

1 Introduction

Market conditions, global competence and care of environment have created the urgent need to improve the energetic efficiency. So, efficient management of energetic resources for big industrial customers shaping up a critical aspect. It is important for industries to economize in the

use of energy and make it sustainable in long terms. It has emerged a significant market and a technological opportunity for the tools and technologies that allow an efficient management of energetic resources.

On the other hand, the quality and velocity of information interchange, requires a cooperation commitment among all the participants in the collecting process. Among the reasons for a non-efficient flux of information are: (1) unfulfilling relations among the different levels that receive and process the information, (2) lack of preventive maintenance programs in the measurement equipment and (3) intrinsic factors of the process.

Data science and its predictive analysis its getting more and more important for the production of goods and services [2], with applications from data monitoring for the energetic consumption [3], system for the prediction of maintenance [4, 5]. Decision Support Systems (DSS) can represent a solution, when human intervention is necessary [6–8], by using of techniques in the improvement of safety [9], and also, in the prediction for the analysis of natural resources consumption [10] and in models of optimization of processes [11]. This document discusses techniques and tools by consumption of energy data in mining, and the possible contribution that can do in the data analysis, for the decision making in the energetic efficiency. Filip (2008) concludes that "the development and application of intelligent DSS can help enterprise cope with problems and uncertainty and complexity, to increase efficiency and competitivity in production networks" [6].

The optimization and the control of the minerals processing plant, such as copper, has generate new research lines. With the increasing availability of data basis coming from the copper productive process, the amount of data gathered at an exponential rhythm. This creates and opportunity for the mining companies to analyse and optimized their operations observing those data. Then as productive operations become bigger, with more variables, it is crucial for the direction and the area managers, to exploit the information in an efficient and timely way. To make this happen, it is important the synchronization among all the areas, through joint planning of all the areas participating in the data collection, and shared information [12]. Other related works related with copper mining can be found in reference [13–15].

A previous preliminary work can be found in reference [1], that Data analysis methods related to energetic consumption in copper mining.

The main contribution of this paper is the generation of a systematic model for the analyses of the data and the use of statistical tools, allowing improvement of the energetic efficiency.

This organization of the paper is: section 2, describes the problem of the sales of the mineral, knowing the energetic costs of the process and the evolution on time of the sales; section 3, details the methodology and the different focuses of big data; section 4, shows the design and analysis of results of the processing of big data, coming from the concentration process of copper, for the energetic efficiency and final conclusions of the research.

2 Description of the problem

With the purposes of improving the decisión making, data mining can be considered a tool of quantitative analysis, that still needs more exploration for the energetic management. In this sense, Data mining and knowledge discovery in data basis have been attracting a significant amount of research [16], particularly to be used in the mining processes for its high data level [17].

The energetic costs hav been a relevant factor in the structure of costs of the mining processes.It is so, that the energy expenditure (electricity and fuel) it is the order of 15% of the operational cost (2008-2011), which increase affected negatively the operational margins, since a maximun of 65% in 2006, up to a 22% in 2014 [18]. So for a better understanding of the costs of

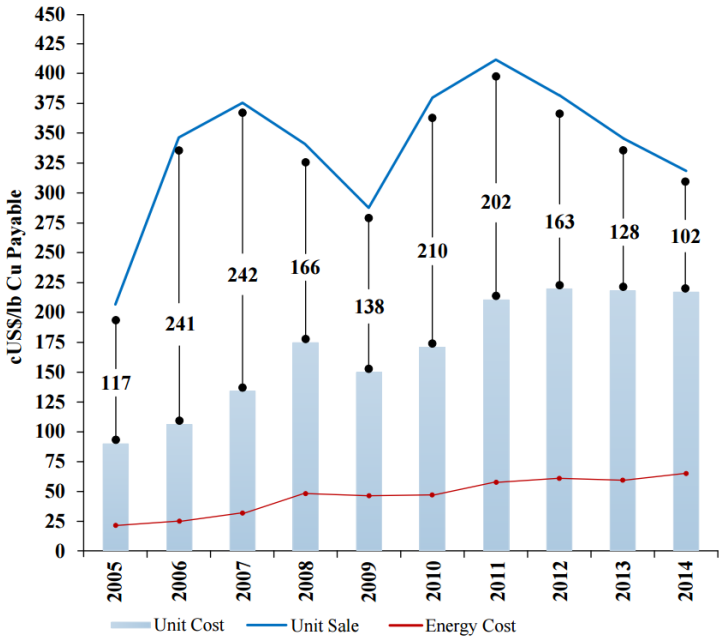


Figure 1: Costs and unit sales in the copper mining in Chile between 2005 and 2014

the process, it is required to study more deeply the behavior of the management of the energetic efficiency, with the intention of founding the optimun energetic consume [19].

On the other hand, analyzing the unit costs of exploitation (dollar cents per pund of payable copper), it is observed that costs have increased, but not in the same direction than sales, which is appreciated comparing costs and unit sales in Figure 1. Besides the same graphic shows the decrease to a little more than the half of the average operational process, from 210 cUS$/lb for the year 2010 up to 102 cUS$/lb in 2014 [18].

The previous graphic, also accounts for the decrease in the operational margins that have experienced the mining companies in the last years, due to the increase in costs and less copper price.

3 Problem formulation

Mining production companies as Codelco, have been collecting and storing increasingly more data of their productive processes, which offer an enormous potential as source of new knowledge [20], given the complexity of the data and the mining process, this document is used to data analysis [21, 22]. The data mining for the analysis of processes efforts to establish correlations between the key indicators a data mining for analysis of process strives to establish correlations between indicators of performance and the consume in the process. If the correlations can establish with the enough precision, it is possible to predict the behavior of energy, given the current consumption and the attributes of the process. In this study, were used several techniques:

1. Markov Chain Monte Carlo
   This paper uses the Markov chain Monte Carlo (MCMC) algorithms for imputing data, are a widely used and well-studied methodology that can be used to draw samples from posterior distributions [23]. The sampling is efficient for estimating parameters [24].

2. Canonical correlation analysis

Canonical correlation analysis (CCA) is a known multivariate analysis method, the objective is to find and quantify the correlations between two sets of multidimensional variables [25]. In this study the CCA, it is used as a right tool to establish the correlation between the energetic consumption as a dependant variable, and the independant variables of the Ball Mills and SAG Mill. CCA can be used recursively depending on the amount of dependant and independant variables that can be related, forming possible variables and canonical correlations. [26, 27], uses CCA in industrial processes to perform the troubleshooting. On the other hand [28], proposes a model of a recursive and adaptative process, to improve the precision of variables of the process in time.

3. Design and Analysis of Experiments
The design and analysis of experiments are classical statistic models whose objective is to find out if some determined factors influence in a variable of interest, if exists influence of any factor and quantify the effect it has on it. The use of the design and analysis of experiments allow us to measure the effect of the mil as a factor on the energetic consume that is the variable of interest of this work, achieving quantify the effect it has the mil factor on the energetic consumption [29]. In this sense it is used in different problems of energy [23, 30, 31].

4 Design and analysis of results

The analysis of the unit of the article is the production process of copper in the biggest state company of Chile, in the Division Codelco Chuquicamata. The data analysis, is done through a three phases procedure: Imputation of data for the missing information; Study to correlate between the variables that influence in the energetic consumption through a multivariable model of Canonical Correlation; and a comparison between means of consumption of the mills to verify potential significant differences between them.

4.1 Variables of study

The study is performed on six mills that operate in the process of the concentrator: Mill of Ball 16A (MOBO 16A); Mill of Ball 16B (MOBO 16B); Mill of Ball 17A (MOBO 17A); Mill of Ball 17B (MOBO 17B), Mill SAG 16 (SAG 16) and Mill SAG 17 (SAG 17), as shown in Figure 2. The defined variables for the analysis of the energetic consumption are divided in two groups. In the first one, defined as mills MOBO 16A, MOBO 16B and SAG 16 and the second group is defined as: MOBO 17A y MOBO 17B and SAG 17, for both cases, the variable of study are: Consumed energy, West Temperature, East Temperature, Winding Temperature and the Tonnage, according with the defined variables with the Maintenance Management of Codelco Chuquicamata.

4.2 Previous Analysis

In Fig. 3 it is presented the potential of, MOBO 16B. It is observed periods without information and the presence of atypical data in the last three years, which should affect the analysis to look for solutions to the proposed problem.

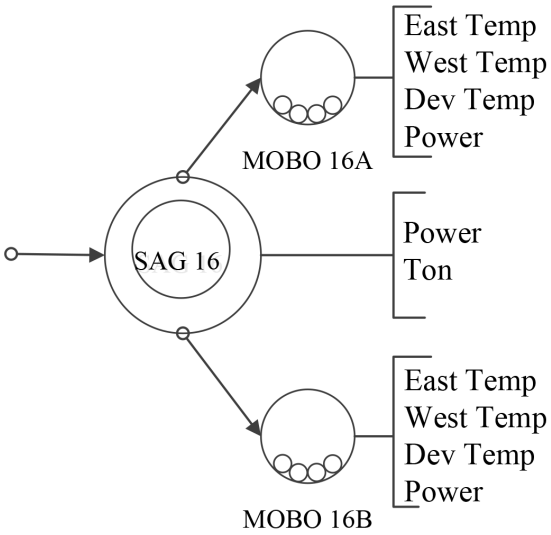


Figure 2: Measure Variables in the SAG mill and the mills of Ball MOBO 16A, MOBO 16B, in the concentrate process

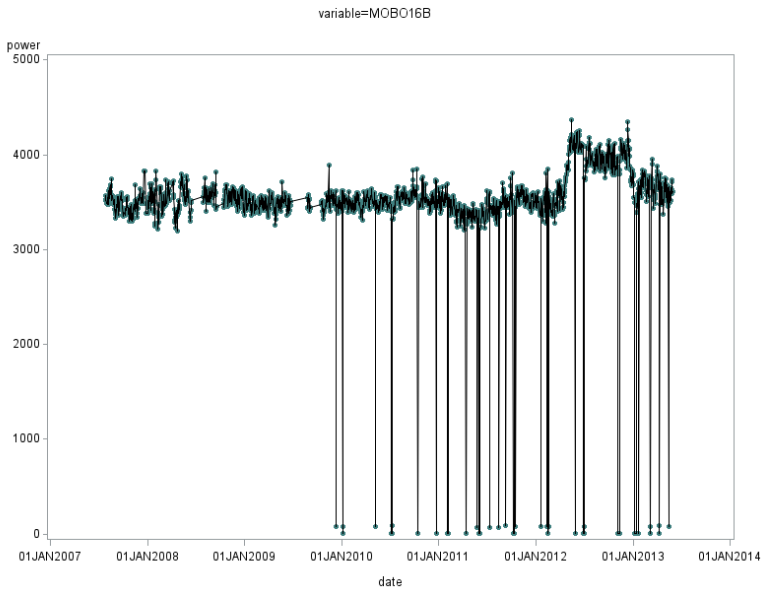


Figure 3: Power of MOBO 16B

As example in Fig. 4 it is shown the values of the power variable of the MOBO 16B. In this graphic are appreciated the outlier values that will be correct thanks to the MCMC procedure. Fig. 5 shows the dispersal among variables (power, tonnage,east temperature,west temperatura and winding temperature), at all times, it is observed that data do not distribute normally since the Kernel distribution does not adjust to the normal distribution.

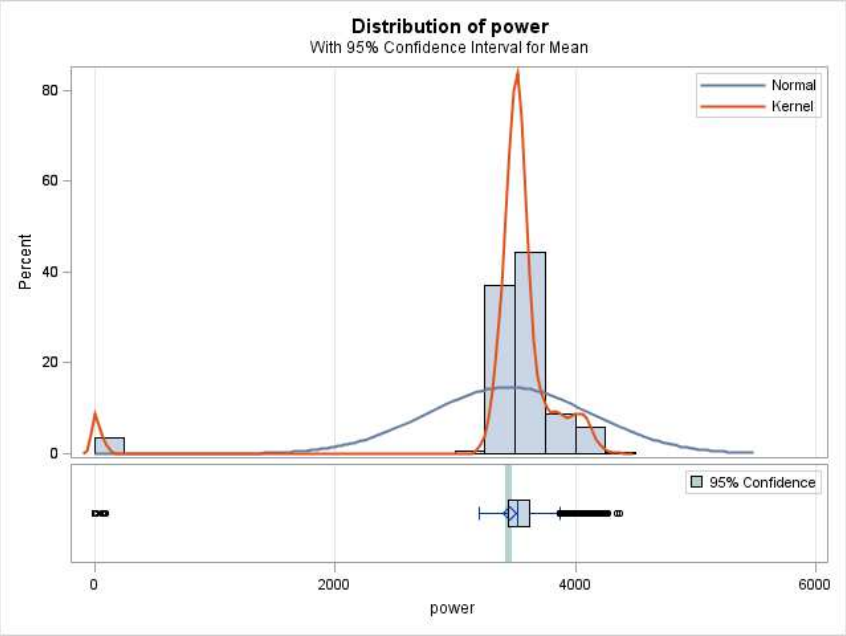


Figure 4: Distribution of the power variable in MOBO 16B

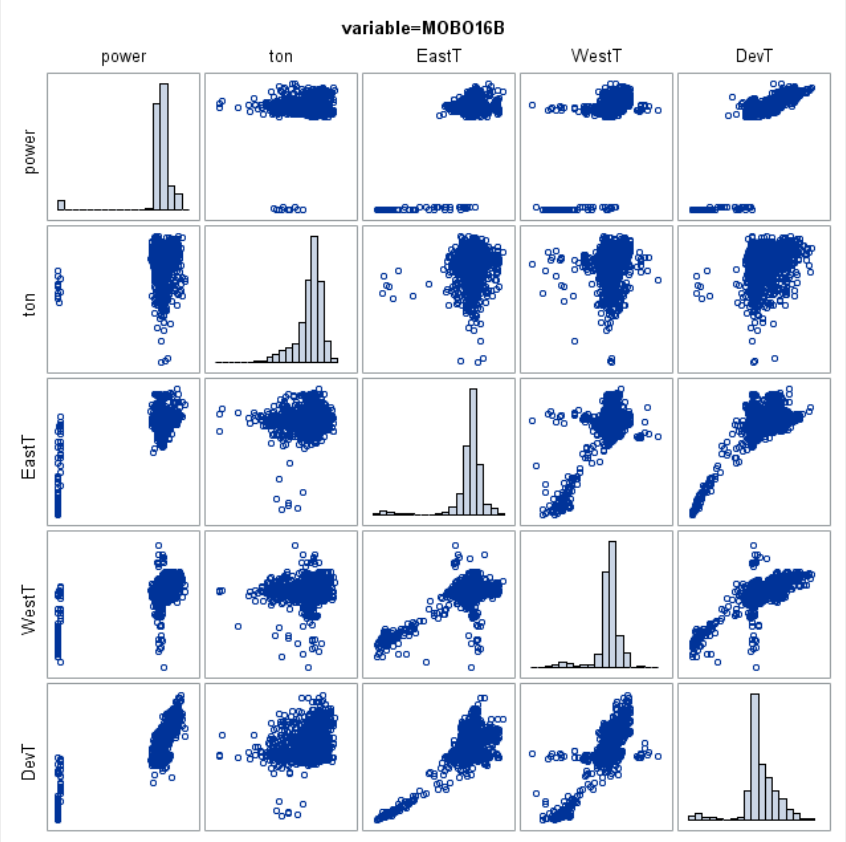


Figure 5: Plot of the correlations of the variables of the MOBO 16B

According to Table 1, a 10.77% of observations posses at least one missing data, so, the multiple imputation is used to recovered the lost information and correct the atypical data.

Table 1: Group of MOBO 16B data, existing and non existing

Group	East T.	West T.	Wind T.	Power	Frequency	Percentage
1	X	X	X	X	1905	89.23%
2	X	X	X	.	60	2.81%
3	X	X	.	X	1	0.05%
4	X	.	X	X	1	0.05%
5	X	.	.	X	3	0.14%
6	.	X	X	X	1	0.05%
7	.	X	.	X	1	0.05%
8	.	.	.	X	1	0.05%
9	.	.	.	.	162	7.59%

4.3 Analysis of imputation

In the case of the previous analysis and of imputation, it is presented the case of the mill of the MOBO 16B, while in the canonical correlation, it is worked with the group of mills of sector 16 (of Ball and SAG); and for the experimental design. For the analysis it is used the software SAS v9.3.

Though it does not exist all the information, and being a great number of atypical data in the period of three years, it is proceed to the imputation, not only for the problem of the missing data, but also “to correct" these atypical data (a 14% of the observations that have at least one missing data).For the imputation it must be deliver, before being used, initial values for the mean vector and the variance and covariance matrix to be able to estimate the coefficients. The matrixes are obtained from the bootstrap procedure, as it is shown shown on table 2:

Table 2: Descriptive Statistical of the variables in study

Variable	EastT.	WestT.	DevT.	Power
Average	47.076154	35.790675	73.20456	3395.297468

Likewise, it is presented the existing correlation between the independant variables or explanatory disposed in a double entry matrix, as it is shown on table 3:

Table 3: Matrix of correlation

Variable	EastT.	WestT.	DevT.	Power
EastT.	102.449985	19.919678	41.713204	3271.145603
WestT.	19.919678	81.730596	51.081713	2880.435635
DevT.	41.713204	51.081713	236.074203	9127.414414
Power	3271.145603	2880.435635	9127.414414	570276

While the procedure is repeated, new mean, variance and covariance matrixes are generated, to estimate the coefficients that constitute the process, this is repeated until the difference between parameters of an iteration with the next one being almost null.

4.4 Test Hypothesis

Below are shown on table 4, the coefficients obtained by the MCMC method, with its corresponding confidence intervals.

Table 4: Test Statistical

Variable	Coefficient	Est. Error	Confidence Interval (5%)	Value t	$p-$value
EastT.	47.104421	0.22795	[46.657; 47.552]	206.64	< 0.0001
WestT.	35.764552	0.204824	[35.362; 36.167]	174.61	< 0.0001
DevT.	73.163543	0.346796	[72.482;73.482]	210.97	< 0.0001
Power	3394.130311	16.935236	[3360.873; 3427.873]	200.42	< 0.0001

It can be observed that each of these values is significant, what can be verified with the hypothesis test $H_0 : \beta_i = 0$ vs $H_1 : \beta_i = 0$, and the test statistical $t-$student, which decision criteria is to reject the hypothesis if $p-$value < 0.05, or if the coefficient value is within the confidence interval. In this case both are true, thus it demonstrate that this procedure is quite efficient and reliable to generate these new data basis.

4.5 Application of the Canonical Correlation

The use of the canonical correlation, allows not only to search a relation from variable to variable, but also the, between one of them and a set of them, as well as between two complete groups. Below it is shown the equation of correlation for the mills 16 (Ball A and B, and SAG), structured in variables and abbreviated as it is seen in parenthesis: Power of MOBO 16A (P16A), MOBO 16B (P16B), SAG 16 (P16S), East Temperatures, MOBO 16A (TE16A), West (TO16A), Winding (TD16A) and of MOBO 16B (TE16B, TO16B y TD16A), and of the Tonnage (TON16S), then:

$P16A+P16B+P16S = TE16A+TO16A+TD16A+TE16B+TO16B+TD16B+TON16S$

So, the process implies to create as much canonical correlations as it is needed to achieve the minimum of present variables between the dependant and the independant (min(# Var. Dependents,# Var. Independants)), that in this case corresponds to 3. Table 5, shows the data obtained through this procedure.

Table 5: Canonical Correlation

	Canonical Correlation	Own value	Proportion	Accumulated
1	0.861034	2.8667	0.9239	0.9239
2	0.435197	0.2336	0.0753	0.9992
3	0.051116	0.0026	0.0008	1.0000

In this case, the canonical correlations are given for each group created inside the program, explains the relation between both groups of variables, that is to say, between the dependants and the independant, where it can be remarked that the first results in having a quite high value in comparison with the others, but to rectify how many groups must be considered, it can be performed the test of Barttlet-Lawley, which results can be appreciated below on Table 6:

Table 6: Statistical of the Barttlet-Lawley Test

	Aprox. F Value.	GL	p$-$value
1	210.57	21	< 0.0001
2	39.74	12	< 0.0001
3	1.11	5	0.3505

This test, for the first line, works on the null hypothesis of $H_0 : \rho_0 > \rho_1 = \rho_2 = \rho_3 = 0$ (with

$\rho_0 = 1$), while the second line works on $H_0 : \rho_1 > \rho_2 = \rho_3$, and so on. Thus, it can be observed that the two first hypothesis are rejected, analyzing the test by means of p−value < 0.05, but it is not the same for the last one, since it does not exists evidence to reject it. This implies that the two first correlations result significant and reliable to be taken in consideration in the study. The canonical correlation allows to modelate the information of the mills, with coefficient. The canonical correlation model allows for each of the variables within each group; to these made-up groups, are nominated as dependant and independant variables.

On one hand, the SAS program, gives two coefficients for each variable in each group, ones called raw coefficients, and the others standardized. The first ones correspond to the real coefficients and, at the same time, common of each variable, that is to say, are interpreted as it were a linear regression, this is, for each unit that increases some value, the response will increase / decrease in β_i units. By the other hand the standardized coefficients are given only with the purpose of facilitating its interpretation, and to establish also a comparison between variables. Avoiding differences between magnitudes or intervals these have.

Table 7: Raw Coefficients and dependant variables

Raw Coefficients	Dependant 1	Dependant 2	Dependant 3
Power MOBO 16A	0.000420051	0.002660012	–0.000485457
Power MOBO 16B	0.0011697651	0.0011781961	–0.000018758
Power SAG 16	–0.049858651	–0.059131666	0.88859719798

Table 8: Raw Coefficients and independant variables

Raw Coefficients	Dependant 1	Dependant 2	Dependant 3
Ton. SAG 16	–0.000117482	–0.000269535	–0.001644198
East Temp MOBO 16A	0.0160939058	0.0454415296	0.0501536296
West Temp MOBO 16A	0.0129238742	0.0075762637	0.0630824701
Dev Temp MOBO 16A	0.0106182145	–0.077979648	–0.005456832
East Temp MOBO 16B	0.0146624833	0.0514678025	–0.004535215
West Temp MOBO 16B	0.0065554753	0.0514678025	–0.034625719
Dev Temp MOBO 16B	0.0467922347	0.0392654956	0.0003705063

Table 9: Standardized Coefficients and dependant variables

Standardized Coefficients	Dependant 1	Dependant 2	Dependant 3
Power MOBO 16A	0.1931	–1.2226	–0.2231
Power MOBO 16B	0.8838	0.8902	–0.0142
Power SAG 16	–0.0577	–0.0685	1.0259

Table 10: Standardized Coefficients and independant variables

Standardized Coefficients	Dependant 1	Dependant 2	Dependant 3
Ton. SAG 16	–0.0368	–0.0845	–0.5155
East Temp MOBO 16A	0.1005	0.2837	0.3131
West Temp MOBO 16A	0.1562	0.0916	0.7626
Dev Temp MOBO 16A	0.1963	–1.4415	–0.1009
East Temp MOBO 16B	0.1481	0.4138	–0.0458
West Temp MOBO 16B	0.0595	0.4668	–0.3140
Dev Temp MOBO 16B	0.7170	0.6017	0.0057

In tables 7 to 10, it can be found important differences between the variables and the mills, with which it can be aroused the equations obtained in the analysis, considering that one of the groups (equation), it is discarded by the results obtained from the test of Barttlet-Lawley. Below are presented two equations, of which it is chosen the second one that represents the greater correlation among the powers of the mills, temperature and tonnage.

Equation 1:

$$0.000420051 * P16A + 0.0011697651 * P16B - 0.049858651 * P16S = 0.0160939058 * TE16A + 0.0129238742 * TO16A + 0.01061821145 * TD16A + 0.146624833 * TE16B + 0.0065554753 * TO16B + 0.0467922347 * TD16B - 0.000117482 * TON16S$$

Equation 2:

$$- 0.002660012 * P16A + 0.0011781961 * P16B - 0.059131666 * P16S = 0.0454415296 * TE16A + 0.0075762637 * TO16A - 0.077979648 * TD16A + 0.0409671048 * TE16B + 0.0514678025 * TO16B + 0.0392654956 * TD16B - 0.000269535 * TON16S$$

4.6 Analysis of the design of experiments

For this case, it is worked with all the mills, both SAG and the ball 16 and 17, but only with the powers, since mills will be measured as factors. The design of the experiment allows to measure the effect of the mill as factor on the total energetic consumption, this to find some difference among the processes, that are expected to be totally independant among the mills.

In the above ANOVA table 11, it is considered the effect of the process, that is to say, it is worked with experimental designs nested, where the mills of Ball A, B and SAG 16, belong to the first group of the process, and the remaining to the second group. Thus, it is concluded that it does not exist difference between mills 16 and 17 in the behaviour, but it exists in the consumption between the mills belonging to a same process (all this concludes starting from the test of the Fisher, by means of its $p-$value, and considering a significance of $\alpha = 0.05$). Due the process in study, it starts in the SAG mills, the energy consumption of the ball mills, it should requiere less consumption. However, there is no difference in the behaviour of the groups of ball mills 16 and SAG 16, respect of the ball mills 17 and SAG 17, which is consistent with the electric energy consumption for both groups of mills.

Table 11: ANOVA the model

FV	GL	SC	MC	Valor F	p−value
Model	5	33306519304	6661303861	23023.6	< 0.0001
Error	12804	3704525732	289326		
Total	12809	37011045036			

In conclusion, if there is a difference in the process variable or group, as observed in Table 12.

Table 12: paired sample MOBO 16

Variable	Means	MOBO 16A v/s MOBO 16B	stat t	p−value
Power	90.2936	A>B	6.89	< 0.0001
EastT.	-7.1599	A<B	-34.63	< 0.0001
WestT.	2.3912	A>B	7.60	< 0.0001
DevT.	18.6154	A>B	59.00	< 0.0001

4.7 Complementary Analysis

The procedure performed with the canonical correlations, does not result to be very different than the factorial analysis. These correlations are given by SAS, and are shown below on Table 13:

Table 13: Correlations and dependant variables

Correlation	Dependant 1	Dependant 2	Dependant 3
Power MOBO 16A	0.7096	–0.7046	–0.0071
Power MOBO 16B	0.9874	0.1440	–0.0652
Power SAG 16	0.1680	–0.1513	0.9741

It is observed a strong correlation of the powers of the mills Ball A and Ball B respect the first group of mills, so it exists a relation between the energies that are consumed in both mills, from this it can be rescue that the work performed in the process inside mill A, it has certain relation with the one done by mill B, what can be due to the type of mineral and/or the tonnage that is given from the SAG mill, which is divided among the ball mills.

Below are given on Table 14, the correlations of the independant variables in relation with the set of variables of temperatures and tonnage.

Table 14: Correlations and independant variables

Correlations	Independant 1	Independant 2	Independant 3
Ton. SAG 16	0.0775	–0.1130	–0.5006
East Temp MOBO 16A	0.6693	–0.1055	0.2723
West Temp MOBO 16A	0.1045	0.0018	0.7730
Dev Temp MOBO 16A	0.8109	–0.5198	–0.0089
East Temp MOBO 16B	0.4764	0.3272	0.1090
West Temp MOBO 16B	0.4805	0.2050	–0.2313
Dev Temp MOBO 16B	0.9219	0.0663	–0.2332

It is observed that a great part of the variables show a strong correlation with the group 1, excepting the tonnage of the mill SAG 16 and the temperature of the west engine of mill of ball 16 A. The fact is that a great part of the factors that act in the milling process (since this group of independant variables are strongly related with group 1 of the dependant variables), have a strong impact over the energetic consumption of the three mills of the process 16.

All the variables considered in this set are from mill of ball A and B, which was expected due to the fact that the first set of dependant variables is strongly related with the power of this two mills.

Other important set to remark (despite of being discarted by the Barttlet-Lawley test) is the third set of variables, since this only considers the energy consumed by the mill SAG 16 for the set of dependant variables. For the tonnage, it only works with the correlations of variables belonging to the mill SAG 16.

The above mentioned (for the first and third set of variables analyzed) can be corroborated with the following values given, where are shown the dependant variables related to the sets of independant variables and viceversa, and are shown below on Table 15, in which are found similar results to the previous presented, with the exception of the third set of variables where it is demonstrated why it was descarted by the Barttlet-Lawley test.

Table 15: Correlations and independant variables

Correlations	Independant 1	Independant 2	Independant 3
Power MOBO 16A	0.6110	–0.3066	–0.0004
Power MOBO 16B	0.8502	0.0627	0.0033
Power SAG 16	0.1446	–0.0658	0.0498

As it was expected, the powers of the mills Ball A and Ball B, have a strong relation with the set of independant variables of the first group, and thus the same independant variables present a strong correlation with the first set of dependant variables, giving in both cases, very similar values to the previous, and are shown below on Table 16.

Table 16: Correlations and dependant variables

Correlations	Dependant 1	Dependant 2	Dependant 3
Ton. SAG 16	0.0667	–0.0492	–0.0256
East Temp MOBO 16A	0.5763	–0.0459	0.0139
West Temp MOBO 16A	0.0900	0.0008	0.0395
Dev Temp MOBO 16A	0.6983	–0.2262	–0.0005
East Temp MOBO 16B	0.4102	0.1424	0.0056
West Temp MOBO 16B	0.4137	0.0892	–0.0118
Dev Temp MOBO 16B	0.7938	0.0289	–0.0119

It is concluded from the above table, that there is a strong relation between the temperatures (except the west engine of the mill of Ball 16A) with the energetic consumption of the mills, mainly in the Mill of Ball 16A and Mill of Ball 16B. Thus, it can be worked without problems for future analysis, descarting those that do not possess any relation with them, since these explain the behaviour of consumption in this machines together.

Conclusions

Despite awareness of the wide potential of the energetic efficiency, there are few mining companies, that make decisions based on the analyzis of data coming from their productive processes, due to obstacles such as the complexity of the process and the lack of clear information. A database that integrates models for the energetic efficiency, it is a great input due to the non existence of them in the industrial explotation of copper. The great volume of data coming from the copper concentrate process, the ball mills and the SAG mills, must be extracted, filtrated and transformed, for the variables related with the energetic consumption. Big data processing, allows to identify the variables of interest (tonnage, temperature and power), and the relation that these variables have with the other variables of the process. For the analyzis of the data it is used the statistical tools, arriving to a model of improvement in the energetic efficiency.

The main contribution of this paper, is obtain a model to improve energy efficiency, through the integrate DSS and the data analysis using different statistical tools such as Markov Chain Monte Carlo, Canonical correlation analysis and design and analysis of experiments. It is expected that the results are applicable in different areas of the production process for large-scale mining.

There is an urgency in the creation and implementation of models to improve the energy efficiency in the mining sector, due to the restriction, environmental challenges and energy cost.

Acknowledgment

The authors acknowledge the financial support of the "Center for Multidisciplinary Research on Signal Processing" (CON-ICYT/ACT1120 Project), the USACH/DICYT 061413SG Project, "Continuous visible light communication Access point for man to machine interaction and detection in high risk mining environments", CORFO code 14IDL2-29919 and "Cooperation agreement between the Universidad de Santiago de Chile and Corporación Nacional del Cobre de Chile, División Chuquicamata", and the support given by Professor Francisco Torres Aviles (R.I.P).

Bibliography

[1] C. Lagos, F. Cordova, S. Gutierrez, G. Fuertes, and R. Carrasco, Data analysis methods related to energetic consumption in copper mining . A test case in Codelco, *Computers Communications and Control (ICCCC), 2016 6th International Conference on*, IEEE Xplore, e-ISSN 978-1-5090-1735-5, doi: 10.1109/ICCCC.2016.7496768, 1241–247, 2016.

[2] J. A. Aloysius, H. Hoehle, S. Goodarzi, and V. Venkatesh (2016), Big data initiatives in retail environments: Linking service process perceptions to shopping outcomes, *Annals of Operations Research*, 1–27.

[3] G. Zucker, J. Malinao, U. Habib, T. Leber, A. Preisler, F. Judex (2014), Improving energy efficiency of buildings using data mining technologies, *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, 2664–2669.

[4] T. Mokfi, M. Almaeenejad, and M. M. Sedighi, A Data Mining Based Algorithm to Enhance Maintenance Management: A Medical Equipment Case Study, *2011 First International Conference on Informatics and Computational Intelligence*, 74–80, IEEE, dec 2011.

[5] P. Bastos, I. d. S. Lopes, and L. Pires (2012), A maintenance prediction system using data mining techniques, *World Congress on Engineering 2012*, 1448–1453.

[6] F. Filip (2008); Decision support and control for large-scale complex systems, *Annual Reviews in Control*, 32(1): 61–70, 2008.

[7] A. Kaklauskas (2015), *Biometric and Intelligent Decision Making Support*, vol. 81, Springer International Publishing AG, 2015.

[8] M. M. Polycarpou and Y. Ohta (2007), Nonlinear Fault Diagnosis of Dynamical Systems: An Intelligent Control Framework, *IFAC Proceedings Volumes*, 40(9):1-1.

[9] L. Martín, L. Baena, L. Garach, G. López, and J. de Oña (2014), Using data mining techniques to road safety improvement in Spanish roads, *Procedia-Social and Behavioral Sciences*, 160:607–614.

[10] M. A. Khan, M. Z. Islam, and M. Hafeez (2012), Evaluating the performance of several data mining methods for predicting irrigation water requirement, *Proceedings of the Tenth Australasian Data Mining Conference*, Australian Computer Society, 134:199–207.

[11] C. Gröger, F. Niedermann, and B. Mitschang (2012), Data mining-driven manufacturing process optimization,*Proceedings of the World Congress on Engineering*, 3:4–6.

[12] N. Altintas and M. Trick (2014), A data mining approach to forecast behavior, *Annals of Operations Research*, 216:3–22.

[13] R. Carrasco, M. Vargas, M. Alfaro, I. Soto, and G. Fuertes (2015); Copper Metal Price Using Chaotic Time Series Forecating, *IEEE Latin America Transactions*, 13(6):1961–1965,

[14] F. Seguel, R. Carrasco, M. Alfaro, P. Adasme, and I. Soto, A Meta-heuristic Approach for Copper Price Forecasting, *Information and Knowledge Management in Complex Systems*, 449: 156–165, 2015.

[15] C. Lagos, G. Fuertes, R. Carrasco, S. Gutierrez, M. Vargas, and R. Rodrigues, Facing the data analysis complexity for the energetic efficiency management at great copper, *IEEE ICA/ACCA 2016*, (Curicó), 1–6, 2016.

[16] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth (1996), From Data Mining to Knowledge Discovery in Databases, *AI Magazine*, v17(3):37-46.

[17] J. Ding, Q. Chen, T. Chai, H. Wang, and C.-Y. Su (2009), Data mining based feedback regulation in operation of hematite ore mineral processing plant, *2009 American Control Conference, IEEE*, 907–912.

[18] D. Rojas, E. Castillo, and J. Cantallopts (2015), *Caracterización de los costos de la gran minería del cobre*, Tech. Rep., Comisión Chilena del Cobre, 2015.

[19] D. Hodouin, S.-L. Jämsä-Jounela, M. Carvalho, and L. Bergh (2001), State of the art and challenges in mineral processing control, *Control Engineering Practice*, 9(9):995–1005.

[20] Codelco, Memoria Anual 2015, tech. rep., Codelco, Santiago of Chile, 2015.

[21] Y. Y. Wen, W. M. Huang, J. Wu, Y. Chen, J. Q. Song (2013), Water consumption analysis system based on data mining, *Applied Mechanics and Materials*, 241:1093–1097.

[22] Y. Liu, J. Zhao, and Z. Wang (2015), Identifying determinants of urban water use using data mining approach, *Urban Water Journal*, 12(8):618–630.

[23] R. C. Leme et al. (2014); Design of experiments applied to environmental variables analysis in electricity utilities efficiency: The Brazilian case, *Energy Economics*, 45:111–119.

[24] Y. Liu and C. Li (2016), Complex-valued Bayesian parameter estimation via Markov chain Monte Carlo, *Information Sciences*, 326:334–349.

[25] X. Xing, K. Wang, T. Yan, Z. Lu (2016), Complete canonical correlation analysis with application to multi-view gait recognition, *Pattern Recognition*, 50: 107–117.

[26] Z. Chen, K. Zhang, S. X. Ding, Y. A. Shardt, and Z. Hu (2016), Improved canonical correlation analysis-based fault detection methods for industrial processes, *Journal of Process Control*, 41:26–34.

[27] Z. Chen, S. X. Ding, K. Zhang, Z. Li, and Z. Hu (2016), Canonical correlation analysis-based fault detection methods with application to alumina evaporation process, *Control Engineering Practice*, 46:51–58.

[28] L. Shang, J. Liu, K. Turksoy, Q. Min Shao, A. Cinar (2015), Stable recursive canonical variate state space modeling for time-varying processes, *Control Engineering Practice*, 36: 113–119.

[29] D. C. Montgomery (2013); *Design and Analysis of Experiments Eighth Edition*, John Wiley & Sons, Inc., 8th ed., 2013.

[30] J. Tang, G. Gong, H. Su, F. Wu, and C. Herman (2016); Performance evaluation of a novel method of frost prevention and retardation for air source heat pumps using the orthogonal experiment design method, *Applied Energy*, 169:696–708.

[31] G. S. dos Reis et al., (2016); The use of design of experiments for the evaluation of the production of surface rich activated carbon from sewage sludge via microwave and conventional pyrolysis, *Applied Thermal Engineering*, 93: 590–597.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(1):76-89, February 2017.

Speed Computation for Industrial Robot Motion by Accurate Positioning

L.M. Matica, H. Oros

Liliana Marilena Matica*
Faculty of Electrical Engineering and Information Technology
University of Oradea
Oradea, Romania
*Corresponding author:lmatica@uoradea.ro

Horea Oros
Faculty of Sciences, Department of Mathematics and Computer Science
University of Oradea
Oradea, Romania
horos@uoradea.ro

Abstract: In this paper we define a new method for speed (velocity) computation, named mixt profile. The mixt profile of speed variation assures an accurate positioning at the end of motion (movement), in a well determinate time lapse. The method is linked with computation of location (position) matrix, about an industrial robot. Mixt profile of speed may be applied about motion on linear or circular trajectories. The paper continues the explanation from [6][a] regarding this method.
Keywords: kinematics of industrial robots; linear or circular trajectory; acceleration and deceleration stage of movement

[a]Reprinted and extended, with permission based on License Number 3979260943291 [2016] IEEE, from "Computers Communications and Control (ICCCC), 2016 6th International Conference on"

1 Introduction

This paper contains others explanation regarding mixt profile of speed, published in [6], those are: more detailed explanations about the computation method named mixt profile of speed; the adaptation of mixt profile method of computation for a circular trajectory and a concrete example of computation; a graphic about deceleration stage explaining software implementation of this computation method; formulas of reverse kinematics about RRRRRR robotic arm, which was done for illustrate the method of computation, figure 1; formulas of direct kinematics about this robotic arm; explanations about the solid kinematics principles that establish the direct kinematics; enunciation of Or algorithm about determining the direct kinematics of an robotic arm (a simple algorithm defined in this paper; it analyses only those two relative positions: parallel or perpendicular).

Concerning movement command of an industrial robot, [1]- [6] it is necessary to define the next location of it. The industrial robot location is defined by the location matrix. Starting with the values of location matrix, the values of kinematics joints parameters of the industrial robot may be computed. Those computation formulas are named, reverse kinematics. The reverse kinematics is the solution of an equation system formed by forward kinematics. Forward kinematics of an industrial robot is the result of a kinematics analysis. An example of kinematics analysis about an industrial robot, (more precise, about a robotic arm type RRRRRR) is illustrated in figure 1 [3].

In figure 1, the notations defines several Cartesian coordinate systems with its axles: X_i, Y_i, Z_i and its origins: O_i (index i goes from 1 to 6; $i = 1..6$); then six rotation driving kinematics

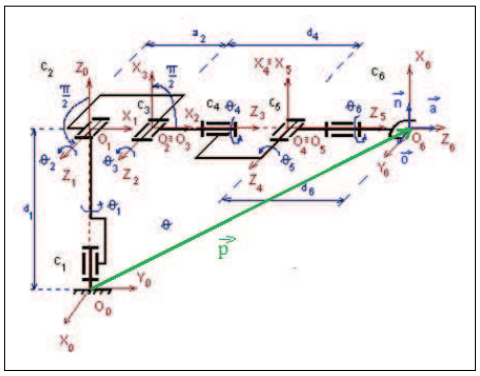


Figure 1: Kinematics analysis of robotic arm, type RRRRRR

couples (d.c.c.) of robotic arm: C_i $(i = 1..6)$; variable parameters of d.c.c.: Θ_i $(i = 1..6)$; constant parameters of the robotic arm: d_1; a_2; d_4; d_6; the versors: $\overrightarrow{n}$; $\overrightarrow{o}$; $\overrightarrow{a}$ (about sense and direction of axles OX_6; OY_6 and OZ_6), the Cartesian coordinate system of index 6 has the origin in the tool centre point (TCP) of the robotic arm. All Cartesian coordinate systems have the position of axles according with Denavit-Hartenberg convention. [3], [6]

About industrial robot movement (motion), if the motion time is defined, the positioning precision at the end of the motion is not very good. A very good positioning precision, at the motion end, can't be obtained, in a defined motion time. Both conditions are very hard to accomplish. The mixt profile speed variation, defined in this paper, during industrial robots motion, may accomplish those two conditions. The method is linked with the location matrix computation of an industrial robot. [6]

2 The location matrix of an industrial robot

The industrial robot location, (position and orientation) is defined by the location matrix that contains axles components of position vector: $\overrightarrow{p}$ and orientation versors: $\overrightarrow{n}$; $\overrightarrow{o}$; $\overrightarrow{a}$; figure 2, [3]; a versors have the module equal to 1.

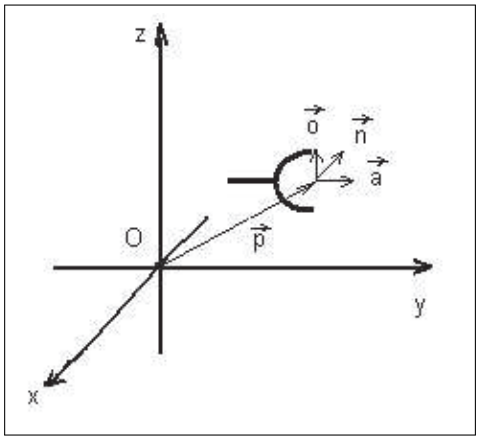


Figure 2: The position vector and orientations versors that define the location matrix of an industrial robot

A versor describes only the orientation; about an industrial robot, the orientation versors: $\overrightarrow{n}$; $\overrightarrow{o}$; $\overrightarrow{a}$; describe the orientation of tool centre point, TCP, regarding the Cartesian coordinate

system considered.

The location matrix of the industrial robot contains three components of those versors and the position vector (three components along well known three axles of a Cartesian coordinate system: OX; OY; OZ).

$$G_i = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Index i of the location matrix makes reference to index i Cartesian coordinate system.

According to Denavit-Hartenberg convention, a coordinate system, index i, is obtained by homogeneous transformations, from previous one, index $i-1$. Those homogeneous transformations are (always in this order):

1. rotation of $\Theta_i(t)+\beta$ angle, around OZ_{i-1} axle (the parameter t show that the angle varies in time, it is programmable, because that d.c.c. is an rotation one);
2. translation of d_i distance, along OZ_{i-1} axle;
3. translation of l_i distance, along OX_i axle;
4. rotation of α_i angle, around OZ_{i-1} axle. [3], [6]

The previous homogeneous transformations define the transformation matrix, ${}^{i-1}A_i$: [3]

$${}^{i-1}A_i = Rot(OZ_{i-1}, \theta_i(t)+\beta_i) \cdot Trans(OZ_{i-1}, d_i) \cdot Trans(OX_i, l_i) \cdot Rot(OX_i, \alpha_i) \tag{2}$$

For example, the forward kinematics formulas of the robotic arm from figure 1 are (a robotic arm is a particular kind of industrial robot, it is similar with the human arm):

$$\begin{array}{rcl} {}^0A_1 & = & Rot(OZ_0, \theta_1(t)+\pi/2) \cdot Trans(OZ_0, d_1) \cdot Rot(OX_1, +\pi/2) \\ {}^1A_2 & = & Rot(OZ_1, \theta_2(t)) \cdot Trans(OX_2, a_2) \\ {}^2A_3 & = & Rot(OZ_2, \theta_3(t)+\pi/2) \cdot Rot(OX_3, +\pi/2) \\ {}^3A_4 & = & Rot(OZ_3, \theta_4(t)) \cdot Trans(OZ_3, d_4) \cdot Rot(OX_4, -\pi/2) \\ {}^4A_5 & = & Rot(OZ_4, \theta_5(t)) \cdot Rot(OX_4, +\pi/2 \\ {}^5A_6 & = & Rot(OZ_5, \theta_6(t) \cdot Trans(OZ_5, d_6) \end{array} \tag{3}$$

Let us discus about transformation matrix ${}^0A_1$.

The kinematics joint, named $C1$, is a rotational one; so, relation 2 is useful and must be adapted (index i is equal with 1):

$${}^0A_1 = Rot(OZ_0, \theta_1 t+\beta_i) \cdot Trans(OZ_0, d_1) \cdot Trans(OX_1, l_1) \cdot Rot(OX_1, \alpha_1) \tag{4}$$

About relation 4, the variable parameter (programmable) is the angle: $\theta_1(t)$; the constant values are named: β_1 ; k_1 ; l_1 and α_1. In purpose to determine the constant value named β_1, the relative position of axle OX_1 to axle OX_0 must be analyzed, figure 3; it is perpendicular (it is not parallel, it define an angle); this parameter has a different from zero value [3].

In purpose to determine the α_1 constant value, it must be analyzed the axle OZ_1 relative position to axle OZ_0; figure 4; it is perpendicular, this parameter has a non-zero value.

The translations are defined by relative position of origins O_0 and O_1; so, it is necessary a translation along axle OZ_1 with constant value named d_1. [3] It result the ${}^0A_1$ transformation matrix, relation 3.

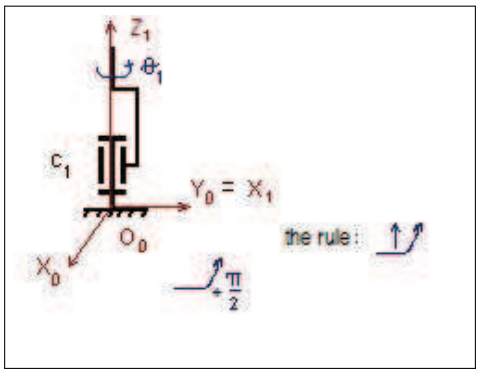


Figure 3: Analysis about β_1 value

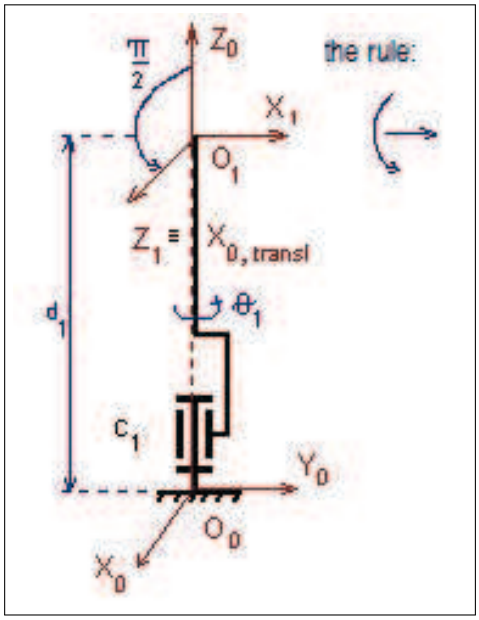


Figure 4: Analysis about α_1 value

Let us discuss other two examples [3]. From figure 5 graphical analysis, it results the ${}^1A_2$ matrix. A constant value (equal with a_2) characterizes a translation along axle OX_2; the constant value named β_2 has a zero value, because axle OX_1 is parallel with axle OX_2; the constant value named α_2 has a zero value, because axle OZ_1 is parallel with axle OZ_2.

From figure 6 graphical analysis, it results ${}^5A_6$ matrix, (relation 3); the significant constant value, d_6, is about a translation along axle OZ_6; β_6 and α_6 have zero value.

Similar analyses are performed for determination of every transformation matrix (of forward kinematics). It results all the transformation matrices (relation 4) and the location matrix of the industrial robot. More precisely, the location matrix defines the position and the orientation of the robotic arm tool center point, TCP.

Lets makes a clear distinction between parameters of a translational or a rotational kinematics joint of an industrial robot. About a rotational kinematics joint, relation 2, the variable parameter is named: $\theta_i(t)$. About a translational kinematics joint, the variable parameter is named: $d_i(t)$; the ${}^{i-1}A_i$ transformation matrix is defined as follow:

$$ {}^{i-1}A_i = Rot(OZ_{i-1}, \beta_i) \cdot Trans(OZ_{i-1}, d_i(t)) \cdot Trans(OX_i, l_i) \cdot Rot(OX_i, \alpha_i) \qquad (5) $$

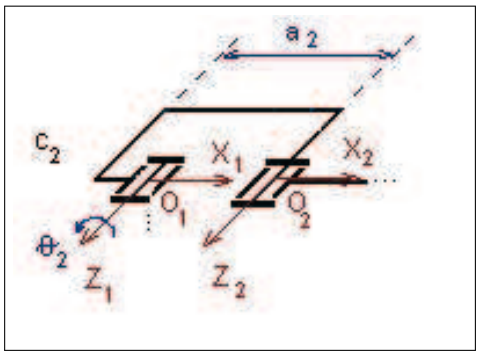


Figure 5: Analysis about ${}^{1}A_2$ matrix

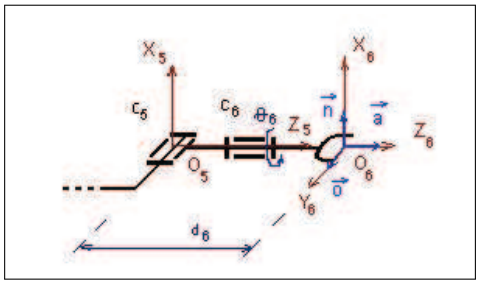


Figure 6: Analysis about ${}^{5}A_6$ matrix

In relation 2 and 5, the constant values of forward kinematics computation are named: β_i; k_i; l_i; α_i, whatever the kinematics joint is a rotational or a translational one.

Usually, about forward kinematics of industrial robots, the explanations work with a single formula (relation 6), without a clear distinction (regarding the parameter name), concerning variable and constant values:

$$ {}^{i-1}A_i = Rot(OZ_{i-1}, \theta_i) \cdot Trans(OZ_{i-1}, d_i) \cdot Trans(OX_i, l_i) \cdot Rot(OX_i, \alpha_i) \tag{6} $$

The relations 2 and 5, described in this paper, make a clear distinction between constant and variable values, involved in forward kinematics computation. According with Denavit-Hartenberg convention, the algorithm explained in this paper, determines the constant values involved in forward kinematics computations, based on observation: two axles are parallel or are not parallel.

The first step of the algorithm consists in Cartesian coordinate systems settlement, identical with Denavit-Hartenberg convention. Every kinematics joint, named C_i, is characterized by a Cartesian coordinate system, named $OXYZ_i$ properly settled; index i goes from 1 value to n, n is the number of kinematics joints. The axle OZ_i is settled along the axis of index $i+1$ kinematics joint, named C_{i+1} ; the axle OX_i is settled perpendicular of the plane formed by axles OZ_{i-1} and OZ_i. The $OXYZ_n$ Cartesian coordinate system is settled linked with TCP position and orientation; in figure 1 $n = 6$. Every Cartesian coordinate system is obtained from the previous one, by several homogenous transformations; those define the transformation matrices, ${}^{i-1}A_i$.

The next step of the algorithm consists on the characteristics identification of each C_i kinematics joint and determination of each ${}^{i-1}A_i$ matrix. Regarding a rotational kinematics joint, relationship 2 is useful for transformation matrix determination; the variable (programmable) parameter is $\theta_i(t)$; the others values are constant. Regarding a translational kinematics joint, relationship 5 is useful for transformation matrix determination; the variable (programmable) parameter is $d_i(t)$; the others values are constant.

Concerning industrial robot motion, the speed variation (as it is defined by mixt profile of speed), had influence (it changes the parameters values, in time) upon programmable (variable) parameters of kinematics joints.

The third step of the algorithm consists on transformation matrixes determination, ${}^{i-1}A_i$, (several analysis about determination of each transformation matrix, were explained on graphical analysis in figures 4, 5 and 6) it may be described as follow: the variable parameter is defined by the kinematics joint type, on step two of the algorithm; about constant values, there is a rotation, named β_i, around axle OZ_{i-1}, if axle OX_{i-1} and axle OX_i are not parallel (those axle may be perpendicular, according with the construction of the industrial robot and the angle may be $\pm\pi/2$; there are translations along axle OZ_{i-1} or along axle OX_i if the Cartesian coordinate systems origins, O_{i-1} and O_i, are not identical (very simple to be identified); there is a rotation around axle OX_i, named α_i, if axle OZ_{i-1} and axle OZ_i are not parallel, (those axle may be perpendicular and the angle may be $\pm\pi/2$).

The previous explanations develop an algorithm about forward kinematics determination, (it asks about each kinematics joint of the robotic arm: is it a translational or a rotational one; it asks about two axles: are those axles parallel or not; this algorithm may be named algorithm Or). It analysis only those two relative positions: parallel or perpendicular; the analysis described by Denavit-Hartenberg convention analysis any relative position about similar Cartesian coordinate axles involved.

The formulas of forward kinematics, to compute position vector components of an industrial robot, more precise, about the robotic arm type RRRRRR from figure 1, are [3] (notations S_i; $i = 1..6$, means sine of θ_i angle and C_i means cosine of same angle, the others notation are identical with those explained):

$$\begin{aligned}
p_x &= (S_1 \cdot C_2 \cdot S_3 + S_1 \cdot S_2 \cdot C_3) \cdot C_4 \cdot S_5 \cdot d_6 + C_1 \cdot S_4 \cdot S_5 \cdot d_6 - S_1 \cdot C_2 \cdot a_2 + \\
&\quad + (S_1 \cdot S_2 \cdot S_3 - S_1 \cdot C_2 \cdot C_3) \cdot (C_5 \cdot d_6 + d_4) \\
p_y &= (-C_1 \cdot C_2 \cdot S_3 - C_1 \cdot S_2 \cdot C_3) \cdot C_4 \cdot S_5 \cdot d_6 + S_1 \cdot S_4 \cdot S_5 \cdot d_6 + \\
&\quad + C_1 \cdot C_2 \cdot a_2 + (C_1 \cdot C_2 \cdot C_3 - C_1 \cdot C_2 \cdot S_3) \cdot (C_5 \cdot d_6 + d_4) \\
p_z &= (-S_2 \cdot S_3 + C_2 \cdot C_3) \cdot C_4 \cdot S_5 \cdot d_6 + (S_2 \cdot C_3 + C_2 \cdot S_3) \cdot (C_5 \cdot d_6 + d_4) + \\
&\quad + S_2 \cdot a_2 + d_1
\end{aligned} \tag{7}$$

The formulas to compute the orientation versors components of the same robotic arm, figure 1, are:

$$\begin{aligned}
n_x &= (S_1 \cdot C_2 \cdot S_3 + S_1 \cdot S_2 \cdot C_3) \cdot (C_4 \cdot C_5 \cdot C_6 - S_4 \cdot S_6) + \\
&\quad + C_1 \cdot (S_4 \cdot C_5 \cdot S_6 + C_4 \cdot S_6) - S_5 \cdot C_6 \cdot (S_1 \cdot S_2 \cdot S_3 - S_1 \cdot C_2 \cdot C_3) \\
n_y &= (-C_1 \cdot C_2 \cdot S_3 - C_1 \cdot S_2 \cdot C_3) \cdot (C_4 \cdot C_5 \cdot C_6 - S_4 \cdot S_6) + \\
&\quad + S_1 \cdot (S_4 \cdot C_5 \cdot C_6 + C_4 \cdot S_6) - S_5 \cdot C_6 \cdot (C_1 \cdot C_2 \cdot C_3 - C_1 \cdot S_2 \cdot S_3) \\
n_z &= (-S_2 \cdot S_3 + C_2 \cdot C_3) \cdot (C_4 \cdot C_5 \cdot S_6 - S_4 \cdot S_6) - S_5 \cdot S_6 \cdot (S_2 \cdot C_3 + C_2 \cdot S_3)
\end{aligned} \tag{8}$$

$$\begin{aligned}
o_x &= (S_1 \cdot C_2 \cdot S_3 + S_1 \cdot S_2 \cdot C_3) \cdot (-C_4 \cdot C_5 \cdot S_6 - S_4 \cdot C_6) + \\
&\quad + C_1 \cdot (-S_4 \cdot C_5 \cdot S_6 + C_4 \cdot C_6) + S_5 \cdot S_6 \cdot (C_1 \cdot C_2 \cdot C_3 - C_1 \cdot S_2 \cdot S_3) \\
o_y &= (-C_1 \cdot C_2 \cdot S_3 - C_1 \cdot S_2 \cdot C_3) \cdot (-C_4 \cdot C_5 \cdot C_6 - S_4 \cdot C_6) + \\
&\quad + S_1 \cdot (-S_4 \cdot C_5 \cdot S_6 + C_4 \cdot C_6) + S_5 \cdot C_6 \cdot (C_1 \cdot C_2 \cdot C_3 - C_1 \cdot S_2 \cdot S_3) \\
o_z &= (-S_2 \cdot S_3 + C_2 \cdot C_3) \cdot (-C_4 \cdot C_5 \cdot S_6 - S_4 \cdot C_6) + S_5 \cdot S_6 \cdot (S_2 \cdot C_3 + C_2 \cdot S_3)
\end{aligned} \tag{9}$$

$$\begin{aligned} a_x &= (S_1 \cdot C_2 \cdot S_3 + S_1 \cdot S_2 \cdot C_3) \cdot C_4 \cdot S_5 + S_1 \cdot S_4 \cdot S_5 + \\ &\quad C_5 \cdot (S_1 \cdot S_2 \cdot S_3 - S_1 \cdot C_2 \cdot C3) \\ a_y &= (-C_1 \cdot C_2 \cdot S_3 - C_1 \cdot S_2 \cdot C_3) \cdot C_4 \cdot S_5 + S_1 \cdot S_4 \cdot S_5 + \\ &\quad + C_5 \cdot (C_1 \cdot C_2 \cdot C_3 - C_1 \cdot S_2 \cdot S_3) \\ a_z &= (-S_2 \cdot S_3 + C_2 \cdot C_3) \cdot C_4 \cdot S_5 + C_5 \cdot (S_2 \cdot C_3 + C_2 \cdot S_3) \end{aligned} \tag{10}$$

The reverse kinematics (as a result of forward kinematics) computes the d.c.c. parameters starting with position matrix elements values; it is the solution of the equations system (12 equations and 12 unknown values) defined by direct kinematic. It results this conclusion: in order to command an industrial robot motion, it is necessary to compute the position matrix components, for every sampling period of time; those components describe the position and orientation of the robotic arm. The speed (velocity) of motion is defined by vector $\overrightarrow{p}$ (position vector) variation. [6]

3 Acceleration, motion on trajectory and deceleration

About motion on a trajectory, a condition could be a certain speed profile. This speed profile may be trapezoidal or parabolic, figure 7 and figure 8 (graphics consider continuous time).

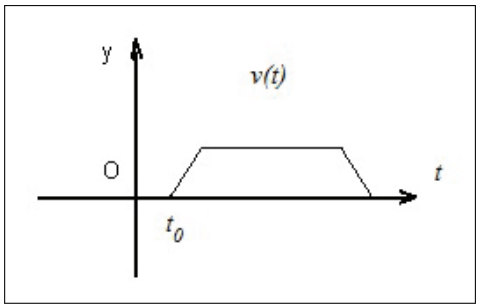


Figure 7: Trapezoidal profile (of speed)

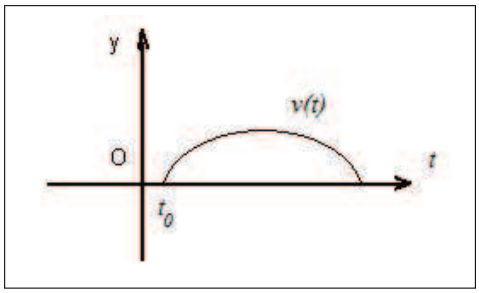


Figure 8: Parabolic profile

The motion of an industrial robot may contain three stages:

1. the acceleration from zero motion speed to the programmed motion speed;
2. the motion with programmed motion speed (constant);
3. the deceleration from programmed speed to zero. [6]

Commonly, acceleration and deceleration depend on the speed profile that was selected. This paper describes another method about deceleration; the method describes another speed profile, named mixt profile of speed, figure 9. [6]

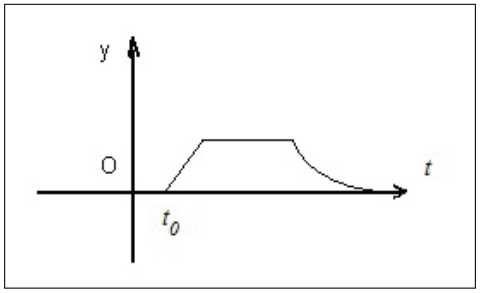


Figure 9: Mixt profile of motion speed

If the trajectory is imposed (linear or circular), it must be computed the position of the intermediary points, named waypoints, (on the trajectory), during acceleration stage, motion on trajectory stage and deceleration stage.

Intermediary positions of the robotic arm are defined by different location matrix. If the trajectory is imposed, we must compute location matrices for every waypoint. Considering reverse kinematics, it results the motion commands for kinematics joints of the robotic arm; starting with location matrix of every waypoint that composes the trajectory, the parameter of every kinematics joint may be computed.

4 Acceleration and deceleration stages for mixt profile of speed

Usually, about acceleration stage, the acceleration variation depends of the maximum acceleration possible, on a sample period of time, considering a numerical computation system with numerical processor.

About an robotic arm motion, the numerical process of command computation, is a discrete one. [1] Variation of robotic arm position, variation of motions speed, acceleration and deceleration values (and others values) depend of a discrete variable defined by relation: $k \cdot T$, where T is the sampling period of time, and k is the number of the sample periods of time considered from the commands beginning (for example, the variable had the value $11 \cdot T$ after eleven sampling periods of time from the start of motion). [6]

About computation described in this paper, the value of maximum possible acceleration in a sample period of time is named a_{max}. About this computation method described in this paper, in the acceleration stage for mixt speed profile, the variation of speed is defined by the relation: [6]

$$v(kT) = v_0 + k \cdot a_{max} \tag{11}$$

Often, the motion speed initial value is zero: $v_0 = 0$; it results: $v(kT) = k \cdot a_{max}$, figure 9, but this speed increasing computation method may be applied for any initial value.

Considering the defined speed increase, relation 11, (in the acceleration stage of mixt profile), the position varies with the values: $T \cdot v(kT) = T \cdot k \cdot a_{max}$; after each sampling period of time. The acceleration value may be considered about axle component of position vector, it results the maximum possible acceleration along every axle, named a_M (instead of a_{max}); the position vector axle components varies, during the acceleration stage: [6]

$$\begin{aligned} p_{k,x} &= p_{k-1,x} + k \cdot T \cdot a_M; k = 1..k_A \\ p_{k,y} &= p_{k-1,y} + k \cdot T \cdot a_M \\ p_{k,z} &= p_{k-1,z} + k \cdot T \cdot a_M \end{aligned} \tag{12}$$

Index k goes from 1 value to k_A value, till the end of the motion stage. The computation starts from axle components values of initial position vector, named: $p_{0,x}$; $p_{0,y}$; $p_{0,z}$.

Let consider a motion with a programmed (imposed) speed value, named v_P; the acceleration stage ends when the speed reach this programed speed value. The programed value of motion speed defines the number of sampling periods of time necessary for the acceleration stage; named k_A, it results:

$$k_A = v_p/T \cdot a_M \tag{13}$$

The variation is a discrete one, so, the value of k_A must be an integer value; the k_A value must be adapted of this condition: it will be the next bigger integer value of the computed value. Because of this aspect, the last step of speed increase value must be adapted, in purpose to reach the programmed speed value (it is obvious that the last step of speed increase value will not be bigger then: a_M).

If speed axle components, named: $v_{P,x}$; $v_{P,y}$; $v_{P,z}$; have different values, the k_A value is determined by the maximum value of speed component:

$$k_A = max(v_{P,x}; v_{P,y}; v_{P,z})/T \cdot a_M \tag{14}$$

The acceleration may be different for each axle, the maximum value of speed component, $max(v_{P,x}; v_{P,y}; v_{P,z})$, defines the axle with maximum acceleration. About other axles, the acceleration is computed, in order to have o constant value for every sampling period of time. The acceleration values are: $v_{P,x}/k_A$; $v_{P,y}/k_A$; $v_{P,z}/k_A$.

About next stage, the motion with a programmed speed, the position vector is described by the relations: [6]

$$\begin{aligned} p_{k+1,x} &= p_{k,x} + T \cdot v_{P,x} \\ p_{k+1,y} &= p_{k,y} + T \cdot v_{P,y} \\ p_{k+1,z} &= p_{k,z} + T \cdot v_{P,z} \end{aligned} \tag{15}$$

In relation 15 index k starts from k_A and goes till is necessary the deceleration stage. This relation, rel. 15 defines those significant values: $\delta_x = T \cdot v_{P,x}$; $\delta_y = T \cdot v_{P,y}$; $\delta_z = T \cdot v_{P,z}$; its mean the linear space steps (because the trajectory is linear), performed at each sampling period of time, during motion with programed speed, on every axle. Those values are named axle steps. Motion on a circular trajectory defines angle steps (about spherical coordinates).

About deceleration stage (the third stage of motion), the speed variation is not a linear one:

$$v(kT) = v_P - T \cdot a_D(kT) = v_P - b \cdot k^2; k = 1..k_D \tag{16}$$

In previous relation, the deceleration value, named a_D, is a variable value and b is a constant value. The b value defines the characteristics about robotic arm motion.

The speed decreases till the motion end; considering the condition: $0 = v_P - b \cdot k_D^2$; it results the number of sampling period of time necessary for the deceleration stage, named k_D (the axle components of speed are: $v_{P,x}$; $v_{P,y}$; $v_{P,z}$):

$$k_D = \sqrt{\frac{max(v_{P,x}; v_{P,y}; v_{P,z})}{b}} \tag{17}$$

About motion on trajectory (the second stage), the necessary distance for deceleration stage, named DD, determine its end (the motion on trajectory ends in the point situated at distance DD before the end point of motion). [6]

The resulting speed profile, named mixt profile, figure 9 (the graphic considers continuous time) ensures a better precision about stop point proximity. Typically, for precise positioning at the motion end, it can't be specified the time needed; the mixt profile of speed specifies exactly the time needed for precise positioning at the motion end. [6]

The described method, named mixt profile (of speed), was implemented at a flexible welding cellule (for manufacture of mining machinery), and the agreed motion characteristics (with the beneficiary) were ok. The maximum weight of processed pieces (with this welding cellule) was 2.5 tons. [6]

About deceleration stage, the software implementation considered 25 values about speed decrease, from speed maximum value possible, those values were written in a table, named: Deceleration table. Inertial reason imposes the number of table values (25). In purpose the determine the deceleration start, the programed speed, v_p, was compared with those values, from Deceleration table; the comparison result defines the value of k_D and every speed value, for every sampling period of time, during deceleration stage; a graphical explanation of this process is described in figure 10.

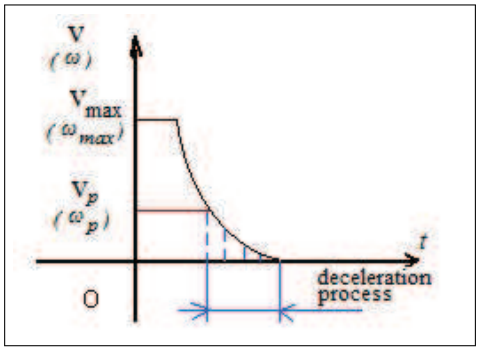


Figure 10: About software implementation of deceleration process

In figure 10, the example works with 4 steps till the end of the motion, (when the speed has zero value).

The software implementation of deceleration stage considered a different Deceleration table for OZ axle component of speed, because a vertical motion has different inertial characteristics, comparing with a horizontal motion (about axle OX and OY the Deceleration table is identical).

5 Example of computation about a linear trajectory

For example, considering a linear trajectory and constant orientation of the robotic arm (along the motion), the value of initial speed zero; the values of programed speed: $v_P = 5\sqrt{2}mm/s$; $v_{P,x} = 3mm/s$; $v_{P,y} = 4mm/s$; $v_{P,z} = 5mm/s$; $a_M = 25mm/s^2$ and $T = 10^{-2}s$; this method of computation determines: $k_A = 20$, the number of sampling periods of time necessary for acceleration stage: [6]

$$k_A = max(3; 4; 5)mm/s/(10^{-2}s \cdot 25mm/s^2) = 20 \tag{18}$$

During acceleration stage, the speed increases with those values: $\delta v_x = 3/20mm/s$; $\delta v_y = 4/20mm/s$; $\delta v_z = 5/20mm/s$; (because of inertial reasons, the acceleration have different values, for each axle components).

Considering the initial values of position vector components: $p_{0,x} = 1,1mm$; $p_{0,y} = 2,2mm$; $p_{0,z} = 3,3mm$, after first sampling period of time, the position vector has the axle components:

$$\begin{array}{rcccccccc} p_{1,x} & = & p_{0,x} + 1 \cdot T \cdot (v_{0,x} + \delta v_x) & = & 1,1 + 10^{-2} \cdot 3/20 & = & 1,1 + 0,0015 & = & 1,1015mm \\ p_{1,y} & = & p_{0,y} + 1 \cdot T \cdot (v_{0,y} + \delta v_y) & = & 2,2 + 10^{-2} \cdot 4/20 & = & 2,2 + 0,002 & = & 2,202mm \\ p_{1,z} & = & p_{0,z} + 1 \cdot T \cdot (v_{0,z} + \delta v_z) & = & 3,3 + 10^{-2} \cdot 5/20 & = & 3,3 + 0,0025 & = & 3,3025mm \end{array} \tag{19}$$

During acceleration stage, after 10 period of time the axle components of position vector differs (from the previous one, in the previous period of time) with: $\delta p_x = 10 \cdot T \cdot \delta v_x = 10 \cdot 3/2000 = 0,015mm$; $\delta p_y = 10 \cdot T \cdot \delta v_y = 10 \cdot 4/2000 = 0,02mm$; $\delta p_z = 10 \cdot T \cdot \delta v_z = 10 \cdot 5/2000 = 0,025mm$.

After 20 periods of time, begin the stage of motion on trajectory. In this moment (considering the initial values of position vector components), the position vector has the axles components: $p_{20,x} = 1,1 + 10^{-2} \cdot 3/20 \cdot (1 + 2 + ... + 20) = 1,1 + 0,315 = 1,415mm$; $p_{20,y} = 2,2 + 10^{-2} \cdot 4/20 \cdot (1 + 2 + ... + 20) = 2,2 + 0,42 = 2,62mm$; $p_{20,z} = 3,3 + 10^{-2} \cdot 5/20 \cdot (1 + 2 + ... + 20) = 3,3 + 0,525 = 3,825mm$.

The stage of motion on trajectory is described by relations relation 15, (it is similar with acceleration stage, but speed has a constant value):

$$\delta p_x = T \cdot v_{P,x}; \delta p_y = T \cdot v_{P,y}; \delta p_z = T \cdot v_{P,z}; \tag{20}$$

For each axle, the axle steps have constant values. Because axle steps have constant values, the algorithm is named: numeric difference analysis, more exactly: interpolate algorithm of numeric difference analysis. [6]

For example, after 80 periods of time, on the stage of motion with constant speed (and after 100 periods of time from the beginning of the motion) the components of position vector have the values: $p_x = 1,415 + 80 \cdot 10^{-2} \cdot 3 = 3,815mm$; $p_y = 2,62 + 80 \cdot 10^{-2} \cdot 4 = 5,82mm$; $p_z = 3,825 + 80 \cdot 10^{-2} \cdot 5 = 4,225mm$.

Let apply this computation (mixt profile of speed), to the robotic arm from figure 1. Let consider the orientation of robotic arm defined by those versors: $\overrightarrow{n} = 1 \cdot \overrightarrow{i}$; $\overrightarrow{o} = 1 \cdot \overrightarrow{j}$; $\overrightarrow{a} = 1 \cdot \overrightarrow{k}$ (where $\overrightarrow{i}$, $\overrightarrow{j}$, $\overrightarrow{k}$ are the versor defining the Cartesian coordinate axles, OX; OY and OZ). After 100 sampling period of time, the location matrix is:

$$G_0(100 \cdot T) = \begin{bmatrix} 1 & 0 & 0 & 3,815 \\ 0 & 1 & 0 & 5,82 \\ 0 & 0 & 1 & 4,225 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{21}$$

During the motion, the location matrix has different values, at every sampling period of time. Knowing the location matrix, it results parameters of robotic arm kinematics joints, considering the formulas of robotic arm reverse kinematics.

The value of $\theta_1(100 \cdot T)$ parameter may be computed with this relationship (formula from reverse kinematics of the robotic arm [3]):

$$\theta_1(100 \cdot T) = \arctan \frac{-(p_{100,x} - d_6 \cdot \sin r_2 \cdot \cos r_1)}{p_{100,y} - d_6 \cdot \sin r_2 \cdot \sin r_1} \tag{22}$$

In previous relationships, the angle r_1 is the polar angle and the r_2 angle is the azimutal angle, about $\overrightarrow{a}$ versor, figure 11.

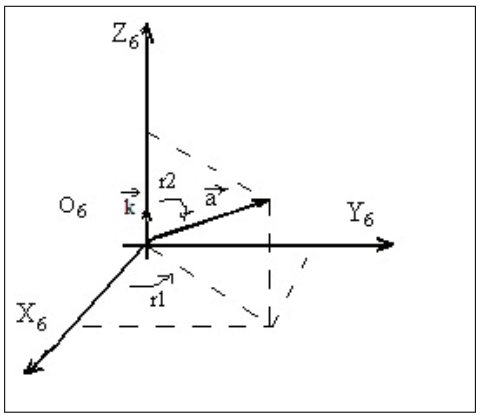


Figure 11: Polar and azimutal angle

Those angles may be computed; the value of n_1 and n_2 depends of the sign of versor components; let remember that arctan can have values in $[-\pi;\pi]$; but those angles may have values in $[0;2\pi]$:

$$r_1 = n_1 \cdot \pi + \arctan\left|\frac{a_y}{a_x}\right|; r_2 = n_2 \cdot \pi + \arctan\frac{\sqrt{a_x^2 + a_y^2}}{|a_z|} \tag{23}$$

From the considered orientation of the robotic arm: $\overrightarrow{a} = 1 \cdot \overrightarrow{k}$; it results the value of those two angles are: $r_1 = 0$; $r_2 = 0$; the parameter value is:

$$\theta_1(100 \cdot T) = \arctan\frac{-(p_{100,x})}{p_{100,y}} = \arctan(-3,815/5,82) = -33,245 \tag{24}$$

Previous relationship defines a negative angle, it means: the rotation sense of motion, about C_1 kinematics joint (the angle is the parameter of this kinematics joint) is opposite considering the positive sense, as it is designed in figure 1.

In purpose to determine the end of the second stage (motion with constant speed), it is necessary to compute the number of sampling period of time for deceleration; let considers $b = 5mm/900s$; it results: [6]

$$k_D = \sqrt{\frac{max(3;4;5)mm/s}{5mm/900s}} = 30 \tag{25}$$

During the deceleration, the computation of waypoints coordinates involves those speed values, from relation 16 it results:

$$\begin{aligned} v_x(kT) &= v_{P,x} - \tfrac{v_{P,x}}{k_D^2} \cdot k^2 = 3 - \tfrac{3}{900} \cdot k^2, k = 1..k_D \\ v_y(kT) &= v_{P,y} - \tfrac{v_{P,y}}{k_D^2} \cdot k^2 = 4 - \tfrac{4}{900} \cdot k^2 \\ v_z(kT) &= v_{P,z} - \tfrac{v_{P,z}}{k_D^2} \cdot k^2 = 5 - \tfrac{5}{900} \cdot k^2 \end{aligned} \tag{26}$$

For each sampling period of time, the position differs with values:

$$\begin{aligned} \delta p_x &= T \cdot v_x(kT), k = 1..k_D \\ \delta p_y &= T \cdot v_y(kT) \\ \delta p_z &= T \cdot v_z(kT) \end{aligned} \tag{27}$$

About deceleration stage, it must be computed the maximum value of distance axle components (necessary for deceleration stage), named DD_{max}:

$$DD_{max} = \sum_{k=1}^{k_D} T \cdot [max(v_{P,x}; v_{P,y}; v_{P,z}) - b \cdot k^2] \tag{28}$$

The DD_{max} value considers the maximum distance of the three axles, necessary for deceleration stage. The deceleration begins when it remains the distance DD_{max}, till the motion end, on respective axle.

According with the considered example, after 24 period of time on deceleration stage, the axle components of speed have the values: [6]

$$\begin{aligned} v_x(24 \cdot T) &= v_{P,x} - \tfrac{v_{P,x}}{k_D^2} \cdot 24^2 = 3 - \tfrac{3}{900} \cdot 24^2 = 1.08[mm/s] \\ v_y(24 \cdot T) &= v_{P,y} - \tfrac{v_{P,y}}{k_D^2} \cdot 24^2 = 4 - \tfrac{4}{900} \cdot 24^2 = 1.44[mm/s] \\ v_z(24 \cdot T) &= v_{P,z} - \tfrac{v_{P,z}}{k_D^2} \cdot 24^2 = 4.0752[mm/s] \end{aligned} \tag{29}$$

6 About computation for a motion on a circular trajectory, with mixt profile of speed

The previous example considered a linear trajectory. A circular trajectory imposes the computation of waypoints on spherical coordinates, named radium: R, polar angle: φ and azimuthal angle: ϕ; figure 11, and conversion on Cartesian coordinates of those values. [6] The acceleration and deceleration is similar with the method described about a linear trajectory, regarding tangential speed. The variation of tangential speed defines the variation of angular speed, named ω, figure 10. [6] Software implementation considered the maximum acceleration possible of motion speed, about rotation around each Cartesian coordinate system axle. A table about deceleration stage was defined about vertical rotations another table was defined about horizontal rotations.

An example of computation may consider $k_A = 3$; this values is defined by imposed values of motion speed. Let consider the motion a rotation of $2\pi/4 = 90°$ around axle OY. In the first period of time, the motion speed is: $v_1 = v_0 + a_M$, it result the angular speed of rotation $\omega_1 = \omega_0 + \Delta\omega$; in the second and third period of time the angular speed increase with the same value $\Delta\omega$; it results: $\omega_2 = \omega_1 + \Delta\omega = \omega_0 + 2 \cdot \Delta\omega$ and $\omega_3 = \omega_2 + \Delta\omega = \omega_0 + 3 \cdot \Delta\omega$. In the fourth period of time, (after motion beginning), the angular speed reaches the programed speed, ω_p.

Comparing the angular programmed speed value with values from Deceleration table, it may results the value $k_D = 5$ and all the values of angular speed, about deceleration stage. The distance necessary for deceleration stage on a circular trajectory, named DEC may be computed, where $\omega_{DEC}(k)$ are the smallest five values from deceleration stage (the Deceleration table simplifies the software implementation of deceleration stage):

$$DEC = \sum_{k=1}^{k_D=5} T \cdot \omega_{DEC}(k) \tag{30}$$

It results the number of sampling period of time for motion on trajectory stage (let consider $\omega_0 = 0$):

$$k_T = \frac{2\pi/4 - (\Delta\omega + 2 \cdot \Delta\omega + 3 \cdot \Delta\omega) \cdot T - DEC}{T \cdot \omega_P} \tag{31}$$

A variable orientation of robot arm, during the motion, involves a similar computation as described for a circular trajectory, but applied about computation of azimuthal and polar angle of each orientation versor; (the orientation versors are: $\vec{n}$; $\vec{o}$; $\vec{a}$).

Conclusions

About a robotic arms motion, those two conditions are very difficult to accomplish: best precision at the motion end and exact defined motion time. Those two conditions are accomplished by mixt profile (of motion speed variation), described in this paper.

The advantages of mixt profile are: the best precision to reach the end point of robotic arms motion, exact determination of motion time and minimum time of acceleration up to programed motion speed.

The method may have others diverse applications, about motion on a linear or circular trajectory; for example about turning or milling process.

Motion execution with exact speed gives processing quality; the described method of mixt profile, about speed variation, was implemented on numerical control equipment and precision was accurate (for example: numerical control equipment for workpieces of sintered metal carbides).

Bibliography

[1] Horsch, T.; Juttler, B.; Cartesian Spline Interpolation for Industrial Robots. University of Technology, Departament of mathematics, Darmstadt, Germany (http://www.ag.jku.at/pubs/csi98.pdf)

[2] Matica, L.M.; Kovendi, Z. (2011); Structure Analysis for an Industrial Robot, *Journal of Computer Science and Control Systems*, ISSN 1844-6043, 4(1): 89-92.

[3] Matica, L.M. (2008); *Conducerea robotilor industriali*, Edit. Univ. din Oradea, ISBN 978-973-759-481-5.

[4] Choset, H. and all, *Principles of Robot Motion*; https://mitpress.mit.edu/books/principles-robot-motion

[5] Laumond, J.P.; *Robot Motion Planning and Control*, ISBN: 978-3-540-76219-5 (Print) 978-3-540-40917-5 (Online), http://link.springer.com/book/10.1007%2FBFb0036069;

[6] Matica, L.M.; Oros, H (2016); Speed Computation in Movement followed by Accurate Positioning of Industrial Robots, *Computers Communications and Control (ICCCC), 2016 6th International Conference on*, IEEE Xplore, e-ISSN 978-1-5090-1735-5, DOI: 10.1109/ICCCC.2016.7496741, 75 - 79.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(1):90-102, February 2017.

Initial Phase Proximity for Reachback Firefly Synchronicity in WSNs: Node Clustering

M. Misbahuddin, R.F. Sari

Misbahuddin Misbahuddin*
Department of Electrical Engineering, Faculty of Engineering,
Universitas Indonesia, Depok, 16424, Indonesia
*Corresponding author: misbahuddin@ui.ac.id

Riri Fitri Sari
Department of Electrical Engineering, Faculty of Engineering,
Universitas Indonesia, Depok, 16424, Indonesia
riri@ui.ac.id

Abstract: Synchronicity is one of the essential basic services to support the main duties of Wireless Sensor Networks (WSNs). Synchronicity is the ability to arrange simultaneously collective actions in WSNs. A high-rate data sampling to analyze the seismic structure and volcanic monitoring is the important applications requiring a synchronicity. However, most of the existing synchronicity algorithm is still executed in a flat network, so that it requires a long time to achieve a synchronous condition. To increase the convergence rate, the synchronicity can be executed concurrently through a clustering scheme approach. In this work, the such scheme is called as the Node Clustering based on Initial Phase Proximity for Reachback Firefly Synchronicity (NCIPP-RFS). The NCIPP-RFS solves in three steps: (1) constructing the node clustering, (2) intra-cluster synchronicity, and (3) inter-cluster synchronicity. The NCIPP-RFS method is based on the firefly-inspired algorithm. The fireflies are a species in the natural system, which are able to manage their flashing for synchronicity in a distributed manner. The NCIPP-RFS was implemented in NS-3 and evaluated and compared with Reachback Firefly Algorithm (RFA). The simulation results show a significant increase in the convergence rate. The NCIPP-RFS can reach a convergence time shorter than the RFA. In addition, the NCIPP-RFS was compared in the various numbers of clusters, where the least number of clusters can reach the fastest convergence rate. Finally, it can also contribute significantly to the increase of the convergence rate if the number of nodes is greater than or equal to 50 nodes.
Keywords: wireless sensor network, synchronicity, node clustering, phase proximity, firefly-inspired algorithm.

1 Introduction

Wireless Sensor Networks (WSNs) is a set of spatially distributed autonomous sensor nodes that could to interact locally, and some nodes of them interact with a sink node or Base Station. A sensor node in WSNs is a device that has the limitations in the processing unit, communication resources, and sensing capabilities. The WSNs has been implemented in various applications for environmental monitoring, condition sensing, and process automation such as battlefield surveillance, habitat monitoring, coordinated target detection and localization, chemical attacked detection, ubiquitous healthcare, and home automation. The duties of the WSNs can be executed properly when it is supported by robust basic requirements such as time synchronization, synchronicity, self-configuration, and self-localization. For example, time synchronization is extremely required to ensure the high accuracy measurement for an event-driven measurement application in an area where an event is detected [1]. Synchronicity inspired by the biological

principle of a firefly is adopted by [2] for handling a dynamic node clustering in data readings. Self-configuration characteristic for a routing protocol of WSNs developed by [3] aimed at discovery the best route for delivering information with minimum energy consumption. Finally, self-localization is a basic feature in WSN for finding out the location of unknown nodes because GPS that is usually used to find the location of a device is not suitable for WSNs [4].

Many phenomena exist in the natural system around us, which have inspired many scientists to solve various problems within the engineering field or the specific problems of WSNs. The basic requirements of WSNs such as synchronization and synchronicity need a self-organized way. Both are a mutually complementing requirement of WSNs. The synchronization is an ability to align the time of node's internal clock referring on the global time to perform a simultaneously collective action. On the other hand, the synchronicity is a way to align the phase of the internal clock of node to conduct a synchronously collaborative action. The application of the synchronicity in WSNs is very useful as a simple sensor network coordinator in sampling the high data rates such as seismic analysis of structure [5], and volcanic monitoring [6]. Moreover, is also used as a scheduling mechanism of the node duty cycles in order to save energy, so that all nodes in a network can wake up at the same time.

There are some challenges of the robust synchronicity requirements, i.e. simple, fast, low energy consumption, self-configuration and high scalability. In this study, we propose a new node-clustering synchronicity method using a firefly-inspired algorithm to address three synchronicity requirements in simple, fast,and self-configured way.

Clustering in the WSNs is often used in some applications because it is extremely useful for various purposes. The purposes are divided as primary and secondary [7]. The first purpose represents the objectives that are the most substantial in the node clustering such as scaling, fault-tolerance, data aggregation, load balancing, network topology stabilization, network lifetime extension. Instead, the second purpose points out the objectives that are not highly important, and they are indirectly achieved by clustering node such as increasing connectivity, reducing routing delay, avoiding collision, and utilizing sleeping schemes. Therefore, in this research we propose the node clustering model as a new approach to synchronize the node in the network, where the node clustering is an extremely important requirement to relieve the load of the network to reach the synchronicity.

Our NCIPP-RFS approach provides two main contributions that consist of (i) in clustering, there are some node subsets that perform an intra-cluster synchronicity in parallel, continued to an inter-cluster synchronicity. The total of periods required to execute both synchronicity processes is smaller than that of without clustering. This emphasizes that our approach is faster than the non-clustered synchronicity algorithm. (ii) Self-configuration is an important problem in the distributed system for executing both its primary duties and basic function as well as for the synchronicity function. In fact, there are many natural phenomena around us in which their population can organize themselves toward a synchronicity state, in which one of them is the firefly. The behavior of firefly flashing has inspired a number of researchers to create the firefly-inspired synchronicity algorithms [8]-[13]. In this research, we utilize the synchronicity algorithm developed by [10] for intra-cluster synchronicity.

The remainder of this paper is organized as follows: Section 2 presents literature review related with the firefly-inspired synchronicity. Section 3 describes the approach used to solve the synchronicity requirements. Section 4 presents the simulation results to show the performance evaluation. Finally, Section 5 concludes this paper and ideas for future work.

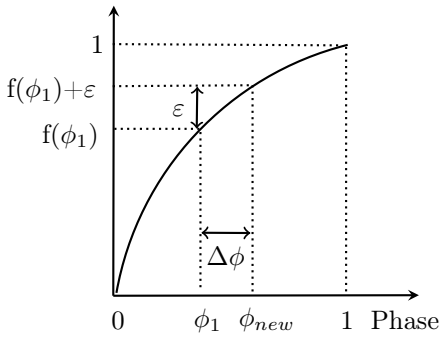


Figure 1: The state function of M&S model

2 Firefly-inspired Synchronicity

In the natural system, there are several synchronicity phenomena that have been observed by many researchers to understand the mutual interaction of a population toward a self-organized synchronicity without a notion of time. Examples of the natural synchronicity include circadian rhythm [14], pacemaker cell of the heart [15], and synchronous flashing of fireflies [16].

Fireflies are one of the species around us, which can interact mutually to fire synchronously. This phenomenon is one of the most spectacular self-organized synchronicity, which was imitated as a firefly-inspired synchronicity algorithm. The population of the fireflies can be analogized as a population of the pulse-coupled biological oscillators (PCO) that was introduced by Mirollow and Strogatz [8], which is known as the M&S model. However, this model cannot be implemented directly in a real WSNs because it still uses some ideal assumptions that are not in accordance with the realistic wireless communication [17]. The assumptions are: (1) the characteristic of oscillators is identical, (2) the node's firing event occurs and other nodes respond instantaneously. (3) Node's computations are conducted perfectly and immediately.

The M&S model presents a basic concept of the firefly-inspired synchronicity algorithm, which is described through two pulse-coupled oscillators. Each oscillator is characterized by a monotonically increasing and concave down function representing a firing function as shown in Figure 1. When the oscillator's phase increase monotonically to reach a threshold, it fires and falls immediately to zero. The mutual interaction occurs between two oscillators when an oscillator fires and sends a firing message to another oscillator that causes another oscillator responds by adjusting its own phase toward firing.

As a result, it will jump to a new phase ϕ_{new} with coupling strength ε. The new phase can be calculated using the following equation [6]:

$$\phi_{new} = \min(1, f^{-1}(\phi) + \varepsilon)) \tag{1}$$

where $f(\phi)=\frac{1}{b}.\ln(1+[e^b\text{-}1].\phi)$, and b is a dissipation parameter.

The essential weakness of the M&S model, which violates the practical WSNs, is when responding immediately a firing message that is sent by its neighbor nodes without considering an unpredictable delay because of the channel contention prior to message transmission. Therefore, the Reachback Firefly Synchronicity (RFA) developed by Werner-Allen et al.[10] overcomes the realistic wireless communication problems of the PCO model. Three problems handled by the RFA are related the wireless communication and one problem of the load computation. They are

the timestamping message, the notion of pre-emptive message staggering, reachback response, and simplified firing function. The amount of time a message that was delayed before being broadcasted can be estimated using the low-level timestamping. The oscillator of the PCO model that reacts shortly to each firing event can be overcome utilizing the notion of reachback response, that is, all firing events as the phase jumps are recorded and are calculated once at the end of each period that is used to jump at the beginning of the cycle. The wireless contention, in the worst case, can be avoided employing the notion of pre-emptive message staggering. Finally, the computation complexity is reduced by simplifying the firing function. The detail description of four ways to be applied in the real wireless network are as follows:

1. A node experiences a delay between when it reaches a firing and when it starts to transmit a message. The delay can be estimated using MAC-layer timestamping. The measurement of the MAC-delay can be started using a trigger to record it in the header of the outgoing message when the message is transmitted. In the receiver node, the information is used to determine the proper firing time by calculating the difference between the MAC-delay and the reception time of the message.

2. In the M&S model, a node responds immediately to each firing message from other nodes. In contrast, the RFA uses the notion of reachback response to record each received firing message and calculates them as the phase jumps once at the end of each period, which is used as a jump at beginning of the next cycle as illustrated in Figure 2. This approach will be discussed in more detail in subsection 3.2.

3. In the M&S model, perhaps many nodes transmit the firing message together when they fire simultaneously. This event is the worst case of the CSMA scheme because they can cause channel collisions. To avoid such a worst case and to control the extent of the message delay, the RFA model introduced a notion of pre-emptive message staggering by adding a random transmission delay to the firing message at the application level of the node. The delay value is assigned to a uniformly random value between 0 up to a constant D before node fires. Furthermore, a random waiting time W (where D<W<<T) is added for a node that has fired to receive the delayed messages before processing the firing message in the queue.

4. The M&S model's firing function can be simplified to make a faster calculation of firing response. Therefore, the RFA algorithm reduces the computation complexity by simplifying the firing function. The new phase obtained after receiving a firing message at phase ϕ_1 is $\phi_2 = \phi_1 + \varepsilon\phi_1$ or $\phi_2 = (1 + \varepsilon)\phi_1$.

3 The Proposed Method of Node Clustering Based on Phase Proximity for Firefly-inspired Synchronicity

The challenges of synchronicity in the wireless sensor network are how to increase the convergence rate, and to handle the latency delay in wireless communication. Both challenges aim to reduce the power consumption. The model proposed by Sun Yi et al. [18] is a centralized node clustering mechanism through sorting the phase value of nodes. This model is difficult to be implemented in WSNs because they require a node as coordinator to gather the phase value of all nodes. Furthermore, the coordinator has to be able to organize the phase order of all nodes in order to construct the node clustering. This research proposes a new decentralized node clustering mechanism based on original phase proximity of nodes where the nodes in the

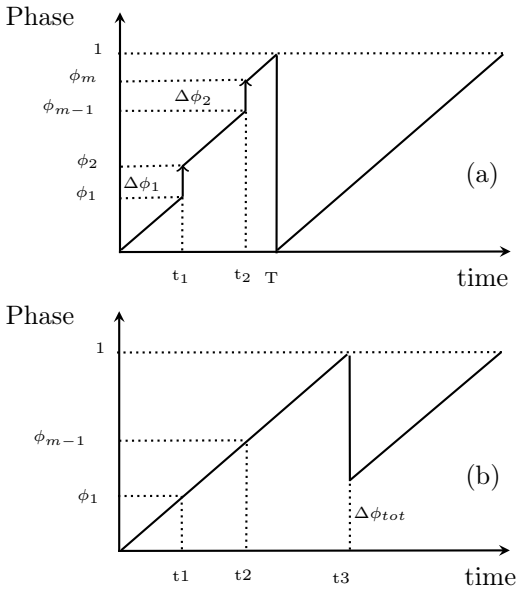


Figure 2: Phase jump in (a) M&S model, and (b) RFA model

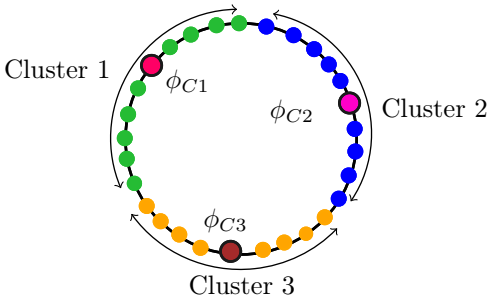


Figure 3: Illustration of the phase cluster for three clusters

network construct the clusters through a self-configuration way. This approach is called Node Clustering based on Initial Phase Proximity for Reachback Firefly Synchronicity (NCIPP-RFS).

Generally, synchronicity steps of the NCIPP-RFS are similar to the CFSA method: cluster-construction, intra-cluster synchronicity, and inter-cluster synchronicity. However, unlike CFSA method, NCIPP-RFS uses a new approach in the cluster construction. The clusters are constructed based on the original phase value proximity of nodes that are organized in a self-configuration way without assistance a node coordinator. Moreover, the inter-cluster synchronicity is added with a Late Sensitive Window (LSW) factor to avoid the overshoot of jump in a group or cluster.

3.1 Construction of the Clusters Based on Phase Proximity

Node Clustering Based on Initial Phase Proximity (NCIPP) is a simple algorithm used for clustering in WSNs. This algorithm partitions the nodes into k clusters. The node clustering is constructed using the following steps:

1. Arbitrarily assign k points as the center clusters, k being the number of clusters desired. The assignment of the center point of the cluster is intended to ensure that each node only

exists in a cluster. Next, each node will internally assign its own cluster by calculating a nearest distance to every center point of the cluster (ϕc_1, ϕc_2, ϕc_j, ...,ϕc_k). The distance can be computed by the following Euclidean equation:

$$d\phi_{ij} = \sqrt{\phi_i^2 - \phi c_j^2} \tag{2}$$

where $d\phi_{ij}$ is the distance between the i^{th} node's phase ϕ_i and the j^{th} center point of the cluster ϕc_j, and the node's phase is $\phi_i \in [0\ 1]$. Each node in the network executes the cluster-initialization function internally to determine its own cluster. The pseudo code is shown in Algorithm 1 in which the cluster number can be obtained through the minimum value of the clusterArray. The center points are determined based on the number of clusters k that is desired, where they are assigned through a balanced division of the distribution of phase values (between 0 and 1) in k groups. For illustration, let's assume that there are three clusters ($k = 3$) where the center point of the clusters are denoted by $\phi c_1 = 0.167$, $\phi c_2 = 0.5$, and $\phi c_3 = 0.833$ respectively. When the uniformly distributed phase values are mapped in a circle window in which the zero and one phase coincide in a point, the nodes will form three clusters as shown in Figure 3.

2. Each node provides an internal data structure called neighborNodes to store the source address (srcAddr) of its neighbor nodes in the same cluster and has a cluster number variable (noCluster).

3. The node clustering is executed by two main functions, i.e., the BroadcastIDCluster and the RecieveIDCluster. The pseudo codes of the both functions are shown in Algorithm 2. The *BroadcastIDCluster* is a function to broadcast the source address, and the cluster number of the transmitter node to other nodes in the same cluster. To avoid simultaneous transmission in the periodic broadcast, it is added with a random interval delay and a beacon expire timer in the *BroadcastIDCluster*. On the contrary, the *ReceiveIDCluster* is a function to receive the broadcasting messages. If the sender's cluster number is equal to the receiver's cluster number, the receiver node stores the sender's address and cluster number into its data structure, the *neighborNodes*. This means that they are in the same cluster.

Algorithm 1 Cluster initialization

```
ClusterInitilization()
Assign the number of cluster k
Assign the center point of the clusters (φc1, φc2, ...,φck )
for j:k do
    clusterArray[j] ←dφij
noCluster ← index of min(clusterArray)
```

3.2 Intra-cluster Synchronicity

After all logical clusters based on the initial phase are constructed. Every logical cluster can start to perform a synchronicity concurrently. There are two firefly-inspired algorithms that can be used to perform the synchronicity, i.e., M&S and RFA model. The M&S model still uses a simple algorithm without considering the delay effect of real communication in WSNs. A firing

Algorithm 2 Node Clustering

```
BroadcastIDCluster(srcAddr, noCluster)
Send(srcAddr, noCluster)
Delay(interval+rand())
BeaconExpireTimer ()

ReceiveIDCluster(srcAddr, noCluster)
if (noClusterOfSender=noCluster) then
    neighborNodes.srcAddr ← srcAddr
    numCluster ← numCluster+1
```

message that is sent by a node is responded instantaneously by its adjacent nodes regardless of its unpredictable delay because of the delay latencies [8]. For illustration, let's assume that in a cluster, node x receives a firing message from another node y at time t_1 or phase ϕ_1. In response, the node x adjusts its oscillator's phase by increasing slightly to a new phase ϕ_2 that can be computed through the following firing function:

$$\phi_2 = \min(1, (1+\varepsilon)\phi_1) \tag{3}$$

Every time node x receives a firing message from any node in same cluster; it calculates its phase jump using the following equation:

$$\Delta\phi_i = \min(1, (\phi_2 - \phi_1)) = \min(1, \varepsilon\phi_1) \tag{4}$$

To address the delay latency in a real network, a reachback response mechanism of the RFA model is a proper choice. In this model, the node x does not jump every time it receives a firing message, but it always stores its entire phase jumps until the period end. Furthermore, the sum of phase jumps in the previous period will be used to jump at the beginning of next period. Figure 2a illustrates the M&S model where the node x receives two firing messages at time t_1 and t_2 respectively. In contrast, in the RFA model node x will jump at time t_3 in the beginning of the next period as shown in Figure 2b. The sum of phase jumps can be calculated through the following equation:

$$\Delta\phi_{tot} = \sum_{i=1}^{n} \Delta\phi_i \tag{5}$$

where n is the number of firing events received in a period.

3.3 Inter-cluster Synchronicity

After all clusters achieve the synchronicity condition, every cluster can be considered as a converging node group. It can be assumed as a firing node if its phase comes near to one. In this case, the RFA mechanism can still be used to reach the synchronicity condition for all nodes in the network. The synchronicity condition can be completed through sending the firing message among the node groups. The number of node groups will decrease, which are caused by the merging of a few of the node groups into one group and finally there is only one group as the synchronicity condition of the network.

In a period the group G_x will receive the firing messages from the firing group G_y equal to the number of the G_y's members. The sum of phase jumps of group G_x is as follows:

$$\Delta\phi_{(tot),G_x} = \sum_{i=1}^{n_y} \Delta\phi_{i,G_x} \tag{6}$$

where n_y is the number of G_y's members. This enlarged the sum of phase jumps of group G_x, so that it can cause an overshooting jump in the nodes of the groups G_x. To avoid this event, the RFA model is added a Late Sensitive Window (LSW) introduced by [11], where the window inhibits the response of node against the firing messages falling outside of the window. When group G_x achieves phase $\phi = 1$ (or $t = T$), it fires and adjusts its phase based on the sum of phase jump in the previous period and jumps in the beginning of next period. The phase jump of the group G_x is calculated by the following equation:

$$\Delta\phi_{G_x} = \begin{cases} 0 & \phi \leq LSW \quad \text{and} \quad \epsilon\phi \geq 1 \\ \varepsilon\phi & \phi \geq LSW \quad \text{and} \quad \epsilon\phi \leq 1 \\ 1 & \text{otherwise.} \end{cases} \tag{7}$$

4 Simulation Results

The performance of both RFA and NCIPP-RFS method is evaluated using a network simulator NS-3 to obtain the convergence rate. A total of 40 simulations were run for each coupling strength in all scenarios. Hence, results presented in Figure 4, 5 and 6 for each coupling strength show the average values and the corresponding 95% confidence interval. The network topology uses a mesh type in a variety of sizes of 25, 50, 75, and 100 nodes spread uniformly in a constricted rectangular area sized 1000 x 1000 m^2. Furthermore, the network is designed to represent a realistic environment to evaluate the convergence rate of the synchronicity. Several simulation parameters used to evaluate the convergence rate of the synchronicity are varied in the simulation both RFA and NCIPP-RFS model as shown in Table 1. The coupling strengths (ε) are varied in the range of 0.01 through 0.1 for cluster 2, 3, 4, and 5. The metric evaluation used to measure the convergence rate is the number of periods required to reach the synchronicity condition.

Table 1: Simulation parameters

Parameter	**Value**
Oscillating period T (milliseconds)	100
Phase of oscillator ϕ	$\phi \in [0,1]$
Phase rate $\frac{d\phi}{dt}=\frac{1}{T}$	0.01
Coupling strength ϵ	$\epsilon \in [0.01, 0.1]$
The number of nodes	25, 50, 75, 100
The number of clusters	2, 3, 4, 5

Figure 4 shows a set of charts that evaluates the influence of the coupling strength parameter on the number of periods in both the RFA and the NCIPP-RFS. It presents some comparison of the convergence rate between the RFA and the NCIPP-RFS with various number of clusters. It can be seen that the convergence rate of the RFA is comparable to the NCIPP-RFS in 25 nodes. However, at the number of nodes, 50, 75 and 100, the convergence rate of the RFA is much slower than the NCIPP-RFS, especially in the lower coupling strength of 0.01 through 0.05. This happens because of the influence of the clustering way in the less the number of nodes to increase the convergence rate does not contribute significantly. Actually, the such way is to synchronize some groups of nodes in a network concurrently to reduce the load of the network. If node has not been synchronized in the network, all other nodes must keep the effort to synchronize it. Therefore, the clustering method of 25 nodes in the RFA results in the convergence rate similar to the NCIPP-RFS. Otherwise, in the larger the number of nodes (50, 75, and 100 nodes), the

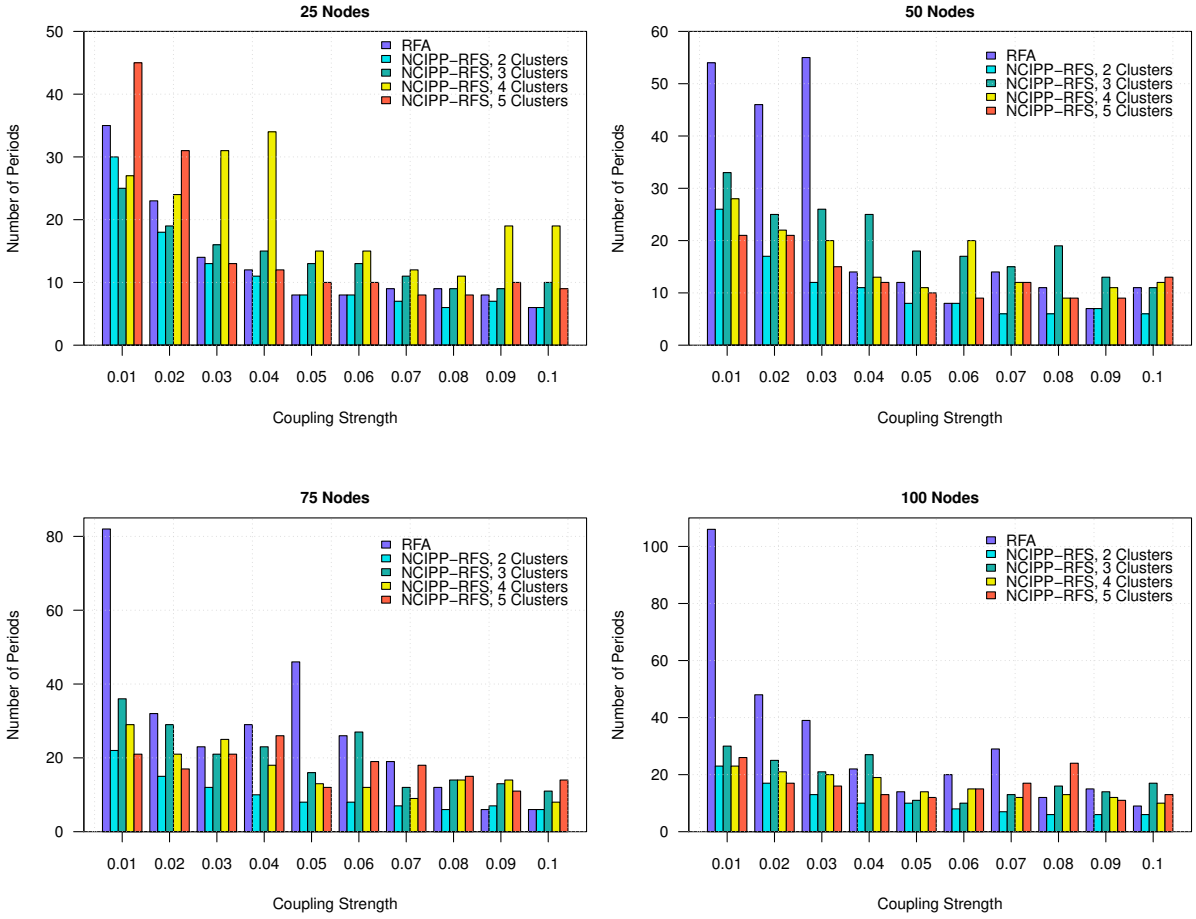


Figure 4: The comparison of the convergence rate between the RFA and the NCIPP-RFS model when the number of clusters is varied for 25, 50, 75, and 100 Nodes.

convergence rate of the NCIPP-RFS declines along with the increasing number of nodes. The influence of the clustering method in synchronicity that contributes significantly to the increase in the convergence rate is in the larger the number of nodes. Finally, the number of nodes which are greater than or equal to 50 nodes generates the best performance of the NCIPP-RFS among all scenarios when the number of node is varied.

A set of charts presented in Figure 5 is to discover the best performance among all scenarios when number of clusters is varied in the NCIPP-RFS model. The chart on the top left shows the convergence rate of the NCIPP-RFS in the two-clusters scenario. It shows that the number of periods declining rapidly along with the gradual increase of the coupling strength value in all scenarios of the number of nodes. This means that the NCIPP-RFS of two clusters will show a better performance. This is in line with the aim of the adjustment of the coupling strength parameter to increase the convergence rate gradually. Next, in the chart for the experiment with three clusters, the convergence rate slows down along with the rising number of nodes to 75 nodes, but it can increase the convergence rate in 100 nodes. Similarly, the larger the number of clusters (4 and 5) the NCIPP-RFS is slower than in the scenario for two clusters. This happens because the increase of the number of clusters causes the increase of the number of periods to reach a convergence in the inter-cluster synchronicity, so that they will slow down the convergence rate of the synchronicity process in the network. Therefore, the best performance of the NCIPP-RFS among all scenarios of the number of clusters is the scenario of two clusters.

An important problem that needs to be understood is the caution of the highly different convergence rate to the others as shown in Figure 5. They are 50 nodes with coupling strength of 0.01 in two clusters, 75 nodes with coupling strength of 0.01 in three clusters, 25 nodes with coupling strength of 0.04 in four clusters, and 25 nodes with coupling strength of 0.01 in five clusters. These problems are caused by the process of inter-cluster synchronicity that is hard to reach a synchronicity percentage of 100 percent as shown in Figure 6. It can be seen that the graphs express the movement of the synchronicity percentage that is constant in some periods causing a difficulty to rise forward to 100 percent. This problem will slow down the convergence rate of the synchronicity.

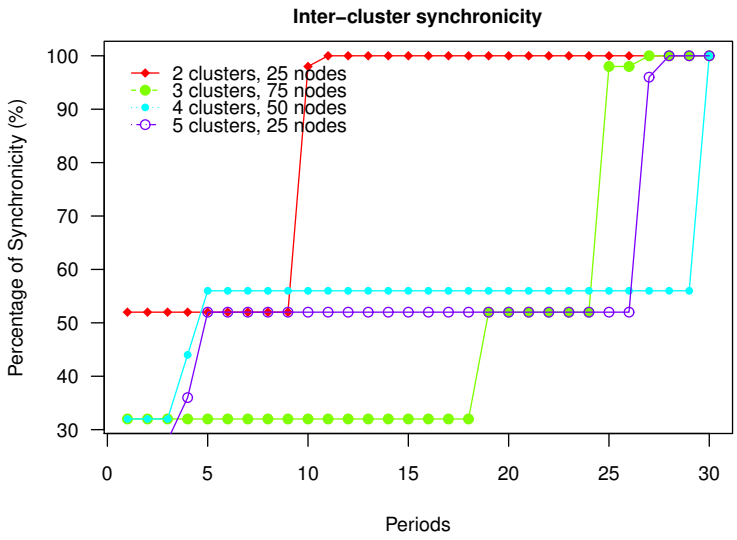


Figure 6: Inter-cluster synchronicity that is difficult to reach a synchronicity convergence

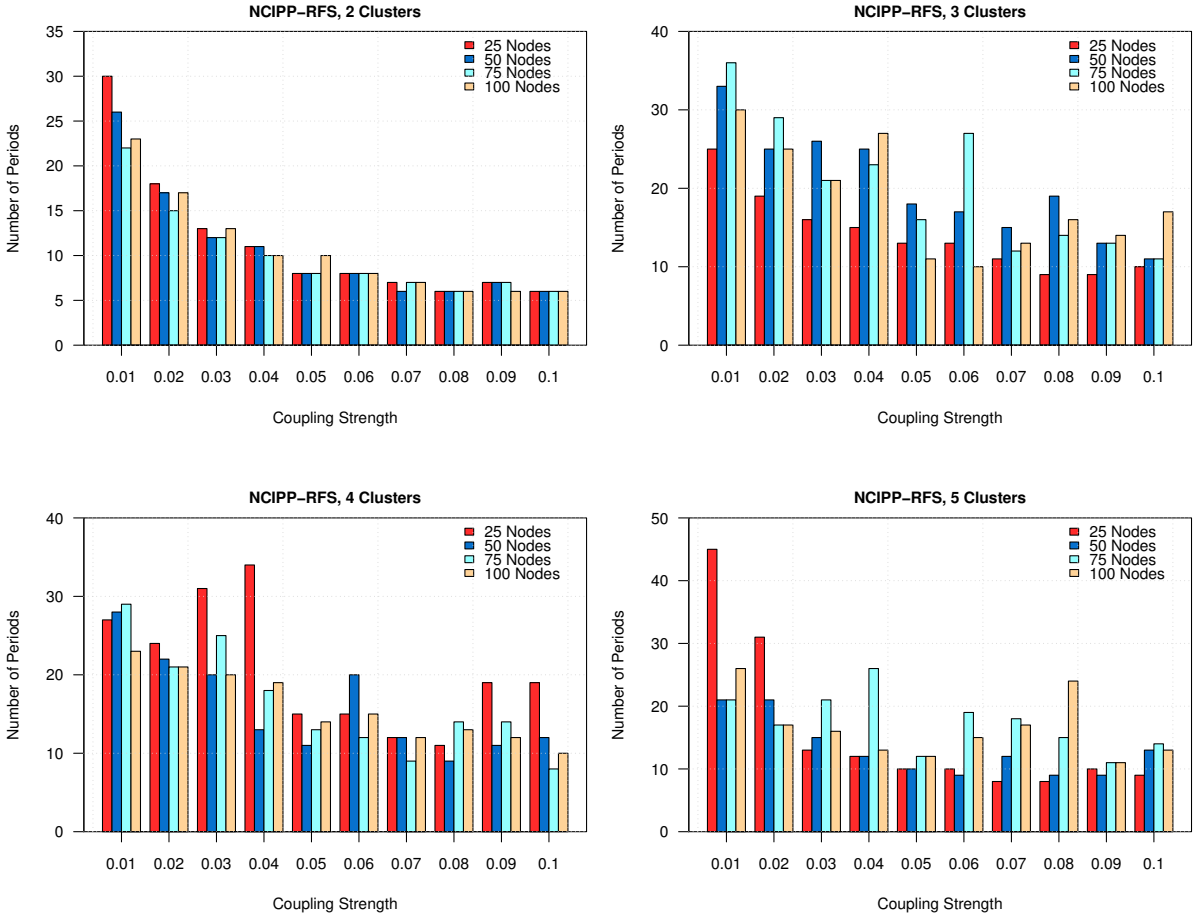


Figure 5: The comparison of the convergence rate among the number of nodes in the varied number of clusters.

Conclusions

The clustering scheme approach based on the initial phase proximity is a way to synchronize all nodes in a network executed concurrently. The clusters are constructed through a self-configured way. The intra-cluster synchronicity uses the Reachback Firefly Algorithm (RFA) simplifying the firing function and considering the realistic transmission effects. The inter-cluster synchronicity utilizes the RFA with the Late Sensitive Windows (RFA with LSW) model to avoid the overshooting jumps due to the flood of the firing messages from a node group to others.

We compare the RFA and the NCIPP-RFS clustering scheme using the Network Simulator (NS-3). The simulation results show that the convergence rate of the NCIPP-RFS can reach a synchronicity faster than the RFA. Moreover, the NCIPP-RFS can contribute significantly to the increase of the convergence rate if the number of nodes is greater than or equal to 50 nodes.

The NCIPP-RFS was also evaluated and compared in the various number of clusters. The best performance of the NCIPP-RFS among all scenarios of the number of clusters is the scenario of the least number of clusters. However, the movement of the synchronicity percentage that is constant in some periods is difficult to rise toward the 100 percent.

As future work, we will implement our approach to test it in a data-gathering application referring a temporal and spatial relationships.

Acknowledgement

The authors gratefully acknowledge the Ministry of Research, Technology and Higher Education of Indonesia for supporting this work through DIKTI's research grant under grant No. 1173/UN2.R12/HKP.05.00/2016

Bibliography

[1] F. Lamonaca, A. Gasparri, E. Garone, and D. Grimaldi (2014), Clock Synchronization in Wireless Sensor Network With Selective Convergence Rate for Event Driven Measurement Applications, *Instruments. IEEE Trans.*, 63(9): 2279-2287.

[2] F. Gielow, G. Jakllari, M. Nogueira, and A. Santos (2015), Data similarity aware dynamic node clustering in wireless sensor networks, *Ad Hoc Networks*, 24: 29-45.

[3] H. Araujo, W. L. T. de Castro, and R. Holanda Filho (2010), A proposal of self-configuration in Wireless Sensor Network for recovery of broken paths, *Sensors Applications Symposium (SAS), 2010 IEEE*, 245-250.

[4] K. Kyungmi, K. Hyunsook, and H. Youngchoi (2009), A Self Localization Scheme for Mobile Wireless Sensor Networks, *Computer Sciences and Convergence Information Technology, 2009. ICCIT 09. Fourth International Conference on*, 774-778.

[5] A. K. Mohapatra, N. Gautam, and R. L. Gibson (2013), Combined Routing and Node Replacement in Energy-Efficient Underwater Sensor Networks for Seismic Monitoring, *Oceanic Engineering, IEEE Journal of*, 38: 80-90.

[6] R. Lara, D. BenĂtez, A. Caamato, M. Zennaro; J. L.Rojo-Alvarez (2015), On Real-Time Performance Evaluation of Volcano-Monitoring Systems With Wireless Sensor Networks, *IEEE Sensors Journal*, 15(6): 3514-3523.

[7] M. M. Afsar and M.-H. Tayarani-N. (2014), Clustering in sensor networks: A literature survey, *J. Netw. Comput. Appl.*, 46: 198-226.

[8] R. E. Mirollo and S. H. Strogatz (1990), Synchronization of pulse-coupled biological oscillator, *SIAM J. Appl. Math.*, 50(6): 1645-1662.

[9] N. Yu, B. J. d'Auriol, W. Xiaoling, W. Jin, C. Jinsung, and L. Sungyoung (2008), Selective Pulse Coupling Synchronicity for Sensor Network, *Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on*,123-128.

[10] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal (2005), Firefly-inspired sensor network synchronicity with realistic radio effects, *Proc. of the 3rd international conference on Embedded networked sensor systems*, ACM, San Diego, California, USA, 142-153.

[11] A. Tyrrell, G. Auer, and C. Bettstetter (2006), Fireflies as Role Models for Synchronization in Ad Hoc Networks, *Bio-Inspired Models of Network, Information and Computing Systems*, 1-7.

[12] J. Yanliang, M. Wei, B. Zhishu, and Z. Xuqin (2013), Blind and buffer phase area based on M&S model fireflies synchronization in WSNs, in Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), *2013 First International Symposium on*, 1-4.

[13] C. Lin and W. HongPeng (2009), Reachback Firefly Synchronicity with Late Sensitivity Window in Wireless Sensor Networks, *Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on*, 1: 451-456.

[14] F. A. dos S. Silva, S. R. Lopes, and R. L. Viana (2016), Synchronization of biological clock cells with a coupling mediated by the local concentration of a diffusing substance, *Commun. Nonlinear Sci. Numer. Simul.*, 35: 37-52.

[15] Y. Yaniv, I. Ahmet, J. Liu, A. E. Lyashkov, T.-R. Guiriba, Y. Okamoto, B. D. Ziman, and E. G. Lakatta (2014), Synchronization of sinoatrial node pacemaker cell clocks and its autonomic modulation impart complexity to heart beating intervals, *Heart Rhythm*, 11(7):1210-1219. doi: 10.1016/j.hrthm.2014.03.049.

[16] D. Kim (2004), A spiking neuron model for synchronous flashing of fireflies., *Biosystems*, 76(1-3): 7-20.

[17] R. Leidenfrost and W. Elmenreich (2009), Firefly clock synchronization in an 802.15. 4 wireless network, *EURASIP J. Embed. Syst.*, 1-17, DOI: 10.1155/2009/186406.

[18] S. Yi, J. Qing, and Z. Kai (2012), A clustering scheme for Reachback Firefly Synchronicity in wireless sensor networks, *Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on*, 27-312.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(1):103-115, February 2017.

Two Flow Problems in Dynamic Networks

C. Schiopu, E. Ciurea

Camelia Schiopu*, Eleonor Ciurea
Transilvania University of Braşov
Romania, 500091 Braşov, Iului Maniu, 50
camelia.s@unitbv.ro, e.ciurea@unitbv.ro
*Corresponding author: camelia.s@unitbv.ro

Abstract: In this paper we study two flow problems: the feasible flow problem in dynamic networks and the maximum flow problem in bipartite dynamic networks with lower bounds. We mention that the maximum flow problem in bipartite dynamic networks with lower bound was presented in paper [8][a]. For these problems we give examples.
Keywords: dynamic networks, feasible flow, bipartite network, maximum flow.

1 Introduction

The theory of flow is one of the most important parts of Combinatorial Optimization. The static network flow models arise in a number of combinatorial applications that on the surface might not appear to be optimal flow problems at all. The problem also arises directly in problems as far reaching as machine scheduling, the assignment of computer modules to computer processor, tanker scheduling etc. [1]. However, in some applications, time is an essential ingredient [3], [4], [5]. In this case we need to use the dynamic network flow model. On the other hand, the bipartite static network also arises in practical context such baseball elimination problem, network reliability testing etc. and hence it is of interest to find fast flow algorithms for this class of networks [1], [6].

The maximum flow problem in bipartite dynamic networks with lower bounds was presented in paper [8]. In this paper, which is an extension of [8], we present in addition the feasible flow problem in dynamic networks. These problem have not been treated so far. Further on, in Section 2.1 we discuss some basic notions and results for the maximum flow problem in general static networks with lower bounds. Section 2.2 deals with the maximum flows problem in general dynamic networks with lower bounds. In Section 2.3 we present algorithms for flow problems in bipartite static network and in Section 2.4 we discuss the feasible flows in static networks. In Section 3 we discuss the feasible flows problem in dynamic network and give some examples. Section 4 deals with maximum flows in bipartite dynamic networks with lower bounds and an example.

2 Terminology and Preliminaries

In this section we discuss some basic notations and results used throughout the paper.

2.1 Maximum flows in static networks with lower bounds

Let $G = (N, A, l, u)$ be a static network with the set of nodes $N = \{1, \ldots, n\}$, the set of arcs $A = \{a_1, \ldots, a_k, \ldots, a_m\}$, $a_k = (i, j)$, $i, j \in N$, the lower bound function $l : A \to \mathbb{N}$, the upper

bound (capacity) function $u : A \to \mathbb{N}$, with $\mathbb{N}$ the natural number set, 1 the source node and n the sink node.

For a given pair of subset X, Y of the set of nodes N of a network G we use the notation:

$$(X, Y) = \{(i, j) | (i, j) \in A, i \in X, j \in Y\}$$

and for a given function f on set of arcs A we use the notation:

$$f(X, Y) = \sum_{(X,Y)} f(i, j)$$

A flow is a function $f : A \to \mathbb{N}$ satisfying the next conditions:

$$f(i, N) - f(N, i) = \begin{cases} v, & \text{if } i = 1 \\ 0, & \text{if } i \neq 1, n \\ -v, & \text{if } i = n \end{cases} \tag{1a}$$

$$l(i, j) \leq f(i, j) \leq u(i, j), \quad (i, j) \in A \tag{1b}$$

for some $v \geq 0$. We refer to v as the value of the flow f.

The maximum flow problem is to determine a flow f for which v is maximum.

We further assume, without loss of generality, that if $(i, j) \in A$ then $(j, i) \in A$ (if $(j, i) \notin A$ we consider that $(j, i) \in A$ with $l(j, i) = u(j, i) = 0$).

A preflow f is a function $f : A \to \mathbb{N}$ satisfying the next conditions:

$$f(N, i) - f(i, N) \geq 0, i \in N - \{1, n\} \tag{2a}$$

$$l(i, j) \leq f(i, j) \leq u(i, j), (i, j) \in A \tag{2b}$$

For a preflow f the excess of each node $i \in N$ is

$$e(i) = f(N, i) - f(i, N) \tag{3}$$

and if $e(i) > 0$, $i \in N - \{1, n\}$ then we say that node i is an active node.

Given a flow (preflow) f, the residual capacity $r(i, j)$ of any arc $(i, j) \in A$ is $r(i, j) = u(i, j) - f(i, j) + f(j, i) - l(j, i)$. The residual network with respect to the flow (preflow) f is $\tilde{G} = (N, \tilde{A}, r)$ with $\tilde{A} = \{(i, j) | (i, j) \in A, r(i, j) > 0\}$. In the residual network $\tilde{G} = (N, \tilde{A}, r)$ we define the distance function $d : N \to \mathbb{N}$. We say that a distance function is valid if it satisfies the following two conditions

$$d(n) = 0 \tag{4a}$$

$$d(i) \leq d(j) + 1, (i, j) \in \tilde{A} \tag{4b}$$

We refer to $d(i)$ as the distance label of node i. We say that an arc $(i, j) \in \tilde{A}$ is admissible if satisfies the condition that $d(i) = d(j) + 1$; we refer to all other arcs as inadmissible. We also refer to a path from node 1 to node k consisting entirely of admissilbe arcs as an admissible path.

Whereas the maximum flow problem with zero lower bounds always has a feasible solution (since the zero flow is feasible), the problem with non-negative lower bounds could be infeasible. Therefore the maximum flow problem with non-negative lower bounds can be solved in two phases:

(P1) this phase determines a feasible flow if one is exists;

(P2) this phase converts a feasible flow into a maximum flow.

The problem in each phase essentially reduce to solving a maximum flow problem with zero lower bounds. Consequently, it is possible to solve the maximum flow problem with non-negative lower bounds by solving two maximum flow problems, each with zero lower bounds.

In the next presentation we assume familiarity with maximum flow algorithms, and we omit many details. The reader interested in further details is urged to consult the book [1].

2.2 Maximum flows in dynamic networks with lower bounds

Dynamic network models arise in many problem settings, including production distribution systems, economic planning, energy systems, traffic systems, and building evacuation systems [3], [4], [5].

Let $G = (N, A, l, u)$ be a static network with the set of nodes $N = \{1, \ldots, n\}$, the set of arcs $A = \{a_1, \ldots, a_m\}$, the lower bound function l, the upper bound (capacity) function u, 1 the source node and n the sink node. Let $\mathbb{N}$ be the natural number set and let $H = \{0, 1, \ldots, T\}$ be the set of periods, where T is a finite time horizon, $T \in \mathbb{N}$. Let use state the transit time function $h : A \times H \to \mathbb{N}$ the time lower bound function $e : A \times H \to \mathbb{N}$, the time upper bound function $q : A \times H \to \mathbb{N}$, $e\,(i, j; t) \leq q(i, j; t)$, for all $(i, j) \in A$ and for all $t \in H$. The parameter $h(i, j; t)$ is the transit time needed to traverse an arc (i, j). The parameters $e(i, j; t)$ and $q(i, j; t)$ represents the minimum and respective maximum amount of flow that can travel over arc (i, j) when the flow departs from i at time t and arrives at j at time $\theta = t + h(i, j; t)$.

The maximal dynamic flow problem for T time periods is to determine a flow function $g : A \times H \to \mathbb{N}$, which should satisfy the following conditions in dynamic network $D = (N, A, h, e, q)$:

$$\sum_{t=0}^{T}(g(1, N; t) - \sum_{\tau} g(N, 1; \tau)) = \bar{w} \tag{5a}$$

$$g(i, N; t) - \sum_{\tau} g(N, i; \tau) = 0, i \neq 1, n,\ t \in H \tag{5b}$$

$$\sum_{t=0}^{T}(g(n, N; t) - \sum_{\tau} g(N, n; \tau)) = -\bar{w} \tag{5c}$$

$$e(i, j; t) \leq g(i, j; t) \leq q(i, j; t), \quad (i, j) \in A \ , \quad t \in H \tag{6}$$

$$max \quad \bar{w}, \tag{7}$$

where $\tau = t - h(k, i; \tau)$, $\bar{w} = \sum_{t=0}^{T} v(t)$, $v(t)$ is the flow value at time t and $g(i, j; t) = 0$ for all $t \in \{T - h(i, j; t) + 1, \ldots, T\}$.

Obviously, the problem of finding a maximum flow in the dynamic network $D = (N, A, h, e, q)$ is more complex than the problem of finding a maximum flow in the static network $G = (N, A, l, u)$. Fortunately, this issue can be solved by rephrasing the problem in the dynamic network D into a problem in the static network $R_1 = (V_1, E_1, l_1, u_1)$ called the reduced expanded network.

The static expanded network of dynamic network $D = (N, A, h, e, q)$ is the network $R = (V, E, l, u)$ with $V = \{i_t | i \in N, t \in H\}$, $E = \{(i_t, j_\theta) | (i, j) \in A, t \in \{0, 1, \ldots, T - h(i, j; t)\}$, $\theta = t + h(i, j; t), \theta \in H\}$, $l(i_t, j_\theta) = e(i, j; t)$, $u(i_t, j_\theta) = q(i, j; t), (i_t, j_\theta) \in E$. The number of nodes in the static expanded network R is $n(T+1)$ and number of arcs is limited by $m(T+1) - \sum_{A} \circ h(i, j)$, where $\circ h(i, j) = min\{h(i, j; 0), \ldots, h(i, j; T)\}$. It is easy to see that any flow in the dynamic network D from the source node 1 to the sink node n is equivalent to a flow in the static expanded network R from the source nodes $1_0, 1_1, \ldots, 1_T$ to the sink nodes $n_0, n_1, \ldots, n_T$ and vice versa. We can further reduce the multiple source, multiple sink problem in the static expanded network R to a single source, single sink problem by introducing a supersource node 0 and a supersink node $n + 1$ constructing the static super expanded network $R_2 = (V_2, E_2, l_2, u_2)$, where $V_2 = V \cup \{0, n + 1\}$, $E_2 = E \cup \{(0, 1_t) | t \in H\} \cup \{(n_t, n + 1) | t \in H\}$, $l_2(i_t, j_\theta) = l(i_t, j_\theta)$, $u_2(i_t, j_\theta) = u(i_t, j_\theta)$, $(i_t, j_\theta) \in E$, $l_2(0, 1_t) = l_2(n_t, n + 1) = 0$, $u_2(0, 1_t) = u_2(n_t, n + 1) = \infty$, $t \in H$.

We construct the static reduced expanded network $R_1 = (V_1, E_1, l_1, u_1)$ as follows. We define the function $h_2 : E_2 \longrightarrow \mathbb{N}$, with $h_2(0, 1_t) = h_2(n_t, n+1) = 0$, $t \in H$, $h_2(i_t, j_\theta) = h(i, j; t)$, $(i_t, j_\theta) \in E$. Let $d_2(0, i_t)$ be the length of the shortest path from the source node 0 to the node i_t, and $d_2(i_t, n+1)$ the length of the shortest path from node i_t to the sink node $n+1$, with respect to h_2 in network R_2. The computation of $d_2(0, i_t)$ and $d_2(i_t, n+1)$ for all $i_t \in V$ are performing by means of the usual shortest path algorithms. The network $R_1 = (V_1, E_1, l_1, u_1)$ have $V_1 = \{0, n+1\} \cup \{i_t | i_t \in V, d_2(0, i_t) + d_2(i_t, n+1) \leq T\}$, $E_1 = \{(0, 1_t) | d_2(1_t, n+1) \leq T, t \in H\} \cup \{(i_t, j_\theta) | (i_t, j_\theta) \in E, d_2(0, i_t) + h_2(i_t, j_\theta) + d_2(j_\theta, n+1) \leq T\} \cup \{(n_t, n+1) | d_2(0, n_t) \leq T, t \in H\}$ and l_1, u_1 are restrictions of l_2, u_2 at E_1.

Now, we construct the static reduced expanded network $R_1 = (V_1, E_1, l_1, u_1)$ using the notion of dynamic shortest path. The dynamic shortest path problem is presented in [3]. Let $d(1, i; t)$ be the length of the dynamic shortest path at time t from the source node 1 to the node i and $d(i, n; t)$ the length of the dynamic shortest path at time t from the node i to the sink node n, with respect to h in dynamic network D. Let as consider $H_i = \{t | t \in H, d(1, i; t) \leq t \leq T - d(i, n; t)\}, i \in N$, and $H_{i,j} = \{t | t \in H, d(1, i; t) \leq t \leq T - h(i, j; t) - d(j, n; \theta)\}, (i, j) \in A$. The multiple sources, multiple sinks static reduced expanded network $R_0 = (V_0, E_0, l_0, u_0)$ have $V_0 = \{i_t | i \in N, t \in H_i\}$, $E_0 = \{(i_t, j_\theta) | (i, j) \in A, t \in H_{i,j}\}$, $l_0(i_t, j_\theta) = e(i, j; t)$, $u_0(i_t, j_\theta) = u_1(i, j; t), (i_t, j_\theta) \in E_0$. The static reduced expanded network $R_1 = (V_1, E_1, l_1, u_1)$ is constructed from network R_0 as follows: $V_1 = V_0 \cup \{0, n+1\}, E_1 = E_0 \cup \{(0, 1_t) | 1_t \in V_0\} \cup \{(n_t, n+1) | n_t \in V_0\}$, $l_1(0, 1_t) = l_1(n_t, n+1) = 0$, $u_1(0, 1_t) = u_1(n_t, n+1) = \infty$, $1_t, n_t \in V_0$, $l_1(i_t; j_\theta) = l_0(i_t, j_\theta)$ and $u_1(i_t, j_\theta) = u_0(i_t, j_\theta)$, $(i_t, j_\theta) \in E_0$.

We notice that the static reduced expanded network $R_1(R_0)$ is always a partial subnetwork of static super expanded network $R_2(R)$. In references [4], [5] it is shown that a dynamic flow for T periods in the dynamic network D with $e = 0$ is equivalent with a static flow in a static reduced expanded network R_1. Since an item released from a node at a specific time does not return to the location at the same or at an earlier time, the static networks R, R_2, R_0, R_1 cannot contain any circuit, and therefore, are acyclic always.

In the most general dynamic model, the parameter $h(i) = 1$ is waiting time at node i, and the parameter $e(i; t)$, $q(i; t)$ are lower bound and upper bound for flow $g(i; t)$ that can wait at node i from time t to $t+1$. This most general dynamic model is not discussed in this paper.

The maximum flow problem for T time periods in the dynamic network D formulated in conditions (5), (6), (7) is equivalent with the maximum flow problem in the static reduced expanded network R_0 as follows:

$$f_0(i_t, V_0) - f_0(V_0, i_t) = \begin{cases} v_t, & i_t = 1_t, t \in H_1 \\ 0, & i_t \neq 1_t, n_t, t \in H_1, \\ & \qquad t \in H_n \\ -v_t, & i_t = n_t, t \in H_n \end{cases} \tag{8}$$

$$l_0(i_t, j_\theta) \leq f_0(i_t, j_\theta) \leq u_0(i_t, j_\theta), \quad (i_t, j_\theta) \in E_0 \tag{9}$$

$$max \sum_{H_1} v_t, \tag{10}$$

where by convention $i_t = 0$ for $t = -1$ and $i_t = n+1$ for $t = T+1$.

In stationary case the dynamic distances $d(1, i;\ t)$, $d(i, n; t)$ become static distances $d(1, i)$, $d(i, n)$.

Many notions from this section are presented and in papers [4], [7].

2.3 Maximum flows in bipartite static networks

In this section we consider that the static network $G = (N, A, l, u)$ is bipartite static network. A bipartite network has the set of nodes N partitioned into two subsets N_1 and N_2, so that for each arc $(i, j) \in A$, either $i \in N_1$ and $j \in N_2$ or $i \in N_2$ and $j \in N_1$. Let $n_1 = |N_1|$ and $n_2 = |N_2|$. Without any loss of generality, we assume that $n_1 \leq n_2$. We also assume that source node 1 belongs to N_2 (if the source node 1 belonged to N_1, then we could create a new source node $1' \in N_2$, and we could add an arc $(1', 1)$ with $l(1', 1) = 0$, $u(1', 1) = \infty$). A bipartite network is called unbalanced if $n_1 << n_2$ and balanced otherwise.

The observation of Gusfield, Martel, and Fernandez-Baca [6] that time bounds for several maximum flow algorithms automatically improves when the algorithms are applied without modification to unbalanced networks. A careful analysis of the running times of these algorithms reveals that the worst case bounds depend on the number of arcs in the longest node simple path in the network. We denote this length by L. For general network, $L \leq n - 1$ and for a bipartite network $L \leq 2n_1 + 1$. Hence for unbalanced bipartite network $L << n$. Column 3 of Table 1 summarizes these improvements for several network flow algorithms.

Table 1: Several maximum flows algorithms

Algorithm	*Running time, general network*	*Running time, bipartite network*	*Running time modified version*
Dinic	n^2m	n_1^2m	*does not apply*
Karazanov	n^3	n_1^2n	$n_1m + n_1^3$
FIFO preflow	n^3	n_1^2n	$n_1m + n_1^3$
Highest label	$n^2\sqrt{m}$	$n_1n\sqrt{m}$	n_1m
Excess scaling	$nm + n^2 \log \bar{u}$	$n_1m + n_1n \log \bar{u}$	$n_1m + n_1^2 \log \bar{u}$

Ahuja, Orlin, Stein, and Tarjan [2] obtained further running time improvements by modifying the algorithms. This modification applies only to preflow push algorithms. They call it the two arcs push rule. According to this rule, always push flow from a node in N_1 and push flow on two arcs at a time, in a step called a bipush, so that no excess accumulates at nodes in N_2. Column 4 of Table 1 summarizes the improvements obtained using this approach.

We recall that the FIFO preflow algorithm might perform several saturating pushes followed either by a nonsaturating push or relabeled operation. We refer to this sequence of operations as a node examination. The algorithm examines active nodes in the FIFO order. The algorithm maintains the list Q of active nodes as a queue. Consequently, the algorithm selects a node i from the front of Q, performs pushes from this node, and adds newly active nodes to the rear of Q. The algorithm examines node i until either it becomes inactive or it is relabeled. In the latter case, we add node i to the rear of the queue Q. The algorithm terminates when the queue Q of active nodes is empty. (see [1])

The modified version of FIFO preflow algorithm for maximum flow in bipartite network is called bipartite FIFO preflow algorithm. A bipush is a push over two consecutive admissible arcs. It moves excess from a node $i \in N_1$ to another node $k \in N_1$. This approach means that the algorithm moves the flow over the path $\tilde{D} = (i, j, k), j \in N_2$, and ensures that no node in N_2 ever has any excess. A push of α units from node i to node j decreases both $e(i)$ and $r_0(i, j)$ by α units and increases both $e(j)$ and $r_0(j, i)$ by α units. (see [2])

In the paper [2] is presented the following bipartite FIFO preflow (BFIFOP) algorithm:

Algorithm 1 The algorithm for a feasible flow in R_0.

```
ALGORITHM BFIFOP;
BEGIN
PREPROCESS;
while Q ≠ ∅ do
    BEGIN
      select the node i from the front of Q;
      BIPUSH/RELABEL(i);
    END
END.

PROCEDURE PREPROCESS;
BEGIN
  f =: 0; Q := ∅;
  push u(1, j) units of flow on each (1, j) ∈ A and add j to rear of Q;
  compute the exact distance labels d(i) from t to i;
END;

PROCEDURE BIPUSH/RELABEL(i);
BEGIN
if there is an admissible arc (i, j) then
    BEGIN
    select an admissible arc (i, j);
    if there is an admissible arc (j, k) then
        BEGIN
          select an admissible arc (j, k);
          push α := min {e(i), r(i, j), r(j, k)} units of flow along the path (i, j, k) and adds k
          to Q if k ∉ Q;
        END
    else
        d(j) := min{d(k) + 1|(j, k) ∈ A, r(j, k) > 0}
    END
else
    d(i) := min{d(j) + 1|(i, j) ∈ A, r(i, j) > 0};
END;
```

For more information see [2].

We notice the fact that have used the notations from this paper and specify that this algorithm runs on networks G with $l = 0$, a single source node 1, a single sink node n.

2.4 Feasible flows in static networks

We consider the flow problem satisfying the conditions (1a) and (1b). The condition (1a) is the flow conservation condition and the condition (1b) is the feasibility condition.

Let f be a flow of value v in the static network $G = (N, A, l, u)$. Let be $\bar{G}$ the static network we get by adding the return arc $(n, 1)$ to the network G. We extend the mappings l, u and f to $\bar{G}$ as follows:

$$\bar{l}(n,1)=0, \bar{u}(n,1)=\infty, \bar{f}(n,1)=v \tag{11}$$

Then $\bar{f}$ is a circulation on $\bar{G}$:

$$\bar{f}(i,N)-\bar{f}(N,i)=0, i\in N \tag{12a}$$

$$\bar{l}(i,j)\le \bar{f}(i,j)\le \bar{u}(i,j) \tag{12b}$$

We notice that the static network $\bar{G}=(\bar{N},\bar{A},\bar{l},\bar{u})$ has $\bar{N}=N$, $\bar{A}=A\cup\{(n,1)\}$, $\bar{l}(i,j)=l(i,j)$, $\bar{u}(i,j)=u(i,j)$, $\bar{f}(i,j)=f(i,j)$, $(i,j)\in A$ and usual, ∞ means a sufficiently large number, for example $\bar{u}(1,N)$.

Therefore the feasible flow problem we are looking for corresponding to an feasible circulation on network $\bar{G}$. However, note that the feasible flow problem might well be unsolvable, because there might not be any feasible circulations.

A cut in the static network $G=(N,A,l,u)$ is an arc set $[X,Y]=(X,Y)\cup(Y,X)$ with $X\subset N$, $Y=N-X$, $(X,Y)=\{(i,j)|(i,j)\in A, i\in X, j\in Y\}$, $(Y,X)=\{(j,i)|(j,i)\in A, j\in Y, i\in X\}$. The set (X,Y) denote the set of forward arcs in the cut and the set (Y,X) denote the set of backward arcs in the cut. We refer to a cut $[X,Y]$ as a $1-n$ cut if $1\in X$ and $n\in Y$. For the maximum flow problem the capacity of $1-n$ cut $[X,Y]$ is

$$c[X,Y]=u(X,Y)-l(Y,X) \tag{13}$$

A $1-n$ cut whose capacity is the minimum among all $1-n$ cuts is a minimum cut.

In [1] the next theorems are proved.

Theorem 1. *(Generalized Max-Flow Min-Cut Theorem). The maximum value of the flow from a source node* 1 *to a sink node* n *in a static network* $G=(N,A,l,u)$ *is equal to the capacity of a minimum* $1-n$ *cut.*

Theorem 2. *(Feasible Circulation Theorem). A necessary and sufficient condition for the existence of a feasible circulation on* $G=(N,A,l,u)$ *is that* $l(Y,X)\le u(X,Y)$ *for any cut* $[X,Y]$ *of* G.

Theorem 3. *(Feasible Flow Theorem). A necessary and sufficient condition for the existence of a feasible flow on* $G=(N,A,l,u)$ *is that* $l(Y,X)\le u(X,Y)$ *for all partitions* $N=X\cup Y$ *with* $1\notin Y$ *or* $n\notin X$.

Theorem 3 result from Theorem 2 if transform the flow f on $G=(N,A,l,u)$ into circulation $\bar{f}$ on $\bar{G}=(\bar{N},\bar{A},\bar{l},\bar{u})$.

3 The feasible flow problem in dynamic networks

3.1 The feasible flow in dynamic networks

We consider the flow problem satisfying the conditions (5) and (6). A flow g on the dynamic network $D=(N,A,h,e,q)$ satisfying the conditions (5) and (6) is equivalent with the flow f_0 on the multiple sources, multiple sinks static reduced expanded network $R_0=(V_0,E_0,l_0,u_0)$ satisfying the conditions (8) and (9).

Recall the construction of sets H_i and $H_{i,j}$: $H_i=\{t|t\in H, d(1,i;t)\le t\le T-d(i,n;t)\}$, $i\in N$, and $H_{i,j}=\{t|t\in H, d(1,i;t)\le t\le T-h(i,j;t)-d(j,n;\theta)\}$, $(i,j\in A)$. This sets make the connection between the dynamic network D and the static network R_0. Therefore we reformulate Theorem 1, Theorem 2, Theorem 3 for the static network R_0.

Let V_0 be $V_0=V_{01}\cup\ldots\cup V_{0i}\cup\ldots V_{0n}$ with $V_{0i}=\{i_t|i\in N, t\in H_i\}, i=1,\ldots,n$. We refer to a cut $[X_0,Y_0]$ with $X_0\subset V_0$, $Y_0=V_0-X_0$ as a $V_{01}-V_{0n}$ cut if $V_{01}\subset X_0$ and $V_{0n}\subset Y_0$.

Theorem 4. *(Generalized Dynamic Max-Flow Min-Cut Theorem). The maximum value of the flow from a source node V_{01} to a sink node V_{0n} in a static network $R_0 = (V_0, E_0, l_0, u_0)$ is equal to the capacity of a minimum $V_{01} - V_{0n}$ cut.*

Theorem 5. *(Feasible Dynamic Circulation Theorem). A necessary and sufficient condition for the existance of a feasible circulation on $R_0 = (V_0, E_0, l_0, u_0)$ is that $l_0(Y_0, X_0) \leq u_0(X_0, Y_0)$ for any cut $[X_0, Y_0]$ of R_0.*

Theorem 6. *(Feasible Dynamic Flow Theorem). A necessary and sufficient condition for the existence of a feasible flow on $R_0 = (V_0, E_0, l_0, u_0)$ is that $l_0(Y_0, X_0) \leq u_0(X_0, Y_0)$ for all partitions $V_0 = X_0 \cup Y_0$ with $V_{01} \notin Y_0$ or $V_{0n} \notin X_0$.*

If h, e, q are constant over time, then a dynamic network $D = (N, A, h, e, q)$ is said to be stationary. Ford and Fulkerson [5] have devised an algorithm that generates a maximum flow in the stationary dynamic network $D = (N, A, H, 0, q)$, the case when $e = 0$. The flow obtained with the algorithm Ford and Fulkerson is called temporally repeated flow. Let $c : A \to \mathbb{N}$ be the fucntion cost. For many details is urged to consult the book [5].

There are two inconveniences for the flow problem in the stationary dynamic network $D = (N, A, h, e, q)$. The first is that although in the planar network $G = (N, A, c, l, u)$ with $c = h, l = e, u = q$ exist a feasible flow, it is possible that in the static network $R_0 = (V_0, E_0, l_0, u_0)$ will not be any feasible flow. The second inconvenience consist in the fact that it is possible that the temporally repeated flow in the network R_0 of a feasible and minimum time flow in the network $G = (N, A, c, l, u)$, will not be any feasible in the network R_0.

3.2 Examples

The stationary dynamic network $D = (N, A, h, e, q)$ is presented in Figure 1 and the time horizon set to $T = 5$, therefore $H = \{0, 1, 2, 3, 4, 5\}$. The transit time $h(i, j)$, the lower bound $e(i, j)$ and the upper bounds $q(i, j)$ for all arcs $(i, j) \in A$ are indicated in this order near arcs.

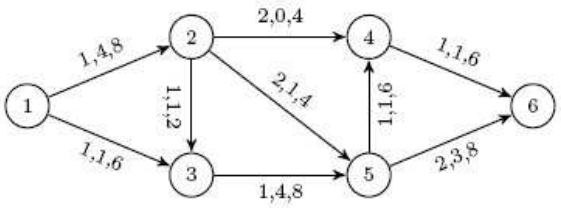


Figure 1: The stationary dynamic network D.

We obtain: $d(1,1) = 0, d(1,6) = 4$, $d(1,2) = 1, d(2,6) = 3$, $d(1,3) = 1, d(3,6) = 3$, $d(1,4) = 3, d(4,6) = 1$, $d(1,5) = 2, d(5,6) = 2$, $d(6,6) = 0$; $H_1 = \{0,1\}$, $H_2 = \{1,2\}$, $H_3 = \{1,2\}$, $H_4 = \{3,4\}$, $H_5 = \{2,3\}$, $H_6 = \{4,5\}$; $H_{1,2} = \{0,1\}$, $H_{1,3} = \{0,1\}$, $H_{2,3} = \{1\}$, $H_{2,4} = \{1,2\}$, $H_{2,5} = \{1\}$, $H_{3,5} = \{1,2\}$, $H_{4,6} = \{3,4\}$, $H_{5,4} = \{2,3\}$, $H_{5,6} = \{2\}$.

The static network $R_0 = (V_0, E_0, l_0, u_0)$ is presented in Figure 2. The lower bounds $l_0(i_t, j_\theta)$ and the upper bounds $u_0(i_t, j_\theta)$ for all arcs $(i_t, j_\theta) \in E_0$ are indicated in this order near arcs.

For all partitions $N = X \cup Y$ with $1 \notin Y$ or $n \notin X$ of the static network $G = (N, A, c = h, l = e, u = q)$ is verified the condition from Theorem 3. Therefore exist a feasible flow on G. Also, is verified Theorem 6 on the network $R_0 = (V_0, E_0, l_0, u_0)$.

Example 1. We replace in the network from Figure 1 $e(1,2) = 4$, $q(2,4) = 4$, $q(2,5) = 4$ with $e(1,2) = 7$, $q(2,4) = 2$, $q(2,5) = 2$. Now, we consider the partition $N = X \cup Y$ with $X = \{2\}$ and

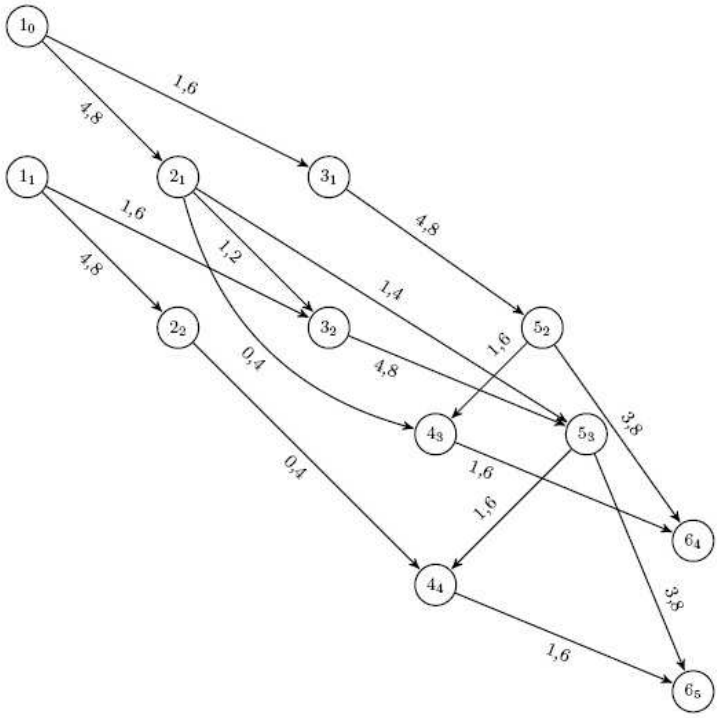


Figure 2: The static network R_0.

$Y = \{1, 3, 4, 5, 6\}$. We have $(X, Y) = \{(2, 3), (2, 4), (2, 5)\}$, $(Y, X) = \{(1, 2)\}$ and in the network $G = (N, A, c = h, l = e, u = q)$ we obtain $l(Y, X) = l(1, 2) = 7 > 6 = u(2, 3) + u(2, 4) + u(2, 5) = u(X, Y)$. Therefore not exist a feasible flow on G.

Example 2. We replace in the network from Figure 1 $q(2, 4) = 4$ with $q(2, 4) = 3$. For $q(2, 4) = 3$ the Theorem 3 is verified. Therefore exist a feasible flow on network $G = (N, A, c = h, l = e, u = q)$. If $q(2, 4) = 3$ then $u_0(2_1, 4_3) = u_0(2_2, 4_4) = 3$. We consider the partition $V_0 = X_0 \cup Y_0$ with $X_0 = \{2_2\}$ and $Y_0 = V_0 - X_0$. We have $(X_0, Y_0) = \{(2_2, 4_4)\}$, $(Y_0, X_0) = \{1_1, 2_2\}$ and $l_0(Y_0, X_0) = l_0(1_1, 2_2) = 4 > 3 = q(2, 4) = u_0(2_2, 4_4) = u_0(X_0, Y_0)$. From Theorem 6 we obtain that the flow problem in the static network R_0 (dynamic network D) is infeasible.

4 Maximum flows in bipartite dynamic networks with lower bounds

4.1 Maximum flows

In this Section the dynamic network $D = (N, A, h, e, q)$ is bipartite.

We construct the static reduced expanded network $R_0 = (V_0, E_0, l_0, u_0)$ and we notice the fact that the network R_0 is a bipartite network with $V_0 = W_1 \cup W_2$, $W_1 = \{i_t | i \in N_1, t \in H\}$, $W_2 = \{i_t | i \in N_2, t \in H\}$. Let w_1, w_2, ε_0 be $w_1 = |W_1|$, $w_2 = |W_2|$, $\varepsilon_0 = |E_0|$. If $n_1 << n_2$ then obvious that $w_1 << w_2$. In the static bipartite network R_0 we determine a maximum flow f_0 with a generalization of bipartite FIFO preflow algorithm.

We generalize the BFIFOP for a network $R_0 = (V_0, E_0, l_0, u_0)$ where $l_0 > 0$, there are multiple source nodes $1_t, t \in H_1$ and there are multiple sink nodes $n_t, t \in H_n$. Also, we present a pseudocode in detail.

We specify that maintain the arc list $E_0^+(i_t) = \{(i_t, j_\theta) \ | (i_t, j_\theta) \in E_0\}$. We can arrange the arcs in these lists arbitrarily, but the order, once decided, remains unchanged throughout the

algorithm. Each node i has a current arc, which is an arc in $E_0^+(i_t)$ and is the next candidate for admissibility testing. Initially, the current arc of node i_t is the first arc in $E_0^+(i_t)$. Whenever the algorithm attempts to find an admissible arc emanating from node i_t, it tests whether the node's current arc is admissible. If not, it designates the next arc in the arc list as the current arc. The algorithm repeats this process until it either finds admissible arc or it reaches the end of the arc list.

The generalised bipartite FIFO preflow (GBFIFOP) is presented in Algorithm 2.

We notice that any path in the residual network $\tilde{R}_0 = (V_0, \tilde{E}_0, r_0)$ can have at most $2w_1 + 1$ arcs. Therefore we set $d(1_t) := 2w_1 + 1$ in PROCEDURE PREPROCES.

The correctness of the GBFIFOP algorithm results from correctness of the algorithm for maximum flow in bipartite network [2].

Theorem 7. *The GBFIFOP algorithm which determines a maximum flow into the bipartite dynamic network $D = (N, A, h, q)$ has the complexity $O(n_1 m T^2 + n_1^3 T^3)$.*

Proof: In Section 3 we specify that the maximum flow problem for T time periods in the dynamic network $D = (N, A, h, e, q)$ is equivalent with the maximum flow problem in the static reduced expanded network $R_0 = (V_0, E_0, l_0, u_0)$. The networks D and R_0 are bipartite with $N = N_1 \cup N_2$, $V_0 = W_1 \cup W_2$. We have $n_1 = |N_1|$, $n_2 = |N_2|$, $m = |A|$, $w_1 = |W_1|$, $w_2 = |W_2|$,$\varepsilon_0 = |E_0|$. The bipartite FIFO preflow algorithm determines a maximum flow into the bipartite static network $G = (N_1 \cup N_2, A, l, u)$ in $O(n_1 m + n_1^3)$ how is specified in table from Section 4. We apply the generalizate bipartite FIFO preflow algorithm in the static reduced expanded bipartite network R_0. Hence the algorithm has the complexity $O(w_1 \varepsilon_0 + w_1^3)$. From Section 4 we have $w_1 = n_1 T$ and $\varepsilon_0 \leq mT$. As a result the algorithm has the complexity $O(n_1 m T^2 + n_1^3 T^3)$.

We specify that in the first phase the feasible flow f_0 is zero flow and in the second phase the feasible flow f_0 is the feasible flow f_0 determined in the first phase. □

Algorithm 2 The generalised bipartite FIFO preflow (GBFIFOP) algorithm

```
ALGORITHM GBFIFOP;
BEGIN
PREPROCESS;
while Q ≠ ∅ do
    BEGIN
        select the node i_t from the front of Q;
        BIPUSH/RELABEL(i_t);
    END;
END.
```

```
PROCEDURE PREPROCESS;
BEGIN
f_0 is a feasible flow in R_0; Q := ∅;
compute the exact distance labels d(i_t);
for t ∈ H_1 do
    BEGIN
        f_0(1_t, j_θ) := u_0(1_t, j_θ) and adds node j_θ to the rear of Q for all (1_t, j_θ) ∈ E_0
        d(1_t) := 2w_1 + 1;
    END;
END.
```

```
PROCEDURE BIPUSH/RELABEL(i_t);
BEGIN
select the first arc (i_t, j_θ) in E_0^+(i_t) with r_0(i_t, j_θ) > 0;
β := 1;
repeat
    if (i_t, j_θ) is admissible arc then
        BEGIN
        select the first arc (j_θ, k_τ) in E_0^+(j_θ) with r_0(j_θ, k_τ) > 0;
        if (j_θ, k_τ) is admissible arc then
            BEGIN
            push α := min {e(i_t), r_0(i_t, j_θ), r_0(j_θ, k_τ)} units of flow over the arcs
             (i_t, j_θ), (j_θ, k_τ);
            if k_τ ∉ Q then
                adds node k_τ to the rear of Q;
            END
        else if (j_θ, k_τ) is not the last arc in E_0^+(j_θ) with r_0(j_θ, k_τ) > 0 then
            select the next arc in E_0^+(j_θ)
        else
            d(j_θ) := min{d(k_τ) + 1|(j_θ, k_τ) ∈ E_0^+(j_θ), r_0(j_θ, k_τ) > 0};
        if e(i_t) > 0 then
            if (i_t, j_θ) is not the last arc in E_0^+(i_t) with r_0(i_t, j_θ) > 0 then
                select the next arc in E_0^+(j_θ)
            else
                BEGIN
                  d(i_t) := min{d(j_θ) + 1|(i_t, j_θ) ∈ E_0^+(j_θ), r_0(i_t, j_θ) > 0};
                  β := 0;
                END;
        END
until e(i_t) = 0 or β = 0
if e(i_t) > 0 then
    adds node i_t to the rear of Q;
END;
```

4.2 Example

We have $N_1 = \{2, 3, 7\}$ and $N_2 = \{1, 4, 5, 6\}$.

The support digraph of the bipartite dynamic network is presented in Figure 3 and time horizon being set $T = 5$, therefore $H = \{0, 1, 2, 3, 4, 5\}$. The transit times $h(i, j; t) = h(i, j), t \in H$, the lower bounds $e(i, j; t) = e(i, j)$ and the upper bounds (capacities) $q(i, j; t) = q(i, j), t \in H$ for all arcs are indicate in Table 2.

Applying the GBFIFOP algorithm in the first phase and the second phase we obtain the flows $f_0(i_t, j_\theta)$, $f_0^*(i_t, j_\theta)$ (the feasible flow, the maximum flow) which are indicated in Figure 4. We have $W_1 = \{2_1, 2_2, 2_3, 3_1, 3_2, 3_3, 7_3, 7_4, 7_5\}$ and $W_2 = \{1_0, 1_1, 1_2, 4_4, 5_2, 5_3, 5_4, 6_2, 6_3, 6_4\}$. A minimum $(1_0, 1_1, 1_2) - (7_3, 7_4, 7_5)$ cut in the static network R_0 is $[Y_0, \bar{Y_0}] = (Y_0, \bar{Y_0}) \cup (\bar{Y_0}, Y_0)$ with $Y_0 = \{1_0, 1_1, 1_2, 2_2, 2_3, 3_1, 3_2, 3_3\}$ and $\bar{Y_0} = \{2_1, 4_4, 5_2, 5_3, 5_4, 6_2, 6_3, 6_4, 7_3, 7_4, 7_5\}$. Hence $[Y_0, \bar{Y_0}] = \{(1_0, 2_1), (2_2, 5_3), (2_2, 6_4), (2_3, 5_4), (3_1, 6_2), (3_1, 4_4), (3_2, 6_3)\} \cup \{(5_2, 3_3)\}$. We have $\bar{w}_0 = f_0^*(Y_0, \bar{Y_0}) - f_0^*(\bar{Y_0}, Y_0) = 40 - 0 = 40 = u_0(Y_0, \bar{Y_0})$. Hence f_0^* is a maximum flow.

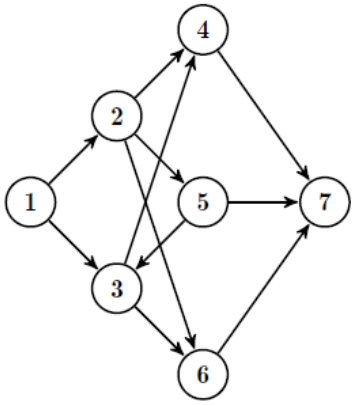


Figure 3: The support digraph of network $D = (N, A, h, e, q)$

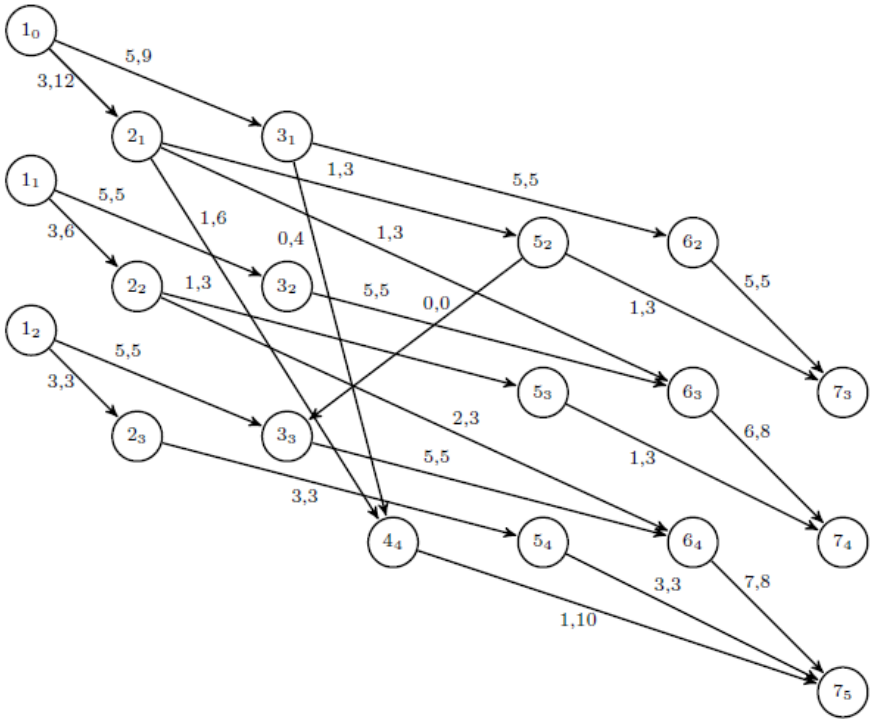


Figure 4: The network $R_0 = (V_0, E_0, f_0, f_0^*)$.

Table 2: The functions h, e, q

(i,j)	$(1,2)$	$(1,3)$	$(2,4)$	$(2,5)$	$(2,6)$	$(3,4)$	$(3,6)$	$(4,7)$	$(5,3)$	$(5,7)$	$(6,7)$
$h(i,j)$	1	1	3	1	2	3	1	1	1	1	1
$e(i,j)$	3	5	1	1	1	0	4	1	0	1	5
$q(i,j)$	12	10	8	3	3	4	5	12	3	4	10

Conclusions

In this paper we approached two flow problems: the feasible flow problem in dynamic networks and the maximum flow problem in bipartite dynamic networks with lower bounds. For the maximum flows problem in bipartite dynamic networks with lower bounds we developed an algorithm. These problem have not been treated so far. We demonstrate the fact that if the dynamic network $D = (N, A, h, e, q)$ is bipartite, then the static reduced expanded network $R_0 = (V_0, E_0, l_0, u_0)$ is bipartite. For solving, we have rephrased the maximum flows problem in bipartite dynamic networks with lower bound into a problem in bipartite static network. Then, we have extended the bipartite FIFO preflow algorithm of Ahuja et al. [2] for the static reduced expanded network with multiple source and multiple sinks $R_0 = (V_0, E_0, l_0, u_0)$. Also we have presented the complexity for the generalization bipartite FIFO preflow algorithm. For each of the two problems we have presented one example.

Flow problems in bipartite dynamic networks like: the parametric maximum flow problem, the minimum cost flow problem, the generalization of the highest label preflow push algorithm or the generalization of the excess scaling algorithm are still open for research.

Bibliography

[1] R. Ahuja, T. Magnanti and J. Orlin (1993), *Network Flows. Theory, algorithms and applications*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1993.

[2] R. Ahuja, J. Orlin, C. Stein and R. Tarjan (1994), Improved algorithms for bipartite network flows, *SIAM Journal of Computing*, 23:906-933.

[3] X. Cai, D. Sha and C. Wong (2007), *Time-varying Nework Optimization*, Springer, 2007.

[4] E. Ciurea (2002), Second best temporally repeated flow, *Korean Journal of Computational and Applied Mathematics*, 9(1):77-86.

[5] L. Ford, D. Fulkerson, *Flow in Networks.*, Princeton University Press, Princenton, New Jersey, 1962.

[6] D. Gusfield, C. Martel, and D. Fernandez-Baca (1987), Fast algorithms for bipartite network flow, *SIAM Journal of Computing*, 16:237-251.

[7] C. Schiopu, E. Ciurea (2016), The maximum flows in planar dynamic dynamic networks, *International Journal of Computers Communications & Control*, 11(2):282-291.

[8] C. Schiopu, E. Ciurea, The maximum flows in bipartite dynamic networks with lower bounds. The static approach, *Computers Communications and Control (ICCCC), 2016 6th International Conference on*, IEEE Xplore, e-ISSN 978-1-5090-1735-5, doi: 10.1109/ICCCC.2016.7496731, 10-15.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(1):116-130, February 2017.

Feature Analysis to Human Activity Recognition

J. Suto, S. Oniga, P. Pop Sitar

Jozsef Suto*
Department of Information Systems and Networks
University of Debrecen, Debrecen, Hungary
*Corresponding author: suto.jozsef@inf.unideb.hu

Stefan Oniga
1. Department of Information Systems and Networks
University of Debrecen, Debrecen, Hungary
oniga.istvan@inf.unideb.hu
2. Department of Electronic and Computer Engineering
Technical University of Cluj-Napoca,
North University Center at Baia Mare, Baia Mare, Romania
stefan.oniga@cunbm.utcluj.ro

Petrica Pop Sitar
Department of Mathematics and Informatics
Technical University of Cluj-Napoca,
North University Center at Baia Mare, Baia Mare, Romania
petrica.pop@cunbm.utcluj.ro

Abstract: Human activity recognition (HAR) is one of those research areas whose importance and popularity have notably increased in recent years. HAR can be seen as a general machine learning problem which requires feature extraction and feature selection. In previous articles different features were extracted from time, frequency and wavelet domains for HAR but it is not clear that, how to determine the best feature combination which maximizes the performance of a machine learning algorithm. The aim of this paper is to present the most relevant feature extraction methods in HAR and to compare them with widely-used filter and wrapper feature selection algorithms. This work is an extended version of [1][a] where we tested the efficiency of filter and wrapper feature selection algorithms in combination with artificial neural networks. In this paper the efficiency of selected features has been investigated on more machine learning algorithms (feed-forward artificial neural network, k-nearest neighbor and decision tree) where an independent database was the data source. The result demonstrates that machine learning in combination with feature selection can overcome other classification approaches.
Keywords: human activity recognition, feature extraction, feature selection, machine learning.

[a]Reprinted and extended, with permission based on License Number 3958150787732 [2016] IEEE, from "Computers Communications and Control (ICCCC), 2016 6th International Conference on"

1 Introduction

This paper is an extended version of our previous work where only artificial neural networks have been used to the efficiency investigation of feature selection in the human activity recognition (HAR) problem [1]. However, in this paper by the involvement of more machine learning techniques we can examine the relation between classifier and feature selection methods.

The appearance of data mining was a milestone of modern biomedical applications. HAR is an interesting and rapidly expanding part of this area. In this type of problem, we want to determine

the activity of people from the information which comes from one or more accelerometer-based data collector devices. Generally, sensors are placed to different parts of the body and provide information about the functional ability and lifestyle of an observed person. Although many articles have been presented in this topic, some questions are unanswered yet. *Which feature selection algorithm generate the best feature combination that maximize the recognition rate of a machine learning algorithm? Does a general feature combination which similarly efficient independently of the person exist? Can a machine learning algorithm in combination with feature selection overcome other classification approaches?*. The aim of this study is to give reliable answers for the above questions.

The rapidly growing rate of elderly population in our society has a huge impact to health care systems. Obviously, everyone wants to stay in a familiar environment where they feel comfortable during the observation. This new challenge motivated the development of different kind of home care services, assistive systems and wearable sensor networks in order to increase the autonomy and life quality of an observed person [2,3]. Today, the miniaturized sensor technology (MEMS) makes it possible for a person to wear data acquisition devices on predetermined body segments. In the past decade, more research groups developed different kinds of devices for HAR purposes which ensure continuous observation in both indoor and outdoor environments [4]- [7]. Such a wearable sensor network was used to the construction of the WARD 1.0 database. It is a benchmark database to the HAR research which was collected at University of California, Berkeley [7]. This public data set gives an opportunity for qualitative comparison of existing HAR algorithms. The database contains information about 13 different activities which were acquired from 20 people aged between 19 and 75 years. During the data acquisition, 5 sensor nodes were placed at multiple body locations of each person: left and right forearm, waist, left and right ankle. In this study we used the data of only one sensor which has been placed on the right ankle because Ertugrul et al. and Oniga et al. demonstrated that one sensor is enough for appropriate HAR recognition while Preece et al. claimed that the ankle is an optimal placement for single sensor [8, 28, 31].

Beyond data acquisition, efficient algorithms are also necessary to interpret the collected data. Previous works have shown that, machine learning algorithms are efficient for human movement classification. Khan et al., Oniga et al. and Yang et al. used artificial neural network (ANN) to their HAR research and archived 97.9%, 95% and 99% recognition rates (with single sensor), respectively [9]- [12]. Duarte et al. and Preece et al. used k-nearest neighbor (kNN) method and measured 97.8% and 95% recognition rates [8, 13]. Finally, Gao et al. and Maurer et al. reached similarly good results with decision tree 96.4%, 92.8% [4, 5]. It was the main motivation to use those three machine learning methods in this study.

2 Raw data pre-processing and classification

2.1 Feature extraction

Many machine learning application require feature extraction and feature selection; see for example [23, 25]. Feature extraction can be seen as a data pre-processing step where different kinds of features will be extracted from the raw data. In the first step the raw data (time series) will be split into short intervals - windows. Usually, a window covers one or two seconds long time interval and its size depends on the sampling frequency. For instance, Preece at al. used 2 seconds long window with 50% overlapping while Gao et al. and Karantonis et al. used 1 second long window without overlapping [4, 8, 14]. In our case, a window contains 32 samples and there is a 50% overlapping between windows in the training phase and no overlapping in the test phase. This size covers 1.6 seconds time interval because the sampling frequency of the

Table 1: Most common feature extraction methods in HAR

Category	Feature	Abbreviation	References
Time	Mean	M	[4, 8]
	Variance	V	[4, 8]
	Mean absolute deviation	MAD	[5, 15]
	Root mean square	RMS	[5, 15]
	Zero Crossing Rate	ZCR	[4, 5]
	Interquartile Range	IQR	[5, 15]
	75'th percentile	PE	[5, 8]
	Kurtosis	KS	[15, 16]
	Signal magnitude area	SMA	[4, 15]
	Min-max	MM	[17, 18]
Frequency	Spectral energy	SE	[4, 8]
	Spectral entropy	E	[8, 15]
	Spectral centroid	SC	[4, 19]
	Principal frequency	PF	[8, 20]
Other	Correlation between axis	CORR	[4, 17]
	Autoregressive coefficients	AR1, AR2	[9, 19]
	Tilt Angle	TA	[9, 18]

WARD database is approximately 20 Hz.

After the windowing step, features will be extracted from each window. In previous activity classification studies, the researchers used different kinds of features from different domains. Most of them come from the time a frequency domains but exist some other types of features. Table 1 summarizes the most common features from the literature and their references. Some features are redundant and those were omitted from the table. For instance, standard deviation because variance is in the table. Moreover, wavelet features have been similarly omitted because time-frequency features are more efficient than wavelets, see [8]. At the rest of this subsection we give a short description to the methods in Table 1. In the equations T indicates the window size, F is the number of frequency components, i refers to the accelerometer dimensions (x, y, z), $a_i(t)$ is an element of time series and $A_i(f)$ is a frequency component.

Mean, variance, mean absolute deviation and root mean square are statistical indicators which give information about sample distribution.

ZCR measures sing changes along the time series. In the formula, the $I\{x\}$ function returns 1 if its argument is true and 0 otherwise.

$$ZCR_i = \frac{1}{T-1} \sum_{t=1}^{T-1} I\{a_i(t)a_i(t+1) < 0\}. \tag{1}$$

Quartiles ($Q1$, $Q2$ and $Q3$) divide an ordered time series into quarters. IQR is the difference between the upper and lower quartiles: $IQR = Q3$ - $Q1$. It measures the spread of a data set over a range. Percentiles are similar as quartiles, except that those divide a data set into arbitrary parts (given in percentage). Therefore, the 25^{th} percentile is equal to the lower quartile ($Q1$) and the 75^{th} percentile is equal to the upper quartile ($Q3$). In this study we used only the 75^{th} percentile.

Kurtosis measures the peakedness of probability distribution of collected data.

$$KS_i = \frac{\frac{1}{T}\sum_{t=1}^{T}(a_i(t) - M_i)^4}{(\frac{1}{T}\sum_{t=1}^{T}(a_i(t) - M_i)^2)^2}. \tag{2}$$

SMA equals to the normalized sum of accelerometer components.

$$SMA = \frac{1}{T}\sum_{t=1}^{T} |a_x(t)| + |a_y(t)| + |a_z(t)|. \tag{3}$$

Min-max is the difference between the maximum and the minimum values of time series.

Generally, signal energy is the area under the squared signal. In this case, SE measures the sum of the squared frequency components. (Since, the spectrum is symmetric, in the following three formulas F can be replaced by F/2).

$$SE_i = \sum_{f=1}^{F} |A_i(f)|^2. \tag{4}$$

The entropy is a measure for uncertainty. The following formula is the entropy of the normalized power spectrum density.

$$E_i = -\sum_{f=1}^{F} \frac{|A_i(f)|^2}{\sum_{j=1}^{F}|A_i(j)|^2} log_2(\frac{|A_i(f)|^2}{\sum_{j=1}^{F}|A_i(j)|^2}). \tag{5}$$

Spectral centroid measures the average frequency, weighted by the amplitude of spectrum.

$$SC_i = \frac{\sum_{f=1}^{F} f|A_i(f)|}{\sum_{j=1}^{F}|A_i(j)|}. \tag{6}$$

Principal frequency refers to the most significant frequency component which has the highest amplitude (DC component has been omitted).

$$PF_i = max_f|A_i(f)| \qquad f \neq 0. \tag{7}$$

Autoregressive model is another representation of signal. In the model, an element in the time series can be estimated by a linear weighted sum of previous elements where the weights are the coefficients. In (8), (9) and (10) p indicates model order, ϕ_k is the AR coefficient, $\varepsilon(t)$ is the noise and τ_k refers to the autocorrelation. In Table 1, AR1 and AR2 abbreviations refer to the first and second autoregressive coefficients, respectively.

$$AR(p) \longrightarrow a_i(t) = \sum_{k=1}^{p} \phi_k a_i(t-k) + \varepsilon(t). \tag{8}$$

$$\phi = \begin{pmatrix} 1 & r_1 & r_2 & \cdots & r_{p-1} \\ r_1 & 1 & r_1 & \cdots & r_{p-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{p-1} & r_{p-2} & r_{p-3} & \cdots & 1 \end{pmatrix}^{-1} * \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_p \end{pmatrix}. \tag{9}$$

$$r_k = \frac{\frac{1}{T}\sum_{t=1}^{T-k}(a_i(t) - M_i)(a_i(t-k) - M_i)}{V_i}. \tag{10}$$

Correlation measures linear relationship between two axes. Actually, $CORR_{i,j}$ is the cosine of angle between normalized vectors. Therefore, -1 $\leq CORR_{i,j} \leq$ 1. The sing indicates the

direction of correlation. If $CORR_{i,j}$ is near to $\pm$ 1 implies that there is strong linear correlation between vectors.

$$CORR_{i,j} = \frac{\sum_{t=1}^{T}(a_i(t) - M_i)(a_j(t) - M_j)}{\sqrt{\sum_{t=1}^{T}(a_i(t) - M_i)^2(a_j(t) - M_j)^2}}. \tag{11}$$

Finally, tilt angle indicates the relative tilt of the sensor. It is the angle between z-axis and gravitational vector. In this survey, the absolute values of tilt angles have been summarized.

$$TA = arccos(\frac{a_z(t)}{\sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2}}). \tag{12}$$

2.2 Feature selection

Feature selection, also called feature reduction, is the process of choosing a subset of original features according to a well-defined evaluation criterion. It is a frequently used dimensionality reduction technique which removes irrelevant and redundant features. This approach has more useful effects for real applications because it accelerates algorithms, improves the performance and simplifies the model. In contrast to other dimensionality reduction techniques like linear discriminant analysis (LDA) or principal component analysis (PCA) which are based on projection, feature selection does not alter the original representation of feature sets, see [21].

According to the training data which are may be tagged, untagged or partially tagged, there have been developed three categories of algorithms: supervised, unsupervised and semi-supervised feature selection where a tag refers to a given class. In addition, depending on the feature evaluation process, feature selection algorithms belong to three different groups: filter, wrapper and embedded. Filter algorithms calculate scores for all features and select features according to the score. Wrapper methods require a predefined classification technique and use its performance as ranking criteria of feature subsets. In embedded models, feature selection takes place at the training process. In this article we confine on the supervised category and particularly we are interested in filter and wrapper methods.

In real applications filter methods are frequently used for feature selection because they have some significant advantages. Firstly, those methods are independently applicable with any types of machine learning techniques. Secondly, filter methods are faster than wrappers. However, wrapper methods are more efficient than feature ranking algorithms in some cases because they take into consideration the classifier hypothesis. This also means that, wrapper techniques can handle feature dependencies. So, both types of feature selection methods have advantages and disadvantages [22]. Essentially, independently of categories and groups, the goal is to find the most appropriate hyperplane of the n-dimensional feature space in all cases where the sample distributions can be separable. For example, Fig. 1 illustrates a proper separation between six 2 dimensional sample distributions (as the colour indicates) where each distribution (samples) comes from different classes. Obviously it is an ideal case and the classification will be 100%. More information about feature selection and their application opportunities in bioinformatics can be found in [24, 25].

Today's, a reach literature exists concerning the feature (or variable) selection. During the last decades, there have been developed a large amount of filter algorithms. The proposed algorithms are based on different kinds of approaches such as statistical, information theory, rough set, etc. [23]. However, to the best of our knowledge, the number of articles which utilizes feature selection algorithm on HAR is very small. We found only two papers where the minimum redundancy maximum relevance and the correlation feature selection techniques have been utilised, see [5, 26]. Therefore, we collected the most relevant filter techniques and examined them on the HAR problem. Fortunately, Zhao et al. proposed a generally applicable repository to feature

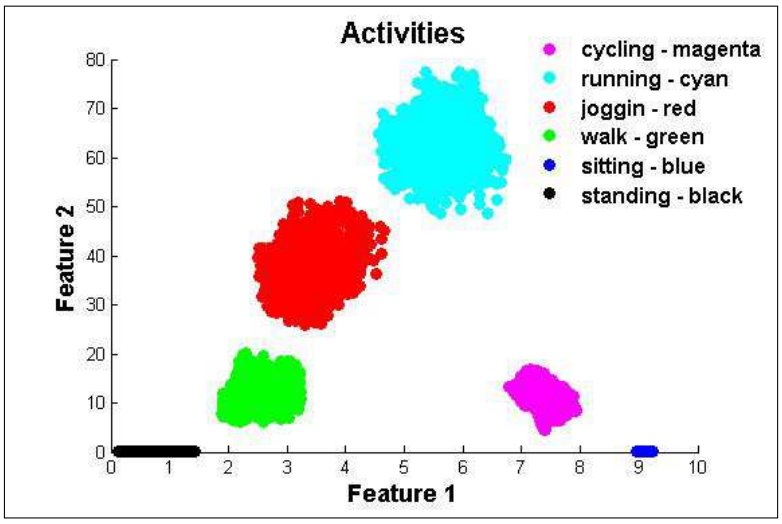


Figure 1: An ideal separation between sample distributions.

selection research which contains 13 supervised and unsupervised filter methods [27]. Moreover, the repository suggests references and implementations (in Matlab) to the algorithms which have been applied in this investigation. In addition, we similarly utilised the naive Bayesian wrapper method from [1].

So, in this work the following 9 feature selection methods have been tested: (1) Correlation feature selection (CFS); (2) Chi square (CHI); (3) Fast correlation-based filter (FCBF); (4) Fisher score (FIS); (5)Information gain (IG); (6) Kruskal-Wallis (KW); (7) Minimum redundancy maximum relevance (MRMR); (8) T-test; (9) Naive Bayesian (Bayes).

2.3 Classification

A classification system has to use an algorithm that is capable to learn and tolerate errors which come from noise. Previous studies have shown that artificial neural network (ANN), k-nearest neighbor (kNN) and decision tree (DT) are well applicable for HAR, see for example [5, 11, 13]. Therefore, to the efficiency measurement of selected features those classifiers have been applied.

As the name indicates, in feed-forward networks the input data go through all layers where the incoming data will be modified according to the weights and biases of a layer. The ANN theory gives some advices to architecture construction but general rules do not exist.

Finding the right architecture of an ANN for a specific purpose is a time-consuming task because it requires lots of simulations. Our ANN architecture design is based on the work of Oniga et al. where the authors demonstrated that a simple feed-forward ANN with only one hidden and one output layers is enough to HAR [28]. Therefore, in this study, some feed-forward ANNs were generated with the same architecture as in their work. The number of neurons on the input, hidden and output layers ere equal to I, $2I$ and C respectively where I indicates the number of inputs and C is the number of activities (classes). The activation functions on the hidden and output layers were sigmoid and linear and the training algorithm was the Levemberg-Marquardt.

The kNN classifier generation is based on the work of Duarte et al. [13]. In this research, the authors reached approximately 98% recognition rate by a 1NN classifier with the Euclidean distance metrics. Finally, we used the default decision tree in Matlab without any modification.

3 Results

At the beginning of the investigation, 7 volunteers were selected from the WARD database with different and similar ages and their raw data were the input of the feature extraction step in the training and test phases: (Subject 1) 19 years old; (Subject 2) 75 years old; (Subject 3) 27 years old; (Subject 4) 29 years old; (Subject 5) 20 years old; (Subject 6) 29 years old; (Subject7) 34 years old.

In Table 1 two features are one dimensional while others are multi-dimensional according to the axis of the accelerometer sensor. Therefore, multi-dimensional features were separated into one dimensional. This step generated 50 one dimensional features from the 17 feature extraction methods. After feature extraction, we got a feature matrix (*windows x features*) where rows contain the features of the windows. This matrix was the input of each feature selection method and the selected feature vectors were the input of the classifiers. In this study we applied the first 5 and 6 selected features because Gao et al. and Oniga et al. demonstrated that approximately 5 or 6 features are enough to an effective classification [4, 28]. Finally, according to the selected features the performance of the classifiers was measured. Table 2-10 contain the selected features by the methods in selection order while the measured recognition rates (in percentage) can be seen in Table 11-17.

Table 2: Selected features by CFS

Subject	Features
Subject 1	M_x, M_z, MAD_x, RMS_x, RMS_y, RMS_z
Subject 2	M_x, M_y, MAD_z, RMS_x, RMS_y, RMS_z
Subject 3	M_x, M_y, V_z, MAD_y, RMS_x, RMS_z
Subject 4	M_x, M_y, M_z, RMS_y, RMS_z, IQR_x
Subject 5	M_x, M_y, M_z, V_x, V_z, RMS_x
Subject 6	M_x, M_y, M_z, MAD_y, RMS_x, RMS_z
Subject 8	M_x, M_y, M_z, MAD_y, RMS_x, IQR_y

Table 3: Selected features by CHI

Subject	Features
Subject 1	PE_y, PE_z, MM_z, MM_y, PE_x, SMA
Subject 2	PE_y, MM_x, PE_x, MM_y, MM_z, RMS_x
Subject 3	PE_y, MM_x, MM_z, MM_y, PE_x, PE_z
Subject 4	PE_y, MM_x, MM_y, PE_z, MM_z, PE_x
Subject 5	MM_y, PE_y, MM_x, MM_z, PE_z, PE_x
Subject 6	MM_y, PE_y, MM_z, MM_x, PE_x, PE_z
Subject 8	PE_y, MM_x, MM_y, PE_x, MM_z, PE_z

Table 4: Selected features by FCBF

Subject	Features
Subject 1	RMS_x, PE_y, RMS_y, M_z, RMS_z, SC_x
Subject 2	SE_x, PE_x, SE_z, RMS_y, M_y, SMA
Subject 3	PE_y, SMA, RMS_x, M_y, RMS_z, MM_x
Subject 4	$2PE_y$, SE_y, SE_x, M_z, SC_x, MAD_x
Subject 5	RMS_x, PE_y, M_x, TA, E_y, SC_x
Subject 6	SMA, RMS_z, PE_x, PE_y, M_x, MAD_z
Subject 8	PE_y, RMS_x, M_z, MAD_y, SC_x, IQR_x

Table 5: Selected features by FIS

Subject	Features
Subject 1	RMS_x, MAD_x, M_x, PE_x, SC_z, E_y
Subject 2	SE_z, RMS_x, M_x, TA, PE_x, M_z
Subject 3	RMS_x, PE_x, M_x, MAD_x, E_y, RMS_z
Subject 4	MM_x, MAD_x, MAD_y, MM_y, SC_x, SC_z
Subject 5	E_y, SC_z, RMS_x, SC_y, E_z, M_x
Subject 6	RMS_x, PE_x, M_x, SE_x, SC_y, MM_z
Subject 8	MM_x, MAD_y, RMS_x, IQR_y, V_y, SE_x

Table 6: Selected features by IG

Subject	Features
Subject 1	ZCR_z, ZCR_x, MM_x, E_x, E_z, $AR2_x$
Subject 2	$AR2_z$, KS_z, PF_z, $AR1_y$, ZCR_y, $CORR_y$
Subject 3	ZCR_z, IQR_z, E_x, $AR1_x$, $AR2_x$, $AR2_y$
Subject 4	ZCR_y, ZCR_z, E_x, E_y, $AR1_x$, $AR2_x$
Subject 5	KS_z, ZCR_x, ZCR_z, ZCR_y, E_x, $AR1_x$
Subject 6	$AR1_z$, $AR2_y$, PF_z, $AR2_z$, ZCR_y, ZCR_z
Subject 8	$AR2_y$, ZCR_z, E_x, E_y, $AR1_x$, $AR2_x$

Table 7: Selected features by KW

Subject	Features
Subject 1	E_y, SC_y, SC_z, SC_x, IQR_x, MAD_x
Subject 2	SC_z, SC_y, E_y, IQR_y, KS_z, MAD_y
Subject 3	SC_y, SC_z, IQR_x, IQR_y, MM_y, MAD_x
Subject 4	SC_x, SC_y, SC_z, IQR_x, IQR_z, MAD_x
Subject 5	E_y, E_z, SC_y, SC_z, SC_x, IQR_y
Subject 6	SC_y, SC_z, E_y, E_z, MAD_x, PE_y
Subject 8	SC_y, SC_z, SC_x, M_y, IQR_y, IQR_z

Table 8: Selected features by MRMR

Subject	Features
Subject 1	PE_y, TA, $AR2_y$, $AR2_z$, $AR2_x$, $AR1_z$
Subject 2	PE_y, TA, SC_z, SC_y, E_y, E_z
Subject 3	MM_x, TA, $AR2_z$, $AR1_z$, $AR2_x$, $AR2_y$
Subject 4	PE_x, TA, $AR2_y$, $AR2_z$, $AR2_x$, $AR1_z$
Subject 5	MM_y, TA, $AR2_z$, $AR2_y$, $AR2_x$, $AR1_z$
Subject 6	MM_y, TA, $AR2_z$, $AR2_y$, SC_z, E_z
Subject 8	PE_y, TA, $AR2_z$, $AR2_y$, $AR2_x$, $AR1_z$

Table 9: Selected features by T-test

Subject	Features
Subject 1	MM_x, E_x, E_z, $AR2_x$, SE_z, RMS_z
Subject 2	ZCR_z, RMS_z, TA, SE_y, SE_z, RMS_y
Subject 3	IQR_z, E_x, $AR1_x$, $AR2_x$, $AR2_y$, RMS_z
Subject 4	E_x, E_y, $AR1_x$, $AR2_x$, ZCR_y, ZCR_z
Subject 5	E_x, $AR1_x$, RMS_y, RMS_z, SE_y, SE_z
Subject 6	TA, SE_z, RMS_z, ZCR_y, $AR2_x$, RMS_y
Subject 8	E_y, E_z, $AR1_x$, $AR2_x$, ZCR_z, ZCR_y

Table 10: Selected features by Bayesian

Subject	Features
Subject 1	PE_z, SMA, $CORR_y$, KS_x, $CORR_x$, MM_y
Subject 2	RMS_y, M_y, MM_z, KS_x, PE_z, PE_x
Subject 3	PE_y, MM_x, SMA, PF_y, IQR_x, E_y
Subject 4	PE_y, MM_x, PE_z, MM_y, M_x, PF_y
Subject 5	PE_y, TA, $CORR_y$, MM_x, PF_y, MM_z
Subject 6	PE_x, PE_y, PF_y, PE_z, MM_x, MM_y
Subject 8	PE_y, PE_x, MM_x, IQR_x, SMA, PF_x

Table 11: Recognition rates for subject 1

	ANN		1NN		DT	
	n=5	n=6	n=5	n=6	n=5	n=6
CFS	93.1	94.7	98.4	99.7	94.2	95.9
CHI	84.9	92.6	100	100	98.6	98.2
FCBF	86.2	90.7	99.7	99.9	90.9	96.3
FIS	87.2	93.4	91.9	94.6	90.1	91.8
IG	18.0	24.7	12.5	12.5	18.0	24.7
KW	83.9	89.0	97.4	99.6	87.2	89.7
MRMR	87.9	90.8	95.9	98.7	93.5	95.0
T-test	12.7	51.7	43.5	77.4	44.0	73.9
Bayesian	92.2	93.0	96.3	98.3	80.7	94.3

Table 12: Recognition rates for subject 2

	ANN		1NN		DT	
	n=5	n=6	n=5	n=6	n=5	n=6
CFS	96.5	97.8	100	100	85.1	86.6
CHI	96.6	97.9	99.9	100	97.6	85.8
FCBF	82.5	85.5	99.7	99.9	97.1	97.8
FIS	71.3	84.7	81.7	83.4	94.0	94.6
IG	63.2	69.0	95.9	88.0	73.6	74.1
KW	89.9	92.2	99.8	99.9	93.2	94.1
MRMR	94.4	94.3	100	100	95.1	96.5
T-test	43.1	72.8	98.5	99.8	93.6	96.0
Bayesian	96.4	98.6	98.7	99.4	96.9	97.5

Table 13: Recognition rates for subject 3

	ANN		1NN		DT	
	n=5	n=6	n=5	n=6	n=5	n=6
CFS	95.0	96.0	88.5	99.8	79.0	87.1
CHI	93.7	94.6	100	100	99.7	99.8
FCBF	96.2	96.1	99.0	99.9	88.8	90.0
FIS	94.0	89.4	76.2	93.9	92.6	94.0
IG	62.3	65.4	62.2	62.9	62.2	63.1
KW	88.1	89.7	99.4	99.4	97.1	97.4
MRMR	82.6	83.9	79.5	79.7	93.0	93.0
T-test	62.3	65.5	62.2	66.4	62.2	70.6
Bayesian	95.3	96.3	99.9	99.9	97.8	98.9

Table 14: Recognition rates for subject 4

	ANN		1NN		DT	
	n=5	n=6	n=5	n=6	n=5	n=6
CFS	92.1	94.6	99.9	100	94.2	97.2
CHI	92.2	93.4	100	100	98.9	99.5
FCBF	62.6	84.5	84.2	87.8	88.9	89.5
FIS	75.0	87.4	71.0	77.1	72.1	76.7
IG	57.9	66.7	51.5	62.5	57.9	66.7
KW	82.3	83.0	68.1	68.9	65.7	66.2
MRMR	82.4	86.9	85.2	89.7	88.3	88.6
T-test	62.7	66.8	62.5	66.5	62.8	66.8
Bayesian	93.7	95.0	100	100	99.4	99.6

Table 15: Recognition rates for subject 5

	ANN		1NN		DT	
	n=5	n=6	n=5	n=6	n=5	n=6
CFS	90.7	95.5	99.6	99.5	96.9	96.7
CHI	92.3	94.5	100	100	99.5	99.6
FCBF	93.1	94.2	70.0	74.1	84.0	62.1
FIS	85.4	92.6	93.5	97.5	92.6	94.2
IG	38.4	47.9	26.2	59.3	38.4	58.3
KW	82.8	87.8	88.2	95.6	82.8	88.7
MRMR	85.8	88.2	62.7	74.4	44.1	49.6
T-test	59.5	61.1	82.0	92.3	76.3	87.7
Bayesian	94.3	96.5	70.6	82.9	46.1	50.3

Table 16: Recognition rates for subject 6

	ANN		1NN		DT	
	n=5	n=6	n=5	n=6	n=5	n=6
CFS	93.0	97.6	100	100	97.7	97.8
CHI	93.8	96.9	100	100	99.5	99.6
FCBF	92.2	95.3	100	100	97.3	97.8
FIS	58.7	62.3	83.1	91.8	88.6	91.5
IG	46.9	67.3	58.1	70.4	53.8	56.6
KW	81.4	91.9	85.5	95.7	81.9	90.6
MRMR	82.6	85.3	87.4	90.7	84.8	88.6
T-test	66.0	68.1	93.6	99.5	83.6	90.4
Bayesian	94.5	96.5	100	100	99.7	97.7

Table 17: Recognition rates for subject 8

	ANN		1NN		DT	
	n=5	n=6	n=5	n=6	n=5	n=6
CFS	92.5	93.8	99.2	99.7	91.4	91.9
CHI	92.4	89.8	100	100	99.7	99.5
FCBF	91.8	95.7	94.7	98.8	93.7	96.7
FIS	88.0	70.5	98.3	99.9	98.2	98.4
IG	47.9	57.3	42.5	48.6	47.9	52.9
KW	87.8	91.0	83.3	87.7	80.0	81.6
MRMR	84.4	87.0	96.7	99.0	94.6	95.0
T-test	47.9	57.9	42.5	46.9	47.9	57.9
Bayesian	95.8	96.8	100	100	99.5	99.6

4 Discussion

Table 18: Average recognition rates

	ANN		1NN		DT	
	n=5	n=6	n=5	n=6	n=5	n=6
CFS	93.3	95.7	97.9	99.8	91.2	93.3
CHI	92.3	94.2	100	100	99.1	97.4
FCBF	86.3	91.7	92.5	94.3	91.5	90.0
FIS	79.2	82.9	85.1	91.2	89.7	91.6
IG	47.8	56.9	49.8	57.7	50.3	56.6
KW	85.2	89.2	88.8	92.4	84.0	86.9
MRMR	85.7	88.1	86.8	90.3	84.8	86.6
T-test	50.6	63.4	69.3	78.4	67.2	77.6
Bayesian	94.6	96.1	95.1	97.2	88.6	91.1

The results in Table 2-17 provide large amount of information about features, feature selection and classification algorithms. Moreover, Table 18 shows the average recognition accuracies where the best rates are highlighted by red. According to the results, we summarized our observations into the following points:

- A generally applicable feature set does not exist because the selected features for the subjects are just partially overlapping.
- More features do not guarantee performance improvement, see CHI and IG in Table 12.
- The performance of some feature selection methods is person dependent. For instance, in the case of subject 2 the machine learning algorithms with MRMR reached higher recognition rate than in the case of subject 3.
- The performance of machine learning algorithms is feature selection and person dependent. For example, in the case of subject 5, the MRMR algorithm reached higher recognition rates with ANN while in other cases, the 1NN or DT were better. In addition, the 1NN produced better results than ANN and DT with CHI in all cases.
- As Table 18 shows, a relation exists between machine learning and feature selection algorithms. The 1NN and DT reached the highest recognition rate with the CHI while the ANN in combination with the Bayesian was the most efficient.
- Already 5 features are enough for good classification, because the difference of recognition rates with 5 and 6 features is rather small.
- Table 18 shows that, the 1NN produced the best result however the decision of the 1NN is more slower than in the case of ANN or DT, see for example [4]. Therefore, the usage of DT with CHI can be a good decision when the calculation capacity is strongly limited.
- With only one training, the performance of the ANN is not as good as we expected. The outcome of an ANN training depends on more hyperparameters, therefore a right ANN construction requires hyperparameter search and more trainings. However, it is a very time-consuming process.
- Since CHI used only time domain features, in this case frequency domain features are negligible.

- The result does not show strong relation between selected features and measured recognition rates for subjects of similar age.

Conclusion

In this article, we investigated the performance of feature selection methods on the HAR problem in combination with three well-known machine learning algorithms. To the survey two external sources have been utilised: WARD 1.0 database was the data source and the feature selection methods derived from an open source repository. At the beginning of the article, the most common feature extraction methods from time, frequency and other domains have been collected from the literature. Thereafter we selected 7 volunteers with different ages from the WARD and applied the feature extraction methods on their data. The selected features were the input of the ANN, 1NN and DT classifiers and the recognition rates have been archived.

Duarte et al., Gao et al., Maurer et al., Khan et al., Oniga et al. and Yang et al. used similar research conditions and environment to their study (they used single tri-axial accelerometer attached to one part of the body; they applied machine learning algorithm(s); they split time series into windows; etc.) and reached 95%, 97.9%, 99%, 97.8%, 96.4% and 92.8% recognition rates, respectively [4, 5, 9, 10, 12, 13]. As Table 10-16 demonstrate, we reached approximately 100% recognition rates in each cases. It is better than previous results and clearly indicates the efficiency of the feature selection and machine learning combination.

Pinardi et al., Su et al. and Yang et al. are also used the WARD database in their research [7,29,30]. They archived 97.8%, 98.5% and 93.5% recognition rates with five sensors and different kinds of classifiers (majority voting, distributed sparsity and support vector machine). Against their works, we showed that less sensors are enough for good classification and a general classifier with an appropriate feature selection algorithm can overcome other classification approaches in the HAR problem.

Bibliography

[1] Suto, J.; Oniga, S.; Pop Sitar, P. (2016); Comparison of wrapper and filter feature selection algorithms on human activity recognition, *Computers Communications and Control (ICCCC), 2016 6th International Conference on*, IEEE Xplore, e-ISSN 978-1-5090-1735-5, DOI: 10.1109/ICCCC.2016.7496749, 124-129.

[2] Chernbumroong, S.; Cang, S.; Atkins, A.; Yu, H. (2013); Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications*, ISSN: 0957-4174, DOI: 10.1016/j.eswa.2012.09.004 40(5):1662-1676.

[3] Sebestyen, G.; Tirea, A.; Albert, R. (2012); Monitoring human activity trough portable devices. *Carpathian Journal of Electronic and Computer Engineering*, ISSN 2343-8908, 5(1):101-106.

[4] Gao, L; Bourke, A.K.; Nelson, J. (2014); Evaluation of accelerometer based multi−sensor versus single−sensor activity recognition systems. *Medical Engineering & Physics*, ISSN: 1350-4533, DOI: 10.1016/j.medengphy.2014.02.012, 36(6):779-785.

[5] Maurer, U.; Smailagic, A.; Siewiorek, D,P; Deisher, M. (2006); Activity recognition and monitoring using multiple sensors on different body positions. *International Workshop on Wearable and Implementable Body Sensor Networks*, ISBN: 0-7695-2547-4, DOI: 10.1109/BSN.2006.6, Cambridge, USA, 112-116.

[6] Orha, I.; Oniga, S. (2015); Wearable sensor network for activity recognition using inertial sensors, *Carpathian Journal of Electronic and Computer Engineering*, ISSN 2343-8908, 8(2):3-6.

[7] Yang, A.Y.; Jafari, L.; Systry, S.S.; Bajcsy, R. (2009); Distributed recognition of human actions using wearable motion sensor networks. *Journal of Ambient Intelligence and Smart Environments*, ISSN 1876-1372, DOI: 10.3233/AIS-2009-0016, 1(1):1-5.

[8] Preece, J.S.; Goulermas, J.Y.; Kenney, L.P.J.; Howard, D. (2009); A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Transactions on Biomedical Engineering*, ISSN: 1558-2531, DOI: 10.1109/TBME.2008.2006190, 56(3):871-879.

[9] Khan, A.M.; Lee, Y.K.; Lee, S.Y.; Kim, T.S. (2010); A triaxial accelerometer−based physical−activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE Transactions on Information Technology in Biomedicine*, ISSN: 1558-0032, DOI: 10.1109/TITB.2010.2051955, 14(5):1166-1172.

[10] Oniga, S.; Suto, J. (2015); Optimal recognition method of human activities using artificial neural networks. *Measurement Science Review*, ISSN 1335-8871, DOI: 10.1515/msr-2015-0044, 15(5):323-327.

[11] Oniga, S., Suto, J. (2014); Human activity recognition using neural networks. *15th International Carpathian Control Conference*, ISBN: 978-1-4799-3528-4, DOI: 10.1109/CarpathianCC.2014.6843636, Velke Karlovice, Czech Republic, 403-406.

[12] Yang, J.Y.; Wang, J.S.; Chen, Y.P. (2008); Using acceleration measurements for activity recognition: an effective learning algorithm for constructing neural classifiers. *Pattern Recognition Letters*, ISSN: 0167-8655, DOI: 10.1016/j.patrec.2008.08.002, 29(16):2213-2220.

[13] Duarte, F.; Lourenco, A.; Abrantes, A. (2014); Classification of physical activities using a smartphone: evaluation study using multiple users. *Procedia Technology*, ISSN: 2212-0173, DOI: 10.1016/j.protcy.2014.10.234, 17(1):239-247.

[14] Karantonis, D.M.; Narayanan, M.R.; Mathie, M.; Lovell, N.H.; Celler, B.G. (2006); Implementation of a real−time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Transactions on Information Technology in Biomedicine*, ISSN: 1558-0032, DOI: 10.1109/TITB.2005.856864, 10(1):156-167.

[15] Lara, D.O.; Labrador, M.A. (2013); A survey on human activity recognition using wearable sensors. *IEEE Communications Survey & Tutorials*, ISSN: 1553-877X, DOI: 10.1109/SURV.2012.110112.00192, 15(3):1192-1209.

[16] Godfrey, A.; Conway, R.; Meagher, D.; Olaighin, G. (2008); Direct measurement of human movement by accelerometry. *Medical Engineering & Physics*, ISSN: 1350-4533, DOI: 10.1016/j.medengphy.2008.09.005, 30(10):1364-1386.

[17] Bayat, A.; Pomplun, M.; Tran, D.A. (2014); A study on human activity recognition using accelerometer data from smartphones. *Procedia Computer Science*, ISSN: 1877-0509, DOI: 10.1016/j.procs.2014.07.009, 34(1):450-457.

[18] Kavanagh, J.J.; Menz, B.H. (2008); Accelerometry: a technique for quantifying movement patterns during walking. *Gait Posture*, ISSN: 0966-6362, DOI: 10.1016/j.gaitpost.2007.10.010, 28(1):1-15.

[19] Shoaib, M.; Bosch, S.; Incel, O.D.; Scholten, H.; Havinga, P.J.M. (2015); A survey of online activity recognition using mobile phones. *Sensors*, ISSN 1424-8220, DOI: 10.3390/s150102059, 15(1):2059-2085.

[20] Suto, J.; Oniga, S.; Buchman, A. (2015); Real time human activity monitoring. *Annales Mathematicae et Informaticae*, ISSN 1787-6117, 44(1):187-196.

[21] Cheng, C.H.; Wang, P.S.P. (2005); *Handbook of Pattern Recognition and Computer Vision, 3th ed.*, World Scientific, ISBN 978-981-4505-21-5.

[22] Liu, H.; Motoda, H.; Setiono, R.; Zhao, Z. (2010); Feature selection: an ever evolving frontier in data mining, *4th Workshop on Feature Selection in Data Mining*, ISSN 1533-7928, Hyderabad, India, 4-13.

[23] Liu, H.; Motoda, H. (2008); *Computational Methods of Feature Selection*, CRC Press Taylor & Francis Group, ISBN 978-158-488-878-9.

[24] Hall, M.A.; Smith, L.A. (1999); Feature selection for machine learning: Comparing a correlation−based filter approach to the wrapper, *Florida Artificial Intelligence Symposium*, Florida, ISBN 978-1-57735-756-8, USA, 235-239.

[25] Saeys, Y.; Inza I.; Larranaga P. (2007); A review of feature selection techniques in bioinformatics, *Bioinformatics*, DOI: 10.1093/bioinformatics/btm344, ISSN 1460-2059, 23(19):2507-2517.

[26] Jatoba, C.L.; Grobmann, U.; Kunze, U.; Ottenbacher J.; Stork, W. (2008); Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. *30th Annual International IEEE EMBS Conference*, ISBN: 978-1-4244-1814-5, DOI: 10.1109/IEMBS.2008.4650398, Vancouver, Canada, 5250-5253.

[27] Zhao, Z.; Morstatte, F.; Sharma, S.; Alelyani, S.; Anand, A.; Liu, H. (2011); *Advancing feature selection research- ASU feature selection repository.* Technical Report, Arizona State University, $http://featureselection.asu.edu/old/featureselection_techreport.pdf$

[28] Oniga, S.; Suto, J. (2016); Activity recognition in adaptive assistive systems using artificial neural networks. *Elektronika ir Elektrotechnika*, ISSN: 2029-5731, DOI: http://dx.doi.org/10.5755/j01.eee.22.1.14112, 22(1):68-72.

[29] Pinardi, S.; Bisiani, R. (2010); Movement recognition with intelligent multisensor analysis, a lexical approach. *6th International Conference on Intelligent Environments*, ISBN 978-1-60750-639-3, DOI: 10.3233/978-1-60750-638-6-170, Kuala Lumpur, Malaysia, 170-177.

[30] Su, B.; Tang, Q.; Wang, G.; Sheng, M. (2016); The recognitions of human daily actions with wearable motion sensor system, *Lecture Notes in Computer Science: Transactions on Edutainment XII*, ISBN 978-3-662-50544-1, DOI: 10.1007/978-3-662-50544-16, 9292(1):68-77.

[31] Ertugrul, O.F.; Kaya, Y. (2016); Determining the optimal number of body-worn sensors for human activity recognition. *Soft Computing*, ISSN 1433-7479, DOI: 10.1007/s00500-016-2100-7, 20(2):1-8.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(1):131-147, February 2017.

Delay/Disruption Tolerant Networking-Based Routing for Rural Internet Connectivity (DRINC)

C. Velásquez-Villada, Y. Donoso

Carlos Velásquez-Villada*, Yezid Donoso
Systems and Computing Engineering Department
Universidad de los Andes
Bogotá D.C., Colombia, South America
*Corresponding author: ce.velasquez917@uniandes.edu.co
ydonoso@uniandes.edu.co

Abstract: Rural networking connectivity is a very dynamic and attractive research field. Nowadays big IT companies and many governments are working to help connect all these rural, disconnected people to Internet. This paper introduces a new routing algorithm that can bring non-real-time Internet connectivity to rural users. This solution is based on previously tested ideas, especially on Delay/Disruption Tolerant Networking technologies, since they can be used to transmit messages to and from difficult to access sites. It introduces the rural connectivity problem and its context. Then, it shows the proposed solution with its mathematical model used to describe the problem, its proposed heuristic, and its results.
The advantage of our solution is that it is a low-cost technology that uses locally available infrastructure to reach even the most remote towns. The mathematical model describes the problem of transmitting messages from a rural, usually disconnected user, to an Internet connected node, through a non-reliable network using estimated delivery probabilities varying through time. The forwarding algorithm uses local knowledge gathered from interactions with other nodes, and it learns which nodes are more likely to connect in the future, and which nodes are more likely to deliver the messages to the destination. Our algorithm achieves an equal or better performance in delivery rate and delay than other well-known routing protocols for the rural scenarios tested.
This paper adds more simulation results for the proposed rural scenarios, and it also extends the explanation of the mathematical model and the heuristic algorithm from the conference paper "Delay/Disruption Tolerant Networks Based Message Forwarding Algorithm for Rural Internet Connectivity Applications" [1] (doi: 10.1109/ICCCC.2016.7496732).[a]
Keywords: dtn routing, opportunistic forwarding, rural connectivity, rural internet, non-real-time communications.

[a]Reprinted (partial) and extended, with permission based on License Number 3958950667073 ©[2016] IEEE, from "Computers Communications and Control (ICCCC), 2016 6th International Conference on".

1 Introduction

The United Nations declared access to Internet as a human right in 2011 [2]; however, around 60% of the world population [3], especially in rural and underdeveloped regions does not have any Internet access. Access to Internet can improve life quality through access to more information, education and applications that will ease the life of citizens in remote areas [2]. Giving access to Internet to these communities is not an easy task. There are many technical and social challenges, such as a reliable supply of electricity, access to electronic devices, network coverage, education and training, that have to be overcome before a complete solution can be given.

Internet connectivity around the world is estimated at 43.4% (individuals using the Internet) for 2015 [3] and it is even lower for developing countries (35.3%) and population in rural areas (see Figure 1). Many benefits of Internet connectivity for rural communities in areas such as education, health-care, agriculture, economics and politics, among others, have already been discussed in one of our previous papers [4]. There are also benefits based on the local infrastructure deployed, since users can create contents and share them with neighbors or other local users, local news can be reported on a local website, and people can share their music, documents, videos or libraries. In some communities, people have created local websites for their businesses and even local radio stations over the infrastructure that also allows them to connect to Internet. These applications depend on a strong deployment of a local infrastructure and the presence of people willing to train and educate local users. Internet connectivity, then, seems associated with more possibilities of bridging the social divide and reducing the social and economic gaps around the world.

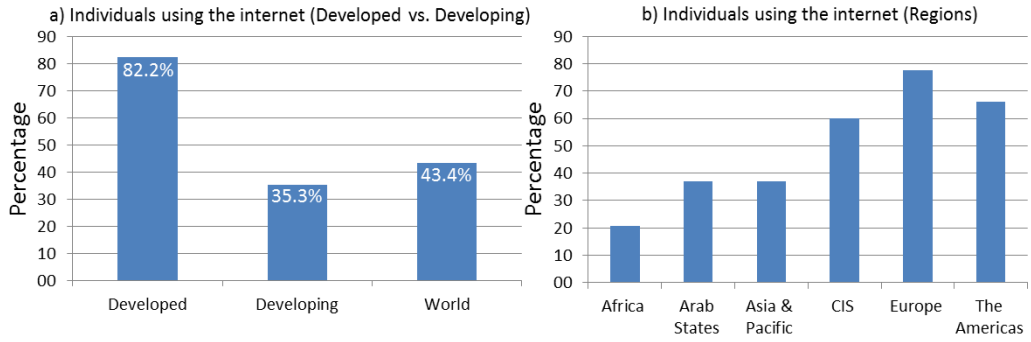


Figure 1: Individuals using the Internet. a) Developed Countries, Developing Countries and the World. b) Statistics by World Region. Information taken from [3]

How to connect world's rural population to Internet is a current and important global issue. Many big companies in Information Technologies, like Google and Facebook, are testing techniques and methods to help connect these currently disconnected rural users. Governments are trying to cover more of their territories, through contracts to deploy the necessary infrastructure and pushing local Telecommunications Companies to cover more towns. Networking equipment providers also expect an increase in coverage through more energy efficient equipment. Google has recently launched an initiative to use stratospheric balloons to connect the whole world at low costs, and Facebook is developing a model of free restricted internet access in partnership with governments and cellular carriers. These and other initiatives for rural connectivity, will be discussed in more detail in Section 2.

This paper proposes a routing and forwarding algorithm for a low cost technology that can be used over local user devices for non-real-time communications. This solution will allow people to ask for and retrieve information from Internet, for non-real-time applications, using transportation vehicles to carry their requests and replies. Figure 2 offers a schematic view of the possible scenario for this connectivity problem. The proposed solution will need an additional infrastructure that can be installed on buses or other mechanical transportation means to remote towns, and a device that can be installed at the main building in the town to serve as a request center. It will also need an application for people to install on their devices so they could be able to send requests to the request center or to the transportation device, and obtain their answers when the transportation returns to town; however, there will be no guarantees about delivery times or even deliveries at all. Users will be connected in a non-real-time fashion to Internet. This technology uses Delay and Disruption Tolerant Networks (DTN) to connect

everyone to Internet using non real-time communications based on the DTN Architecture [5] and inspired on Daknet [6]. Daknet is a technology used to interconnect rural communities in India using mechanical transportation means. Daknet uses PROPHET [7] as its routing protocol to deliver and receive messages between users and Internet. However, as far as we know, there are not publications about its performance on field. We tested the official implementation of PROPHET in The ONE (Opportunistic Network Environment) Simulator [8] and compared our proposal against it and other well-known DTN routing protocols, in rural connectivity scenarios for developing countries.

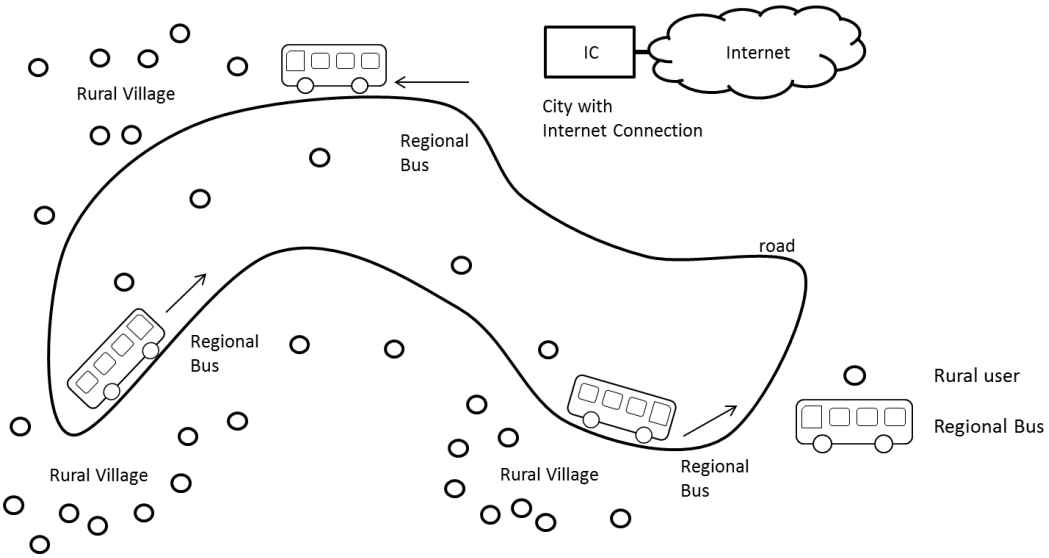


Figure 2: Proposed scenario for the rural connectivity problem

This paper extends the conference paper [1]. The key additions of this journal version are as follows. First, Figure 4, has a small correction. It has an additional device in the Internet Connected town, to receive user's requests and Internet replies. Section 2 includes and describes more related works, and DTN implementations for rural solutions. Besides, this paper contains an extended explanation of the mathematical model and of the heuristic algorithm from Section 3. Finally, this paper contains additional results for the simulated scenarios.

This document is divided as follows: Section 2 presents a review about rural internet networking. It reviews some connectivity technologies and initiatives that are currently being tested and developed around the world for rural internet connectivity. It also contains a brief description of the most important protocols in the state-of-the-art of DTN routing protocols that can be used for rural connectivity. Section 3 gives a detailed description of our proposed mathematical model, solution architecture, and forwarding algorithm developed in this research. Section 4 analyses the simulation results, including the comparison with other protocols. Finally, Section 4 presents general conclusions, highlighting the most important achievements of this research, and it also states some possible directions for future work.

2 Delay/Disruption tolerant networks and rural networking

2.1 Delay/Disruption tolerant networks

Delay/Disruption Tolerant Networks (DTN) are a recent kind of networking technology for environments with difficult conditions. DTN were developed for spatial communications to over-

come the long distances (long delays) and frequent disruptions that are usual for this environment. From this research, the DTN Architecture [5] and the Bundle Protocol [9] were developed. The DTN Architecture describes a networking technology that has to be able to work on environments with long delay and frequent disruptions. It relaxes several assumptions of the traditional Internet, including the ideas that there exists an end to end path between source and destination, that error correction based on acknowledgements is effective, that packet switching is the most appropriate abstraction for interoperability and performance, and that endpoint-based security mechanisms are enough. The Bundle protocol is the technical implementation of the DTN architecture. It proposes a new intermediate layer between the application layer and the transport layer. The Bundle protocol is independent of the networking and network access technologies, and it allows DTN nodes to exchange messages between neighbors as a relay for a transmission between distant DTN nodes.

Figure 3 shows a graphical explanation of the bundle protocol position in the networking layered model. It shows that bundles can communicate between intermediate DTN nodes or between end DTN nodes. It also shows that transmitting a bundle from a DTN node to another is made via lower layer protocols, starting with the transport layer and finishing in the other node's transport layer, and that this communication can be made over TCP/IP protocols or any other networking stack. It also shows that transmitting a single bundle between two DTN nodes can require the exchange of several segments, datagrams and frames between the inferior layers.

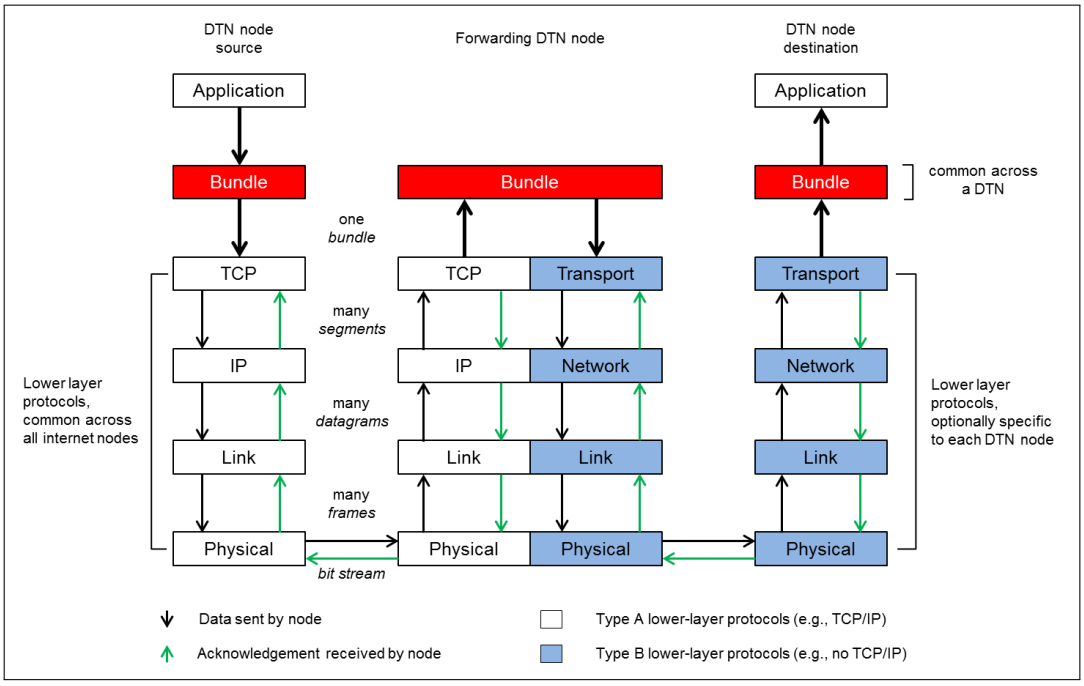


Figure 3: Bundle layer as middleware between DTN applications and traditional networking protocols [10]

2.2 Rural internet networking

Recently, there have been many initiatives to connect the unconnected part of the world. These unconnected citizens are, primarily, located at rural areas in developing countries. Main players in the IT world, like Google, and Facebook, are heading these initiatives. The next paragraphs summarize a review of a couple of their projects and a DTN-Based, rural connectivity

technology, the DakNet project.

Google's Loon [11] and Link [12] projects are some of Google initiatives to connect people to internet. These projects are focused on rural users in developing countries, but they are part of a Google's initiative to connect more users to Internet and expand its business. The Loon project uses aero-static balloons that create a mesh network between them and several earth-based stations. These earth-based stations have internet connectivity via satellital internet or wireless links. The Loon project allows local users to send request to the balloon network (via LTE technology) and obtain a response in almost real time through the connection at the earth-based station. The Link project deploys optical fiber networks in metropolitan regions, and it builds Wi-Fi networks as last-mile solution for potential users. Google started the Link project in Uganda and it is expanding it in Ghana. It has installed or is currently installing five fiber optical networks in these two countries. They report that these networks have increased Internet connectivity availability and bandwidth. They allow users to access educational information, share medical results, expand the impact of their businesses and build cooperation networks between universities.

Facebook's Internet.org [13] and ARIES [14] projects are some of Facebook initiatives to connect rural people to internet. The Internet.org project, also called Free Basics, is an initiative in developing countries and rural areas where local users with a mobile phone can access a limited version of Internet via a local mobile carrier. Facebook, in agreement with local governments and mobile carriers, give this limited internet access free of charge. The ARIES project is one of the Facebook's initiatives to bring connectivity to areas where there is not networking infrastructure available. ARIES is an antenna array that uses Multiple Input - Multiple Output (MIMO) technology to transmit up to 24 simultaneous streams over the same spectrum. It is a base station with 96 antennas that tries to improve the spectral efficiency of wireless communications, working at different frequencies and reaching 71 bps/Hz. Facebook is developing this technology with the idea to reduce deployment costs of networking in urban areas; besides, it is aiming at connecting, through this antenna, rural population living in a radius of 40 km. of urban centers. Facebook estimates that 97% of the world's population live inside these areas.

Governments are also investing in rural internet connectivity. In Colombia, the ICT Ministry created in 2011 the Proyecto Nacional de Fibra Óptica (National Optical Fiber Project) with the aim to connect to Internet all Colombian municipalities. This governmental contract had a cost of around 230 million USD. It was planned to be executed between 2012 and 2014 [15]. In recent declarations from the Colombian ICT Minister, he stated that the project currently covers around 90% of the Colombian municipalities. Although this is a big investment, it covers the main urban developments in Colombia; rural areas, near these municipalities, are not being covered by this project.

DakNet [6], [16] is a rural connectivity solution based on DTN. DakNet has been in use for several years in India. DakNet installs kiosks in rural towns, where users can go and use a computer to communicate with other rural users or with Internet servers in a non-real-time fashion. Messages are stored in the local computer until a motorcycle or a bus stops by the town, collecting the messages via an access point installed on it. The vehicle will carry the messages to an Internet connected spot, where they will be delivered and the desired answers retrieved. DakNet uses PROPHET [7] as its routing protocol. DakNet is the main inspiration for the present work.

2.3 DTN routing protocols

Epidemic routing [17] is the most known routing protocol for opportunistic networks. It replicates the messages it wants to deliver giving copies to every node it meets until the message

gets to its destination. Epidemic guarantees the delivery of a message if enough resources, including time, are given.

Spray and Wait routing [18] is the evolution of epidemic routing. It also is a replication-based routing protocol. It works in two phases, a replication phase where the node creates copies of the messages it wants to deliver, and give the copies to every node it meets, just like epidemic routing. The second phase is the delivery phase, in this phase the nodes stop the replication of messages and they will only pass the message to the destination node.

PROPHET [7] is a probabilistic routing protocol based on the history of contacts between the nodes and transitivity. PROPHET estimates a delivery probability for every node it meets and decreases it by a constant factor over a constant time interval. It also can calculate the delivery probability for nodes it has not met, using intermediary nodes to deliver the messages (transitivity).

A more detailed classification and comparison of DTN routing protocols can be found in reviews [19], [20] and [21]. The heuristic proposed in this paper uses ideas from some of the protocols described in this section.

3 DTN for rural internet connectivity

3.1 Proposed architecture

This section presents a rural networking connectivity scenario, where communication requests will be originated from nodes in an usually disconnected rural area, to other nodes or towards Internet servers. The originating rural nodes make a mobile, sparsely distributed network. They will be able to communicate with each other and they could serve as relays to deliver messages to a static, always-on node in the nearest town, the Access Point (AP). The AP can store users' requests until a Mobile Access Point (MAP) comes to the town and retrieves them. The MAP will store the requests from the small, disconnected towns it visits and it will deliver them at the drop point, an Internet Connected Node (IC) in a bigger, Internet connected town. The IC is the connection between the DTN for rural users and the usual Internet. The IC can deliver the rural users' messages to the respective Internet Servers and retrieve their replies. All users' replies will be stored in the IC until a MAP comes by and collects the replies for the users in any of the towns in its route. See Figure 4, for a general scheme of the proposed architecture. DTN nodes representing rural users are expected to get more sparsely distributed as they are located farther away from the town's AP. They can use other DTN nodes to convey requests to the AP and to deliver replies from Internet to the final user.

3.2 Optimization Model for the Rural Internet Connectivity Problem

This subsection introduces a new bidirectional, multiobjective, non-linear mathematical model for a DTN in a rural networking connectivity scenario. This model is extended and corrected from our previous models in [4] and [22]. This mathematical model maximizes the delivery probabilities over the paths that may exist through time, and at the same time it tries to minimize the mean delivery time for each message. This model optimizes through time the routing of messages based on the availability probability of each link in a possible path. A path from a source node to a destination may not exist at a single moment but over various intervals. Also, the mathematical model will try to deliver all messages to their destinations, restricted to the links and buffers capacities and nodes' availability probabilities given by their movements. See Figure 5 for a graphical representation of the mathematical abstraction. The mathematical model is presented in equations (1) to (7).

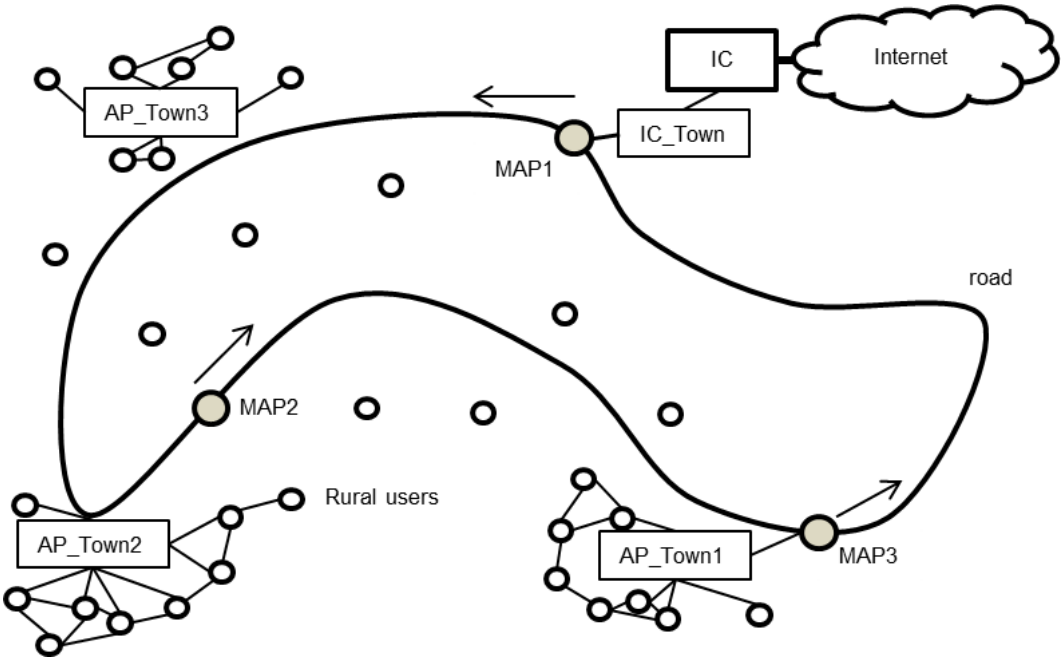


Figure 4: Proposed architecture for the rural internet connectivity solution

The mathematical model assumes that there is a set of rural nodes (N), with intermittent connections (E) between a subset of them and the local AP. These connections are modeled through time dependant availability probabilities ($a_{ij}(t)$). If the availability probability at any given time for a couple of nodes is bigger than zero, a path can be formed using them. The set P represents the possible paths that can exist in the network through time. Delivery probabilities ($d_{ij}(t)$) are calculated using availability probabilities of neighboring nodes and their knowledge of a path to, or a node with previous contacts with, the destination. Nodes and links have capacities that should be respected. We assume that these capacities do not change over time, then, parameters c_{ij} and c_{ii} represent the link capacity from node i to node j and the node i capacity to store messages through time, respectively. Parameter $b_i(t)$ is a time based vector with demands and supplies for node i through time. All b vectors are grouped together in matrix B, representing the desired flow of information for the system. The model uses discrete times t.

Table 1: Mathematical notation for the Rural Internet Connectivity problem

Var./Param.	Definition
N	Set of nodes, $i \in N$
E	Set of links, $(i,j) \in E$
T	Set of discrete time intervals $t \in T$
P	Set of paths, $(i,j) \in P$, $a_{ij}(t) > 0$
B	Matrix of information demands and supplies, $b_i(t) \in B$
$a_{ij}(t)$	Availability Probability of $(i,j) \in E$ at time t
$d_{ij}(t)$	Delivery Probability of node i through node j at time t
c_{ij}	Capacity of link $(i,j) \in E$
c_{ii}	Storage capacity of node i
δ_t	Time interval duration
$x_{ij}(t)$	Data flow through link $(i,j) \in E$ at time t

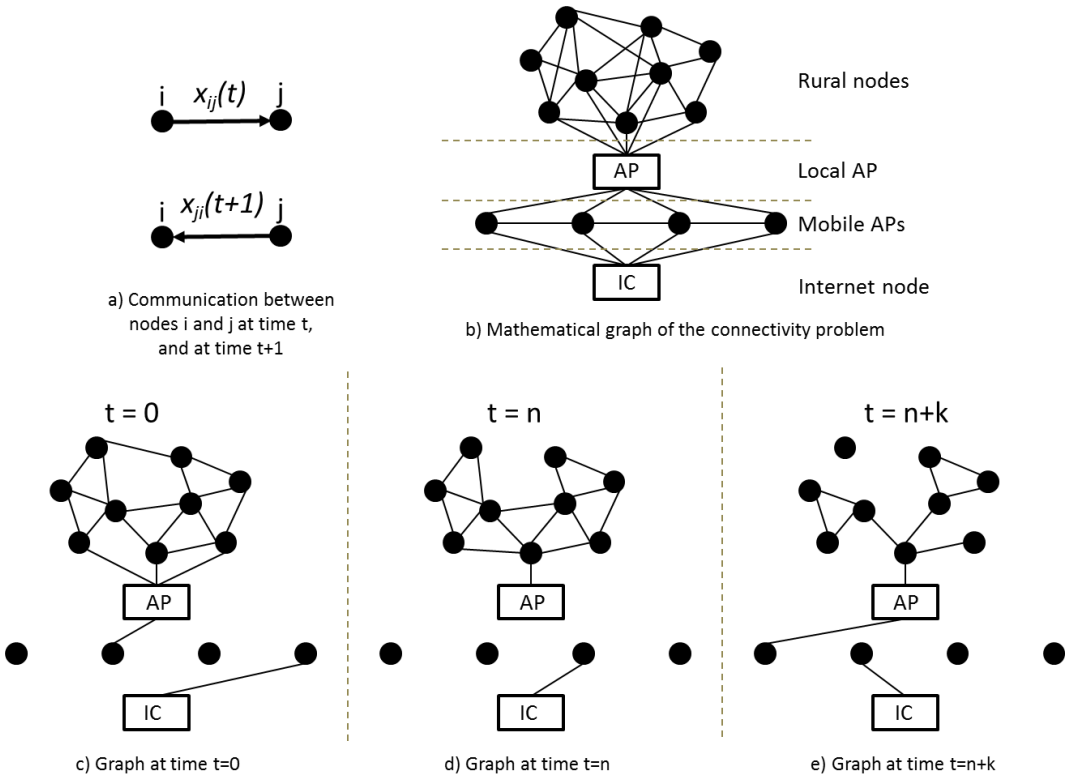


Figure 5: Mathematical abstraction for the Rural Internet Connectivity problem

Each of them represents a moment lasting long enough for a node to reliably transfer a message to a neighbor. The lasting of each time interval is modeled through the parameter δ_t. Forwarding decisions are made based on links' availability probabilities, the size and the delivery time of the message. These decisions are represented by the positive integer variable $x_{ij}(t)$, which represents how much information should flow over a link at any given time. Table 1 summarizes all the model's parameters and variables.

Objective Functions

$$\max \prod_{(i,j)\in P} d_{ij}(t)x_{ij}(t) \tag{1}$$

$$\min \sum_{t\in T} \frac{\delta_t}{\sum_{i\in N} |b_i(t)|} \tag{2}$$

Constraints

$$\sum_{j\in N} x_{ij}(t) - \sum_{j\in N} x_{ji}(t) + x_{ii}(t) - x_{ii}(t-1) = b_i(t) \qquad \forall (i,j) \in E, i \neq j, t \geq 1 \tag{3}$$

$$\sum_{j\in N} x_{ij}(t) - \sum_{j\in N} x_{ji}(t) + x_{ii}(t) = b_i(t) \qquad \forall (i,j) \in E, i \neq j, t = 0 \tag{4}$$

$$x_{ij}(t) + x_{ji}(t) \leq c_{ij}\delta_t \qquad \forall (i,j) \in E, i \neq j \tag{5}$$

$$x_{ii}(t) \leq c_{ii} \qquad \forall i \in N \tag{6}$$

$$x_{ij}(t) \in \mathbb{Z}_{\geq 0} \qquad \forall (i,j) \in E, p \in P, t \in T \tag{7}$$

Equations (1) and (2) represent the objective functions intended to maximize the delivery probability of messages over all paths through time, and to minimize the mean time taken to

deliver all messages. Equations (3) to (7) state the mathematical model constraints. Equations (3) and (4) are data flow constraints, requiring all messages to be delivered to their destinations. Equations (5) and (6) are capacity constraints, requiring all transmission to be in the limits of buffers and links capacities. Finally, equation (7) states that the decision variable is a positive integer variable defined over all available paths through time and representing the data flow over an edge for a time interval.

3.3 Heuristic approach of the proposed solution for the rural internet connectivity problem

Every user in the rural networking connectivity scenario, shown in Figure 4, should use a distributed routing algorithm to deliver all its messages to their destinations. The proposed algorithm introduced in this subsection works with limited and local information. A node only knows where it is based on a general location abstraction given by a Uniform Resource Identifier (URI) described in the DTN Architecture [5]. Based on this general location it can direct its requests to the nearest AP. Each node must create two tables that it should save to a non-volatile memory. One table is for saving the name, address, availability probability, last contact time, average time between contacts and centrality of every node it meets (neighbors' table). The centrality is a measure of how many nodes a neighbor has met, and it can be used to make forwarding decisions. Nodes with better centrality metrics will be preferred when no delivery probability to a destination is known. Delivery probabilities (d) are calculated from nodes that have a path to the destination. They can give a delivery probability based on previous contacts with the destination, and neighboring nodes of these nodes can have a delivery probability through them. Delivery probabilities are kept at a deliveries table.

When a node wants to send a message to another node or to Internet (through the AP), it will create an entry in its deliveries table with the information about the neighbors with paths to the destination, the average time between contacts for each one of these neighbors, the delivery probability to the destination through the neighbors, and the estimated time to deliver the message given by the neighbor. DTN nodes will store in memory these tables, so they can forward messages in an effective way, minimizing time and maximizing the delivery probability.

Nodes should have enough capacity to store own's and neighbors' messages. Since the communication model is not a real time one, this should not be a problem. Nodes will communicate using wireless links (IEEE 802.11g/n/ac or a similar technology) that will give them at least 50 Mbps, so a contact of a few minutes (a MAP visiting the location of an AP in the rural town) should be enough for a couple of nodes to exchange the messages (requests and replies) that they have for each other. The AP should have a significantly bigger memory capacity, since it is the bottleneck for all messages leaving the rural network and going to Internet. The memory size should depend on the frequency of delivery and reception of messages. If MAPs pass frequently by the AP's position, the memory could be smaller than the memory requirement for APs where MAPs pass once a day or even less frequently. Figure 6 depicts a flow diagram representing the routing and forwarding algorithm for each DTN node.

4 Results for Rural Internet Connectivity Scenarios

This section presents the results for different Rural Internet connectivity scenarios. These scenarios have several rural users distributed over a map, one AP, one IC, and one or more mobile APs, connecting the AP to the IC.

These proposed scenarios are located around a small and remote town in Colombia, called La Macarena. It is a small community of about 500 inhabitants, with difficult access conditions.

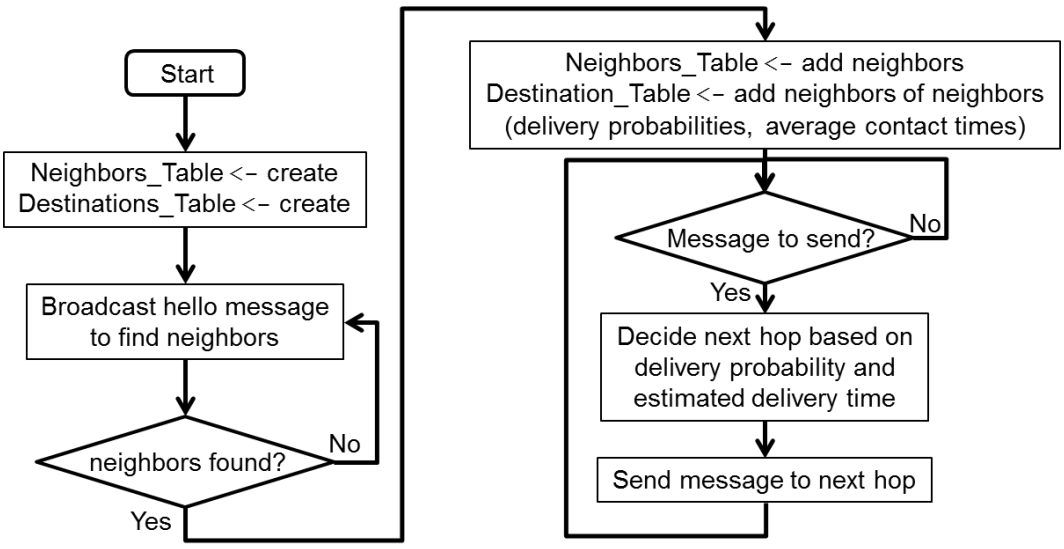


Figure 6: Flow diagram for the proposed algorithm for the Rural Internet Connectivity Solution

There are currently three different ways to reach the town, a road to the nearest town (5 hours), two daily flights in small airplanes (for 5 to 10 people), and small canoes by the Guayabero river. This town also has a military base, where communication antennas work all day long. Rural users, then, have to go to the town in order to communicate to the capital or any other part of the country. We scaled the map around La Macarena using OpenJump [23] to use the mobility models of The ONE simulator. Rural DTN users are scattered around the town and they move around from the town to small farms nearby. There are roads from the town to most farms, also, the river is a very popular way of transportation.

For our simulations, DTN users can generate messages at any time and they can receive replies from Internet (through the static node at the military base or from a MAP). Messages have to get to the center of the town, at the military base, to be transmitted to Internet. There are DTN nodes moving in their farms, neighboring farms and the town, MAPs going from the farms to the town and viceversa over roads and the river, and the AP in the military base in the center of the town. This simulation scenario can be seen in Figure 7.

We created several scenarios changing the number of nodes (from 10 to 100 rural users), the size of messages (from 2 kB to 2 MB), and the time between messages (from 1 to 12 hours). These scenarios are described in Table 2. Messages are generated from rural DTN user nodes and the AP. These messages can be sent from user node to user node, from user node to AP or from AP to user node. MAPs only serve as relays to carry the messages from one location to another. Every scenario was tested for 3 simulated days (72 hours) with 5 different random seeds. The simulation results were obtained by changing the nodes' buffers and changing the user nodes movement model from random to MapRoute (a model where nodes follow a predefined path at a constant speed defined randomly at the beginning of the simulation). We changed the nodes' buffers from 10MB to 2000MB, but the results were always the same, they were not affected by the buffer size. We also changed the number of message replicas that our proposed algorithm uses; however, results were neither affected by replication number changes. At the end, our algorithm does not use replication.

Results are shown in Figures 8 to 15. They show the delivery rate, delay, overhead, and hop count of our DRINC algorithm against PROPHET, Epidemic, and SprayAndWait routing protocols. It can be seen that DRINC achieves the same delivery rate or even a better one in some cases than Epidemic routing and PROPHET for the proposed scenarios, and that all

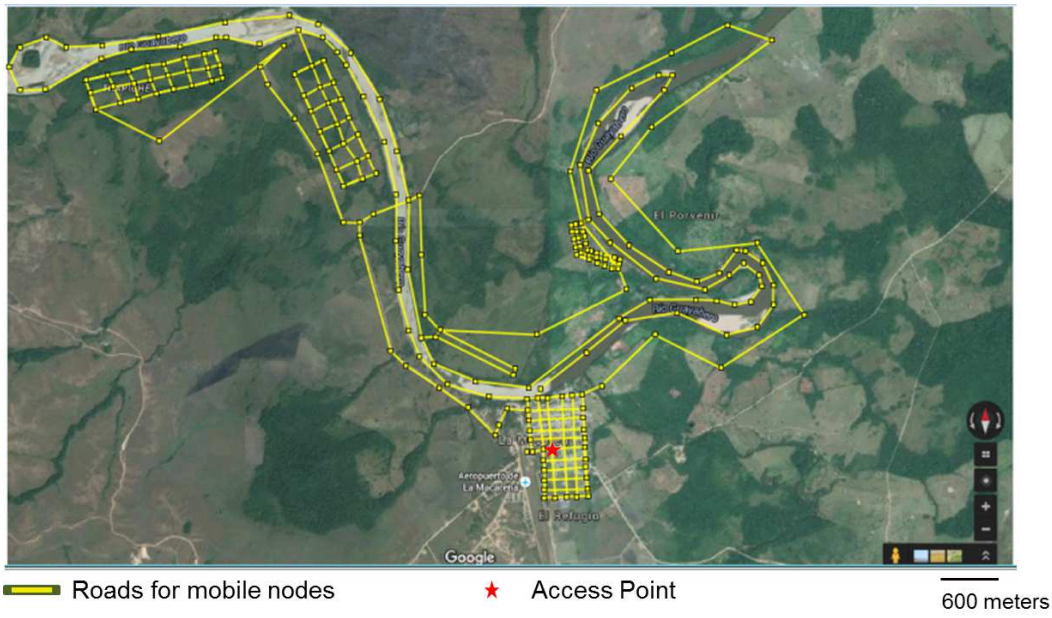


Figure 7: Scenario for the Rural Internet Connectivity problem based on La Macarena town in OpenJump

Table 2: Simulation scenarios the Rural Internet Connectivity problem based on La Macarena town

Scenario	Parameters				
Number	DTN nodes	AP	MAPs	Message size	Frequency
1	10	1	3	2 MB	1-2 hours
2	10	1	3	2 kB	1-2 hours
3	10	1	3	2 MB	1-12 hours
4	10	1	3	2 kB	1-12 hours
5	100	1	3	2 MB	1-12 hours
6	100	1	3	2 kB	1-12 hours

protocols deliver more messages with the random movement of the user nodes than they do with the MapRoute movement model of the nodes. This could be due to the size of the scenario and that the random movement model gives more contact opportunities in this scenario. The 95% confidence intervals were calculated for every result and they appear in the figures.

Figures 8 and 9 show simulation results for Epidemic, PROPHET, and SprayAndWait routing protocols against our DRINC algorithm. See Table 2 for a summary of the scenarios. Figure 8 shows delivery results for scenarios 1 to 4, these are scenarios with 10 nodes as rural users, generating messages every 1 to 2 hours or every 1 to 12 hours. Figure 9 shows delivery results for scenarios 5 and 6, these scenarios are the big ones, with 100 nodes as rural users, and they are generating new messages every 1 to 12 hours. When nodes have enough opportunities to communicate (random movement of the nodes), almost all messages are delivered by all protocols; however, when nodes are kept apart most of the time, by using MapRoute movement, all protocols deliver between a 50% to 60% of the messages, with SprayAndWait performing a little bit below this percentage.

Figures 10 and 11 show simulation results for delivery delay for Epidemic, PROPHET, and SprayAndWait routing protocols against our DRINC algorithm. See Table 2 for a summary of

Scenario	DTN nodes	IC	MAPs	Message size	frequency
1	10	1	3	2MB	1-2 hours

	Random	MapRoute
DRINC	0.97±0.01	0.49±0.10
Epidemic	0.97±0.01	0.49±0.10
ProphetV2	0.97±0.01	0.43±0.09
SprayAndWait	0.85±0.02	0.38±0.05

Scenario	DTN nodes	IC	MAPs	Message size	frequency
3	10	1	3	2MB	1-12 hours

	Random	MapRoute
DRINC	1.00±0.00	0.56±0.09
Epidemic	1.00±0.00	0.56±0.09
ProphetV2	0.96±0.07	0.53±0.14
SprayAndWait	0.89±0.00	0.44±0.09

Scenario	DTN nodes	IC	MAPs	Message size	frequency
2	10	1	3	2kB	1-2 hours

	Random	MapRoute
DRINC	0.98±0.00	0.54±0.10
Epidemic	0.98±0.00	0.54±0.10
ProphetV2	0.97±0.01	0.46±0.11
SprayAndWait	0.85±0.04	0.39±0.06

Scenario	DTN nodes	IC	MAPs	Message size	frequency
4	10	1	3	2kB	1-12 hours

	Random	MapRoute
DRINC	1.00±0.00	0.56±0.09
Epidemic	1.00±0.00	0.56±0.09
ProphetV2	1.00±0.00	0.53±0.14
SprayAndWait	0.89±0.00	0.44±0.09

Figure 8: Delivery rate for the Rural Internet Connectivity scenarios with 10 nodes. The scenarios change the message size between 2 MB and 2 KB, and the frequency of message generation from 1 to 2 hours or from 1 to 12 hours

Scenario	DTN nodes	IC	MAPs	Message size	frequency
5	100	1	3	2MB	1-12 hours

	Random	MapRoute
DRINC	1.00±0.00	1.00±0.00
Epidemic	1.00±0.00	1.00±0.00
ProphetV2	1.00±0.00	0.98±0.06
SprayAndWait	1.00±0.00	0.33±0.00

Scenario	DTN nodes	IC	MAPs	Message size	frequency
6	100	1	3	2kB	1-12 hours

	Random	MapRoute
DRINC	1.00±0.00	1.00±0.00
Epidemic	1.00±0.00	1.00±0.00
ProphetV2	1.00±0.00	0.98±0.06
SprayAndWait	1.00±0.00	0.33±0.00

Figure 9: Delivery rate for the Rural Internet Connectivity scenarios with 100 nodes. The scenarios change the message size between 2 MB and 2 KB

Scenario	DTN nodes	IC	MAPs	Message size	frequency
1	10	1	3	2MB	1-2 hours

	Random	MapRoute
DRINC	2.09±0.20	7.19±1.13
Epidemic	2.06±0.17	7.16±1.06
ProphetV2	2.66±0.20	6.95±1.26
SprayAndWait	1.60±0.21	5.91±1.28

Scenario	DTN nodes	IC	MAPs	Message size	frequency
3	10	1	3	2MB	1-12 hours

	Random	MapRoute
DRINC	2.12±0.18	5.17±1.65
Epidemic	2.12±0.18	5.17±1.65
ProphetV2	2.17±0.49	6.12±1.57
SprayAndWait	1.72±0.41	4.14±1.48

Scenario	DTN nodes	IC	MAPs	Message size	frequency
2	10	1	3	2kB	1-2 hours

	Random	MapRoute
DRINC	1.51±0.17	7.32±1.09
Epidemic	1.51±0.17	7.32±1.09
ProphetV2	1.85±0.19	6.87±1.40
SprayAndWait	1.21±0.14	5.77±1.66

Scenario	DTN nodes	IC	MAPs	Message size	frequency
4	10	1	3	2kB	1-12 hours

	Random	MapRoute
DRINC	1.85±0.18	5.13±1.63
Epidemic	1.85±0.18	5.13±1.63
ProphetV2	2.43±0.48	6.09±1.55
SprayAndWait	1.38±0.17	4.11±1.46

Figure 10: Delay in thousands of seconds for the Rural Internet Connectivity scenarios with 10 nodes. The scenarios change the message size between 2 MB and 2 KB, and the frequency of message generation from 1 to 2 hours or from 1 to 12 hours

Scenario	DTN nodes	IC	MAPs	Message size	frequency
5	100	1	3	2MB	1-12 hours

	Random	MapRoute
DRINC	0.15 ±0.02	4.78 ±1.09
Epidemic	0.15 ±0.02	4.78 ±1.09
ProphetV2	0.26 ±0.03	5.25 ±0.82
SprayAndWait	0.28 ±0.04	0.14 ±0.09

Scenario	DTN nodes	IC	MAPs	Message size	frequency
6	100	1	3	2kB	1-12 hours

	Random	MapRoute
DRINC	0.15 ±0.02	4.78 ±1.09
Epidemic	0.15 ±0.02	4.78 ±1.09
ProphetV2	0.26 ±0.03	5.25 ±0.82
SprayAndWait	0.28 ±0.04	0.14 ±0.09

Figure 11: Delay in thousands of seconds for the Rural Internet Connectivity scenarios with 100 nodes. The scenarios change the message size between 2 MB and 2 KB

Scenario	DTN nodes	IC	MAPs	Message size	frequency
1	10	1	3	2MB	1-2 hours

	Random	MapRoute
DRINC	11.97 ±0.11	11.32 ±0.98
Epidemic	11.95 ±0.12	11.29 ±0.98
ProphetV2	8.67 ±0.16	5.65 ±0.33
SprayAndWait	6.97 ±0.37	7.77 ±0.81

Scenario	DTN nodes	IC	MAPs	Message size	frequency
3	10	1	3	2MB	1-12 hours

	Random	MapRoute
DRINC	12.00 ±0.00	11.25 ±1.16
Epidemic	12.00 ±0.00	11.25 ±1.16
ProphetV2	8.69 ±0.49	4.73 ±0.73
SprayAndWait	6.10 ±0.52	7.34 ±1.51

Scenario	DTN nodes	IC	MAPs	Message size	frequency
2	10	1	3	2kB	1-2 hours

	Random	MapRoute
DRINC	11.97 ±0.02	10.93 ±0.89
Epidemic	11.97 ±0.02	10.92 ±0.90
ProphetV2	8.67 ±0.24	5.66 ±0.33
SprayAndWait	7.00 ±0.50	7.84 ±1.29

Scenario	DTN nodes	IC	MAPs	Message size	frequency
4	10	1	3	2kB	1-12 hours

	Random	MapRoute
DRINC	12.00 ±0.00	11.53 ±1.04
Epidemic	12.00 ±0.00	11.53 ±1.04
ProphetV2	8.40 ±0.24	4.73 ±0.73
SprayAndWait	6.28 ±0.27	7.29 ±1.53

Figure 12: Overhead (messages generated/messages delivered) for the Rural Internet Connectivity scenarios with 10 nodes. The scenarios change the message size between 2 MB and 2 KB, and the frequency of message generation from 1 to 2 hours or from 1 to 12 hours

Scenario	DTN nodes	IC	MAPs	Message size	frequency
5	100	1	3	2MB	1-12 hours

	Random	MapRoute
DRINC	102.09 ±0.10	101.53 ±1.30
Epidemic	102.09 ±0.10	101.53 ±1.30
ProphetV2	98.07 ±0.11	46.94 ±2.79
SprayAndWait	6.76 ±0.14	20.93 ±0.17

Scenario	DTN nodes	IC	MAPs	Message size	frequency
6	100	1	3	2kB	1-12 hours

	Random	MapRoute
DRINC	102.09 ±0.10	101.53 ±1.30
Epidemic	102.09 ±0.10	101.53 ±1.30
ProphetV2	98.07 ±0.11	46.94 ±2.79
SprayAndWait	6.76 ±0.14	20.93 ±0.17

Figure 13: Overhead (messages generated/messages delivered) for the Rural Internet Connectivity scenarios with 100 nodes. The scenarios change the message size between 2 MB and 2 KB

Scenario	DTN nodes	IC	MAPs	Message size	frequency
1	10	1	3	2MB	1-2 hours

	Random	MapRoute
DRINC	2.69 ±0.17	2.81 ±0.47
Epidemic	2.71 ±0.19	2.79 ±0.42
ProphetV2	2.11 ±0.12	2.44 ±0.26
SprayAndWait	2.21 ±0.11	2.20 ±0.21

Scenario	DTN nodes	IC	MAPs	Message size	frequency
3	10	1	3	2MB	1-12 hours

	Random	MapRoute
DRINC	2.58 ±0.67	2.53 ±0.32
Epidemic	2.58 ±0.67	2.53 ±0.32
ProphetV2	1.97 ±0.38	2.17 ±0.53
SprayAndWait	2.02 ±0.47	2.01 ±0.35

Scenario	DTN nodes	IC	MAPs	Message size	frequency
2	10	1	3	2kB	1-2 hours

	Random	MapRoute
DRINC	2.61 ±0.18	2.95 ±0.45
Epidemic	2.61 ±0.18	2.95 ±0.45
ProphetV2	2.07 ±0.10	2.55 ±0.35
SprayAndWait	2.18 ±0.06	2.21 ±0.32

Scenario	DTN nodes	IC	MAPs	Message size	frequency
4	10	1	3	2kB	1-12 hours

	Random	MapRoute
DRINC	2.29 ±0.14	2.44 ±0.43
Epidemic	2.29 ±0.14	2.44 ±0.43
ProphetV2	1.93 ±0.31	2.17 ±0.53
SprayAndWait	1.85 ±0.18	2.01 ±0.35

Figure 14: Hop Count for the Rural Internet Connectivity scenarios with 10 nodes. The scenarios change the message size between 2 MB and 2 KB, and the frequency of message generation from 1 to 2 hours or from 1 to 12 hours

Scenario	DTN nodes	IC	MAPs	Message size	frequency
5	100	1	3	2MB	1-12 hours

	Random	MapRoute
DRINC	4.27±0.70	6.18±0.91
Epidemic	4.22±0.47	6.20±0.86
ProphetV2	2.73±0.11	4.38±0.33
SprayAndWait	2.40±0.27	2.93±0.55

Scenario	DTN nodes	IC	MAPs	Message size	frequency
6	100	1	3	2kB	1-12 hours

	Random	MapRoute
DRINC	4.27±0.70	6.18±0.91
Epidemic	4.22±0.47	6.20±0.86
ProphetV2	2.73±0.11	4.38±0.33
SprayAndWait	2.40±0.27	2.93±0.55

Figure 15: Hop Count for the Rural Internet Connectivity scenarios with 100 nodes. The scenarios change the message size between 2 MB and 2 KB

the scenarios. Figure 10 shows delay results for scenarios 1 to 4. And, Figure 11 shows delay results for scenarios 5 and 6. These results show that all protocols have similar performance regarding delay when the movement model is random. SprayAndWait has the best delay and PROPHETv2 has the worst. DRINC is tied in second place with Epidemic.

Figures 12 and 13 show simulation results for messages generated over messages delivered (overhead) for Epidemic, PROPHET, and SprayAndWait routing protocols against our DRINC algorithm. See Table 2 for a summary of the scenarios. Figure 12 shows overhead results for scenarios 1 to 4. And, Figure 13 shows overhead results for scenarios 5 and 6. These results show that in small sized scenarios (Scenarios 1 to 4, with 10 rural nodes) PROPHETv2 has the best overhead when the movement model is MapRoute, and SprayAndWait has the best overhead when the movement model is Random. SprayAndWait has also the best performance in overhead when the scenario has 100 nodes. Our DRINC algorithm has the same overhead as Epidemic.

Figures 14 and 15 show simulation results for the number of hops used to deliver the messages for Epidemic, PROPHET, and SprayAndWait routing protocols against our DRINC algorithm. See Table 2 for a summary of the scenarios. Figure 14 shows hop count results for scenarios 1 to 4. And, Figure 15 shows hop count results for scenarios 5 and 6. These results show that all protocols have a similar performance regarding hop count for small sized scenarios (Scenarios 1 to 4, with 10 rural nodes). SprayAndWait has the best performance regarding hop count when

the scenario has 100 nodes. Our DRINC algorithm has the same hop count as Epidemic.

Results suggest that our proposed algorithm uses more messages to find paths and gather knowledge from the network than other protocols. This can be seen in a higher overhead metric for our DRINC algorithm than it is for other known protocols. DRINC delivers the same or more messages than all protocols used for comparison and it does it in the same time or less, with exception of SprayAndWait that has the best delay performance for nodes using the MapRoute movement model; although, it delivers less messages.

Conclusions and future work

Conclusions

- Delay/Disruption Tolerant Networking is a technically feasible way to bring Internet connectivity to remote rural areas in the world. Simulation results in section 4 show promising results for several DTN routing protocols. They also show that a new protocol, specifically tailored for rural connectivity scenarios, based on the ideas presented in this paper can enhance the performance of current DTN routing protocols, delivering more messages with less delay. To get to a product solution that could be deployed in rural areas, an implementation on an electronic platform should be done. A development with electronics and software will be very useful for tuning to obtain a robust solution.

- The DRINC algorithm presented in Section 3 combines the best characteristics of some of the most known DTN routing protocols. From results in Section 4, it can be seen that our DRINC algorithm performs as well as these protocols. Simulation results show that our DRINC algorithm solution delivers almost all messages when the mobility model given by the simulator allows rural nodes to interact more frequently. When the simulator is configured to use a mobility model that keeps rural nodes apart from each other most of the time, the delivery rate of all protocols drops significantly. This effect is minimized when there are 100 rural nodes, creating more contact opportunities. It can also be seen that in this scenario, all protocols have a very similar performance regarding all metrics. Only Spray and Wait differentiates from the others by delivering less messages with less delay and having less overhead in general.

Future work

This paper provides the basic research to test the feasibility of a technological solution based on DTN technologies for Rural Internet Connectivity for non-real-time applications. Technology development, implementation and testing is necessary before a complete solution can be given. Simulation results for our DRINC algorithm suggest that it improves the delivery rate and delay when compared with well known DTN routing protocols for rural connectivity scenarios; however, more tuning should be done regarding the updating of delivery probabilities, in a way that faster routes or newly discovered routes can be fairly compared with previous ones, reducing the number of hops, and the overhead, in the network. Also, it is necessary to perform tests in real implementations and tune the inner working of the DRINC algorithm in electronic platforms and in real life conditions, taking into account errors and disruptions given by the physical world, and network access' technologies.

Acknowledgment

I would like to thank the Colombian "Departamento Administrativo de Ciencia, Tecnología e Innovación - Colciencias Conv. 528/2011" for the funding of the research leading to this paper. It was also partially funded by:

- Systems and Computing Engineering Department, Universidad de los Andes in Colombia;
- European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 269985.

Bibliography

[1] Velásquez-Villada, C.; Donoso, Y. (2016); Delay/Disruption Tolerant Networks Based Message Forwarding Algorithm for Rural Internet Connectivity Applications, *Computers Communications and Control (ICCCC), 2016 6th International Conference on*, IEEE Xplore, e-ISSN 978-1-5090-1735-5, doi: 10.1109/ICCCC.2016.7496732, 16-22.

[2] La Rue, F. (2011); Report of the Special Rapporteur on the promotion and protection of the right to freedom of opinion and expression Special Procedures of the Human Rights Council, United Nations.

[3] International Telecommunication Union. (2016); Measuring the Information Society. http://www.itu.int/en/ITUD/Statistics/Pages/stat/default.aspx

[4] Velásquez-Villada, C.; Solano, F.; Donoso Y. (2015); Routing Optimization for Delay Tolerant Networks in Rural Applications Using a Distributed Algorithm. *International Journal of Computers, Communications and Control*. 10(1):100-111.

[5] Cerf, V. et al. (2007); RFC4838-Delay-Tolerant Networking Architecture. *IETF RFC 4838*.

[6] Pentland, A. S.; Fletcher, R.; Hasson, A. (2004); Daknet: Rethinking connectivity in developing nations. *Computer*, 37(1):78-83.

[7] Lindgren, A. et al. (2012); Probabilistic routing protocol for intermittently connected networks. Technical report, *IETF RFC 6693*, Experimental.

[8] Keranen, A.; Ott, J.; Karkkainen T. (2009); The ONE simulator for DTN protocol evaluation, *Proc. of the 2nd intl. conf. on simulation tools and techniques*, ICST, art.55, 1-10.

[9] Scott, K.; Burleigh, S. (2007); RFC5050: Bundle protocol specification. *IETF RFC 5050*.

[10] Warthman, F. (2015); Delay and Disruption-Tolerant Networks (DTNs) A Tutorial. Based on Technology Developed by the DTN Research Group (DTN-RG), Version 3.2, 1-35.

[11] Google Inc. (2016); Loon Project. https://www.google.com/loon/

[12] Google Inc. (2016); Link Project. https://www.google.com/get/projectlink/

[13] Facebook Inc. (2016); Internet.org. https://www.internet.org/projects

[14] Choubey, N.; Panah, A. Y. (2016); Introducing Facebook's new terrestrial connectivity systems Terragraph and Project ARIES.

[15] Ministerio de Tecnologías de la Información y las Comunicaciones, Gobierno de Colombia. (2011); Proyecto Nacional de Fibra Óptica.

[16] Watkins, J.; Tacchi, J.; Kiran, M.S. (2009); The role of intermediaries in the development of asynchronous rural access. *In Universal Access in Human-Computer Interaction. Applications and Services*, 451-459. Springer.

[17] Vahdat, A.; Becker, D. (2000); Epidemic routing for partially connected ad hoc networks.

[18] Spyropoulos, T.; Psounis, K.; Raghavendra, C. S. (2005); Spray and wait: an efficient routing scheme for intermittently connected mobile networks. *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, 252-259.

[19] Khabbaz, M. J.; Assi, C. M.; Fawaz, W. F. (2012); Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges. *Communications Surveys & Tutorials, IEEE*, 14(2):607-640.

[20] Cao, Y.; Sun, Z. (2013); Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *Communications Surveys & Tutorials, IEEE*, 15(2):654-677.

[21] Psaras, I.; Wood, L.; Tafazolli, R. (2010); Delay-/disruption-tolerant networking: State of the art and future challenges. *University of Surrey, Technical Report.*

[22] Velásquez-Villada C; Donoso Y. (2016); Delay/Disruption Tolerant Network-Based Message Forwarding for a River Pollution Monitoring Wireless Sensor Network Application. *Sensors*, 16:1-25.

[23] OpenJUMP GIS (2016) http://www.openjump.org/

Author index