INTERNATIONAL JOURNAL

of

COMPUTERS COMMUNICATIONS & CONTROL

ISSN 1841-9836

A Bimonthly Journal
With Emphasis on the Integration of Three Technologies

Year: 2016 Volume: 11 Issue: 6 Month: December

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

C O P E
Member since 2012
JM08090

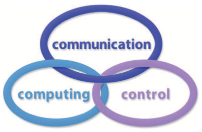
CCC Publications - Agora University

CCC Publications

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

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/

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 11, Issue 6, December, 2016.

Contents

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(6):755-775, December 2016.

Incremental and Decremental SVM for Regression

H. Gâlmeanu, L.M. Sasu, R. Andonie

Honorius Gâlmeanu*
1. Siemens Corporate Technology
honorius.galmeanu@siemens.com
2. Faculty of Mathematics and Informatics
Transilvania University of Braşov
*Corresponding author: galmeanu@unitbv.ro

Lucian Mircea Sasu
1. Faculty of Mathematics and Informatics
Transilvania University of Braşov
lmsasu@unitbv.ro
2. Siemens Corporate Technology
lucian.sasu@siemens.com

Răzvan Andonie
1. Computer Science Department
Central Washington University, Ellensburg, USA
andonie@cwu.edu
2. Electronics and Computers Department
Transilvania University of Braşov

Abstract: Training a support vector machine (SVM) for regression (function approximation) in an incremental/decremental way consists essentially in migrating the input vectors in and out of the support vector set with specific modification of the associated thresholds. We introduce with full details such a method, which allows for defining the exact increments or decrements associated with the thresholds before vector migrations take place. Two delicate issues are especially addressed: the variation of the regularization parameter (for tuning the model performance) and the extreme situations where the support vector set becomes empty. We experimentally compare our method with several regression methods: the multilayer perceptron, two standard SVM implementations, and two models based on adaptive resonance theory.

Keywords: support vector machine, incremental and decremental learning, regression, function approximation

1 Introduction

The approximation of continuous functions that are known only at a certain number of discrete points (also known as regression), a standard procedure in statistics, can be also approached from a machine learning perspective. In the context of supervised training, incremental learning means learning each input-output sample pair, without keeping it for subsequent processing. The topic addressed in this paper is supervised incremental/decremental learning of regression models, in particular SVM models. The fundamental issues in incremental/decremental learning are: *i)* how can a learning system adapt to new information without corrupting or forgetting previously learned information (the *stability-plasticity* dilemma addressed by Carpenter and Grossberg [2]), and *ii)* how can a learning system "forget" information without corrupting previously learned information and without adding new information. In other words, how can a system *learn without unlearning* and *unlearn without learning*.

Support Vector Regressions (SVRs) are SVM learning models used for regression. These algorithms solve the quadratic optimization problem using a decomposition method based on Sequential Minimum Optimization [4], or on an incremental learning method. The incremental method considers that input patterns are added to the solution (or removed from the solution), one at a time, without affecting the learning process of the other patterns [13, 14].

After introducing some basic notations, we aim review in this introductory section related work on incremental/decremental SVRs and also highlight our contribution.

1.1 SVR notations

For a non-linear SVR, regression translates into the following optimization problem [14]:

Approximate a given set of training pairs (x_i, y_i), $i = 1 \dots N$, $x_i \in \mathbb{R}^m$, $y_i \in \mathbb{R}$, with an SVM defined by

$$g(x_i) = \mathbf{w}^T \Phi(x_i) + w_0, \tag{1}$$

where $\Phi(\cdot)$ is a non-linear function and w_0 is an offset. The objective function $g(x) : \mathbb{R}^m \to \mathbb{R}$ requires the minimization [21] of

$$\min_{\mathbf{w}, w_0, \xi_i, \xi_i^*} J(\mathbf{w}) = \min_{\mathbf{w}, w_0, \xi_i, \xi_i^*} \left(\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) \right), \tag{2}$$

and the constraints are given by

$$-\epsilon - \xi_i^* \leq g(x_i) - y_i \leq \epsilon + \xi_i \qquad \text{for} \;\; \xi_i, \xi_i^* \geq 0, \; i = 1 \dots N \tag{3}$$

SVR matches the output value of the objective function $g(x_i)$ as close as possible to the given y_i. The output value is constrained to lay inside a "tube" of values, given by $-\epsilon + y_i \leq g(x_i) \leq \epsilon + y_i$, where ϵ is a given constant. The slack variables ξ_i and ξ_i^* allow the system to cope with patterns that do not fall inside the "tube" and also to minimize the acceptable error for such patterns.

To minimize the expression of $J(\mathbf{w})$ in equation (2) with respect to $\mathbf{w}$ and w_0, after several manipulations, the Lagrangian can be written as:

$$L(\mathbf{w}, w_0, \alpha_i, \alpha_i^*) = \\ -\frac{1}{2} \sum_i \sum_j (\alpha_i^* - \alpha_i) \Phi(x_i)^T \Phi(x_j) (\alpha_j^* - \alpha_j) + \sum_i (\alpha_i^* - \alpha_i) y_i - \sum_i (\alpha_i^* - \alpha_i) w_0 - \sum_i (\alpha_i + \alpha_i^*) \epsilon \tag{4}$$

subject to $\alpha_i, \alpha_i^* \geq 0$. The Karush-Kuhn-Tucker (KKT) conditions (imposed by minimizing the Lagrangian) are given by:

$$\mathbf{w} = \sum_i (\alpha_i^* - \alpha_i) \Phi(x_i) \tag{5}$$

$$\sum_i (\alpha_i - \alpha_i^*) = 0 \qquad 0 \leq \alpha_i \leq C \qquad 0 \leq \alpha_i^* \leq C \tag{6}$$

The Wolfe dual form states that this minimization with respect to $\mathbf{w}$ and w_0 can be transformed into a maximization with respect to α_i and α_i^*. This leads to the minimization of $L'(\mathbf{w}, w_0, \alpha_i, \alpha_i^*) = -L(\mathbf{w}, w_0, \alpha_i, \alpha_i^*)$.

The expression of $L'(\mathbf{w}, w_0, \alpha_i, \alpha_i^*)$ depends only on the inner product $\Phi(x_i)^T\Phi(x_j)$. Therefore, we use the notation $Q_{ij} = \Phi(x_i)^T\Phi(x_j)$. However, SVM uses the "kernel trick": the inner products of the vectors $\Phi(x)$, belonging to an infinite-dimension feature space, are replaced by the non-linear kernel function $K(x_i, x_j) = e^{-\sigma(x_i - x_j)^2} = \Phi(x_i)^T\Phi(x_j)$ [19]. Generally, kernel functions are used for non-linear SVMs, so we use the notation $Q_{ij} = K(x_i, x_j)$ to designate the kernel function.

The extremum conditions for $L'(\mathbf{w}, w_0, \alpha_i, \alpha_i^*)$, considering also the conditions (5, 6), are given by:

$$\frac{\partial L'}{\partial \alpha_i} = -\sum_j Q_{ij}(\alpha_j^* - \alpha_j) - w_0 + y_i + \epsilon - \delta_i + u_i = 0 \tag{7}$$

$$\frac{\partial L'}{\partial \alpha_i^*} = \sum_j Q_{ij}(\alpha_j^* - \alpha_j) + w_0 - y_i + \epsilon - \delta_i^* + u_i^* = 0 \tag{8}$$

with the following additional KKT conditions:

$$\delta_i, \delta_i^* \geq 0 \qquad \delta_i\alpha_i = 0, \quad \delta_i^*\alpha_i^* = 0 \tag{9}$$

$$u_i, u_i^* \geq 0 \qquad u_i(C - \alpha_i) = 0, \quad u_i^*(C - \alpha_i^*) = 0 \tag{10}$$

From equation (1), using the condition (5) and the notation $\alpha_j^* - \alpha_j = \theta_j$, the expression of the "margin" function h_i becomes:

$$h_i = g(x_i) - y_i = \sum_j Q_{ij}\theta_j + w_0 - y_i \tag{11}$$

$$\text{where} \qquad -C \leq \theta_i \leq C \tag{12}$$

From equations (7) - (12), observing in the KKT conditions that at most one of the α_i, α_i^* values is nonzero, it follows that there are three possible vector sets:

- *error vectors* (E), for the following cases:

  $h_i < -\epsilon$ (on $\theta_i = C$), or $h_i > \epsilon$ (on $\theta_i = -C$);

- *support vectors* (S), for the cases:

  $h_i = -\epsilon$ (on $0 < \theta_i < C$), or $h_i = \epsilon$ (on $-C < \theta_i < 0$);

- *other vectors* (O), for the cases where

  $-\epsilon < h_i < \epsilon$, with $\theta_i = 0$.

SVMs can be trained in an incremental/decremental way. Using equation (11), we can decompose h_i into two parts: the contribution of other vectors and the contribution of its own vector:

$$h_i = \sum_{j \neq i} Q_{ij}\theta_j + Q_{ii}\theta_i + w_0 - y_i \tag{13}$$

Given that $Q_{ii} > 0$, an increasing negative contribution of other vectors θ_j (for $j \neq i$) would be compensated by increasing the value for θ_i up to the point where $\theta_i = C$. From this point onward, since the increasing of negative contribution for θ_j would no longer be counteracted by the increase of θ_i, the net result is that h_i would decrease past the $-\epsilon$ value, and x_i would be treated as an error vector.

In equation (13), only the error and support vectors contribute to the expression h_i. For all the sets, the associated threshold value θ_i is bounded by the *regularization parameter* C (equation (12)). $R = \{E \cup O\}$ is the set of *reserve vectors*. We denote by r, s, and e the reserve, support and error vectors, respectively. The whole training set is defined as $\{E \cup S \cup O\}$.

1.2 Related work and our contribution

The *adiabatic* incremental/decremental training algorithm for SVMs (and SVRs) was introduced in [13, 14] and it follows from a method proposed by Cauwenberghs and Poggio [3, 5]. For each pattern being part of the solution (called *support vector*), the associated threshold value is $\theta_i \in \{-\epsilon, \epsilon\}$. Following equation (13), as one new vector is "learned", its associated threshold θ_i increases/decreases, starting from 0, on the expense of other vectors' thresholds. During this process, the rest of the vectors migrate between the sets of support, error and other vectors. When the same vector is "unlearned", its corresponding threshold value θ_i is decreased (when positive) or increased (when negative), in order to reach 0 (when the vector is removed). During unlearning a vector, the vectors which previously migrated between sets when learning that vector will now migrate in the reverse order. An implementation of the adiabatic algorithm can be found in [12], with further developments in [7].

During incremental training, while changing the threshold of the newly introduced vector, the vectors will migrate among sets. The value of the threshold increment determines how many vectors will migrate. Therefore, we are interested to determine the largest threshold increment $\Delta\theta_i$ which is small enough to cause only one vector to migrate. Proceeding with fixed small increments of $\Delta\theta_i$ is not an efficient search technique. We discussed in [7] the conditions for vector migrations between the sets and how to determine the optimal increment. We gave the incremental/decremental context, but we did not describe particular situations, like the empty support vector set situation and the variation of the regularization parameter.

An incremental/decremental SVR, which can simultaneously add batches of new samples and also remove the obsolete ones, was applied for online time series prediction [11]. The ν-SVR, presented in [21], is an incremental/decremental SVR which determines a priori the proportion ν of support vectors from the total set. The algorithms in [9, 11] are variations of the adiabatic algorithm.

For a classification problem, an incremental/decremental SVM operates in the following way: To add vector x_c to the solution, the associated threshold θ_c is set from 0 to C, whereas to remove x_c, θ_c is set to 0. For SVRs, this operations are not symmetric anymore: To add x_c, θ_c is set from 0 to C (increase) or to $-C$ (decrease); to remove x_c, θ_c is also increased or decreased. The direction (incremental or decremental) is not directly given by the intention - adding or removing the vector, but by the margin h_c. We will discuss this aspect in Section 2.

We name our method Incremental/Decremental Support Vector Machine Regression (IDSVMR). It is based on the adiabatic training algorithm. Compared to [13, 14], the IDSVMR contains all aspects needed for an implementation: *a)* We provide the relations that give the exact amount needed for increasing the threshold θ_c for the new vector, before any vector migrates; *b)* We determine the exact threshold variation that leads from one migration to the next. This produces a robust incremental algorithm, that creates a sequence of vector migrations, as part of the learning process; *c)* We deduct the expressions for the variation of the threshold θ_c, from one migration to the next. During training, the support vectors' set may become empty. We give the exact expressions needed to increase θ_c also considering this particular case.

Compared to [7], we contribute with: *i)* a complete description of the increment/decrement operations of the regularization parameter, showing that the expressions to be maximized (minimized) are similar, and *ii)* we expand the results from [12] by giving an exact procedure on how to continue the migration in the empty support vector set cases, for both the regular procedure and the increment/decrement of the regularization parameter.

Section 2 describes the expressions of the threshold variables θ_i and the expression of the vectors margins before the first migration. Section 3 introduces the expressions of the maximum threshold variations that occur before vectors migrations. Section 4 analyzes the maximum variation of the regularization parameter. Section 5 discusses the variations of the free parameter w_0 and the regularization parameter in case of an empty support vector set. Section 6 presents the results of the IDSVMR implemented as a Weka [10] plugin, compared with the multilayer perceptron (MLP), two models based on adaptive resonance theory, and two classical SVM–derived regression models. Finally, Section 6 contains our conclusions.

2 Incremental and decremental updates of thresholds

We start this section with summarizing the adiabatic training algorithm, which is at the core of the IDSVMR method. Then, we will introduce IDSVMR implementation details which are not included in the original adiabatic algorithm.

To adapt to a new vector x_c, the adiabatic algorithm migrates specific vectors between the sets, in order to reestablish the KKT conditions [5]. At the beginning, x_c has the threshold $\theta_c = 0$. The threshold can evolve both toward positive or negative directions. First, this direction is determined. Then, the threshold is modified considering, with each update, the migration between the sets. The migration is identified by observing the maximum (or minimum) variation of the margins h_i for each vector. Considering the KKT conditions, specifically the conditions for the thresholds (6) and margin function (11), their variations can be expressed as:

$$\Delta h_i = \sum_j Q_{ij}\Delta\theta_j + \Delta w_0 \tag{14}$$

$$0 = \sum_j \Delta\theta_j \tag{15}$$

When adding vector x_c to the set, the migrations that take place strive to keep KKT conditions in place. Thus, the variations of thresholds and margins can be further expanded to:

$$\Delta h_i = \sum_j Q_{ij}\Delta\theta_j + \Delta w_0 + Q_{ic}\Delta\theta_c \tag{16}$$

$$0 = \sum_j \Delta\theta_j + \Delta\theta_c \tag{17}$$

These relations can be written more compact using $\mathbf{e}$, the vector of ones:

$$\begin{bmatrix} \Delta h_S \\ \Delta h_R \\ \Delta h_c \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{e}_S & Q_{SS} \\ \mathbf{e}_R & Q_{RS} \\ 1 & Q_{cS} \\ 0 & \mathbf{e}_S^T \end{bmatrix} \begin{bmatrix} \Delta w_0 \\ \Delta\theta_S \end{bmatrix} + \Delta\theta_c \begin{bmatrix} Q_{Sc} \\ Q_{Rc} \\ Q_{cc} \\ 1 \end{bmatrix} \tag{18}$$

The modification of $\Delta\theta_c$ would be absorbed by the modification of $\Delta\theta_S$ (the set of support vectors), Δw_0, and the variation of margin functions. The margins of support vectors do not change: they are either $-\epsilon$ or $+\epsilon$. Hence, we have $\Delta h_S = 0$ and we obtain:

$$\begin{bmatrix} \Delta w_0 \\ \Delta\theta_S \end{bmatrix} = \underbrace{-\begin{bmatrix} 0 & \mathbf{e}_S^T \\ \mathbf{e}_S & Q_{SS} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ Q_{Sc} \end{bmatrix}}_{\beta} \Delta\theta_c \tag{19}$$

Considering the definitions of Δw_0 and $\Delta\theta_S$, the margin functions of the current and reserve vectors are:

$$\begin{bmatrix} \Delta h_c \\ \Delta h_R \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & Q_{cS} \\ \mathbf{e}_R & Q_{RS} \end{bmatrix} \beta + \begin{bmatrix} Q_{cc} \\ Q_{Rc} \end{bmatrix}}_{\gamma} \Delta\theta_c \tag{20}$$

To prevent the recalculation of the inverse matrix in equation (19), the Sherman-Morrison-Woodbury formula was used in [12]. If used repeatedly, this leads to error accumulation. Therefore, we prefer to calculate the inverse by LU decomposition, which is in $O(n^3)$ but numerically stable [8].

3 Learning direction and extremum increment/decrement before vector migration

In what follows, we describe the conditions for vector migration between the sets and establish migration conditions for all possible situations. We explain how we prevent immediate cycling, mentioned in [12]. Although laborious, we give full coverage of all particular situations.

To "learn" a new vector x_c, the training process starts with the initial value $\theta_c = 0$. Its margin, h_c, is first determined using relation (11). Since $\Delta h_c = \gamma_c \Delta\theta_c$, the aim is to modify θ_c such that the margin function is updated to fit as well as possible into the "tube" according to relation (3). The following cases occur:

1. if $h_c > \epsilon$, then the margin decreases to fit into the $-\epsilon < h_c < \epsilon$ "tube" and $\Delta h_c < 0$:
   - if $\gamma_c > 0$, then $\Delta\theta_c < 0$, θ_c decreases to $-C$, and the direction is **decremental**;
   - else, if $\gamma_c < 0$, then $\Delta\theta_c > 0$, θ_c increases to C, and the direction is **incremental**;
2. if $h_c < -\epsilon$, then the margin increases and $\Delta h_c > 0$:
   - if $\gamma_c > 0$, then $\Delta\theta_c > 0$, θ_c increases to C and the direction is **incremental**;
   - else, if $\gamma_c < 0$, then $\Delta\theta_c < 0$, θ_c decreases to $-C$ and the direction is **decremental**;

From these relations we can establish the direction:

- if $sgn(h_c) \neq sgn(\gamma_c)$, the approach is **incremental** and $\Delta\theta_c > 0$;
- if $sgn(h_c) = sgn(\gamma_c)$, the approach is **decremental** and $\Delta\theta_c < 0$.

To "unlearn" a vector x_c from the support or error sets, the direction only depends on the value of θ_c: if $\theta_c < 0$, the direction is **incremental**; if $\theta_c > 0$, the direction is **decremental**.

Once the direction (the variation of θ_c) is established, we have to determine the variation increment $\Delta\theta_c$. We have the following cases (details are provided in Appendix A):

1. Migration of support (S) vectors.

   Considering only the $\Delta\theta_S$ components from equation (19) and following the β notation, the expression of the increment for only one support vector is $\Delta\theta_s = \beta_s \cdot \Delta\theta_c$. It gives the limits for the θ_s updates (we considered β_s to be the specific component from the β vector). On the other hand, considering the allowed variation interval for θ_s, we can determine the limit for θ_c before the support vector associated with θ_s leaves the set:

   - the **incremental** case:
     - if $sgn(h_s) = sgn(\beta_s)$, θ_s may reach 0, so we have the migration to set O:

       $$\Delta\theta_c \leq -\frac{\theta_s}{\beta_s} \qquad (S \rightarrow O)$$

     - if $sgn(h_s) \neq sgn(\beta_s)$, θ_s may reach C or $-C$:

       $$\Delta\theta_c \leq \frac{sgn(\beta_s) \cdot C - \theta_s}{\beta_s} \qquad (S \rightarrow E)$$

   - the **decremental** case:
     - if $sgn(h_s) \neq sgn(\beta_s)$, θ_s may reach 0:

       $$\Delta\theta_c \geq -\frac{\theta_s}{\beta_s} \qquad (S \rightarrow O)$$

     - if $sgn(h_s) = sgn(\beta_s)$, θ_s may reach C or $-C$:

       $$\Delta\theta_c \geq \frac{-sgn(\beta_s) \cdot C - \theta_s}{\beta_s} \qquad (S \rightarrow E)$$

2. Migration of other (O) vectors. These are the vectors with $-\epsilon < h_e < \epsilon$:

   - for the **incremental** case:

     $$\Delta\theta_c \leq \frac{sgn(\gamma_r) \cdot \epsilon - h_r}{\gamma_r} \qquad (O \rightarrow S)$$

   - for the **decremental** case:

     $$\Delta\theta_c \geq \frac{-sgn(\gamma_r) \cdot \epsilon - h_r}{\gamma_r} \qquad (O \rightarrow S)$$

3. Migration of error (E) vectors. These are the vectors with $h_e < -\epsilon$ or $h_e > \epsilon$:

   - for the **incremental** case:
     - if $sgn(h_r) \neq sgn(\gamma_r)$:

       $$\Delta\theta_c \leq \frac{-sgn(\gamma_r) \cdot \epsilon - h_r}{\gamma_r} \qquad (E \rightarrow S)$$

     - if $sgn(h_r) = sgn(\gamma_r)$: vector does not migrate
   - for the **decremental** case:
     - if $sgn(h_r) = sgn(\gamma_r)$:

       $$\Delta\theta_c \geq \frac{sgn(\gamma_r) \cdot \epsilon - h_r}{\gamma_r} \qquad (E \rightarrow S)$$

- if $sgn(h_r) \neq sgn(\gamma_r)$: vector does not migrate

4. Migration of current x_c vector. The current vector x_c newly added to the set is checked for possible migration to the support set (S):

- for the **incremental** case:
  - if $sgn(h_c) \neq sgn(\gamma_c)$:

$$\Delta\theta_c \leq \frac{-sgn(\gamma_c) \cdot \epsilon - h_c}{\gamma_c} \qquad (E \to S)$$

  - if $sgn(h_r) = sgn(\gamma_c)$: vector does not migrate
- for the **decremental** case:
  - if $sgn(h_c) = sgn(\gamma_c)$:

$$\Delta\theta_c \geq \frac{sgn(\gamma_c) \cdot \epsilon - h_c}{\gamma_c} \qquad (E \to S)$$

  - if $sgn(h_c) \neq sgn(\gamma_c)$: vector does not migrate

For removing x_c from the training set, the objective is to increase/decrease θ_c to zero, so that the variation of the margin does not determine vector migration.

5. The variation of θ_c from/to zero should be $-C \leq \theta_c \leq C$, and we have the following conditions:

- for **adding** x_c:
  - for the **incremental** case, $\Delta\theta_c \leq C - \theta_c$
  - for the **decremental** case, $\Delta\theta_c \geq -C - \theta_c$
- for **removing** x_c:
  - for the **incremental** case, where $\theta_c < 0$, $\Delta\theta_c \leq -\theta_c$
  - for the **decremental** case, where $\theta_c > 0$, $\Delta\theta_c \geq -\theta_c$

The maximum variation of $\Delta\theta_c$ for the **incremental** case, before the vectors change sets, is determined by computing the minimum of all these values. Conversely, the minimum variation for the **decremental** case is established by determining the maximum of all these (negative) values deduced in the five cases presented above.

4 Incrementing and decrementing the regularization parameter

Training a SVM also requires at some stage a fine-tuning of the regularization parameter. Many times, this is done by trial-and-error. In order to avoid re-training from scratch, the adiabatic algorithm modifies the regularization parameter in an incremental or decremental way, in small increments, watching for vectors migrations between sets. In the following, we aim to determine the maximum values for these increments.

Starting from equations (14) and (15), and adopting the notation $\theta_k = b_k \cdot C$ for the threshold of error vectors, the relations for the margin functions (when only the regularization parameter is modified) can be written as:

$$\Delta h_i = \sum_{j \in S} Q_{ij} \Delta\theta_j + \left(\sum_{k \in E} Q_{ik} b_k \right) \Delta C + w_0 \qquad (21)$$

$$0 = \sum_{j \in S} \Delta\theta_j + \left(\sum_{k \in E} b_k \right) \Delta C \tag{22}$$

Or, more compact:

$$\begin{bmatrix} \Delta h_S \\ \Delta h_R \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{e}_S & Q_{SS} \\ \mathbf{e}_R & Q_{RS} \\ 0 & \mathbf{e}_S^T \end{bmatrix} \cdot \begin{bmatrix} \Delta w_0 \\ \Delta\theta_S \end{bmatrix} + \begin{bmatrix} \sum_{k \in E} b_k Q_{Sk} \\ \sum_{k \in E} b_k Q_{Rk} \\ \sum_{k \in E} b_k \end{bmatrix} \cdot \Delta C \tag{23}$$

Since the margins for the support vectors remain constant, $\Delta h_S = 0$, the variations of support vectors threshold and the margin variations can be computed as

$$\begin{bmatrix} \Delta w_0 \\ \Delta\theta_S \end{bmatrix} = \underbrace{- \begin{bmatrix} 0 & \mathbf{e}_S^T \\ \mathbf{e}_S & Q_{SS} \end{bmatrix}^{-1} \cdot \underbrace{\begin{bmatrix} \sum_{k \in E} b_k \\ \sum_{k \in E} b_k Q_{Sk} \end{bmatrix}}_{\eta}}_{\beta} \Delta C \tag{24}$$

$$\Delta h_R = \underbrace{\left(\begin{bmatrix} \mathbf{e}_R & Q_{RS} \end{bmatrix} \beta + \sum_{k \in E} b_k Q_{Rk} \right)}_{\gamma} \Delta C \tag{25}$$

Let us summarize the migration conditions for each vector type (details can be found in Appendix B).

1. Migration of support vectors.

   - for **incremental** ($\Delta C > 0$, cumulative conditions):
     - if $sgn(\beta_s + sgn(h_s)) \neq sgn(h_s)$, then the limit is imposed by (migration $S \to E$):
     $$\Delta C \leq \frac{-sgn(h_s) \cdot C - \theta_s}{\beta_s + sgn(h_s)}$$
     - if $sgn(\beta_s) = sgn(h_s)$, then the limit is imposed by (migration $S \to O$):
     $$\Delta C \leq -\frac{\theta_s}{\beta_s}$$
   - for **decremental** ($\Delta C < 0$, cumulative):
     - if $sgn(\beta_s + sgn(h_s)) = sgn(h_s)$, then the limit is imposed by (migration $S \to E$):
     $$\Delta C \geq \frac{-sgn(h_s) \cdot C - \theta_s}{\beta_s + sgn(h_s)}$$
     - if $sgn(\beta_s) \neq sgn(h_s)$, then the limit is imposed by (migration $S \to O$):
     $$\Delta C \geq -\frac{\theta_s}{\beta_s}$$

2. Migration of other vectors.

   For other (O) vectors, $-\epsilon < h_r < \epsilon$:

- for **incremental** ($\Delta C > 0$):

$$\Delta C \leq \frac{sgn(\gamma_r)\epsilon - h_r}{\gamma_r}$$

- for **decremental** ($\Delta C < 0$):

$$\Delta C \geq \frac{-sgn(\gamma_r)\epsilon - h_r}{\gamma_r}$$

3. Migration of error (E) vectors.

   - for **incremental** ($\Delta C > 0$):

     (a) if $sgn(h_r) \neq sgn(\gamma_r)$:

$$\Delta C \leq \frac{-sgn(\gamma_r)\epsilon - h_r}{\gamma_r}$$

     (b) if $sgn(h_r) = sgn(\gamma_r)$, the limit is not active;

   - for **decremental** ($\Delta C < 0$):

     (a) if $sgn(h_r) = sgn(\gamma_r)$:

$$\Delta C \geq \frac{sgn(\gamma_r)\epsilon - h_r}{\gamma_r}$$

     (b) if $sgn(h_r) \neq sgn(\gamma_r)$, the limit is not active;

4. Variation of C is also limited by a target value C_{target}:

   - for **incremental**:

$$\Delta C \leq C_{target} - C$$

   - for **decremental**:

$$\Delta C \geq C_{target} - C$$

When varying C, the maximum variation before the vectors change sets is given by the minimum of these previously described ΔC values for the **incremental** scenario, or as the maximum, for the **decremental** case.

5 Migration when the support vector set is empty

During training process, the support vector set may become empty. In this circumstance, the variations of the free parameter w_0 and the regularization parameter are very particular.

5.1 Regular training when the support vector set is empty

For this case, equations (19) and (20) cannot be used to determine new values for the threshold parameters. Equation (16) can be written considering that there are no support vectors and that the threshold values for the error and other vectors do not change:

$$\Delta h_c = \Delta h_r = \Delta w_0 \tag{26}$$

The variation of the margin for the current vector is the same as the variation for the margins of all vectors. Imposing limits on the allowed change in margins, before some vector change sets, will lead to determine the specific vector that migrates first. Depending on desired direction (increase or decrease of θ_c), we have two cases:

1. if $h_c < -\epsilon$, then $\theta_c > 0$ increases, we have the following limits for margins:
   - for other vector x_o, $-\epsilon < h_o < \epsilon$, $\Delta h_o \leq \epsilon - h_o$;
   - for error vector x_e with $h_e > \epsilon$, the restriction is not active;
   - for error vector x_e with $h_e < -\epsilon$, $\Delta h_e \leq -\epsilon - h_e$;
   - for x_c, $\Delta h_c \leq -\epsilon - h_c$.
2. if $h_c > \epsilon$, then $\theta_c < 0$ decreases, we have the following limits for margins:
   - for other vector x_o, $-\epsilon < h_o < \epsilon$, $\Delta h_o \geq -\epsilon - h_o$;
   - for error vector x_e with $h_e > \epsilon$, $\Delta h_e \geq \epsilon - h_e$;
   - for error vector x_e with $h_e < -\epsilon$, the restriction is not active;
   - for x_c, $\Delta h_c \geq \epsilon - h_c$.

The rules are valid for either the incremental or the decremental case. The margin of the x_c vector is sought to either increase (first situation) or decrease (second).

It is sufficient to take Δw_0 to be the smallest (or largest, depending on the desired direction) of these quantities. This way of varying the margins would always guarantee that some of them would be equal to $-\epsilon$ or ϵ limit, ensuring that the support vector set would not become empty.

5.2 Updating the regularization parameter when the support vector set is empty

When the support vector set is empty, the regularization parameter cannot modify on the expense of the variation of support vectors' threshold values. Relations (21) and (22) will change to:

$$\Delta h_i = \left(\sum_{k \in E} Q_{ik} b_k \right) \Delta C + \Delta w_0 \tag{27}$$

$$0 = \sum_{k \in E} b_k \Delta C \tag{28}$$

Relation (28) is always valid, regardless of C's variation. As opposed to how we proceeded previously, we assume that w_0 parameter is kept unchanged, thus $\Delta h_i = \mu_i \Delta C$, where we made the notation $\mu_i = \sum_{k \in E} Q_{ik} b_k$.

1. when C increases ($\Delta C > 0$, incremental):
   (a) for other vectors $x_r \in O$, $-\epsilon < h_r < \epsilon$:
   - if $\mu_r > 0$ then $\Delta h_r > 0$, thus $\Delta h_r \leq \epsilon - h_r$ or $\mu_r \Delta C \leq \epsilon - h_r$, which leads to $\Delta C \leq \frac{\epsilon - h_r}{\mu_r}$;
   - if $\mu_r < 0$ then $\Delta h_r < 0$, thus $\Delta h_r \geq -\epsilon - h_r$ or $\mu_r \Delta C \geq -\epsilon - h_r$, which leads to $\Delta C \leq \frac{-\epsilon - h_r}{\mu_r}$;

   (b) for error vectors $x_r \; inE$ and $\theta_r = C$, $h_r < -\epsilon$:
   - if $\mu_r > 0$ then $\Delta h_r > 0$, thus $\Delta h_r \leq -\epsilon - h_r$ or $\mu_r \Delta C \leq -\epsilon - h_r$, which leads to $\Delta C \leq \frac{-\epsilon - h_r}{\mu_r}$;
   - if $\mu_r > 0$ then $\Delta h_r < 0$, the condition is not active;

   (c) for error vectors $x_r \; inE$ and $\theta_r = -C$, $h_r > \epsilon$:
   - if $\mu_r > 0$ then $\Delta h_r > 0$, the condition is not active;

- if $\mu_r < 0$ then $\Delta h_r < 0$, thus $\Delta h_r \geq \epsilon - h_r$ or $\mu_r \Delta C \geq \epsilon - h_r$, which leads to $\Delta C \leq \frac{\epsilon - h_r}{\mu_r}$;

In brief, a minimum will be computed among all the thresholds computed for vectors, imposed by conditions like:

$$\Delta C \leq \frac{set \cdot sgn(\mu_r)\epsilon - h_r}{\mu_r} \tag{29}$$

where $set = 1$ if $x_r \in O$ and $set = -1$ if $x_r \in E$. The condition will be inactive if $sgn(\mu_r) = sgn(h_r)$ for $x_r \in E$.

2. when C decreases ($\Delta C < 0$, incremental): Proceeding in the same manner, a maximum will be computed among all the thresholds determined by similar conditions:

$$\Delta C \geq \frac{-set \cdot sgn(\mu_r)\epsilon - h_r}{\mu_r} \tag{30}$$

where set is defined like above. The condition will be inactive if $sgn(\mu_r) \neq sgn(h_r)$, for $x_r \in E$.

6 Experimental results

In our experiments, we compare the IDSVMR with the MLP, two SVM–based regression models, and two incremental learning models derived by us in previous work from adaptive resonance theory: the Fuzzy ARTMAP with Relevance (FAMR) and the Bayesian ARTMAP for Regression (BAR). Since the last two models are less known, we included here a short description of them.

FAMR was introduced in [1], as an extension of the classical Fuzzy ARTMAP (FAM) [15] and PROBART [16]. FAMR can be used as classifier, regression model and posterior probability estimator. Each training pattern can come with its own relevance factor assigned to it, which influences the probabilistic links formed between the input and output categories. A relevance factor allows for ranking of sample pairs according to the confidence one has in the information source. FAMR builds one–to–many mappings between input and output categories through maximum likelihood, approximating the probability of associations between input and output categories. FAMR comes with a stochastic approximation procedure, which allows making use of the relevance factor associated to the training patterns. Under some mild constraints, FAMR's Mapfield values converge both in mean square and with probability one to the posterior probability $P(k|j)$ between the jth input category and the kth output category.

Bayesian ARTMAP (BA) [17] is a neural architecture which uses a combination of FAM competitive learning and Bayesian learning. BA uses Gaussian categories and FAM competitive learning. BA modifies some of the characteristics of the FAM algorithm mainly by replacing the hyperrectangular categories with Gaussian categories. The categories may grow or shrink, and they are probabilistically associated to classes. BA probabilistically infers the classes associated to input categories, by using all input categories, unlike the competitive approach used by FAM and FAMR. In [18], the BA is extended for function approximation. The resulted architecture – Bayesian ARTMAP for Regression (BAR) – generalizes the BA algorithm using the clustering functionality of both input and output modules. The BAR has the universal approximation capability and also the best approximation property; this is very important in regression [18]. Both the FAMR and the BAR can be trained incrementally, but not decrementally. This is typical for FAM models.

For benchmarking, we use the following public datasets [6]: CPU Computer Hardware (CPU), Boston Housing (BH), Wisconsin Breast Cancer (WBC), and Communities and Crime (CC). For the CPU datasets we removed the following input features: vendor name, model name, and estimated relative performance. From the WBC dataset, we removed the four instances with missing values and also the outcome attribute. The first five features of the CC dataset are acknowledged (by the dataset donors) not to be predictive. We removed them, together with the features with missing values.

The original datasets are described in [6]. The filtered datasets, as explained above[1], are synthetically described in Table 1. To support performance comparisons across various benchmarks, both input and output values are independently scaled between 0 and 1.

Table 1: Synthetic description of the four benchmark datasets used. Some instances and/or attributes were removed from the original datasets

Dataset	Input attributes	Instances
CPU	6	209
BH	13	506
WBC	32	194
CC	99	1994

We compare the regression capabilities of IDSVMR, BAR, and FAMR. We are also interested in comparing these incremental models with the popular MLP model and two classical SVM-based regression algorithms, namely ε-SVR [20] and ν-SVR [21]. This is of interest because IDSVMR, ε-SVR and ν-SVR have overlapping inductive bias, and beyond that they cover incremental and non-incremental adaptive models. We include MLP in the group of non–incremental models. Even if is trained with mini–batches of data, the MLP does not learn new information without possibly corrupting previously learned information (the stability-plasticity condition is not addressed).

For FAMR, BAR and IDSVMR, we use our own Weka plugins, whereas for ε-SVR, ν-SVR and MLP we use the Weka–provided implementations [10].

We use ten random permutations (shuffles) of each datasets. For each permutation, 66% of the data is used as training/validation set and the rest serves exclusively as test set. A ten–fold cross-validation on the training/validation set is performed, for fine-tuning the hyperparameters of each model. For MLP, BAR and FAMR, paper [18] enumerates the sought hyperparameters and their corresponding search ranges. For IDSVMR, the hyperparameters are: the regularization coefficient C (shown in equation 2) and σ, the width of the Gaussian kernel[2]. For the CC dataset, we seek an optimal value of C in $\{4, 4.5, \ldots, 8\}$, and σ is sought over the candidate values $\{0.5, 0.7, 0.9\}$. For the other three datasets, C is sought in the set $\{0.2, 0.4, \ldots, 10\}$, and the candidate values for σ are in the set $\{0.5, 0.6, \ldots, 0.9\}$. The same hyperparameters and corresponding search ranges as for IDSVMR are used for both ε-SVR and for ν-SVR.

After cross-validation, we train the optimized models on the whole learning/validation set, and their generalization capability is assessed on the test set. The test set is used solely in this final assessment. The reported scoring values are the root mean squared error (RMSE) and the mean absolute error (MAE). Table 2 contains the average RMSE and MAE values over the ten permutations, measured on the test sets. We split the table into two sections: the former contains only incremental models, the latter is devoted to MLP and to the two SVM–based regression models.

[1] available at http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html

[2] Note that Section 1.1 and the subsequent ones "hide" this hyperparameter under the notation Q_{ij}. Nevertheless, one ought to seek for a proper value of it as well.

The results for MLP, FAMR, and BAR are the ones found in [18]. All the steps for cross-validation, learning, and performance assessment are performed under the Weka Experimenter framework [10].

Table 2: The performance assessment results for ten random permutations of each of the four benchmarks datasets. The results are split into two groups: the former refers the incremental models (FAMR, BAR and IDSVMR), the latter presents the scores for ε-SVR, ν-SVR and MLP. Every cell of the table contains the pair of RMSE and MAE values averaged on the test sets. The boldfaced values are optimal in their corresponding group. The loss values for MLP, FAMR and BAR are the ones reported in [18].

Model	CPU	BH	WBC	CC
FAMR	**0.09**, 0.06	0.15, 0.12	0.31, 0.27	0.21, 0.14
BAR	0.12, 0.06	0.17, 0.12	0.27, 0.23	0.20, 0.14
IDSVMR	0.12, **0.05**	**0.09**, **0.05**	**0.26**, **0.22**	**0.14**, **0.09**
ε-SVR	0.07, 0.05	**0.09**, 0.06	**0.26**, **0.22**	0.16, 0.12
ν-SVR	**0.05**, **0.02**	**0.09**, **0.05**	**0.26**, **0.22**	**0.15**, **0.11**
MLP	0.19, 0.14	0.15, 0.11	0.30, 0.25	**0.15**, 0.12

For all four datasets, in the group of incremental models, the minimum MAE is obtained by IDSVMR. Except for the CPU dataset, the lowest RMSE is produced by IDSVMR; for CPU, IDSVMR produces the median RMSE value, at tie with the BAR. For the BH dataset, IDSVMR largely outperforms the other models. For all four datasets, IDSVMR exhibits a consistent tendency to produce the lowest error scores.

In case of non–incremental models, ν-SVR always obtains the lowest scores. Except for the CPU dataset, the ε-SVR and ν-SVR models have a similar performance. MLP's scores are close to the ones obtained by ν-SVR for CC, while for the other ones MLP shows rather modest performances.

Finally, we compare the incremental and non–incremental models. For CPU, ν-SVR obtains better performance scores than any incremental model. For BH and WBC, ν-SVR is at par with IDSVMR. For CC, IDSVMR outperforms any non–incremental model.

Conclusion

The IDSVMR proves to be a performant and functional implementation of the adiabatic incremental/decremental SVR model. We have described it here with complete implementation details and we have implemented it as a Weka plugin.

In conclusion, the newly–introduced IDSVMR shows itself as a promising incremental regression model, favorably comparing with the other three incremental algorithms in terms of performance. Due to the differences between the hyperparameters number and ranges, it is rather meaningless to consider execution time for the cross-validation stage used for hyperparameter optimization.

SVMs (and SVRs) are known be relatively slow during training. Using incremental/decremental training is therefore very attractive when dealing with large datasets and with data streams.

Bibliography

[1] Andonie, R.; Sasu, L. (2006); Fuzzy ARTMAP with input relevances, *IEEE Transactions on Neural Networks*, 17: 929-941.

[2] Carpenter, G.A.; Grossberg, S. (1988); The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network, *IEEE Computer*, 77–88.

[3] Cauwenberghs, G.; Poggio, T. (2000); Incremental and Decremental Support Vector Machine Learning, *Neural Information Processing Systems*, 409-415.

[4] Chang, C.C.; Lin, C.J. (2001); LIBSVM: a Library for Support Vector Machines, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[5] Diehl, C.P.; Cauwenberghs, G. (2003); SVM Incremental Learning, Adaptation and Optimization, *Proceedings of the IJCNN*, 4: 2685-2690.

[6] Frank, A., Asuncion, A. (2010); UCI Machine Learning Repository, http://archive.ics.uci.edu/ml

[7] Galmeanu, H.; Andonie, R. (2008); Incremental/Decremental SVM for Function Approximation, *Proceedings of the 11th Intl. Conf. on Optimization of Electrical and Electronic Equipment*, 2: 155-160.

[8] Golub, G.H.; Van Loan, C.F. (1996); *Matrix Computations*, JHU Press, Baltimore and London, 1996.

[9] Gu, B.; Wang, J.D.; Yu, Y.; Zheng, G.S.; Yu Fan.; Xu, T. (2012); Accurate on-line v-support vector learning, *Neural Networks*, 27: 51–59.

[10] Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. H. (2009); The WEKA data mining software: an update, *SIGKDD Explorations Newsletter*, 11 (1):10–18.

[11] Karasuyama, M.; Takeuchi, I. (2010); Multiple incremental decremental learning of support vector machines, *IEEE Transactions on Neural Networks*, 21(7): 1048–1059.

[12] Laskov, P.; Gehl, C.; Krüger, S.; Müller, K.R. (2006); Incremental Support Vector Learning: Analysis, Implementation and Applications, *Journal of Machine Learning Research*, 7: 1909–1936.

[13] Martin, M. (2002); *On-line Support Vector Machines for Function Approximation*, Technical Report LSI-02-11-R, Software Department, Universitat Politecnica de Catalunya.

[14] Ma, J.; Thelier, J.; Perkins, S. (2003); Accurate On-line Support Vector Regression, *Neural Computation*, 15(11): 2683–2703.

[15] Carpenter, G. A.; Grossberg, S.; Markuzon, N.; Reynolds, J. H.; Rosen, D. B., (1992), Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Transactions on Neural Networks*, 3: 698-713.

[16] Marriott, S.; Harrison, R. F. (1995), A modified fuzzy ARTMAP architecture for the approximation of noisy mappings, *Neural Networks*, 8(4): 619–641.

[17] Vigdor, B.; Lerner, B. (2007). The Bayesian ARTMAP, *IEEE Transactions on Neural Networks* 18: 1628–1644.

[18] Sasu, L.M.; Andonie, R. (2013); Bayesian ARTMAP for regression, *Neural Networks*, ISSN 0893-6080, 46: 23-31.

[19] Shashua, A. (2009); Introduction to Machine Learning: Class Notes 67577, https://arxiv.org/abs/0904.3664v1

[20] Vapnik, V. (1998); *Statistical learning theory*, New York: Wiley.

[21] Schölkopf, B; Smola, A. J; Williamson, R. C.; Bartlett, P. L. (2000); New support vector algorithms, *Neural computation*, 12 (5): 1207–1245.

Appendix A

Migration of vectors on learning and unlearning

1. Migration of support (S) vectors.

   From equation (19), $\Delta\theta_s = \beta_s \cdot \Delta\theta_c$, the following cases are possible:

   (a) $h_s = \epsilon$, for $-C < \theta_s < 0$:

   i. if **incremental** ($\Delta\theta_c > 0$) and $\beta_s > 0$:
   $\Delta\theta_s > 0$, θ_s can reach 0, $\beta_s \cdot \Delta\theta_c = \Delta\theta_s \leq 0 - \theta_s$, thus $\Delta\theta_c \leq -\frac{\theta_s}{\beta_s}$ $(S \to O)$

   ii. if **incremental** ($\Delta\theta_c > 0$) and $\beta_s < 0$:
   $\Delta\theta_s < 0$, θ_s can reach $-C$, $\beta_s \cdot \Delta\theta_c = \Delta\theta_s \geq -C - \theta_s$, thus $\Delta\theta_c \leq \frac{-C-\theta_s}{\beta_s}$ $(S \to E)$

   iii. if **decremental** ($\Delta\theta_c < 0$) and $\beta_s > 0$:
   $\Delta\theta_s < 0$, θ_s can reach -C, $\beta_s \cdot \Delta\theta_c = \Delta\theta_s \geq -C - \theta_s$, thus $\Delta\theta_c \geq \frac{-C-\theta_s}{\beta_s}$ $(S \to E)$

   iv. if **decremental** ($\Delta\theta_c < 0$) and $\beta_s < 0$:
   $\Delta\theta_s > 0$, θ_s can reach 0, $\beta_s \cdot \Delta\theta_c = \Delta\theta_s \leq 0 - \theta_s$, thus $\Delta\theta_c \geq -\frac{\theta_s}{\beta_s}$ $(S \to O)$

   (b) $h_s = -\epsilon$, for $0 < \theta_s < C$:

   i. if **incremental** ($\Delta\theta_c > 0$) and $\beta_s > 0$:
   $\Delta\theta_s > 0$, θ_s can reach C, $\beta_s \cdot \Delta\theta_c = \Delta\theta_s \leq C - \theta_s$, thus $\Delta\theta_c \leq \frac{C-\theta_s}{\beta_s}$ $(S \to E)$

   ii. if **incremental** ($\Delta\theta_c > 0$) and $\beta_s < 0$:
   $\Delta\theta_s < 0$, θ_s can reach 0, $\beta_s \cdot \Delta\theta_c = \Delta\theta_s \geq 0 - \theta_s$, thus $\Delta\theta_c \leq -\frac{\theta_s}{\beta_s}$ $(S \to O)$

   iii. if **decremental** ($\Delta\theta_c < 0$) and $\beta_s > 0$:
   $\Delta\theta_s < 0$, θ_s can reach 0, $\beta_s \cdot \Delta\theta_c = \Delta\theta_s \geq 0 - \theta_s$, thus $\Delta\theta_c \geq -\frac{\theta_s}{\beta_s}$ $(S \to O)$

   iv. if **decremental** ($\Delta\theta_c < 0$) and $\beta_s < 0$:
   $\Delta\theta_s > 0$, θ_s can reach C, $\beta_s \cdot \Delta\theta_c = \Delta\theta_s \leq C - \theta_s$, thus $\Delta\theta_c \geq \frac{C-\theta_s}{\beta_s}$ $(S \to E)$

2. Migration of other (O) vectors.

   These are the vectors with $-\epsilon < h_e < \epsilon$, $\theta_r = 0$, $\Delta h_r = \gamma_r \cdot \Delta\theta_c$.

   (a) if **incremental** ($\Delta\theta_c > 0$) and $\gamma_r > 0$:
   $\Delta h_r > 0$, h_r can reach ϵ, $\gamma_r \cdot \Delta\theta_c = \Delta h_r \leq \epsilon - h_r$, thus $\Delta\theta_c \leq \frac{\epsilon-h_r}{\gamma_r}$ $(O \to S)$

   (b) if **incremental** ($\Delta\theta_c > 0$) and $\gamma_r < 0$:
   $\Delta h_r < 0$, h_r can reach $-\epsilon$, $\gamma_r \cdot \Delta\theta_c = \Delta h_r \geq -\epsilon - h_r$, thus $\Delta\theta_c \leq \frac{-\epsilon-h_r}{\gamma_r}$ $(O \to S)$

   (c) if **decremental** ($\Delta\theta_c < 0$) and $\gamma_r > 0$:
   $\Delta h_r < 0$, h_r can reach $-\epsilon$, $\gamma_r \cdot \Delta\theta_c = \Delta h_r \geq -\epsilon - h_r$, thus $\Delta\theta_c \geq \frac{-\epsilon - h_r}{\gamma_r}$ $(O \to S)$

   (d) if **decremental** ($\Delta\theta_c < 0$) and $\gamma_r < 0$:
   $\Delta h_r > 0$, h_r can reach ϵ, $\gamma_r \cdot \Delta\theta_c = \Delta h_r \leq \epsilon - h_r$, thus $\Delta\theta_c \geq \frac{\epsilon - h_r}{\gamma_r}$ $(O \to S)$

3. Migration of the error (E) vectors.

   (a) for $h_r > \epsilon$, $\theta_r = -C$, $\Delta h_r = \gamma_r \cdot \Delta\theta_c$:

      i. if **incremental** ($\Delta\theta_c > 0$) and $\gamma_r > 0$:
      $\Delta h_r > 0$, h_r increases even further, vector does not migrate;

      ii. if **incremental** ($\Delta\theta_c > 0$) and $\gamma_r < 0$:
      $\Delta h_r < 0$, h_r may reach ϵ, $\gamma_r \cdot \Delta\theta_c = \Delta h_r \geq \epsilon - h_r$, $\Delta\theta_c \leq \frac{\epsilon - h_r}{\gamma_r}$ $(E \to S)$

      iii. if **decremental** ($\Delta\theta_c < 0$) and $\gamma_r < 0$:
      $\Delta h_r > 0$, h_r increases even further, vector does not migrate;

      iv. if **decremental** ($\Delta\theta_c < 0$) and $\gamma_r > 0$:
      $\Delta h_r < 0$, h_r may reach ϵ, $\gamma_r \cdot \Delta\theta_c = \Delta h_r \geq \epsilon - h_r$, $\Delta\theta_c \geq \frac{\epsilon - h_r}{\gamma_r}$ $(E \to S)$

   (b) for $h_r < -\epsilon$, $\theta_r = C$, $\Delta h_r = \gamma_r \cdot \Delta\theta_c$:

      i. if **incremental** ($\Delta\theta_c > 0$) and $\gamma_r > 0$:
      $\Delta h_r > 0$, h_r may reach $-\epsilon$, $\gamma_r \cdot \Delta\theta_c = \Delta h_r \leq -\epsilon - h_r$, $\Delta\theta_c \leq \frac{-\epsilon - h_r}{\gamma_r}$ $(E \to S)$

      ii. if **incremental** ($\Delta\theta_c > 0$) and $\gamma_r < 0$:
      $\Delta h_r < 0$, h_r decreases even further, vector does not migrate;

      iii. if **decremental** ($\Delta\theta_c < 0$) and $\gamma_r > 0$:
      $\Delta h_r < 0$, h_r decreases even further, vector does not migrate;

      iv. if **decremental** ($\Delta\theta_c < 0$) and $\gamma_r < 0$:
      $\Delta h_r > 0$, h_r may reach $-\epsilon$, $\gamma_r \cdot \Delta\theta_c = \Delta h_r \leq -\epsilon - h_r$, $\Delta\theta_c \geq \frac{-\epsilon - h_r}{\gamma_r}$ $(E \to S)$

4. Migration of the current x_c vector. The current newly added vector x_c is checked for possible migration to the support set (S):

   (a) for $h_c > \epsilon$:

      i. if **incremental** ($\Delta\theta_c > 0$) and $\gamma_c > 0$:
      $\Delta h_c > 0$, h_c increases even further, vector does not migrate;

      ii. if **incremental** ($\Delta\theta_c > 0$) and $\gamma_c < 0$:
      $\Delta h_c < 0$, h_c may reach ϵ, $\gamma_c \cdot \Delta\theta_c = \Delta h_c \geq \epsilon - h_c$, $\Delta\theta_c \leq \frac{\epsilon - h_c}{\gamma_c}$ $(E \to S)$

      iii. if **decremental** ($\Delta\theta_c < 0$) and $\gamma_c > 0$:
      $\Delta h_c < 0$, h_c may reach ϵ, $\gamma_c \cdot \Delta\theta_c = \Delta h_c \geq \epsilon - h_c$, $\Delta\theta_c \geq \frac{\epsilon - h_c}{\gamma_c}$ $(E \to S)$

      iv. if **decremental** ($\Delta\theta_c < 0$) and $\gamma_c < 0$:
      $\Delta h_c > 0$, h_c increases even further, vector does not migrate;

   (b) for $h_c < -\epsilon$:

      i. if **incremental** ($\Delta\theta_c > 0$) and $\gamma_c > 0$:
      $\Delta h_c > 0$, h_c may reach $-\epsilon$, $\gamma_c \cdot \Delta\theta_c = \Delta h_c \leq -\epsilon - h_c$, $\Delta\theta_c \leq \frac{-\epsilon - h_c}{\gamma_c}$ $(E \to S)$

      ii. if **incremental** ($\Delta\theta_c > 0$) and $\gamma_c < 0$:
      $\Delta h_c < 0$, h_c decreases even further, vector does not migrate;

      iii. if **decremental** ($\Delta\theta_c < 0$) and $\gamma_c > 0$:
      $\Delta h_c < 0$, h_c decreases even further, vector does not migrate;

      iv. if **decremental** ($\Delta\theta_c < 0$) and $\gamma_c < 0$:
      $\Delta h_c > 0$, h_c may reach $-\epsilon$, $\gamma_c \cdot \Delta\theta_c = \Delta h_c \leq -\epsilon - h_c$, $\Delta\theta_c \geq \frac{-\epsilon - h_c}{\gamma_c}$ $(E \to S)$

Appendix B

Migration of vectors on varying regularization parameter

1. Migration of the support vectors.

   (a) for $\Delta C < 0$, when C is decreasing (decremental):

   i. for support vector with $0 \leq \theta_s \leq C$, $h_s = -\epsilon$, the variation of C is limited by $0 \leq \theta_s + \Delta\theta_s \leq C + \Delta C$:

   A. for $\beta_s \leq 0$:
   - $\beta_s \cdot \Delta C = \Delta\theta_s \geq 0$, $\theta_s + \Delta\theta_s \geq 0$ is always fulfilled;
   - $\theta_s + \beta_s \cdot \Delta C \leq C + \Delta C$; so $(\beta_s - 1)\Delta C \leq C - \theta_s$ leads to $\Delta C \geq \frac{C-\theta_s}{\beta_s-1}$, $\Delta\theta_s \geq 0$, θ_s can reach C (migration $S \to E$)

   B. for $0 \leq \beta_s \leq 1$, both conditions are active:
   - $0 \leq \theta_s + \beta_s\Delta C$, leading to $\Delta C \geq -\frac{\theta_s}{\beta_s}$, $\Delta\theta_s \leq 0$, θ_s can reach 0 (migration $S \to O$)
   - $\theta_s + \beta_s\Delta C \leq C + \Delta C$, $(\beta_s - 1)\Delta C \leq C - \theta_s$, leading to $\Delta C \geq \frac{C-\theta_s}{\beta_s-1}$, $\Delta\theta_s = \beta_s\Delta C$, θ_s can reach C (migration $S \to E$)

   C. for $\beta_s \geq 1$:
   - $\theta_s + \beta_s\Delta_C \geq 0$ leads to $\Delta C \geq -\frac{\theta_s}{\beta_s}$, θ_s decreases (migration $S \to O$)
   - $\theta_s + \beta_s\Delta C \leq C + \Delta C$, or $(\beta_s - 1)\Delta C \leq C - \theta_s$, is always fulfilled.

   Summary:
   - if $\beta_s - 1 \leq 0$, we have $S \to E$ migration:

   $$\Delta C \geq \frac{C - \theta_s}{\beta_s - 1}$$

   - if $\beta_s \geq 0$, we have $S \to O$ migration:

   $$\Delta C \geq -\frac{\theta_s}{\beta_s}$$

   ii. for support vector with $-C \leq \theta_s \leq 0$, $h_s = +\epsilon$, the variation of C is limited by $-C - \Delta C \leq \theta_s + \Delta\theta_s \leq 0$:

   A. for $\beta_s \leq -1$:
   - $-C - \Delta C \leq \theta_s + \beta_s\Delta C$, or $-C - \theta_s \leq (\beta_s + 1)\Delta C$ is always fulfilled;
   - $\theta_s + \beta_s\Delta C \leq 0$ leads to $\Delta C \geq -\frac{\theta_s}{\beta_s}$, $\Delta\theta_s \geq 0$, θ_s can reach 0 (migration $S \to O$)

   B. for $-1 \leq \beta_s \leq 0$, both conditions are active:
   - $-C - \Delta C \leq \theta_s + \beta_s\Delta C$, leading to $\Delta C \geq \frac{-C-\theta_s}{\beta_s+1}$ (migration $S \to E$)
   - $\theta_s + \beta_s\Delta C \leq 0$ leads to $\Delta C \geq -\frac{\theta_s}{\beta_s}$ (migration $S \to O$)

   C. for $\beta_s \geq 0$:
   - $-C - \Delta C \leq \theta_s + \beta_s\Delta C$, or $-C - \theta_s \leq (\beta_s + 1)\Delta C$, leading to $\Delta C \geq \frac{-C-\theta_s}{\beta_s+1}$ (migration $S \to E$)
   - $\theta_s + \beta_s\Delta C \leq 0$ is always fulfilled.

Summary:

- if $\beta_s \leq 0$, we have $S \to O$ migration:

$$\Delta C \geq -\frac{\theta_s}{\beta_s}$$

- if $\beta_s + 1 \geq 0$, we have $S \to E$ migration:

$$\Delta C \geq \frac{-C - \theta_s}{\beta_s + 1}$$

(b) for $\Delta C > 0$, when C is increasing (incremental):

i. for support vector with $0 \leq \theta_s \leq C$, $h_s = -\epsilon$, the variation of C is limited by $0 \leq \theta_s + \Delta\theta_s \leq C + \Delta C$:

A. for $\beta_s \leq 0$:

- $\theta_s + \beta_s \Delta C \geq 0$, or $\beta_s \Delta C \geq -\theta_s$ leads to $\Delta C \leq -\frac{\theta_s}{\beta_s}$ (migration $S \to O$)
- $\theta_s + \beta_s \Delta C \leq C + \Delta C$ or $(\beta_s - 1)\Delta C \leq C - \theta_s$ is always fulfilled;

B. for $0 \leq \beta_s \leq 1$:

- $\theta_s + \beta_s \Delta C \geq 0$, or $\beta_s \Delta C \geq -\theta_s$ is always fulfilled;
- $\theta_s + \beta_s \Delta C \leq C + \Delta C$ or $(\beta_s - 1)\Delta C \leq C - \theta_s$ is also always fulfilled;

C. for $\beta_s \geq 1$:

- $\theta_s + \beta_s \Delta C \geq 0$, or $\beta_s \Delta C \geq -\theta_s$ is always fulfilled;
- $\theta_s + \beta_s \Delta C \leq C + \Delta C$ or $(\beta_s - 1)\Delta C \leq C - \theta_s$ leads to $\Delta C \leq \frac{C-\theta_s}{\beta_s - 1}$ (migration $S \to E$)

Summary:

- if $\beta_s \leq 0$, we have $S \to O$ migration:

$$\Delta C \leq -\frac{\theta_s}{\beta_s}$$

- if $\beta_s - 1 \geq 0$, we have $S \to E$ migration:

$$\Delta C \leq \frac{C - \theta_s}{\beta_s - 1}$$

ii. for support vector with $-C \leq \theta_s \leq 0$, $h_s = +\epsilon$, the variation of C is limited by $-C - \Delta C \leq \theta_s + \Delta\theta_s \leq 0$:

A. for $\beta_s \leq -1$:

- $-C - \Delta C \leq \theta_s + \beta_s \Delta C$, or $-C - \theta_s \leq (\beta_s + 1)\Delta C$ leads to $\Delta C \leq \frac{-C-\theta_s}{\beta_s + 1}$ (migration $S \to E$)
- $\theta_s + \beta_s \Delta C \leq 0$ or $\beta_s \Delta C \leq -\theta_s$ is always fulfilled;

B. for $-1 \leq \beta_s \leq 0$:

- $-C - \Delta C \leq \theta_s + \beta_s \Delta C$, is always fulfilled;
- $\theta_s + \beta_s \Delta C \leq 0$ is also always fulfilled;

C. for $\beta_s \geq 0$:

- $-C - \Delta C \leq \theta_s + \beta_s \Delta C$ is always fulfilled;
- $\theta_s + \beta_s \Delta C \leq 0$ or $\beta_s \Delta C \leq -\theta_s$ leads to $\Delta C \leq -\frac{\theta_s}{\beta_s}$ (migration $S \to O$)

Summary:

- if $\beta_s + 1 \le 0$, we have $S \to E$ migration:

$$\Delta C \le \frac{-C - \theta_s}{\beta_s + 1}$$

- if $\beta_s \ge 0$, we have $S \to O$ migration:

$$\Delta C \le -\frac{\theta_s}{\beta_s}$$

For migration of support vectors, these conditions are summarized in section 4. Hence, we have (conditions may be cumulative):

- for **incremental** ($\Delta C > 0$):
  - if $sgn(\beta_s + sgn(h_s)) \neq sgn(h_s)$, then the limit is imposed by (migration $S \to E$):

$$\Delta C \le \frac{-sgn(h_s) \cdot C - \theta_s}{\beta_s + sgn(h_s)}$$

  - if $sgn(\beta_s) = sgn(h_s)$, then the limit is imposed by (migration $S \to O$):

$$\Delta C \le -\frac{\theta_s}{\beta_s}$$

- for **decremental** ($\Delta C < 0$):
  - if $sgn(\beta_s + sgn(h_s)) = sgn(h_s)$, then the limit is imposed by (migration $S \to E$):

$$\Delta C \ge \frac{-sgn(h_s) \cdot C - \theta_s}{\beta_s + sgn(h_s)}$$

  - if $sgn(\beta_s) \neq sgn(h_s)$, then the limit is imposed by (migration $S \to O$):

$$\Delta C \ge -\frac{\theta_s}{\beta_s}$$

2. Migration of other vectors.

   For other (O) vectors, $-\epsilon < h_r < \epsilon$, where $\theta_r = 0$; $\Delta h_r = \gamma_r \cdot \Delta C$.

   (a) for $\Delta C < 0$, when C is decreasing (decremental):

   i. for $\gamma_r > 0$: $\Delta\gamma_r < 0$, h_r may reach $-\epsilon$, $\gamma_r \Delta C \ge -\epsilon - h_r$, it leads to $\Delta C \ge \frac{-\epsilon - h_r}{\gamma_r}$ (migration $O \to S$)

   ii. for $\gamma_r < 0$: $\Delta\gamma_r > 0$, h_r may reach ϵ, $\gamma_r \Delta C \le \epsilon - h_r$, it leads to $\Delta C \ge \frac{\epsilon - h_r}{\gamma_r}$ (migration $O \to S$)

   (b) for $\Delta C > 0$, when C is decreasing (decremental):

   i. for $\gamma_r > 0$: $\Delta\gamma_r > 0$, h_r may reach ϵ, $\gamma_r \Delta C \ge \epsilon - h_r$, it leads to $\Delta C \le \frac{\epsilon - h_r}{\gamma_r}$ (migration $O \to S$)

   ii. for $\gamma_r < 0$: $\Delta\gamma_r < 0$, h_r may reach $-\epsilon$, $\gamma_r \Delta C \ge -\epsilon - h_r$, it leads to $\Delta C \le \frac{-\epsilon - h_r}{\gamma_r}$ (migration $O \to S$)

   Resuming the $O \to S$ migration:

- for **incremental** ($\Delta C > 0$):

$$\Delta C \leq \frac{sgn(\gamma_r)\epsilon - h_r}{\gamma_r}$$

- for **decremental** ($\Delta C < 0$):

$$\Delta C \geq \frac{-sgn(\gamma_r)\epsilon - h_r}{\gamma_r}$$

3. Migration of error (E) vectors.

   (a) for error vectors with $h_r > \epsilon$, where $\theta_r = -C$; $\Delta h_r = \gamma_r \cdot \Delta C$:

   i. for $\Delta C < 0$ and $\gamma_r < 0$, or $\Delta C > 0$ and $\gamma_r > 0$, then $\Delta h_r > 0$, h_r increases even further, there is no migration;

   ii. for $\Delta C < 0$ and $\gamma_r > 0$, then $\Delta h_r < 0$, h_r may reach ϵ; $\gamma_r \Delta C \geq \epsilon - h_r$ leads to $\Delta C \geq \frac{\epsilon - h_r}{\gamma_r}$ (migration $E \to S$)

   iii. for $\Delta C > 0$ and $\gamma_r < 0$, then $\Delta h_r < 0$, h_r may reach ϵ; $\gamma_r \Delta C \geq \epsilon - h_r$ leads to $\Delta C \leq \frac{\epsilon - h_r}{\gamma_r}$ (migration $E \to S$)

   (b) for error vectors with $h_r < -\epsilon$, where $\theta_r = C$; $\Delta h_r = \gamma_r \cdot \Delta C$:

   i. for $\Delta C < 0$ and $\gamma_r > 0$, or $\Delta C > 0$ and $\gamma_r < 0$, then $\Delta h_r < 0$, h_r decreases even further, there is no migration;

   ii. for $\Delta C < 0$ and $\gamma_r < 0$, then $\Delta h_r > 0$, h_r may reach $-\epsilon$; $\gamma_r \Delta C \leq -\epsilon - h_r$ leads to $\Delta C \geq \frac{-\epsilon - h_r}{\gamma_r}$ (migration $E \to S$)

   iii. for $\Delta C > 0$ and $\gamma_r > 0$, then $\Delta h_r > 0$, h_r may reach $-\epsilon$; $\gamma_r \Delta C \leq -\epsilon - h_r$ leads to $\Delta C \leq \frac{-\epsilon - h_r}{\gamma_r}$ (migration $E \to S$)

   Resuming the $E \to S$ migration:

   - for **incremental** ($\Delta C > 0$):

     (a) if $sgn(h_r) \neq sgn(\gamma_r)$:

     $$\Delta C \leq \frac{-sgn(\gamma_r)\epsilon - h_r}{\gamma_r}$$

     (b) if $sgn(h_r) = sgn(\gamma_r)$, the limit is not active;

   - for **decremental** ($\Delta C < 0$):

     (a) if $sgn(h_r) = sgn(\gamma_r)$:

     $$\Delta C \geq \frac{sgn(\gamma_r)\epsilon - h_r}{\gamma_r}$$

     (b) if $sgn(h_r) \neq sgn(\gamma_r)$, the limit is not active;

4. Variation of C is also limited by a target value C_{target}:

   - for **incremental**:

   $$\Delta C \leq C_{target} - C$$

   - for **decremental**:

   $$\Delta C \geq C_{target} - C$$

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(6):776-788, December 2016.

Lossless Compression of Data Tables in Mobile Devices using Co-clustering

B. Han, B. Li

Bo Han*, Bolang Li
International School of Software, Wuhan University
37 Luoyu Road, Wuhan, Hubei, China, 430079
*Corresponding author: bhan@whu.edu.cn

Abstract: Data tables have been widely used for storage of a collection of related records in a structured format in many mobile applications. The lossless compression of data tables not only brings benefits for storage, but also reduces network transmission latencies and energy costs in batteries. In this paper, we propose a novel lossless compression approach by combining co-clustering and information coding theory. It reorders table columns and rows simultaneously for shaping homogeneous blocks and further optimizes alignment within a block to expose redundancy, such that standard lossless encoders can significantly improve compression ratios. We tested the approach on a synthetic dataset and ten UCI real-life datasets by using a standard compressor 7Z. The extensive experimental results suggest that compared with the direct table compression without co-clustering and within-block alignment, our approach can boost compression rates at least 21% and up to 68%. The results also show that the compression time cost of the co-clustering approach is linearly proportional to a data table size. In addition, since the inverse transform of co-clustering is just exchange of rows and columns according to recorded indexes, the decompression procedure runs very fast and the decompression time cost is similar to the counterpart without using co-clustering. Thereby, our approach is suitable for lossless compression of data tables in mobile devices with constrained resources.

Keywords: data tables, lossless compression, co-clustering, redundancy.

1 Introduction

Internet of things and mobile Internet churn out huge volume of real-time data in mobile computing devices, such as smartphones, tablets and e-readers. By using ubiquitous mobile sensors, these digital devices constantly collect environment information or persons' behavior information. The information is generally recorded in a structured data table. For example, in a table of a healthcare application, rows represent the records in a sequential sampling time series and columns show the values of multiple measured parameters, such as heart rates. By cumulating records with the time, multi-columns mobile data tables can grow sharply in size, varying from few hundred kilobytes to few hundred megabytes dependent upon different applications. In the Internet context and big data era, these sensor data will generally be sent to cloud servers for further processing in a time frequency.

For a mobile device, a larger size of a data table not only takes more storage space, but its transportation to cloud servers takes more network transmission latencies and more energy costs in batteries. Therefore, lossless compression of data tables is critical to optimize the usage of system resources and thus play an important role in optimizing system performance.

Traditional lossless compression methods view a dataset as a large sequence of strings and employ the occurrence frequency of duplicated symbols to compress them in variable length codes in a dictionary, such as Huffman, Lempel-Ziv, run length encoding and other compression encoders [1]. These methods do not account for dependency patterns in a table and therefore they cannot compress a table with the size less than the limit given by the Shannon entropy [2–4].

Recently, some researchers studied data dependency and exploited the resulted redundancy for compression. Mielikainen et al. applied spatial dependency to propose an adaptive prediction length in a clustered differential pulse code modulation method for lossless compression of hyperspectral data [5]. Venugopal et al. applied Hadamard transformation to eliminate the correlation inside local blocks in medical data [6]. Patauner et al. combined vector quantization, delta calculation and a Huffman coding algorithm together to reduce the correlation among a sequence of data records acquired from pulse digitizing electronics and then apply lossless compression [7]. Kolo et al. proposed an adaptive lossless data compression algorithm for wireless sensor networks [8]. The network data sequence is partitioned into consecutive blocks, and the optimal compression scheme is applied for each block by dynamically analyzing data dependency in a block. They further improved their approach and proposed a fast lossless adaptive compression scheme with low memory requirements for wireless sensor networks [9]. The approach can generate its coding tables on the fly and compress data blocks very fast. Buchsbaum et al. proposed a novel dynamic programming algorithm to discover column dependencies in a table by a one-time, offline learning procedure from a small number of training examples [10]. By exploiting the dependencies, they contiguously partitioned table columns into disjoint groups and found that compressing each group of columns separately can significantly improve compression rates. In their further research, Buchsbaum et al. applied a TSP (traveling salesman problem) tour method to reorder table columns prior to partitioning for further improving compression rates [11]. It provided a unified theory of entropy-like functions to explain both contiguous partitioning and column rearrangement. However, the time cost of the algorithm is expensive and is not linearly proportional to a data table size. Yang et al. studied a transform compression approach for a Boolean matrix [12]. It firstly located the largest columnwise-constant submatrix, and next rearranged columns such that the columnwise-constant is moved to the left-upper corner of the matrix. Following the procedure, the approach recursively applied transformation on the rest of the matrix until the partition resulted in a matrix smaller than a user-defined threshold. However, its running-time cost is also very expensive and the choice of the user-defined threshold is a tricky problem and some improper choices can negatively affect the compressibility.

In a word, the above methods exploited data dependencies and correlations among columns in a table (or called a matrix in mathematics). However, rows in a table also can show dependencies and redundancies. A distinct characteristic of tables is that reordering of columns and rows will not lead to information loss. Therefore, we aim to apply an approach to group dependent or similar columns and rows together, such that the redundancy in grouped homogeneous blocks can be exposed and then we apply compression on these blocks.

Co-clustering is an approach simultaneously clustering of similar rows and columns for revealing hidden structures of a data matrix [13, 14]. It provides great potential for compression. Firstly, co-clustering reorders and groups rows and columns into similar or homogeneous rectangular regions, such that redundant information gets exposed and can be removed by a statistical encoder (such as a Huffman compressor). Next, the inverse transform of co-clustering is very simple and fast. This property will be desirable for uncompressing data. In addition, the time complexity of the algorithm is $O((m+n)kl)$ [15], where m is the number of rows, n is the number of columns, k is the number of row-clusters and l is the number of column-clusters. Since k and l are much smaller than m and n, the algorithm can run in linear time with the size of a data table. In another word, the algorithm is scalable to big size of data tables. Co-clustering has been widely studied for information clustering, pattern structure discovery, et al [16–19].

In this paper, we propose a novel lossless compression approach for data tables. Unlike traditional compression methods by exploiting data dependencies in a single view of either columns or rows, the proposed approach can shape homogeneous rectangular blocks by reordering columns and rows simultaneously via co-clustering. To the best of our knowledge, the proposed approach is the first algorithm that integrates co-clustering and information coding theory for the purpose of lossless compression of data tables. It not only constructs homogeneous blocks by co-clustering, but also optimizes columns/rows alignment in a block to further expose block redundancy, such that the downstream compression with a standard lossless encoder becomes more efficient. We compare empirically the performance of a standard compressor 7Z before and after the application of co-clustering and within-block alignment on a synthetic dataset and several public datasets with properties similar to tables in mobile systems, such as information collected from wearable sensors, clinical care, customer reviews, et al. The extensive experimental results suggest that the proposed approach can effectively improve compression ratios for data tables. In addition, the decompression procedure runs very fast and thus it is very suitable for lossless compression of data tables in mobile devices with constrained resources.

2 Methodology

In this paper, we propose a novel data table compression approach consisting of three steps: 1. reorder table columns and rows by co-clustering; 2. refine table columns and rows alignment to further expose redundancy; 3. compress the resorted data table by a standard compressor.

2.1 Reorder table columns and rows by co-clustering

Given a data table T with m rows and n columns, co-clustering transforms T into another table T' with k row-clusters and l column-clusters (here k and l are smaller than m and n respectively), where each element $b_{ij}(i \in [1,k], j \in [1,l])$ in T' corresponds to a homogeneous two-dimensional block after reordering columns and rows. The adjacent elements in the block have the same or similar values and they provide potentials to boost compression rates. The co-clustering transform can be illustrated in Figure 1 as below,

$$\begin{array}{ccc} T & \longrightarrow & T' \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} & \rightarrow & \begin{bmatrix} b_{11} & \dots & b_{1l} \\ \dots & \dots & \dots \\ b_{k1} & \dots & b_{kl} \end{bmatrix} \end{array}$$

Figure 1: Co-clustering of a data table

In a statistical way, the original table T can be viewed as a joint probability distribution between two discrete random variables denoting rows and columns respectively. Let R and C be such two discrete random variables that take values in the set $\{r_1, \dots, r_m\}$ and $\{c_1, \dots, c_n\}$ respectively. Co-clustering aims to simultaneously quantize R into k disjoint clusters, and C into l disjoint clusters. In other words, co-clustering will generate mappings M_{row} and M_{col} as below,

$M_{row} : R = \{r_1, r_2, \dots, r_m\} \rightarrow \hat{R} = \{\hat{r_1}, \hat{r_2}, \dots, \hat{r_k}\}$
$M_{col} : C = \{c_1, c_2, \dots, c_n\} \rightarrow \hat{C} = \{\hat{c_1}, \hat{c_2}, \dots, \hat{c_l}\}$
The mappings also can be represented in functional forms:$\hat{R} = M_{row}(R)$ and $\hat{C} = M_{col}(C)$.

For obtaining homogeneous blocks in those clusters and further be applied compression, an optimal co-clustering minimizes the loss in mutual information, that is,

$$Minimize \quad I(R;C) - I(\hat{R};\hat{C}) \quad by \quad subject \quad to \quad the \quad given \quad k \quad and \quad l; \tag{1}$$

Here, the mutual information $I(R;C)$ measures the amount of information random variable R contains about C.

To facilitate the search for the optimal co-clustering, the above objective function can be expressed as the following "distance" of $p(R,C)$ to an approximation $q(R,C)$,

$$I(R;C) - I(\hat{R};\hat{C}) = KL(p(R,C)||q(R,C)) \tag{2}$$

Here, $KL(\cdot||\cdot)$ denotes the Kullback-Leibler divergence, $q(R,C)$ is a distribution of the form

$$q(r,c) = p(\hat{r},\hat{c})p(r|\hat{r})p(c|\hat{c}), where \quad r \in \hat{r}, c \in \hat{c} \tag{3}$$

By following Kullback-Leibler divergence, the following equations (4) and (5) can be derived,

$$\begin{aligned} &KL(p(R,C,\hat{R},\hat{C})||q(R,C,\hat{R},\hat{C})) \\ &= \sum\nolimits_{\hat{r}} \sum\nolimits_{r:M_{row}(r)=\hat{r}} p(r)KL(p(C|r)||q(C|\hat{r})) \end{aligned} \tag{4}$$

$$\begin{aligned} &KL(p(R,C,\hat{R},\hat{C})||q(R,C,\hat{R},\hat{C})) \\ &= \sum\nolimits_{\hat{c}} \sum\nolimits_{c:M_{col}(c)=\hat{c}} p(c)KL(p(R|c)||q(R|\hat{c})) \end{aligned} \tag{5}$$

Here, the distribution $q(C|\hat{r})$ can be defined as "row-clustering prototype". Similarly, the distribution $q(R|\hat{c})$ can be defined as "column-clustering prototype". In this way, the above equations show that the objective function in (1) can solely be expressed in terms of row-clustering or column-clustering.

With this intuition, the co-clustering algorithm is listed in Figure 2.

In the first step, co-clustering algorithm starts with an initial mapping function $M_{row}^{(0)}$ and $M_{col}^{(0)}$, and then computes the approximation distributions $q^{(0)}$, including the initial row-cluster prototype $q^{(0)}(C|\hat{r})$. For every column c, the row-cluster is computed as,

$$q^{(t)}(c|\hat{r}) = q^{(t)}(c|\hat{c})q^{(t)}(\hat{c}|\hat{r}) \quad Where, \hat{c} = M_{col}(c). \tag{6}$$

From line 6 to 16, the algorithm keeps iterative computing of row-clusters and column-clusters until a desired convergence condition is satisfied.

Specifically, from line 7 to 9, we re-assign each row r into a row-cluster by using the mapping function $M_{row}^{(t+1)}(r)$. In line 10, the algorithm recomputes the required marginal of $q^{(t+1)}$ based on the updated row-clustering results. It also recomputes the column-cluster prototype by (7). For every row r, we have,

$$q^{(t+1)}(r|\hat{c}) = q^{(t+1)}(r|\hat{r})q^{(t+1)}(\hat{r}|\hat{c}) \quad Where, \hat{r} = M_{col}(r). \tag{7}$$

From line 11 to 13, we re-assign each column c into a column-cluster by using the mapping function $M_{col}^{(t+2)}(c)$, while keeping the row-cluster fixed.

In line 14, the algorithm recomputes marginals of $q^{(t+2)}$ and the row-cluster prototype $q^{(t+2)}(C|\hat{r})$ by (6).

In line 15, we let the iterative variable $t = t + 2$.

The processes from line 6 to 16 are repeated and the row/column clusters are updated until the change of objective function is very small. Actually, the co-clustering algorithm monotonically decreases the objective function given in (2) [15,16] . It grantees an optimal co-clustering results will be achieved by the iterative procedure in the algorithm.

By co-clustering, similar columns/rows are grouped into one cluster. It facilitates coding and provides compression potentials by using a standard compressor.

```
**Algorithm** Co-clustering of a Data Table
**Input:** Table $T$ with a joint probability distribution $p(R, C)$
           The number of row-clusters $k$, The number of column-clusters $l$
**Output:** The cluster mapping functions $M_{row}$ and $M_{col}$
//Initialization:
Let iterative variable $t = 0$
Start with some initial mapping functions $M_{row}^{(0)}$ and $M_{col}^{(0)}$
Compute $q^{(0)}(R|\hat{R})$ , $q^{(0)}(C|\hat{C})$ , $q^{(0)}(\hat{R}, \hat{C})$ and $q^{(0)}(C|\hat{r})(1 \leq \hat{r} \leq k)$ using (6)
//Compute new row-cluster index and column-cluster index by iterative rounds:
repeat do
    for each row $r$ do
        $M_{row}^{(t+1)}(r) = argmin_{\hat{r}} KL(p(C|r)||q^{(t)}(C|\hat{r}))$, $M_{col}^{(t+1)} = M_{col}^{(t)}$
    end for
    Compute $q^{(t+1)}(R|\hat{R})$, $q^{(t+1)}(C|\hat{C})$, $q^{(t+1)}(\hat{R}, \hat{C})$ and $q^{(t+1)}(R|\hat{c})$ $(1 \leq \hat{c} \leq l)$ using (7)
    for each column $c$ do
        $M_{col}^{(t+2)}(c) = argmin_{\hat{c}} KL(p(R|c)||q^{(t+1)}(R|\hat{c}))$, $M_{row}^{(t+2)} = M_{row}^{(t+1)}$
    end for
    Compute $q^{(t+2)}(R|\hat{R})$, $q^{(t+2)}(C|\hat{C})$, $q^{(t+2)}(\hat{R}, \hat{C})$ and $q^{(t+2)}(C|\hat{r})$ $(1 \leq \hat{r} \leq k)$ using (6)
    $t = t + 2$
until convergence of the change in objective function
(that is, $|KL(p(R, C)||q^{(t)}(R, C)) - KL(p(R, C)||q^{(t+2)}(R, C))| < \mu$, $\mu$ is a small threshold)
```

Figure 2: Co-clustering algorithm

2.2 Refine table columns and rows by alignment to further expose redundancy

Next, a series of refinements are applied on a co-clustered table T' to further expose information redundancy. For each column, we compute the minimum value Col_min and their standard derivation value Col_std. If Col_min/Col_std is larger than a threshold α, it shows all elements in the column have similar values. Thereby, by deducting Col_min from all column elements, we can only code their differences more efficiently in a smaller range of numbers. In a column-cluster, we sort columns by their column mean values such that neighborhood columns show the maximum similarity. This will help to expose the redundancy among columns in a cluster. In addition, we compute the standard derivation value for each column to obtain the homogenous status in each column. Next, in a row-cluster, rows are sorted by the most similar columns according to the minimum standard derivation. The above transformation helps form the homogenous blocks. Their same or similar values will facilitate a statistical encoder (such as the run-length coding) in a standard compressor to improve compression effects.

Furthermore, for the same reason, if the number of rows is larger than the number of columns (i.e.$m > n$), we transpose the matrix, such that more number of similar neighbor elements are listed in a row and more redundancy can be exposed.

2.3 Compress the resorted data table by a standard compressor

After the above transformation, we use a statistical encoder for data compression. 7-zip is a popular and fast file archiver with a high compression ratio [20]. We select it since it supports several different data compression file formats and encryption algorithms, such as .7Z format with LZMA algorithm and .GZip format with DEFLATE algorithm [21]. LZMA is a variation of LZ77

```
**Algorithm** Compressing a Data Table
**Input:** a Table T with $m$ rows and $n$ columns
           the number of row clusters $k$, the number of column clusters $l$
           a Threshold $\alpha$, a Standard Compressor $SC$
**Output:** an index $R'$ recording the original order of rows
            an index $C'$ recording the original order of columns
            a Mean Vector $V'$, a Compressed File $F'$
//Co-clustering:
Generate a measure matrix $W$ with $m$ rows and $n$ columns, let all entries in $W$ are $1s$
Setup co-clustering parameters (Options) according to guidance in [15]
Let $[R, C] = co - clusting(T, W, k, l, Options)$
$T' = reorder(T, R, C)$; //Get a Reordered Data Table
// Refine Table Columns and Rows in a co-clustering block:
//If a column has little variance, each element is deducted from the column minimum value
for each column i do
    $Col\_min = min(i^{th}$ column of $T')$; $Col\_std =$standard_derivation($i^{th}$ column of $T'$)
    if $Col\_min/Col\_std > \alpha$ then deduct each element in the ith column of $T'$ by $Col\_min$
    record $Col\_min$ in $V'$
end for
//For those columns in a column cluster,sort them by their mean
for each column-clustering i do
    $Temp = T'(:, C == i)$ //find columns in column cluster i
    $Col\_mean = mean(Temp)$, $Temp = sort(T', Col\_mean)$
    Add sorted column-cluster Temp into new matrix SortT
    Record the updated column index in $C'$
end for
//For those rows in a row cluster, sort them by the column with the minimum std value
$[order\_index] = sort(std(SortT))$
for each row-clustering i do
    $Temp = SortT(R == i, :)$ //find those rows in row cluster i
    //sort rows by the column with the minimum std value
    $Temp = sort(Temp(:, order\_index(1)))$
    //use the reordered rows in row cluster i to form a matrix
    Add sorted row-cluster Temp into new matrix SortT'
    Record the updated row index in $R'$
end for
//Compress the Resorted Table by a Standard Compressor:
$F' = Compress(SC, SortT')$
```

Figure 3: Compression algorithm using co-clustering

algorithm and it uses entropy coding with a Markov chain based range coder and binary trees. DEFLATE is a standard algorithm based on LZ77 and Huffman coding. In the above two steps, our approach applies co-clustering to expose redundancy in a data table and the redundancy can be removed by any of these encoding algorithms. Thereby, we use 7-zip to test our approach.

2.4 Complete compression and decompression algorithm

The details of the compression algorithms are shown in Figure 3.

Accordingly, the decompression algorithm consists of two steps: decompression and reorder back to the original table. Its details are shown in Figure 4,

```
Algorithm Decompressing a File to its Original Data Table
Input: a Compressed File $F'$, a Mean Vector $V'$
       an index $R'$ recording the original order of rows
       an index $C'$ recording the original order of columns
       a Standard Compressor $SC$
Output: The Original Table $T$ with $m$ rows and $n$ columns
// Decompression:
$DT = Decompress(SC, F')$
//Reorder Back to the Original Table:
Reorder rows in $DT$ according to $R'$, Reorder columns in $DT$ according to $C'$
For a column with a $Col\_min$ value in $V'$, add each element in the column with $Col\_min$
```

Figure 4: Decompression algorithm using co-clustering

3 Experimental results

To test the effectiveness of our approach, we designed two experiments. One is the compression on a synthetic dataset. We would like to show how co-clustering can help to reorder and group columns and rows in a data table such that redundancy is exposed. Since compression ratio results are dependent upon different data tables used in mobile applications, we perform the other experiment which is compressing public benchmark datasets. These benchmark datasets are selected from UCI real-life datasets. They cover many typical topics in mobile applications, such as wearable sensors, clinical care, customer reviews, and they have the similar table structures and contents as data in mobile systems. We aim to compare the compression performance before and after applying co-clustering transformation for the benchmark datasets. The co-clustering software we used is Co_Cluster (Version 1.1) [15]. The compression software used is 7-zip. All experiments were performed on a machine with an Intel core i7 2.30GHz processor and 8GB main memory.

For comparing the compression effectiveness, we define a measure called improved compression rate (ICR) based on compression rate (CR) as below,

$$CR = \frac{Original \quad File \quad Size}{Compressed \quad File \quad Size} \tag{8}$$

$$ICR = \frac{CR \quad with \quad Co-clustering \quad - CR \quad without \quad Co-clustering}{CR \quad without \quad Co-clustering} \tag{9}$$

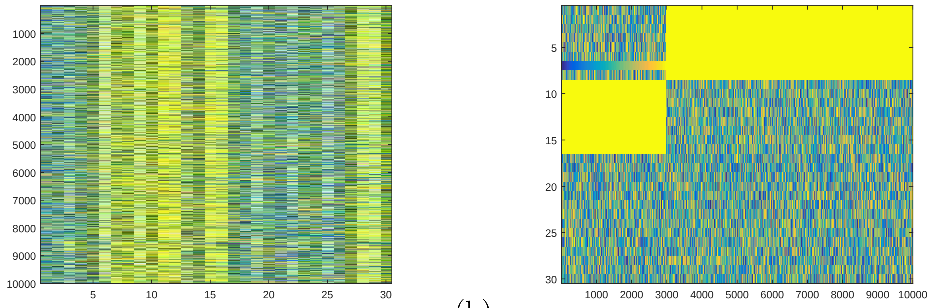
Another two measures are compression time and decompression time.

3.1 Experiment on a synthetic dataset

We firstly generate a data table with 10000 rows and 30 columns. It includes 2 row-clusters and 3 column-clusters. Thereby, the data table is designed with six blocks. Two of them are homogenous (entries values all are 1). The other four blocks contain entries with random values ranging in [0, 1]. By random permutation on rows and columns, the data table mixes six blocks together. The image of the designed data table is shown in Figure 5(a) and its original text file size is 946KB.

In compression algorithm, the parameter m is set to 10000, n is 30, k is 2, l is 3, and OPTIONS vector is set to default values.α is configured with 10.

Figure 5(b) shows the reordered data table images after co-clustering. It illustrates clearly that the two homogenous blocks (or equivalently say, redundancy) are discovered by reordering columns and rows. The homogenous blocks are helpful boost the compression effects by a statistical encoder. In addition, in this case $m > n$, so we transpose the transformed table such that the number of consecutive entries with the same or similar values in a row increases for improving the performance of an encoder. By comparing Figure 5(a) with Figure 5(b), we see that the redundancy information has been well exposed by co-clustering. It has been proved by final compression results. In 7-zip, by setting compression format as 7Z, compression level as maximum and compression algorithm as LZMA, other parameters as default values, the compressed file sizes before and after co-clustering are 268KB and 216KB respectively.



(a) (b)

Figure 5: Synthetic data matrix image: (a) Before co-clustering, (b) After co-clustering and transpose

By (8), CR with Co-clustering and without Co-clustering is 4.38 and 3.53 respectively. Consequently, ICR for the synthetic dataset is 24%, a significant improvement on the original compression rate.

The compression procedure without Co-clustering takes 0.55 seconds, and the counterpart compression procedure with Co-clustering takes 4.77 seconds. For decompression, they both take 0.21 seconds.

3.2 Experiments on ten UCI datasets

We further test our approach on ten real-life benchmark datasets in multiple fields, such as wearable sensors, clinics, ecommerce, et al. They are from the discretized version of UCI Machine Learning Repository [22] and their structures and contents are very similar to data tables in mobile applications. The list of the ten datasets is described in Table 1. For each dataset, we list the number of rows and number of columns in a table, its original file size and its brief description.

Obviously, the ten datasets differ in table rows and columns, from 700 rows to 1000000 rows and from 6 columns to 10000 columns. Subsequently, their file sizes vary from hundreds of KB to tens of MB. The diversified tables are good to test the general compression and decompression performance of our approach.

Table 1: Dataset description

Dataset	Rows*Columns	Original Size	Brief Description
Nursery	12960*8	288KB	Applications for Schools
Waveform	5000*21	304KB	Attributes of waves
ChessKRvK	28056*6	463KB	Chess positions
Gait Freeze	151987*11	6018KB	Data from wearable acceleration sensors
Connect4	67557*42	8775KB	8-ply positions in a game
Diabetes130	101766*48	21482KB	10 years of clinical care data at 130 US hospitals
Arcene_test	700*10000	18560KB	Biomedical Features
Poker_hand	1000000*11	22978KB	A hand of playing cards
Opportunity	51116*250	39022KB	Dataset for Human Activity Recognition
Amazon	1500*10000	29396KB	Customer reviews in a commercial website

In co-clustering settings, m and n are set to the original dimension information of a data table. For simplicity, k is fixed to 4. If a table contains at least 10 columns, l is set to 4, otherwise, l is set to 2. α is configured with 10.

Table 2 compares the compression performance by using LZMA algorithm in 7Z without co-clustering or with co-clustering. In 7Z, the compression level is set to maximum. Other parameters in 7Z are configured with default values. The 3^{rd} and 4^{th} columns in Table 2 (column header "size by 7Z" and "CR by 7Z") show the compressed file sizes and compression rates by using 7Z without co-clustering. And the 5^{th} and 6^{th} columns (column header "size by 7Zco" and "CR by 7Zco") give the compressed file sizes and compression rates by using co-clustering. From the 7^{th} column, we see clearly that co-clustering can significantly improve the compression rates with $ICR \in [21\%, 59\%]$. The 8^{th} and 9^{th} columns compare the compression time difference between 7Z and 7Z with co-clustering. It suggests co-clustering takes more time when a table size increases. The 10th and 11th columns show decompression times between 7Z and 7Z with co-clustering are similar with each other.

For testing the compression performance by using other algorithms in 7Z, Table 3 list the compression results by using GZip algorithm. The compression level is also set to maximum and other parameters are configured with default values. Compared with the results in Table 2, the dataset compression rates by GZip are mostly lower than those by LZMA. However, our co-clustering approach still can improve the compression rates by the range from 23% to 68%.

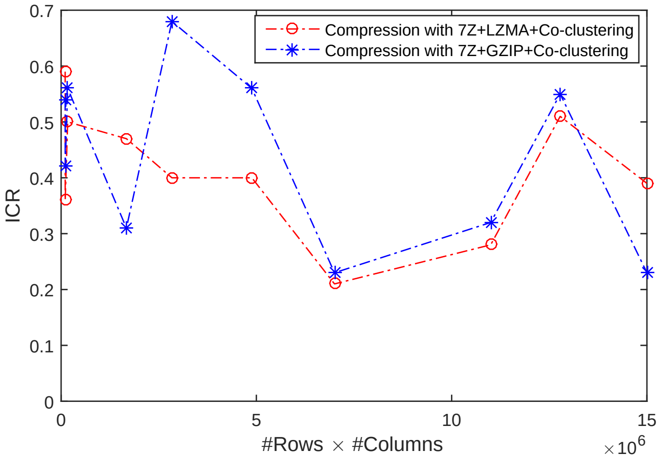


Figure 6: ICRs obtained using co-clustering with LZMA and GZip

Table 2: File compression comparison by using LZMA algorithm in 7Z

Dataset	Original Size (KB)	Size by 7Z (KB)	CR by 7Z	Size by 7Zco (KB)	CR by 7Zco	ICR	CT by 7Z (s)	CT by 7Zco (s)	DT by 7Z (s)	DT by 7Zco (s)
Nursery	288	35	8	22	13	59%	1.88	5.66	0.20	0.20
Waveform	304	30	10	22	14	36%	1.84	3.74	0.23	0.25
ChessKRvK	463	78	6	52	9	50%	3.00	7.13	0.32	0.30
Gait Freeze	6018	1167	5.16	792	7.60	47%	67.64	149.13	0.42	0.41
Connect4	8775	385	23	275	32	40%	46.73	98.70	0.44	0.40
Diabetes130	21482	2177	9.84	1554	13.79	40%	187.69	427.70	3.33	2.85
Arcene_test	18560	5572	3.33	4598	4.04	21%	260.46	539.58	3.81	4.09
Poker	22978	4794	4.79	3749	6.13	28%	444.35	855.87	3.63	3.98
Opportunity	39022	8403	4.64	5573	7.00	51%	522.43	1026.00	4.01	4.56
Amazon	29396	1691	17.38	1220	24.10	39%	170.13	437.01	3.96	4.24

Table 3: File compression comparison by using GZip algorithm in 7Z

Dataset	Original Size (KB)	Size by 7Z (KB)	CR by 7Z	Size by 7Zco (KB)	CR by 7Zco	ICR	CT by 7Z (s)	CT by 7Zco (s)	DT by 7Z (s)	DT by 7Zco (s)
Nursery	288	37	8	24	12	54%	2.00	4.02	0.22	0.23
Waveform	304	34	9	24	13	42%	2.06	3.88	0.24	0.24
ChessKRvK	463	84	6	54	9	56%	2.78	6.24	0.28	0.26
Gait Freeze	6018	1339	4.49	1025	5.87	31%	80.24	178.79	1.78	1.61
Connect4	8775	499	18	297	30	68%	50.01	106.13	2.23	1.74
Diabetes130	21482	2615	8.21	1680	12.79	56%	237.95	490.94	6.71	6.69
Arcene_test	18560	5878	3.16	4792	3.87	23%	307.20	643.88	6.80	6.23
Poker	22978	5342	4.30	4056	5.67	32%	433.88	986.79	7.58	8.27
Opportunity	39022	10256	3.80	6617	5.90	55%	524.55	1124.00	11.51	12.34
Amazon	29396	1908	15.41	1550	18.97	23%	194.45	497.12	8.87	7.42

Figure 6 illustrates that with the increased products of number of rows and columns in the ten tables, the ICRs are waved around 40%. It shows our algorithms keep stable of improved compression rates without much affected by table properties.

Figure 7 visualizes the compression times by using two different compression algorithms in 7Z with co-clustering or without co-clustering. It shows the two algorithms LZMA and GZip have no much difference from compression time view. Except for the last dataset "Amazon", we see that with the product of rows number and columns number increasing, the corresponding compression times increase linearly, without regarding to using co-clustering algorithm. It confirms that the time cost of co-clustering algorithm is just linearly related to the product of rows number and columns number in a table. Thereby, our approach is faster than other approaches where time costs are nonlinearly proportional to a data table size. The last dataset digitalizes customer reviews in Amazon website. By digitalization representation of text reviews, the table is a high dimensional sparse matrix and it contains a lot of consecutive 0s. It leads to a sharp decrease on compression time though it contains a huge number of columns.

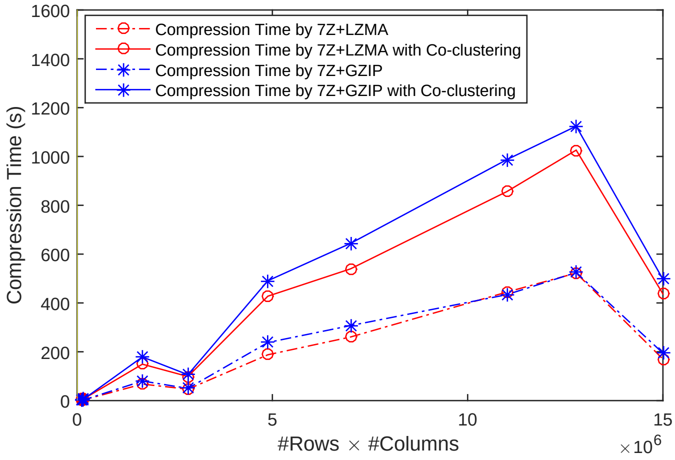


Figure 7: Compression time comparisons

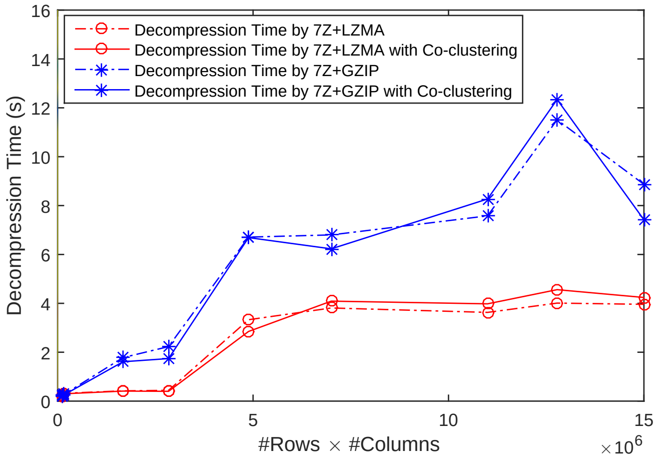


Figure 8: Decompression time comparisons

Figure 8 compares the decompression time costs with different compression settings. It shows LZMA algorithm runs consistently faster than GZIP on decompression for the ten datasets. The co-clustering approaches, either using LZMA or GZIP, take similar time on decompression than those counterparts without using co-clustering. One reason is that the compressed file after co-clustering has a smaller size than the packaged file without co-clustering. In addition, the inverse transform of co-clustering is just reordering rows and columns from recorded indexes in a compression procedure. Thereby, it is very simple and fast in table transformation. By integrating the two factors, we see in Figure 8 that our co-clustering compression approach takes very similar time costs on decompression as original 7Z approach. It is favorable for compressed data tables in a mobile device, which will mostly be sent to a cloud server and be decompressed.

Conclusions

Data tables are widely used in many mobile applications. Lossless compression of a data table can not only reduce storage sizes and network transmission latencies, but also save energy in batteries. In this paper, we propose a novel co-clustering compression approach consisting of three steps: reordering and grouping columns and rows by co-clustering; refine table rows and columns to further expose redundancy; data compression by using a standard compressor. We tested the approach on a synthetic dataset and ten UCI real-life datasets. The experimental results suggest that our approach can significantly improve compression rates at least 21% and up to 68%. The approach is robust both to LZMA and GZip encoders. In addition, the time cost of our approach is linearly proportional to the product of number of columns and rows in a data table. Furthermore, since the inverse transform of co-clustering is just exchange of rows and columns according to records, the decompress procedure is fast and takes similar time on decompression as the original compressor. Therefore, our approach is suitable for compressing a data table in a mobile device where the table requires a limited storage size and reduced communication latency with constrained energy cost.

In next step, we would like to test our co-clustering approach on lossy compression of multimedia files in mobile devices, such as images and videos. We are also interested in incorporating other dimension reduction techniques, such as principle component analysis and independent component analysis, to further improve the effectiveness of multimedia compression in the mobile systems with constrained resources.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grants 61272272 and U1531122, and by the Natural Science Foundation of Hubei province under Grant 2015CFA058.

Bibliography

[1] Rao S., Bhat P. (2015); Evaluation of lossless compression techniques, *IEEE International Conference on Communications and Signal Processing (ICCSP)*, 1655-1659.

[2] Kontoyiannis I., Verdu S. (2014); Optimal lossless data compression: Non-asymptotics and asymptotics, *IEEE Transactions on Information Theory*, 60(2): 777-795.

[3] D. Salomon (2007); *Data Compression: The Complete Reference*, 4th ed. New York: Springer, 2007.

[4] K. Sayood (2000); *Introduction to Data Compression*, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2000.

[5] Mielikainen J., Huang B.(2012); Lossless compression of hyperspectral images using clustered linear prediction with adaptive prediction length, *IEEE Geoscience and Remote Sensing Letters*, 9(6): 1118-1121.

[6] Venugopal D., Mohan S., Raja S.(2016); An efficient block based lossless compression of medical images, *Optik-International Journal for Light and Electron Optics*, 127(2): 754-758.

[7] Patauner C. et al. (2011); A lossless data compression system for a real-time application in HEP data acquisition, *IEEE Transactions on Nuclear Science*, 58(4): 1738-1744.

[8] Kolo J.G. , et al. (2012); An adaptive lossless data compression scheme for wireless sensor networks, *Journal of Sensors*, vol. 2012, Article ID 539638, http://dx.doi.org/10.1155/2012/539638, 1-20.

[9] Kolo J.G. et al. (2015); Fast and efficient lossless adaptive compression scheme for wireless sensor networks, *Computers & Electrical Engineering*, 41: 275-287.

[10] A.L. Buchsbaum et al. (2000); Engineering the compression of massive tables: an experimental approach, *Proceedings of the ACM-SIAM Annual Symposium on Discrete Algorithms*, San Francisco, CA, 213-222.

[11] A.L. Buchsbaum, G.S. Fowler, R. Giancarlo (2002); Improving table compression with combinatorial optimization, *Proceedings of the ACM-SIAMAnnual Symposium on Discrete Algorithms*, San Francisco, CA, 175-184.

[12] Q. Yang, S. Lonardi (2005); A compression-boosting transform for 2D data, *Data Compression Conference (DCC'05)*, 492-492.

[13] H. Shan, A. Banerjee (2008); Bayesian co-clustering, *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08)*, 530-539.

[14] F. Pan, X. Zhang, W. Wang (2008); CRD: fast co-clustering on large datasets utilizing sampling-based matrix decomposition, *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'08)*, 173-184.

[15] A. Banerjee et al. (2007); A generalized maximum entropy approach to Bregman coclustering and matrix approximation, *Journal of Machine Learning Research*, 1919-1986.

[16] I.S. Dhillon, S. Mallela, D.S. Modha (2003); Information-theoretic co-clustering, *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, 89-98.

[17] R.G. Pensa, J.F. Boulicaut, F. Cordero, M. Atzori (2010); Co-clustering numerical data under user-defined constraints, *Statistical Analysis and Data Mining*,3(1): 38-55.

[18] R.G. Pensa, J.F. Boulicaut (2008); Constrained co-clustering of gene expression data, *Proceedings in Applied Mathematics 130, 8th SIAM International Conference on Data Mining 2008*, 1:25-36.

[19] Wu X., Zurita-Milla R., Kraak M.J. (2015); Co-clustering geo-referenced time series: exploring spatio-temporal patterns in Dutch temperature data, *International Journal of Geographical Information Science*, 29(4): 624-642.

[20] http://www.7-zip.org

[21] http://en.wikipedia.org/wiki/7z

[22] F. Coenen, The LUCS-KDD Discretised/normalized ARM and CARM Data Library, In http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS-KDD-DN/DataSets/dataSets.html

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(6):789-803, December 2016.

Analytical Modelling of a New Handover Algorithm for Improve Allocation of Resources in Highly Mobile Environments

Y. Kirsal

Yonal Kirsal*
Electrical and Electronic Engineering
European University of Lefke
North Cyprus, Mersin 10, Turkey
*Corresponding author: ykirsal@eul.edu.tr

Abstract: Wireless and mobile communication systems have evolved considerably in recent years. Seamless mobility is one of the main challenges facing mobile users in wireless and mobile systems. However, highly mobile users lead to a high number of handover failures and unnecessary handovers due to the limited resources and coverage limitations with a high mobile speed. The traditional handover models are unable to cope with high mobile users in such environments. This paper proposes, an intelligent handover decision approach to minimize the probability of handover failures and unnecessary handovers whilst maximizing the usage of resources in highly mobile environments. The proposed approach is based on modelling the system using a Markov chain to enhance the system's performance in terms of blocking probability, mean queue length and transmission delay. The results are compared with the traditional handover model. Simulation is also employed to validate the accuracy of the proposed model. Numerical results have shown that the proposed method outperforms the traditional algorithm over a wide range of handover failures and significantly reduced the number of such failures and unnecessary handovers. The results of this study show that quality if service (QoS) measures of such systems can be evaluated efficiently and accurately using the proposed analytical model. However, the performance results have also shown that it is still necessary to explore an effective model for operational spaces. In addition, the proposed model can also be adapted to various types of networks considering the high speed of the mobile user and the radius of the network.

Keywords: analytical modelling, mobility, handover decision algorithm, quality of service (QoS), highly mobile environments.

1 Introduction

With the rapid development and deployment of wireless technologies, next-generation wireless networks are expected to provide seamless mobility and ubiquitous access to the networks [1,2,4]. Researchers have focused on the improved quality of service (QoS) and performance evaluation of 4G/5G networks in highly mobile environments [6,7,9]. One of the main challenges for seamless mobility in next-generation wireless networks is the availability of the resources in the networks which allow mobile users to roam among heterogeneous environments [4,9]. Mobility between wireless networks may lead to a high number of unnecessary handovers and handover failures when a mobile user is in highly mobile environments [10,11]. The mobile user requires less time to across the coverage area of the network when the speed increases. Thus, the mobile user does not have enough time to acquire the network resources to do the handover. In addition, mobile users can join the queue by requesting resources from the system. However, mobile users will never get access to a channel for communication due to the user's velocity as well as the lack of resources. Hence, the queue will have unnecessary handovers by allowing such users to access the system. Unnecessary handover occurs when the mobile user's travelling time within the system is less than the handover process from the neighbouring networks to the system. Thus, the mobile user leaves the network coverage area before the handover process is executed [2, 9, 10]. This causes network connection breakdown [11] and interrupts the service [10]. Unnecessary handover is undesirable because it wastes network resources [7, 10]. On the other hand, if the user's call holding time is equal to or less than the total time of handover into and out of the system, then handover failure occurs [11]. In this situation, the mobile user does not transmit or receive any data packet to system; however, the mobile user might enter the system just after triggering the handover process. In other words, the system will be available after a short time when the mobile user triggers the handover process.

Many network characteristics (such as power consumption, received signal strength, and network conditions) affect the on handover process in wireless and mobile systems. Most existing decision algorithms have focused on the vertical handover process, such as optimization problems [12], a policy-enabled schemes [13, 14], and fuzzy logic [15]. On the other hand, in [16] the main focus is various mathematical models used in vertical handover decisions for heterogeneous networks. However, most previous works consider vertical handover decision algorithms. More recently, the literature has proposed various handover decision mechanisms [1–6] but none of the existing works consider user's the high mobility in the handover decision procedure. In [1] two-dimensional (2-D) Markov queuing models have been constructed to enhance multiple service requirements in the LTE network. The proposed models decrease the call blocking rate, especially for handover users. However, the mobility issues are not considered in [1]. The authors in [2] demonstrated their overall approach by describing the VANET Testbed in vehicular environments. The results obtained in [2] showed that it is necessary to consider a new handover model based on a probabilistic rather than traditional approach. In addition, proactive queueing approach for handover is also considered in [2], but the parameters (e.g., two channels in the systems) and scenario used for the analysis are rather simple in order to obtain realistic QoS results in highly mobile environments. In [4] a simple and robust two-step vertical handover decision algorithm is proposed for wireless and mobile networks. The new call blocking probability of the proposed model is modelled as M/M/B_i/B_i based on the Erlang-B model. However, [5] showed that the Erlang-B model is not suitable for the handover schemes of highly mobile environments due to user's mobility in the system. In addition, as the reserved bandwidth may not be utilized effectively in low handover rates, the traditional reservation-based schemes are not efficient for future networks, especially in 4G vehicular networks [16]. In [17] the time before the vertical handover and the network dwell time are calculated and presented for any network topology. However, both parameters have not been used in previous studies to improve unnecessary handovers and handover failures.

In order to achieve seamless handover, when the mobile user has a high speed, it is important to predict the network availability, the time before handover and network dwell time with coverage boundaries. The Time before handover is the time after which the handover occurs, and the network dwell time is the time that the mobile user spends in the coverage area of the network. These two parameters are important in order to obtain the best time and place for the handover for mobile users. It is also possible to improve resource allocation in mobile and wireless systems by using these two parameters. Traditional handover models have been studied for wireless and mobile systems [1,3–5,8,9] but they lead to the degradation of the QoS due to the network's small coverage area and the velocity of the users, especially in highly mobile environments. Hence, the new handover approach is necessary for providing ubiquitous communication in next-generation systems. An analytical modelling and performance evaluation of mobile and wireless system using queueing theory has recently been performed [1–6, 8, 9]. Modelling limited resources and enhancing the QoS of the systems are the basis of the queuing phenomenon. Thus, the queuing theory can be used to model and analyse such problems [1, 7, 9]. Therefore, developing an analytical model of a new handover approach and resource allocation model for such systems would be the best option for obtaining more efficient QoS measurements.

This paper develops a new analytical model for handover and resource allocation in highly mobile environments based on the time the mobile user needs to acquire network resources for the handover. The main contribution of the paper can be summarized as follows:

- The QoS degradation of the traditional approach due to the handover failures and unnecessary handovers can be improved by the proposed algorithm considering call holding time, mobile user dwell time and time before handover.
- The performance improvement of high mobile users and management of the high mobile users within the cell and/or between neighbouring cells can be based on the acceptance factor.
- The results show that the proposed method gives better performance results than the traditional approach. However, a statistical model is still necessary to predict the degree of contention in highly mobile environments.

The main purpose of this paper is to develop a useful analytical model based on time before handover, call holding time and network dwell time by using a new decision algorithm to improve the QoS of real networks. The analysis done in this paper is based on modelling the system using a Markov chain to enhance the system performance in terms of blocking probability, mean queue length and transmission delay. This proposed model is applicable for most wireless communication systems. The rest of the paper is organised as follows: Section II presents the traditional handover approach. Section III describes the proposed handover approach. Section IV discusses the performance evaluation of the proposed algorithm with the traditional approach and simulation results. Finally, Section V provides the conclusions and future works.

2 The traditional approach

This section explains and represents the traditional handover approach for wireless and mobile environments. Figure 1 shows the handover process in such environments. The traditional handover approach introduces two types of threshold circles in the coverage area of the system [3, 8, 9, 17]. The handover threshold and the exit threshold circles are shown in Figure 1 for the traditional handover approach.

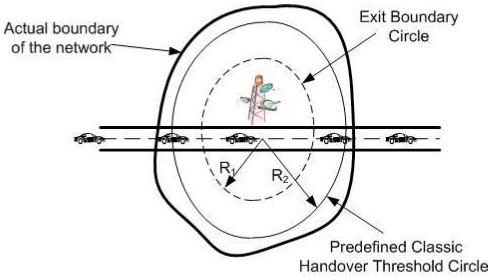

Actual boundary of the network
Exit Boundary Circle
R_1
R_2
Predefined Classic Handover Threshold Circle

Figure 1: The traditional handover approach with threshold circles

Based on the traditional approach, the cell can be divided into different regions depending on the radius. The continuous circle with radius R_2 represents the handover threshold circle. In addition, the dotted circle with R_1 represents the exit threshold circle as shown in Figure 1. The exit threshold circle is the starting point for the the handover process in the traditional handover approach. In order to process successful mobility, mobile users have to finish the handover before reaching the classic handover threshold circle. If the mobile user is not handed over successfully before the circle, the mobile user will lose the connection. The traditional handover approach is currently being used in [1, 3–5, 8, 9] and [17] for wireless and mobile systems.

2.1 Traditional handover queueing model

In mobile and wireless communication networks, the service is provided by the base station (BS) and/or access point(AP) depending on the network. Mobile users communicate via radio links with BSs/APs [3, 5]. A single and arbitrary shape of cell is assumed. There are S channels that the system can provide for the service. In addition, the length of the queue is Q. The maximum number of calls allowed into the system is a combination of the number of users being served (S) and the number of users in the queue (Q). Hence, the maximum number of calls in the system is given by L where $L = S + Q$. The traditional handover queueing model is given in Figure 2 [3, 8, 9].

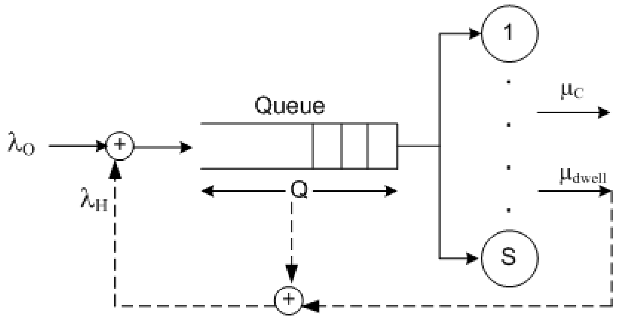


Figure 2: Traditional handover queueing model

The originating calls and handover calls are two kinds of arrival rates defined in the system, with mean arrival rates given as λ_O and λ_H, respectively. λ_O is newly generated calls in the system and λ_H is the mobile calls from one cell to another. If the channels are available and idle in the system, both call's arrival can be assigned to any channel. Otherwise, the incoming call request is added to the queue if the channels are busy [3,8,9]. In addition, if the queue is full, the incoming call is blocked. The channel requests in the system are served by first in first out (FIFO) rule. The inter-arrival times of the incoming call requests are assumed to follow an exponential distribution. λ is defined as the total arrival rate of calls in the cell, where $\lambda = \lambda_O + \lambda_H$. In traditional handover, the mobile users are moving at a velocity V, and there is a probability that it can also leave the network when being served due to the mobility. Moreover, a mobile user is placed in the queue waiting for the channel to be served. However, the mobile user can leave the system due to the mobility shown in Figure 2. A formula is given in [3] and [9] for λ_H. In traditional handover, T_C is call holding time in the system. An exponentially distributed T_C with a mean rate of μ_C is assumed. In addition, T_{dwell} is the dwell time, indicating the time that mobile users spend in the cell. This is also assumed to be exponentially distributed with a mean rate of μ_{dwell}. The equation 1 is used in the literature for the dwell time in wireless and mobile systems [3, 8, 9] for traditional handover queuing model. Thus, μ_{dwell} can be calculated and described as follows:

$$\mu_{dwell} = \frac{E[V] \cdot L}{\pi \cdot A} \tag{1}$$

where $E[V]$ is the average of the random variable, V is the speed of mobile users, L is the length of the perimeter of cell (a cell with an arbitrary shape is assumed), and A is the area of the cell [3,8,9]. The total channel holding time of a call is exponentially distributed with mean $1/\mu$ where, $\mu = \mu_C + \mu_{dwell}$. The state transition diagram of the traditional handover queueing model is shown in Figure 3. The states are defined as i (i=0,1,2,$\cdots$,S+Q) the number of calls in the system at time t.

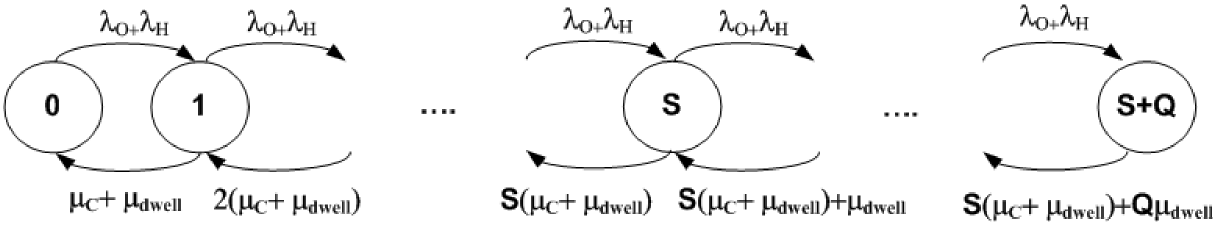


Figure 3: The state diagram of the traditional handover queueing model

ρ is the traffic intensity in the system, where ρ=λ/μ. Assuming a system in a steady state, the state probabilities, P_i's, can be obtained as in equation 2 [3,8,9].

$$P_i = \begin{cases} \frac{(\lambda_O+\lambda_H)^i}{i!} \cdot P_0 & 0 \le i \le S \\ \frac{\frac{(\lambda_O+\lambda_H)^S}{S!} \cdot (\lambda_O+\lambda_H)^{i-S} \cdot P_0}{\prod\limits_{j=S+1}^{i} [S\mu+(j-S)\mu_{dwell}]} & S < i \le S+Q \end{cases} \tag{2}$$

In equation 2, P_i is the probability that there are i calls in the system. P_0 can be defined as follows:

$$P_0 = \left[\sum_{i=0}^{S} \frac{(\lambda_O + \lambda_H)^i}{i!} + \sum_{i=S+1}^{S+Q} \frac{\frac{(\lambda_O+\lambda_H)^S}{S!} \cdot (\lambda_O + \lambda_H)^{i-S}}{\prod\limits_{j=S+1}^{i} [S\mu + (j - S)\,\mu_{dwell}]} \right]^{-1} \tag{3}$$

Once all the steady state probabilities P_i are computed, the rest of the performance measures can be easily obtained. More information about the traditional handover queueing model can be found in [3,8,9].

3 The proposed approach

This section presents an abstract intelligent handover algorithm for high mobile environments applying queuing theory. The accurate knowledge of network availability, coverage boundaries (radius of the cell) and the velocity of mobile users are fundamental factors that play an important role in correct decision making during the handover. Hence, in the proposed model, the proposed algorithm determines the time that a mobile user needs before performing a handover. As mentioned in the previous sections, the proposed scheme is based on the current time $T_{current}$, network dwell time T_{dwell} and estimated time before handover $T_{estimated}$ of the mobile users. In order to reduce the number of unnecessary handovers and handover failures, the proposed algorithm determines the required time for the mobile user whether, admitting it into the system or performing a handover as shown in Figure 4. Assuming a mobile user is moving at a velocity V towards the system at $T_{current}$, the user can request a channel for communication. The user needs a channel at $T_{estimated}$ and releases the channel at $(T_{estimated} + T_{dwell})$. Based on the call holding time (T_C) of the calls, three possible conditions are proposed and analysed in Figure 4. Hence, the proposed approach can improve the resource allocation, especially in highly mobile environments.

- First condition: $(T_{estimated} + T_{dwell})_{n-1} < (T_{estimated})_n$
  If the channel needs time of the current user (n) is higher than the channel release time of the user being served (n-1), then the mobile user can enter the system seamlessly. This shows that the mobile user has enough time to get a place in the system to be served. In other words, unnecessary handovers $(T_{current} + T_{dwell}) < (T_{estimated})$ and partial handovers $T_C \leq (T_{dwell} + T_{estimated})$ do not occur.

- Second condition: $(T_{estimated})_{n-1} < (T_{estimated})_n$ and $(T_{estimated} + T_{dwell})_{n-1} < (T_{estimated} + T_{dwell})_n$
  For the second condition, the users are currently using the channels or waiting in the queue to be served. If the channels' release time of the users and/or waiting time in the queue (n-1) are higher than the channel release time of the current user (n), then the system will be partially busy by the time the current user reaches the system. This means that the current user might be admitted to the system after a short time. In other words, the system will be available soon for the service after the current user requests a channel. Hence, there is a partial contention $T_C \leq (T_{dwell} + T_{estimated})$ in the system.

- Third condition: $(T_{estimated})_{n-1} < (T_{estimated})_n$ and $(T_{estimated} + T_{dwell})_{n-1} > (T_{estimated} + T_{dwell})_n$
  If the channel release time of the users being served (n-1) is greater than the channel release time of the current user (n), then the system will be busy during the travel of the current user. Hence, the current user will never get access to the system. The channels and queue will no longer be available and the mobile user will be handed over to another network. Thus, the mobile user leaves the network coverage area before the handover process is executed $(T_{current} + T_{dwell}) < (T_{estimated})$ [11]. This causes a network connection breakdown [11] and interrupts the service [10].

In summary, in the event of the third condition, current mobile users will never join the system. The unnecessary handovers occur due to the high speed of the user as well as the radius of the network. Thus, the proposed algorithm passes the mobile user to the next available network via the acceptance factors. When the first condition is identified, the system (channels plus queue) can be used by the mobile user. In addition, when the second condition is identified and notified before the current user reaches the system, the contention can be signalled and the mobile user might be passed to other available networks nearby instead of waiting for the service. This approach should result in better network performance.

3.1 The proposed handover queuing model

In the proposed approach, the decision algorithm decides whether the mobile user will be admitted to the system based on the analysis described above section (see Figure 4). It is clearly seen that the proposed algorithm ensures that mobile users do not wait and leave the system unserved because of mobility. In other words, all mobile users will be allowed into the system depending on the analysis. Otherwise, the mobile user at a high speed towards the system will not have enough time to enter the system. Hence, the mobile users move at a high speed will be handed over to the another available network. Thus, mobile users do not wait long and leave the system without service. The proposed handover queueing model is shown in Figure 5.

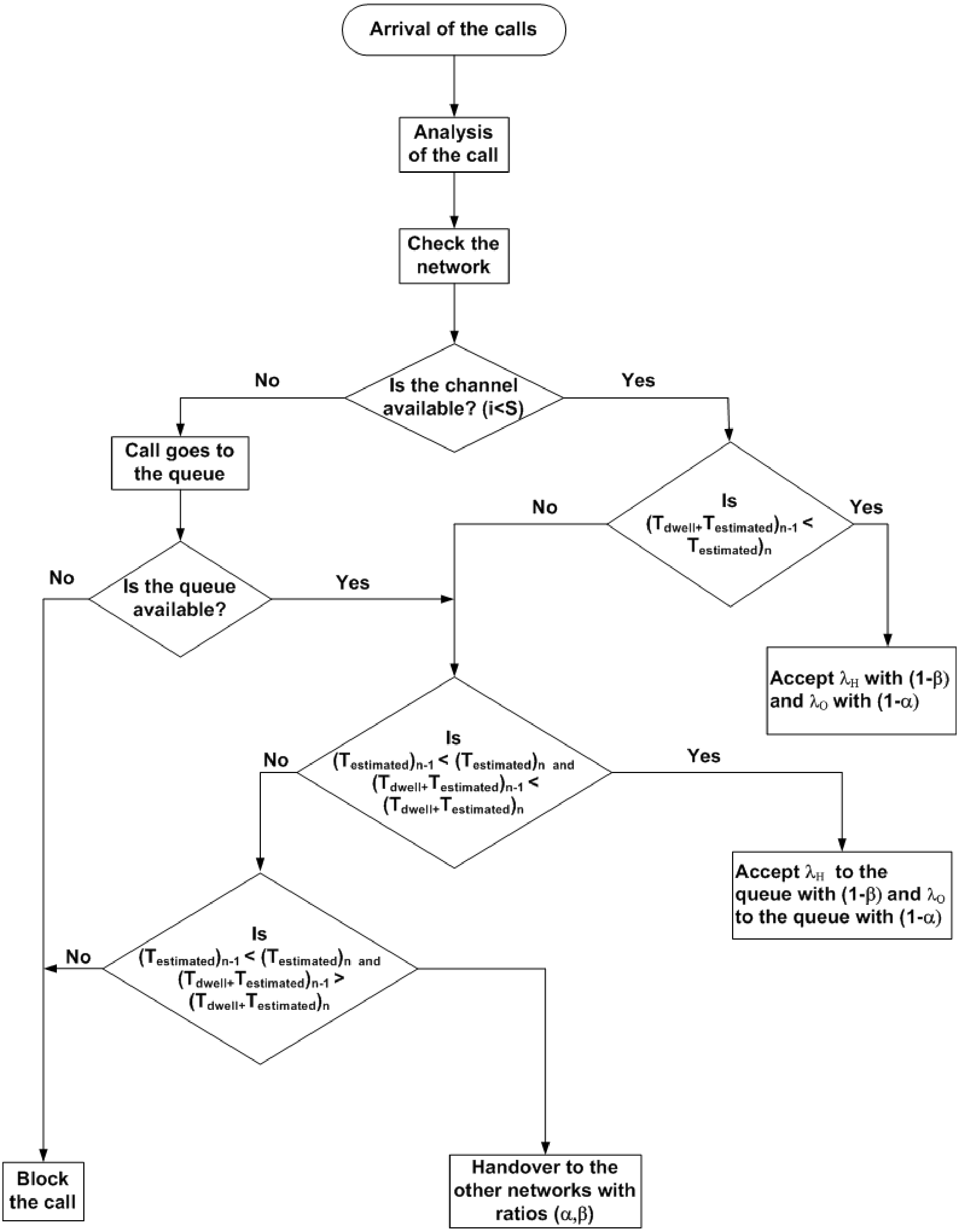

Arrival of the calls
Analysis of the call
Check the network
Is the channel available? (i<S)
No
Yes
Call goes to the queue
Is the queue available?
No
Yes
Is $(T_{dwell}+T_{estimated})_{n-1} < T_{estimated})_n$
No
Yes
Accept λ_H with (1-β) and λ_O with (1-α)
Is $(T_{estimated})_{n-1} < (T_{estimated})_n$ and $(T_{dwell}+T_{estimated})_{n-1} < (T_{dwell}+T_{estimated})_n$
No
Yes
Accept λ_H to the queue with (1-β) and λ_O to the queue with (1-α)
Is $(T_{estimated})_{n-1} < (T_{estimated})_n$ and $(T_{dwell}+T_{estimated})_{n-1} > (T_{dwell}+T_{estimated})_n$
No
Block the call
Handover to the other networks with ratios (α,β)

Figure 4: The proposed handover decision algorithm

The proposed handover queueing model (similar to the traditional handover) considers S number of channels and can allow i requests at time t as shown in Figure 5. The queueing capacity of the system is Q. The arriving requests may be sent from different users to the

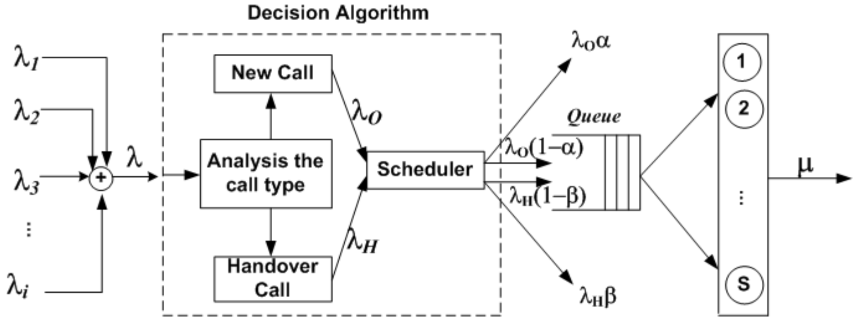


Figure 5: The proposed handover queueing model

system. Hence, the inter-arrival time of consecutive requests follows the Poisson process which can be distributed as an exponential distribution with arrival rate λ. According to [3], for a two-dimensional fluid model, the arrival rate of handover calls can be obtained as follows:

$$\lambda_H \approx \frac{\mu_{dwell}}{\mu_c}\lambda_O \tag{4}$$

The decision algorithm distinguishes the calls (λ_O/λ_H) and decides to send them into the system depending on the acceptance factors. α and β are the acceptance factors of the originating calls and handover calls, respectively. For the purpose of the proposed analytical model, α and β are taken as constant. It is assumed that the originating calls can join the system with an arrival rate of λ_O (1-α). Similarly, the handover calls can join the system with an arrival rate of λ_H (1-β). Hence, the total arrival rate is $\lambda = \lambda_O(1-\alpha)+\lambda_H$ (1-β). As the requests are rejected from entering the system, especially into the queue, the queue can be treated as a normal queuing system. Hence, the service rate is $\mu = \mu_C + \mu_{dwell}$.

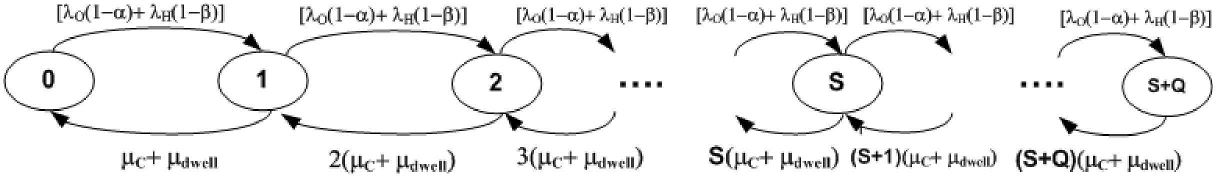

$[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]$
$[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]$
$[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]$
$[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]$
$[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]$
$[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]$
0
1
2
....
S
....
S+Q
$\mu_C+\mu_{dwell}$
$2(\mu_C+\mu_{dwell})$
$3(\mu_C+\mu_{dwell})$
$S(\mu_C+\mu_{dwell})$
$(S+1)(\mu_C+\mu_{dwell})$
$(S+Q)(\mu_C+\mu_{dwell})$

Figure 6: The state diagram of the proposed handover model

It is clearly seen that the M/M/C/K queuing model is fit for the proposed model for the performance evaluation. Thus, the proposed system can be illustrated by the given one-dimensional Markov chain as shown in Figure 6.

Let's define the states i (i=0,1,2,$\cdots$,S+Q) as the number of calls in the system at time t. The arrival rate can be taken as constant for all requests regardless of the number of users in the system. Hence, the arrival rate is the birth rate in the proposed model and can be obtained as [λ_O (1-α) + λ_H (1-β)]. In contrast, the rate of service completions in the proposed scheme depends on the number of calls in the system based on the analysis. If there are S or more requests in the system, then all S channels are busy. As each channel services users at the rate $\mu_C + \mu_{dwell}$, the combined service rate for the system is $S(\mu_C + \mu_{dwell})$. If there are fewer than S requests in the system, $i < S$, only i of the S channels are busy and the combined service rate for the system is $i(\mu_C + \mu_{dwell})$, as shown in Figure 6. Hence μ_i can be calculated as follows:

$$\mu_i = \begin{cases} i(\mu_C + \mu_{dwell}) & 0 \le i < S \\ \\ S(\mu_C + \mu_{dwell}) & S \le i \le S+Q \end{cases} \tag{5}$$

Assuming the system is in a steady state, then using the well-known birth and death process, the steady state probabilities P_i can be obtained and are given in Equation 6:

$$P_i = \begin{cases} \frac{[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]^i}{i!(\mu_C+\mu_{dwell})^i} \cdot P_0 & 0 \leq n < S \\ \\ \frac{[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]^i}{S^{i-S}S!(\mu_C+\mu_{dwell})^i} \cdot P_0 & S \leq i \leq S+Q \end{cases} \tag{6}$$

In order to find P_0 the normalization condition used since the probabilities must sum to 1, which gives:

$$P_0 = \left[\sum_{i=0}^{S-1} \frac{[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]^i}{i!(\mu_C+\mu_{dwell})^i} + \sum_{i=S}^{S+Q} \frac{[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]^i}{S^{i-S}S!(\mu_C+\mu_{dwell})^i} \right]^{-1} \tag{7}$$

The average number of packets in the system, MQL can then be calculated as $MQL = \sum_{i=0}^{S+Q} i \cdot P_i$ which gives:

$$MQL = \left[\sum_{i=0}^{S-1} i \frac{[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]^i}{i!(\mu_C+\mu_{dwell})^i} + \sum_{i=S}^{S+Q} i \frac{[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]^i}{S^{i-S}S!(\mu_C+\mu_{dwell})^i} \right] \cdot P_0 \tag{8}$$

Similarly, the blocking probability P_B can be calculated as:

$$P_B = P(S+Q) = \frac{[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]^{S+Q}}{S^Q S!(\mu_C+\mu_{dwell})^{S+Q}} \cdot P_0 \tag{9}$$

In addition, the average queue length L_Q is:

$$L_Q = \sum_{i=S+1}^{S+Q} (i-S) \cdot P_i \tag{10}$$

Using Little's formula, the mean waiting time of channel requests in the queue can be calculated as follows:

$$E[T_w] = \frac{L_Q}{(1-P_B)[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]} \tag{11}$$

Hence, the average value of time of a call in a cell is:

$$E[T_s] = \frac{MQL}{(1-P_B)[\lambda_O(1-\alpha)+\lambda_H(1-\beta)]} \tag{12}$$

Let us define MQL_H as the mean number of handovers per user during its lifetime which can be calculated as:

$$MQL_H = \frac{MQL_H E[Tw] E[Tw]}{E[Ts]} \tag{13}$$

Therefore, the mean transmission delay of packet is calculated:

$$E[T_D] = MQL_H E[Tw] \tag{14}$$

4 Performance evaluation

This section presents numerical results in order to show the accuracy and effectiveness of the proposed analytical model of the new handover algorithm to improve resource allocation in highly mobile environments. In addition, the results obtained from the solution of the traditional and proposed approaches are validated by using discrete event simulation (DES). The simulation tool is mainly used for the validation of both the traditional and proposed models. As the simulation simulates the actual scenario rather than the Markov models presented in this paper, it can also be used for the performance evaluation of such systems. The DES developed considers the stochastic processes for all types of mobile users' arrivals and departures. Mobile users' arrivals and departures occur one at a time in a random, discrete, event-triggered fashion when an arrival enters the system and service is completed, respectively. In addition, the users waiting in the system are served based on first come first serve (FCFS) basis in the order of their arrival. The channel (and/or channels) becomes idle or remains busy with requests stored in the queue when the service event is completed. While a particular event is handled, the next event is generated. The results obtained from the simulation runs are within the 5% confidence interval with a 95% confidence level [9]. The simulation model was adopted for the scenario considered and implemented in C++ language. In order to validate the proposed analytical model, the results obtained from the analytical model and the simulation results for different performance measures are presented and compared. The numerical study focuses on MQL, P_B and transmission delay of the proposed models. The mean arrival and mean service rates are mainly application dependent. The assumptions in [3, 5, 7, 8] and [9] are employed in this paper as well for consistency, unless stated otherwise.

4.1 Key parameters

The system parameters used are mainly taken from [3, 5, 7, 8] and [9] based on the relevant literature [1, 2, 4, 6, 10–13, 16]. The system has a fixed number of identical channels: S=16. Q is the queuing capacity which represents the number of packets waiting for service. It is assumed that the moving direction of the mobile users can be detected by the BS/AP using a control channel. In addition, a mixed traffic pattern is also assumed, as in [2] where on average a minimum of 2 slots are 0.5 ms. Hence, the rates are translated into packet per second in order to use consistent values. The service rate of the mobile users μ_{dwell} is calculated using Equation 1. The requests are handed over or rejected from entering the system due to the proposed analysis; thus, the arrival rate is $\lambda = \lambda_O(1-\alpha) + \lambda_H(1-\beta)$. However, in this paper α is taken as 0.01 because λ_H passing through in a unit time with a high speed is larger than λ_O. In other words, λ_O calls are assumed to be allocated by the system as they are newly generated in the system. The analysis of α could be explored in future work.

4.2 Results

The Figures 7 and 8 show MQL and P_B results, respectively, as a function of the originating calls λ_O in the system. The parameters are S=16, Q=50, $E[V]$=40m/s (144km/hr), R=1000m, $E[T_c]$=120 packets/sec, and α=0.01 and the λ_O rate per user varies from 0.01 packets per second.

The figures clearly show that the proposed approach works far better than the traditional approach. In the traditional approach, due to the high mobile users, most users will leave the system without being served. In addition, the handover calls from the neighbour cells will request channel allocation at the same time, especially for heavy traffic loads (e.g., λ_O=0.08). This causes an increase in MQL as well as the P_B of the system. Thus, β has an impact on the

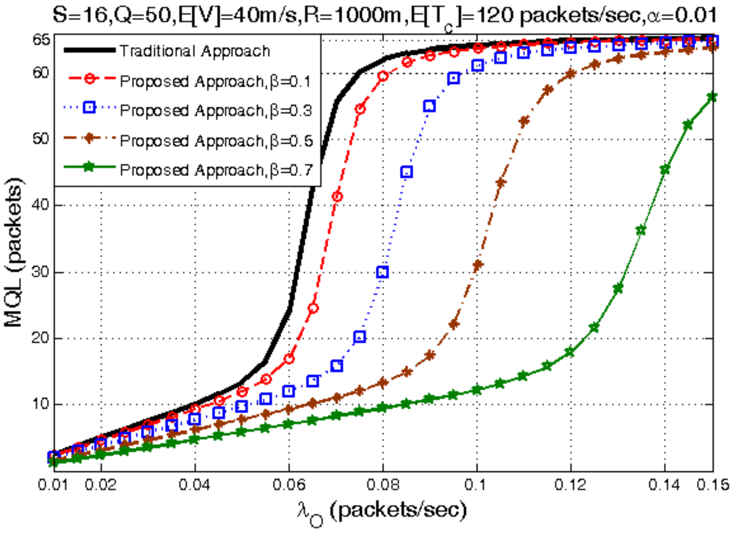


Figure 7: Mean queue length results as a function of originating calls λ_O with different β values

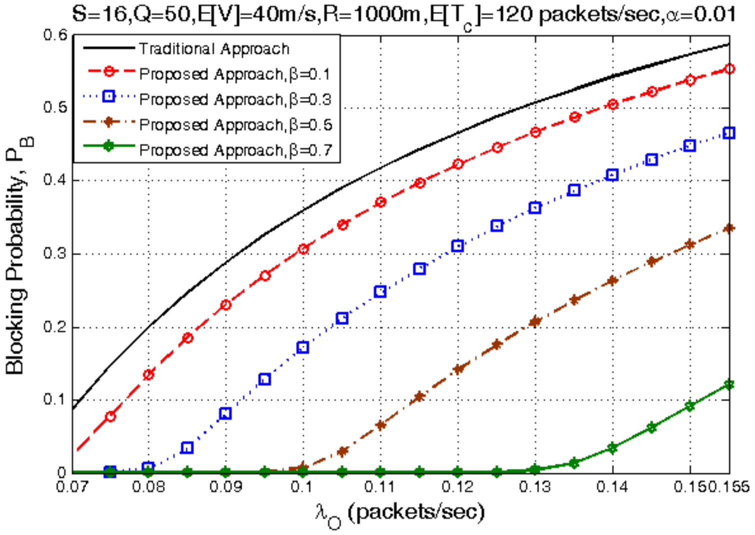


Figure 8: Blocking Probability results as a function of originating calls λ_O with different β values

system. It is clear from the figures that β significantly affects the system performance. Hence, β is an important parameter for the handover management in highly mobile environments.

Figure 9 shows transmission delay as a function of the originating calls λ_O with different β values. In wireless and mobile networks, transmission delay is another important QoS parameter criterion. It can be clearly seen that transmission delay increases rapidly for the traditional handover when λ_O increases due to the number of unnecessary handovers allowed in the system. Such handovers will leave the system without being served. The system is then busy with unnecessary handovers and the transmission delay increases. It can be observed from the graphs in Figures 7, 8 and 9 that the proposed approach gives better QoS results in terms of mean calls in the system, blocking probability and transmission delay, respectively, when β increases. This means that, according to the acceptance factor, highly mobile users are handed over to the neighbour cell and/or served without wasting the network resources.

Table 1 illustrates blocking probability results as a function of queue size. Parameters used

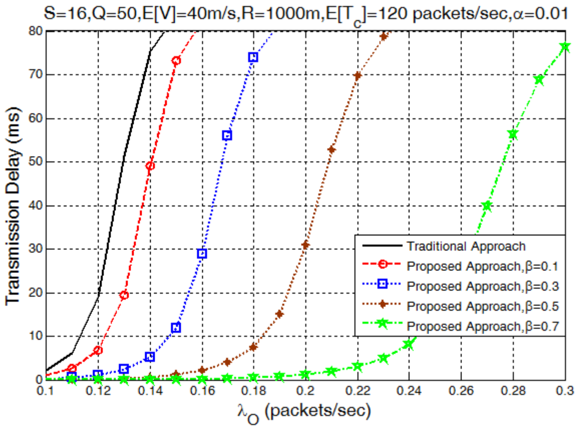


Figure 9: Transmission delay as a function of originating calls λ_O with different β values

Table 1: Blocking Probability results as a function of queue size (Q)

Q	Traditional Approach	Proposed Approach, $\beta = 0.2$	Proposed Approach, $\beta = 0.4$	Proposed Approach, $\beta = 0.6$	Proposed Approach, $\beta = 0.8$
30	0.089398	0.005699	1.57E-05	3.01E-09	1.35E-14
50	0.0863125	0.001211	6.89E-08	9.82E-14	6.58E-22
70	0.0858134	0.000269	3.03E-10	3.20E-18	3.20E-29
90	0.0857308	0.0000601	1.33E-12	1.04E-22	1.56E-36

for Figures 7, 8 and 9 are used for the results presented in Table 1 as well. The parameters are as follows: S=16, λ_O=0.1, $E[V]$=40m/s, R=1000m, $E[T_c]$=120 packets/secs and α=0.01. The blocking probability decreases slightly in the traditional handover because (especially for a loaded system) highly mobile users make the system busy due to the unnecessary handover as well as the handover failures. However, this is not the case when the proposed algorithm is employed. The blocking probability decreases rapidly when Q increases. This means that the proposed approach can handle unnecessary handovers and handover failures. In other words, the proposed approach gives better resource usage than the traditional approach.

On the other hand, MQL results as a function of originating calls for low mobile users are given in Figure 10. The results show that the proposed model performs better than the traditional approach when the system utilisation ($U = \lambda/S\mu$) is less than 0.72. However, as the velocity decreases, the traditional approach outperforms the proposed approach in some situations, especially for a heavy-loaded system (e.g., U=0.88). This is mainly because at such low velocity no one leaves the system due to the mobility. Then, large MQL results are experienced in the proposed system when $\beta = 0.1$ and 0.3. However, the proposed model gives better results when higher values of β are considered (i.e., $\beta = 0.7$). In addition, even at low velocity, most of the mobile users leave the queue without being served in the traditional approach.

The numerical results obtained from the proposed model are also validated by the simulation in Table 2 and Figure 11. The parameters used in Table 2 and Figure 11 are the same parameters used in Figures 7 and 8. Table 2 shows the P_B results of the traditional and proposed approaches with different β. It is obvious in Table 2 that numerical results obtained from the proposed model show agreement with the results obtained from the simulation as the discrepancies are less than 5%. The numerical results show the effectiveness of the proposed model. The MQL results for both approaches as a function of λ_O are shown in Figure 11 and validated by simulations. The results of the proposed analytical approach and simulation results show good agreement. The maximum discrepancy between the analytical model and simulation is 3.42% which is well within

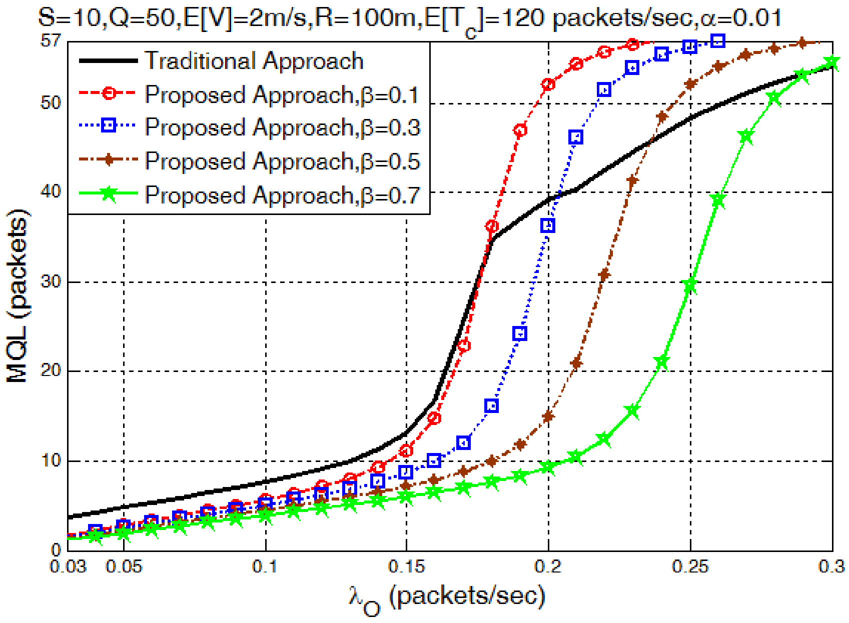


Figure 10: Mean queue length results as a function of originating calls λ_O with different β values for low mobile users

the 5% confidence interval of the simulation.

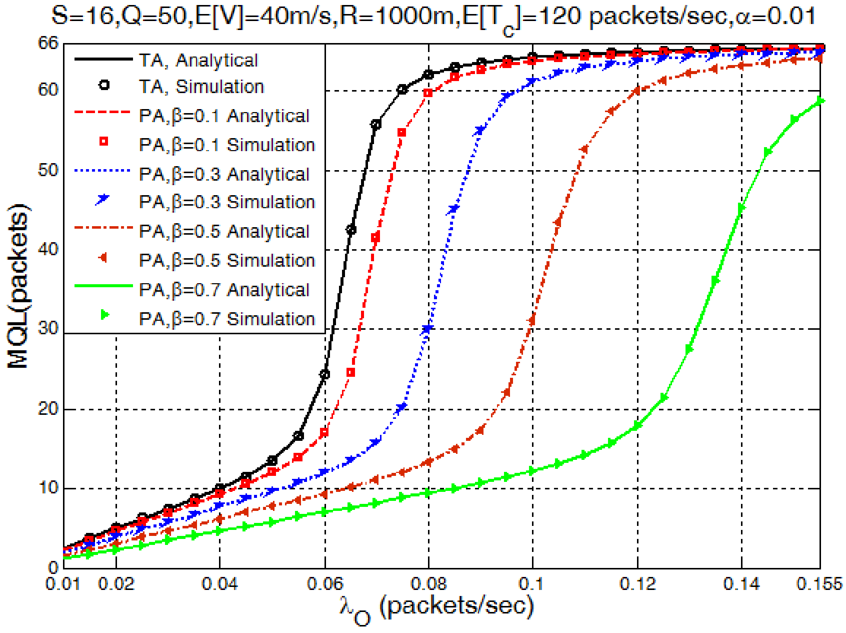


Figure 11: The analytical and simulation MQL results of both traditional approach (TA) and proposed approach (PA) as a function of originating calls λ_O with different β values

Table 2: Validation of P_B results as a function of λ_O for both traditional and proposed approaches. (D is Discrepancy).

P_B,Traditional Approach				P_B,Proposed Approach β= 0.3			P_B,Proposed Approach β =0.5		
λ_O	Analytical	Simulation	D(%)	Analytical	Simulation	D(%)	Analytical	Simulation	D(%)
0.085	0.2471	0.2490	0.78	0.0335	0.0340	1.40	1.20E-12	1.20E-12	0.04
0.09	0.2889	0.2900	0.38	0.0805	0.0810	0.57	2.89E-11	2.89E-11	0.02
0.095	0.3263	0.3180	2.62	0.1282	0.1240	3.42	5.63E-10	5.62E-10	0.05
0.1	0.3600	0.3610	0.28	0.1718	0.1750	1.85	9.06E-09	9.05E-09	0.07
0.105	0.3905	0.3900	0.12	0.2112	0.2130	0.84	1.22E-07	1.22E-07	0.03
0.0.11	0.4182	0.4120	1.50	0.2471	0.2400	2.94	1.39E-06	1.39E-06	0.22
0.115	0.4435	0.4430	0.11	0.2798	0.2790	0.28	1.34E-05	1.35E-05	0.51
0.12	0.4667	0.4660	0.14	0.3098	0.3090	0.26	1.09E-04	1.10E-04	0.91
0.125	0.4880	0.4800	1.67	0.3374	0.3370	0.12	7.23E-04	7.30E-04	0.93
0.13	0.5077	0.5000	1.54	0.3629	0.3620	0.25	3.76E-03	7.70E-03	1.63
0.135	0.5259	0.5250	0.18	0.3865	0.3860	0.13	1.38E-02	1.3E-02	0.23
0.14	0.5429	0.5400	0.53	0.4084	0.4080	0.10	3.433E-02	3.43E-02	0.10
0.145	0.5586	0.5500	1.57	0.4288	0.4300	0.28	6.233E-02	6.23E-02	0.07
0.15	0.5733	0.5700	0.58	0.4478	0.4490	0.26	9.21E-02	9.39E-02	1.96
0.155	0.5871	0.5800	1.22	0.4657	0.4656	0.01	1.21E-01	1.22E-01	0.81

Conclusions and future work

This paper proposed a new analytical modelling approach and QoS management for handovers based on a new handover admission control mechanism in highly mobile environments. The analysis of the handover is an important issue in order to achieve better performance, especially in highly mobile environments. The proposed handover admission control mechanism is useful for achieving better performance in such systems. It offers the perspective of considering the current time $T_{current}$, network dwell time T_{dwell} and estimated time before handover $T_{estimated}$ of the mobile users. The system is modelled as an open queuing network using a Markov chain with continuous time to determine the state probabilities. Based on the proposed approach developed in this paper, computer simulations are also used to assess the accuracy for the proposed model. The proposed model can be used to analyse QoS measures such as MQL, P_B and transmission delay. The presented examples were kept simple for performance evaluation due to the introductory nature of the proposed model for highly mobile environments. The proposed method successfully reduced the number of handover failures and unnecessary handovers to the system by using the proposed algorithm compared to the traditional approach for highly mobile users. It minimizes the number of handover failures and unnecessary handovers to the system by enhancing the usage of the resources. With this approach, resource allocation can be improved in such systems with highly mobile environments. However, there are still specific operational aspects that need to be explored where the proposed approach can be applied to get the best effect. In addition, considering the availability, modelling the proposed model could be considered for future work.

Bibliography

[1] Chen, Y.; Yang, S.; Xu, S.; Xue, P.; Zhou, X.; (2012); Queuing Theory Based Handover Resource Self-Management in LTE Networks, *International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*:1-4.

[2] Ghosh, A.; Paranthaman, V.; Mapp, G.; Gemikonakli, O.; Loo, J. (2015); Enabling Seamless V2I Communications: Towards Developing Cooperative Automotive Applications in VANET Systems, *IEEE Communications Magazine, Special Issue on Towards Autonomous Driving: Advances in V2X Connectivity*, 53(12):80-86.

[3] Zeng, Q.A.; Agrawal, D. P., (2001); Modeling of handoffs and performance analysis of wireless data networks, *International Conference on Parallel Processing Workshops*, 491-496.

[4] He, D. et al. (2010); A simple and robust vertical handoff algorithm for heterogeneous wireless mobile networks, *Wireless Personal Communication*, 59(2):361-373.

[5] Trivedi, K.S.; Dharmaraja, S.; Ma, X.; (2002); Analytic modelling of handoffs in wireless cellular networks, *Information Sciences*, 148:155-166.

[6] Rejeba, S. B.; Nasser, N.; Tabbane, S.; (2014) A novel resource allocation scheme for LTE network in the presence of mobility, *Journal of Network and Computer Applications*, Elsevier, 46:352-361.

[7] Halabian, H.; Rengaraju, P.; Lung, C.H.; Lambadaris, I.; (2015), A reservation-based call admission control scheme and system modeling in 4G vehicular networks, *EURASIP Journal on Wireless Communications and Networking*, 1-12.

[8] Kirsal-Ever, Y.; Kirsal Y.; Ever, E.; Gemikonakli,O.; (2015); Analytical Modelling and Performability Evaluation of Multi-Channel WLANs with Global Failures, *International Journal of Computers Communications and Control*, 10:551-566.

[9] Kirsal Y.; Ever, E.; Kocyigit, A.; Gemikonakli,O.; Mapp, G.; (2015); Modelling and analysis of vertical handover in highly mobile environments, *The Journal of Supercomputing*, 71(12):4352-4380.

[10] Xiaohuan, Y., Mani, N., Sekercioglu, Y.A. (2008); A traveling distance prediction based method to minimize unnecessary handovers from cellular networks to WLANs, *IEEE Communications Letters*, 12(1):14-16.

[11] Kyoung, S.L., Ae-Soon, P. (2014); Reduction of handover failure for small cells in heterogeneous networks, *Intl Conf. on Information and Communication Technology Convergence (ICTC)*: 707-708.

[12] Zhu, F.; MacNair, J.; (2004); Optimizations for Vertical Handoff Decision Algorithms, *IEEE Wireless Communications and Networking Conference*, 2:867-872.

[13] Zhu, F.; McNair, J.; (2006); Multiservice vertical handoff decision algorithms, *EURASIP Journal on wireless communications and networking*, 2; 1-13.

[14] Stevens-Navarro, E.; Lin, Y.; Wong, V.W.S.; (2008); An MDP-based vertical handoff decision algorithm for heterogeneous wireless networks, *IEEE Transactions on Vehicular Technology*, 57(2): 1243-1254.

[15] Ismail, A.; Byeong-hee, R.; (2011) Adaptive handovers in heterogeneous networks using fuzzy MADM, *Intl Conf. on Mobile IT-Convergence*, 99-104.

[16] Yan, X.; Sekercioglu, A. Y.; Narayanan, S.; (2010) A survey of vertical handover decision algorithms in fourth generation heterogeneous wireless networks, *Computer Networks*, 54(11):1848-1863.

[17] Mapp. G. et al. (2009); Exploring Efficient Imperative Handover Mechanisms for Heterogeneous Networks, *International Symposium on Emerging Ubiquitous and Pervasive Systems*, 286-291.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(6):804-818, December 2016.

The Particle Swarm Optimization Algorithm with Adaptive Chaos Perturbation

L. Mengxia, L. Ruiquan, D. Yong

Li Mengxia
1. School of Information and Mathematics, Yangtze University
Jingzhou Hubei 434023, China
2. Petroleum Engineering College, Yangtze University
Wuhan Hubei 430100, China
3. The Branch of Key Laboratory of CNPC for Oil and Gas Production, Yangtze University
Wuhan Hubei 430100, China
4. Key Laboratory of Exploration Technologies for Oil and Gas Resources, Yangtze University
Wuhan Hubei 430100, China
limengxia81@126.com

Liao Ruiquan
1. Petroleum Engineering College, Yangtze University
Wuhan Hubei 430100, China
2. The Branch of Key Laboratory of CNPC for Oil and Gas Production, Yangtze University
Wuhan Hubei 430100, China
3. Key Laboratory of Exploration Technologies for Oil and Gas Resources, Yangtze University
Wuhan Hubei 430100, China
liaoruiquan@263.net

Dong Yong*
1. School of Information and Mathematics,Yangtze University
Jingzhou Hubei 434023, China
2. The Branch of Key Laboratory of CNPC for Oil and Gas Production, Yangtze University
Wuhan Hubei 430100, China
3. Key Laboratory of Exploration Technologies for Oil and Gas Resources, Yangtze University
Wuhan Hubei 430100, China
*Corresponding author: dongyong80@126.com

Abstract: Aiming at the two characteristics of premature convergence of particle swarm optimization that the particle velocity approaches 0 and particle swarm congregate, this paper learns from the annealing function of the simulated annealing algorithm and adaptively and dynamically adjusts inertia weights according to the velocity information of particles to avoid approaching 0 untimely. This paper uses the good uniformity of Anderson chaotic mapping and performs chaos perturbation to part of particles based on the information of variance of the population's fitness to avoid the untimely aggregation of particle swarm. The numerical simulations of five test functions are performed and the results are compared with several swarm intelligence heuristic algorithms. The results shows that the modified algorithm can keep the population diversity well in the middle stage of the iterative process and it can improve the mean best of the algorithm and the success rate of search.

Keywords: Particle Swarm Optimization, inertia weight,population diversity, expected velocity, chaos perturbation.

1 Introduction

Particle swarm optimization (PSO) is an evolutionary algorithms (EAs). It stems from the simulation of group behavior for birds swarm's foraging [1, 2]. The parameters and structures of

PSO are very simple and are apt to be realized. It has parallelism essentially. It has no requirement to the properties of objective functions. It has good adaptation and gains wide concern. It is applied in many fields such as decision feedback equalizer [3, 4], parameter identification [5, 6], power dispatch [7] and mechanical control [8,9], etc.

The particle swarm of the standard PSO is apt to get into local best position and occurs premature convergence phenomenon. The premature convergence has two characteristics that the velocity of particle swarm approach 0 and the particles aggregate in a small region.

In order to avoid early approaching to 0 for the velocity of particle swarm, the inertia weights of standard PSO are modified from fixed values to variation values in the references [10-15], and the performance of PSO is improved in some extent.

The reference [2] proposed a method that the inertia weight decreases within the increasing of iterations. It emphasizes the global search in the earlier stage of evolution and intensifies the local exploitation in the later stage of evolution. But the effect on the complex problem is not obvious. The reference [10] use the fuzzy controller to adjust the inertia weight, but it is complex to build the fuzzy rule. The reference [11] proposed the method to set the inertia weight randomly and obtained a greater effect on pursuing the optimization. The reference [12] used the chaotic mapping to change the inertia weight and the inertia weight changes according to the Logistic chaotic mapping. The reference [13, 14] used different exponential forms to build the setting method of the inertia weight which deceases according to the exponential rule. The reference [15] uniformly adjusted the inertia weight for the all particles according to the mean velocity of the particle swarm, and the numerical results showed that it can keep the population diversity in the middle stage of evolution.

For PSO algorithm, the particle,s position is different. The global best particle and the personal best particle are in the leading position. In the process of evolution, the other particles are affected by the two particles and are close to them. The global best particle and the personal best particle are considered as on group which is called the dominant group. The other particles are taken as another group which is called the ordinary group. Obviously, it is appropriate to deal with the dominant group and the ordinary group respectively.

This paper uses the smaller inertia weight to develop the local development capabilities for the particles in the dominant group. For the particles in the ordinary group, the method in reference [15] that adjusts the inertia weight based on the mean velocity of swarm is modified and the inertia weight corresponding to each particle is adjusted respectively according to the expected values of velocity of particle swarm to avoid the swarm velocity early approaching 0.

For the second characteristics of the premature convergence, the reference [16] considered that the variance of the population's fitness can reflect the concentration degree of population. If the value of is less than some specified threshold, the particle swarm aggregate and the premature convergence occurs.

$$\sigma^2 = \sum_{k=1}^{n} (\frac{f_k - f_m}{f})^2 \tag{1}$$

$$f = max\left\{1, \max_k \left\{|f_k - f_{avg}|\right\}\right\} \tag{2}$$

In Eq. 1 and Eq. 2, $f_k$denotes the fitness value of the k-th particle.$f_m$denotes the mean value of the variance of the population's fitness. $f$denotes the normalized parameter.

The reference [16] didn't point out how to determine the appropriate threshold value. To analyze a large number of numerical simulations, the results showed that the variances σ^2keep stabilization and basically equal in the subsequent several evolutionary when the premature convergence occurs. Hence, this paper compares the corresponding variancesσ^2between two adjacent iterations. If the change of σ^2value is very small, it considers that the premature convergence occurs and needs to set the position of particle swarm to advance the population diversity. One common method is to use the Logistic chaotic mapping performing perturbation to part of particles.

It is easy to verify that the distribution of the chaotic sequence generated by the Logistic chaotic mapping has the characteristic that is big at both ends and small in the middle and the uniformity of the chaotic sequence is poor. But the chaotic sequence generated by the Anderson chaotic mapping has better uniformity. So it is appropriate to use the Anderson chaotic mapping to perform the perturbation to part of particles.

Based on the above analysis, this paper firstly considers the expected value of population velocity, and changes the inertia weights respectively corresponding to the particles according to the category of particles. And then it performs chaos perturbation to part of particles based on the difference of the variances of the population's fitness between two adjacent iterations. Finally, the numerical simulations are performed and the results show that the new algorithm has higher performance.

2 Standard PSO algorithm

PSO algorithm is simulating the foraging process of the bird flock. Particle denotes the individual. It has position and velocity, but has no volume and mass. Multiple particles constitute the particle swarm which denote the bird flock. The objective function value corresponding to the particle's position is called the fitness of the particle which is used to evaluate the good points and the bad points of the particle. The best position which the individual particle went through is called the personal previous best position. The personal previous best position of the kth particle is denoted by p_k.The previous best position of the particle swarm is called the swarm best position which is denoted by p_g.The evolution of the particle is realized through tracing the personal previous best position and swarm best position.

The evolution of standard PSO algorithm is divided into the evolution of the velocity and the evolution of the position. The evolution equations are shown in Eq. 3 and Eq. 4 [1, 2] as below.

$$v_k^{t+1} = \omega \times v_k^t + c_1 \times r_1 \times (p_k - x_k^t) + c_2 \times r_2 \times (p_g - x_k^t) \tag{3}$$

$$x_k^{t+1} = x_k^t + v_k^{t+1} \tag{4}$$

In Eq. 3 and Eq. 4, v_k^tdenotes the velocity of the kth particle in the tth evolution. x_k^tdenotes the position of the kth particle in the tth evolution. r_1 and r_2 denote the independent random numbers on the interval [0, 1]. c_1 and c_2 denote the learning factors which generally equal 2. ω denotes the inertia weight which takes fixed value between 0.1 and 0.9.

The value ranges of the particle's velocity and position are usually restricted. If the maximum of the position is determined by x_{max}, the maximum of the velocity is $v_{max} = k \times x_{max}$ for $0.1 \leq k \leq 1.0$ [16].

3 Modification of the inertia weight

The mean velocity v_a^t of the particle swarm in the kth evolution is denoted by Eq. 5.

$$v_a^t = (\sum_{i=1}^{n} \sum_{k=1}^{m} \left| v_{k,i}^{(t)} \right|)/(n \times m) \tag{5}$$

n is the dimension of the problem which is the dimension of the search space. m is the total number of the particles. $v_{k,i}^{(t)}$ is the velocity of the kth particle in the ith dimensionality as the particle experiences the tth evolution. $v_a^{(t)}$ denotes the evolution amplitude of the particle swarm or the evolution step. The larger the value of $v_a^{(t)}$ is, the larger the evolution step is. It means that the search capability of the particle swarm is great. If the value of $v_a^{(t)}$ is small, it means that the search of particle swarm focuses on the local exploitation. In the earlier stage of evolution, the global search capability is expected to be stronger. It means that the value of $v_a^{(t)}$ is expected to be larger. In the later stage of the evolution, the capability of local exploitation is expected to be intensified to increase the capability of finding the maximal solution by the algorithm, and the value of $v_a^{(t)}$ is expected to be smaller.

This paper firstly introduce the expected value of the mean value of particle swarm in the evolution which is shown in Eq. 6 as follow.

$$v_e^t = v_0 \times exp(-(\lambda_1 \times t/T_{max})^{\lambda_2}) \tag{6}$$

In Eq. 6, v_0 denotes the mean velocity of the initial particle swarm. λ_1 and λ_2 are adjustable parameters which is to control the change rule of the expected value v_e^t of the particle swarm shown in Figure 1. T_{max} denotes the maximal evolution generations.

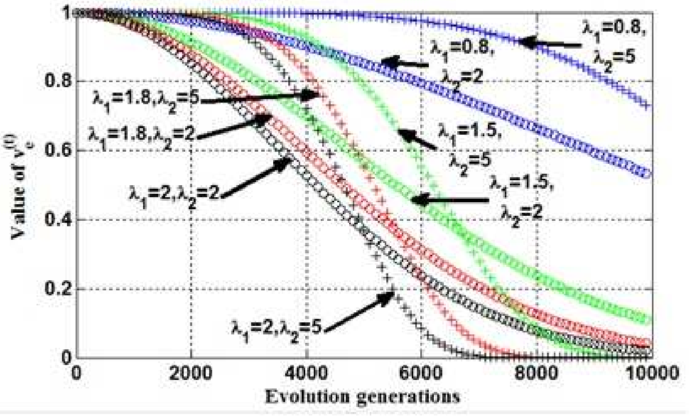


Figure 1: The change rule of v_e^t under different parameters

From Figure 1, it can be seen that the value of v_e^t as $\lambda_1 = 1.8$ and $\lambda_2 = 5$ is suitable to be the expected mean velocity of particle swarm. In this case, in the earlier stage of evolution, the expected value of mean velocity of particle swarm is larger and it helps to keep the global search capability. In the last stage of evolution, the expected value of mean velocity approaches 0 and it helps to improve the local exploitation capability.

ω_0 denotes the initial inertia weight. v_e^t denotes the actual velocity of the kth particle. ω_k^t denotes the inertia weight of the kth particle in the tth evolution. ω_k^{t+1} denotes the inertia weight of the kth particle in the $(t+1)$th evolution. ω_k^{t+1} is determined by the following rule.

(1) If the particle is the better particle, ω_k^{t+1}=ω_{min};

(2) If the particle is the trivial particle, then, define

$$v_{k,a}^{(t)} = (\sum_{i=1}^{n} \left| v_{k,i}^{(t)} \right|)/n \tag{7}$$

there are three cases to be considered as below.

(2.1) If $v_{k,a}^{(t)} > v_e^{(t)}$,then, $\omega_k^{(t+1)} = \omega_k^{(t)} \times p_1$.

(2.2) If $v_{k,a}^{(t)} < v_e^{(t)}$,then, $\omega_k^{(t+1)} = \omega_k^{(t)} \times p_2$.

(2.3) if $v_{k,a}^{(t)} = v_e^{(t)}$,then, $\omega_k^{(t+1)} = \omega_k^{(t)}$.

In the above process, p_1 and p_2 are constants with $0 < p_1 < 1$ and $p_2 > 1$. It also can take $p_1 = 1/p_2$. If $\omega_k^{(t+1)} > \omega_{max}$, p_2 is taken to be 1.05 according to the reference [8] or is determined by the experiment. Then taking $\omega_k^{(t+1)} = \omega_{max}$. ω_{max} and ω_{min} denote the upper limit and lower limit of the inertia weight, respectively.

4 Anderson chaotic mapping

This paper uses the rule that the difference of σ^2 between the adjacent evolution process is less than some given value se to recognize the premature convergence, for example, taking $se = 10^{-3}$. If there exists premature convergence, then it needs to perform the perturbation for the current particle to improve the population diversity.

Generally, the researchers use the Logistic chaotic mapping to perform perturbation [18-21], but the sequence generated by the Logistic chaotic mapping has bad uniformity [22]. The Logistic chaotic mapping is shown in Eq. 8 as below.

$$y_{n+1} = \mu \times y_n \times (1 - y_n) \tag{8}$$

When $\mu = 4$, Eq. 8 will generate the chaotic sequence whose value is taken between 0 and 1.

This paper introduce Anderson chaotic mapping to generate chaotic sequence which has good uniformity [23]. Anderson chaotic mapping is shown in Eq. 9 as below.

$$y_i = (Ln(x_i + 1/2) + Ln(2)))/ln(3) \tag{9}$$

where,

$$x_{n+1} = \begin{cases} \frac{3}{2} \times x_n + 1/4 & 0 \leqslant x_n < \frac{1}{2} \\ \frac{1}{2} \times x_n - 1/4 & \frac{1}{2} \leqslant x_n < 1 \end{cases} \quad (n = 1, 2, ...) \tag{10}$$

The sequence y_i is uniform distribution on $(0, 1)$.

In order to compare the uniformity of the two chaotic mapping, the initial value is taken arbitrarily. And then iterate 1000 times according to Eq. 8 and Eq. 9. The interval [0, 1] is divided into 10 subintervals with equal width. The frequency that the elements of the chaotic sequence locate at each subinterval is considered and shown in Figure 2.

If the other initials are taken except that the value of 0.25 and 0.75 cannot be taken because of the occurring of fixed points for Logistic chaotic mapping. The obtained frequency graph is similar to Figure 2. This means that Anderson chaotic mapping has stable uniformity.

5 Modified PSO algorithm

The objective function is expressed in Eq. 11.

$$minf(x) = f(x_1, x_2, ..., x_n) \tag{11}$$

Where, n is the dimension of the problem.

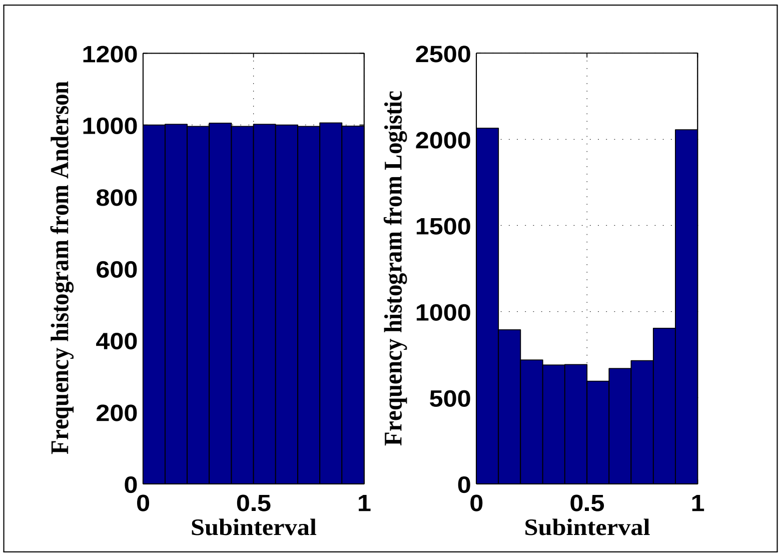


Figure 2: Comparison of the uniformity between two chaotic mappings

5.1 Initial particle swarm generated by the chaotic mapping

Firstly, one m-dimensional vector is generated randomly and its element is between 0 and 1. Secondly, each element is taken as initial value. And using Anderson chaotic mapping iterates $n-1$ times to generate ms n-dimensional vector which are denoted by $y_1, y_2, ..., y_m$. Finally, the value range of element of y_i for $i = 1, 2, ..., m$ is transformed to the search space from the interval $[0, 1]$.

Suppose $y \in [a, b]$,then $x \in [c, d]$. And the relation between x and y is shown in Eq. 12.

$$x = c + \frac{y-a}{b-a}(d-c) \tag{12}$$

The initialization of the velocity can be finished analogously.

5.2 Modification for evolution process

In Figure.3, after finishing the tth evolution process, the variance $\sigma^2(t)$ is obtained by calculating the variance of the population's fitness. And then it is compared with the variance $\sigma^2(t-1)$ in the previous evolution process. If $\left|\sigma^2(t) - \sigma^2(t-1)\right| \leqslant se$, the particles are ordered from small to large according to the fitness value. The particles which account for 61.8% of total particle number after ordering are performed chaotic perturbation. The number of foregoing particles is denoted by S. That is, the positions of Ss particles are initialized again.

5.3 Modified PSO algorithm

Step 1 Give the initial value of the inertia weight $\omega_0 = \omega_{max} = 0.95$,$\omega_{min} = 0.05$ learning factors $c_1 = c_2 = 2$, the number of particles $m = 40$, the maximal evolution generations $T_{max} = 10000$, the maximal evolution generations n, the control value for starting the chaotic perturbation $se = 10^{-3}$, the search space $[x_{min}, x_{max}]$ and the upper limit of velocity v_{max}.

step 2 Randomly generate a m-dimensional particle on $[0, 1]$.Use the Anderson mapping shown in Eq. 9 and Eq. 10 to get ms n-dimensional particles. The ms n-dimensional particles which are transformed into the search space based on Eq. 12 are denoted by x_i for $i = 1, 2, ..., m$.

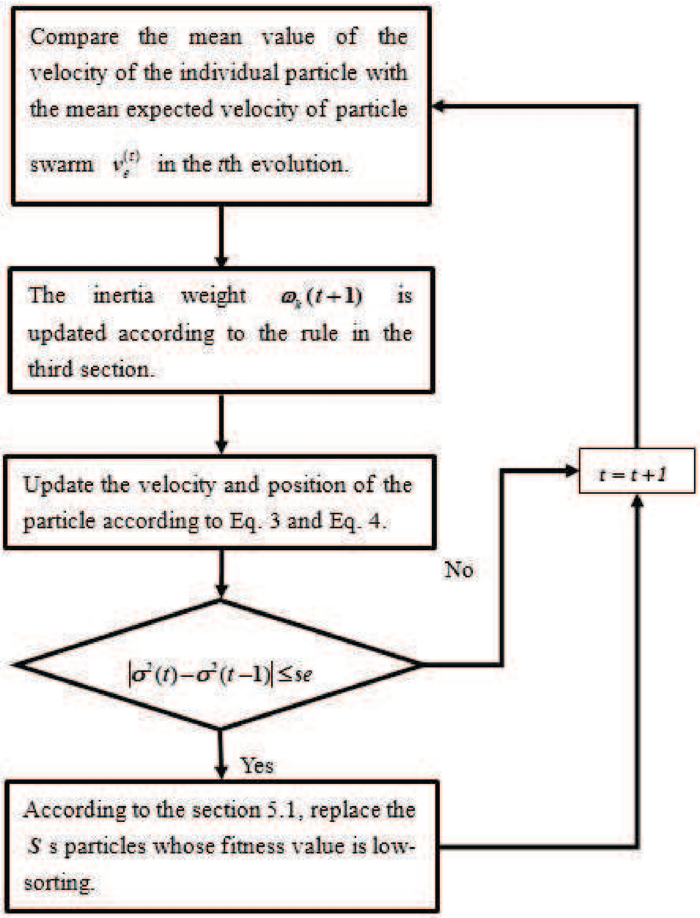

Compare the mean value of the velocity of the individual particle with the mean expected velocity of particle swarm $v_s^{(t)}$ in the tth evolution.
The inertia weight $\omega_k(t+1)$ is updated according to the rule in the third section.
Update the velocity and position of the particle according to Eq. 3 and Eq. 4.
$|\sigma^2(t)-\sigma^2(t-1)|\leq se$
No
Yes
According to the section 5.1, replace the S s particles whose fitness value is low-sorting.
$t=t+1$

Figure 3: The flow chart of chaos perturbation

Similarly, generate the particle's velocity. Take $v_{max} = x_{max}$. Let the evolution generations be 0, and then turn to step 3.

step 3 Calculate the fitness of each particle and the variance of the population's fitness. Determine the swarm best position p_g, the personal previous best position p_i for $i = 1, 2, ..., m$. Turn to step 4.

Step 4 Adjust ω according to the contents in section 3. Evolve the particle's velocity and position according to Eq. 3 and Eq. 4. And then add one to the evolution generations. Update p_g and p_i.Calculate the variance of current population's fitness. Compare the calculating variance with the variance of fitness value before evolution. If the difference is less than se, turn to step 5. Otherwise, turn to step 6.

Step 5 Determine the number of particles which need to be performed perturbation, S. Use Anderson mapping to generate Ss particles to replace the Ss particles whose fitness value order are back. Then turn to step 6.

Step 6 If the evolution generations is less than T_{max}, turn to step 4. Otherwise, turn to step 7.

Step 7 Output the results: p_g and $f(p_g)$.

6 Numerical simulation

In order to test the performance of the PSO algorithm with chaos perturbation proposed in this paper, this paper selects five benchmark function shown in Table 1.

Table 1: Benchmark functions and part parameters

Function	Dimensionality	Search space	Optimal value	Convergence criteria
Sphere Function	30	$[-100, 100]^n$	0	0.01
Rosenbrock's Function	30	$[-30, 30]^n$	0	0.01
Rastrigin's Function	30	$[-5.12, 5.12]^n$	0	0.01
Griewanks's Function	30	$[-600, 600]^n$	0	0.01
Schwefel Function	30	$[-100, 100]^2$	0	0.01

Table 1 gives some parameters of the algorithm, such as dimensionality, search space, theoretical optimum and convergence criteria. The theoretical optimums of these benchmark functions are all zeros. Because the theoretical optimum is difficult to obtained, so this paper uses the convergence criteria to determine the astringency of the algorithm. The convergence criteria equals to 0.01.

Aiming at the above benchmark functions, this paper compares the optimal performance of the following algorithms with each other: the algorithm ACPSO proposed in this paper, the standard PSO algorithm denoted by SPSO, the algorithm in the reference [12] denoted by Ref. 12, the algorithm in the reference [15] denoted by Ref. 15, and the algorithm in the reference [20] denoted by Ref. 20. The algorithm ACPSO proposed in this paper tests the number of two kinds of particles. The SPSO algorithm is shown in the reference [2]. The algorithm Ref. 12 generates the random number of the velocity equation for the SPSO algorithm by using Logistic chaos. The algorithm Ref. 15 uniformly adjusts the inertia weight according to the average velocity of the all particles. The algorithm Ref. 20 is the firefly algorithm. For Ref. 20, it takes the number of particles 40, the evolution generations 1000, initial attractiveness 0.728, the initial absorption coefficient 0.345 and the randomization parameter 0.25. Using the parameters shown in Table 2, each algorithm is run 20 times respectively.

Table 2: The parameters for each algorithm

Algorithm	Number of particles	Evolution generations	Initial ω	p_2	se	Chaos update rate	r_1	r_2
ACPSO	40	1000	0.95	1.05	1	61.8%	/	/
ACPSO	400	1000	0.95	1.05	1	61.8%	/	/
SPSO	40	1000	0.7298	/	/	/	/	/
Ref. 12	40	1000	0.7298	/	/	/	0.2	0.8
Ref. 15	40	1000	3	1.05	/	/	/	/

The performance of each algorithm is compared by adapting the following criterions. (1) The convergence rate I_r. It is the ratio of running time as achieving the convergence criteria to the total running time 20.

(2) Optimal value f_{best}. It is the minimum value of the results in 20 times run.

(3) The mean optimal value denoted by f_{av}. It is defined by the arithmetic mean value of the optimal fitness in 20 times run.

(4) Mean squared deviation denoted by std. It is the mean squared deviation for some algorithm in 20 times run.

(5) Elapsed time. It is the consuming time for each run.

(6) The schematic diagram of mean convergence. Firstly the evolution generations and the corresponding optimal value are extracted. And then take the arithmetic mean value for the optimal value in 20 times run for each evolution generation. Finally, draw the graph by taking the evolution generations and the mean optimal value of each evolution generation to be the abscissa and ordinate respectively.

(7) The schematic diagram of optimal convergence. Aiming at 20 times run, take the running process corresponding the optimal result. And draw the graph by taking the evolution generations and the optimal value of each evolution generation to be the abscissa and ordinate, respectively.

It gives the result respectively for each benchmark function from Table 3 to Table 7 as follow.

Table 3: The result of Sphere Function

Algorithm	I_r /%	f_{best}	f_{av}	std	Elapsed time $/s$
ACPSO	0	0.7204	2.68	1.11	1.66
ACPSO	1	$7.83e-13$	$6.60e-12$	4.77	15.23
SPSO	0	2164.84	6705.48	2772.00	0.97
Ref. 12	0	22237.05	55486.04	14306.06	0.97
Ref. 15	0	41.8138	438.52	448.86	1.14
Ref. 20	1	0.00174	0.00268	0.000495	5.12

The benchmark functions adapted in this paper are all high-dimensional functions of 30-dimension. From Figure 4 to Figure 13, it can be seen that the ACPSO algorithm proposed in this paper has better search capability, especially in the earlier stage of the evolution. From Table 3 to 7, it can be seen that the ACPSO algorithm shows better stability. On the whole,

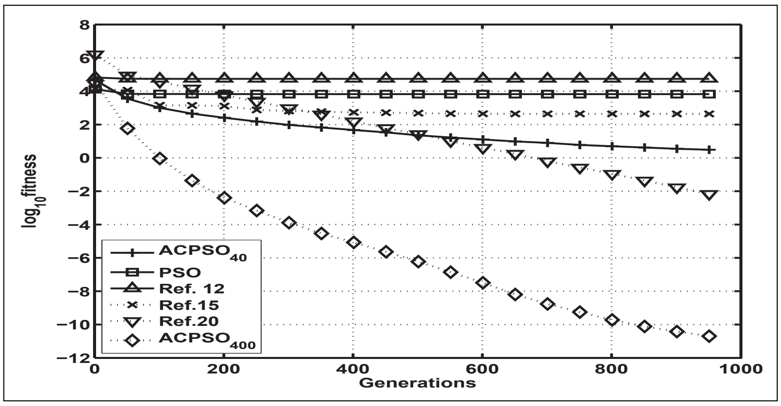


Figure 4: Average evolutionary process for Sphere function

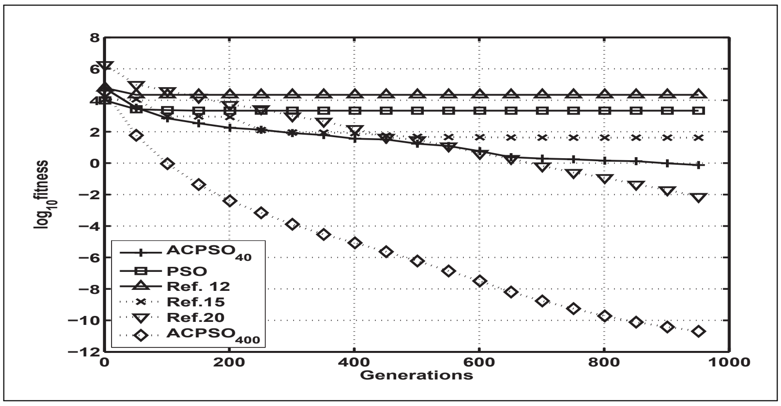


Figure 5: Best evolutionary process for Sphere function

Table 4: The result of Rosenbrock's Function

Algorithm	$I_r/\%$	f_{best}	f_{av}	std	Elapsed time s
ACPSO	0	38.10	174.31	128.14	1.64
ACPSO	1	16.90	23.19	10.99	14.74
SPSO	0	$6.65e+5$	$3.79e+6$	$3.16e+6$	0.90
Ref. 12	0	$1.40e+8$	$2.26e+8$	$4.57e+7$	1.03
Ref. 15	0	138.44	$3.75e+3$	$4.86e+3$	1.05
Ref. 20	1	28.07	$1.76e+5$	$7.00e+5$	5.27

Table 5: The result of Rastrigin's Function

Algorithm	$I_r/\%$	f_{best}	f_{av}	std	Elapsed time s
ACPSO	0	44.32	105.88	39.94	1.63
ACPSO	0	29.85	80.35	29.34	15.05
SPSO	0	205.15	262.46	39.06	0.93
Ref. 12	0	304.99	392.91	41.42	0.94
Ref. 15	0	29.01	62.14	24.09	1.09
Ref. 20	0	17.39	34.47	9.39	4.96

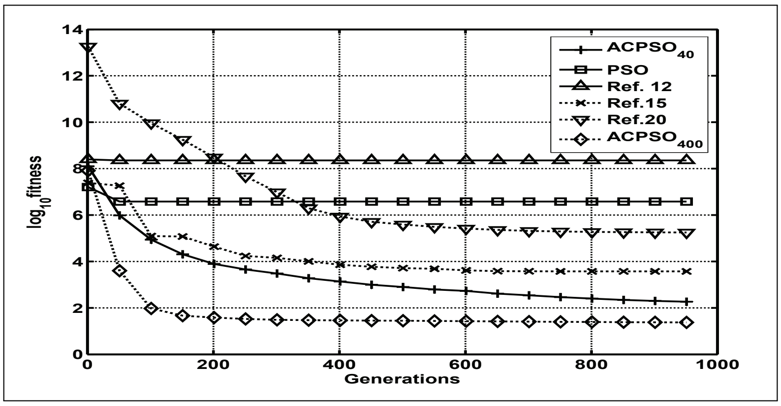


Figure 6: Average evolutionary process for Rosenbrock's function

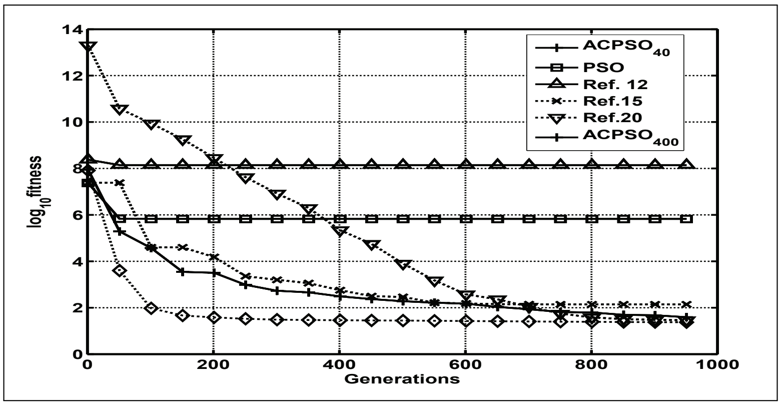


Figure 7: Best evolutionary process for Rosenbrock's function

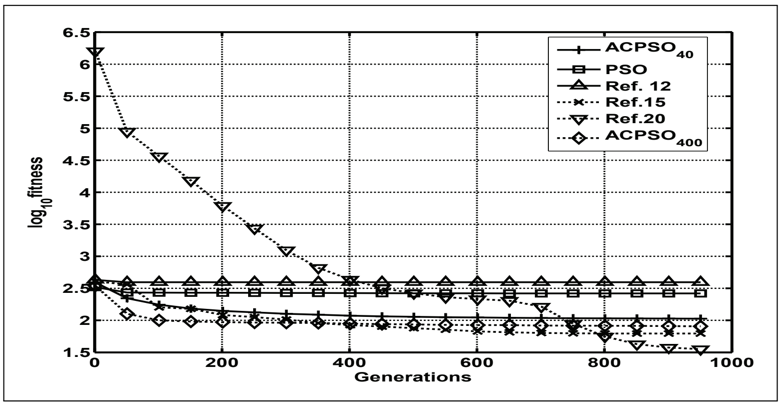


Figure 8: Average evolutionary process for Rastrigin's function

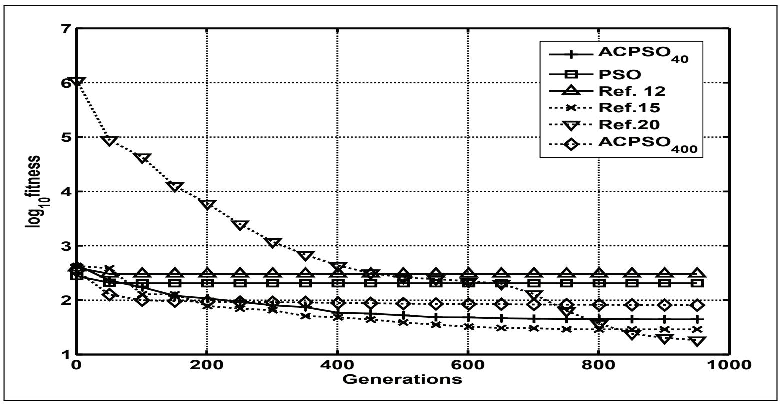


Figure 9: Best evolutionary process for Rastrigin's function

Table 6: The result of Griewanks's Function

Algorithm	$I_r/\%$	f_{best}	f_{av}	std	Elapsed time s
ACPSO	0	0.7952	0.9675	0.0068	1.87
ACPSO	35	$5.66e-10$	0.0214	0.0169	19.52
SPSO	0	19.303	51.182	18.643	1.10
Ref. 12	0	278.85	527.87	116.75	1.11
Ref. 15	0	1.2985	6.2802	5.9002	1.36
Ref. 20	85	$1.04e-4$	0.0041	0.0082	6.21

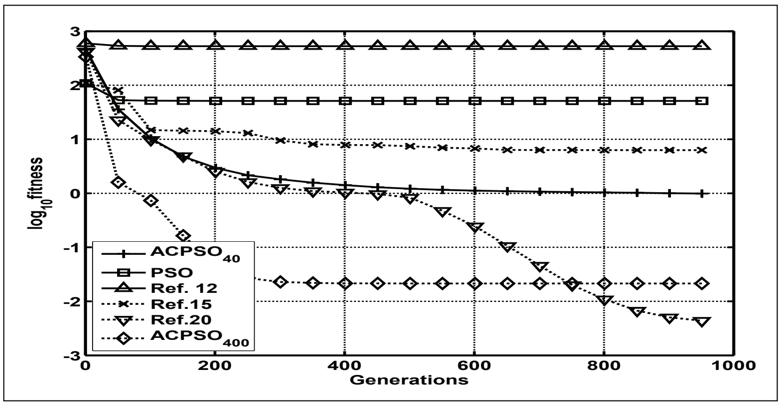


Figure 10: Average evolutionary process for Griewanks's function

Table 7: The result of Griewanks's Function

Algorithm	$I_r/\%$	f_{best}	f_{av}	std	Elapsed time s
ACPSO	0	1.0486	2.0469	0.6447	1.84
ACPSO	1	0.00038	0.00038	$1.91e-08$	17.08
SPSO	0	207.8326	453.7501	130.34	1.10
Ref. 12	0	2014.7359	3771.971	852.53	1.11
Ref. 15	0	12.2179	44.2053	32.499	1.27
Ref. 20	0	4621.4715	6014.141	767.825	5.38

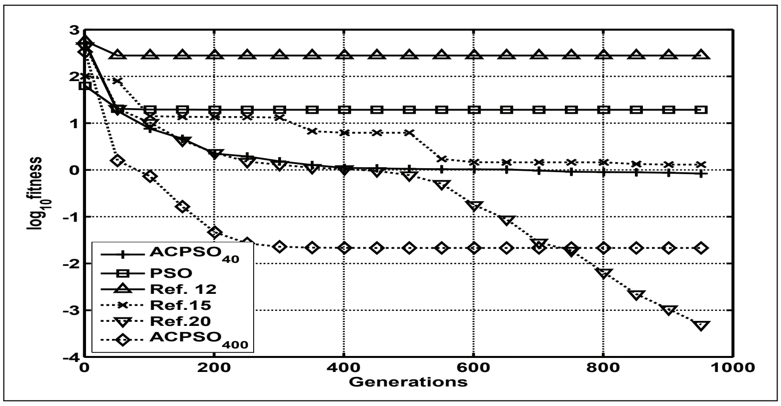


Figure 11: Best evolutionary process for Griewanks's function

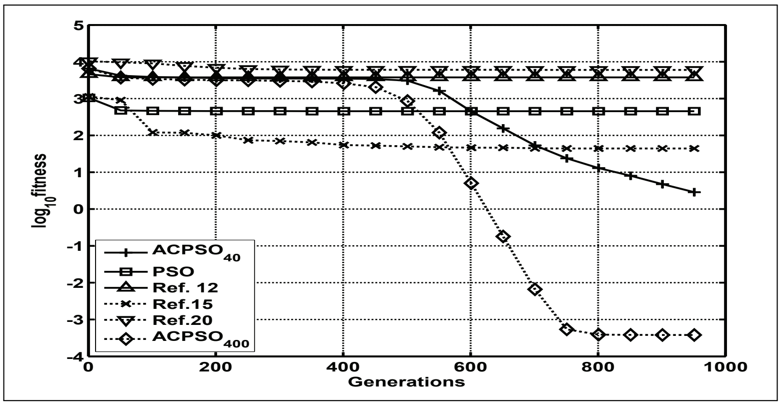


Figure 12: Average evolutionary process for Schwefel function

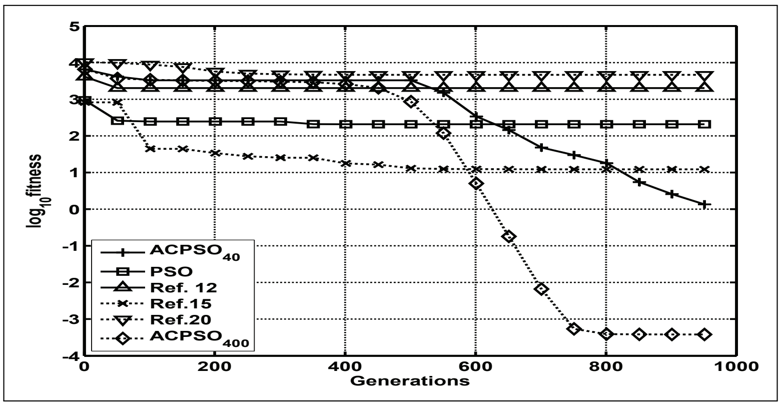


Figure 13: Best evolutionary process for Schwefel function

the search capability of the algorithm proposed in this paper is superior to the standard PSO algorithm and is better than the algorithm Ref. 20 for part benchmark functions.

Conclusions

(1) The algorithm proposed in this paper introduces the Anderson chaotic mapping to realize the initial of the particle swarm and the perturbation of part particles. It also introduces the concept of the expected mean velocity of particle swarm to realize the adjustment of the inertia weight of particles.

(2) The results of Benchmark functions show that the algorithm proposed in this paper reflects preferable search ability.

(3) The analyses of the mean optimal fitness and evolution generations show that the algorithm of this paper keeps the population diversity well in the middle stage of the evolution and it has stronger local search ability in the later stage of evolution.

Acknowledgment

The authors will thank people in the Branch of Key Laboratory of CNPC for Oil and Gas Production and Key Laboratory of Exploration Technologies for Oil and Gas Resources for their great help. This paper is supported by Educational Commission of Hubei Province of China (B2015449) and National Natural Science Foundation of China (61572084 and 51504038).

Bibliography

[1] Kennedy J, Eberhart R,(1995); Particle swarm optimization. *IEEE Int. Conf. on Neural Networks.* Piscataway, NJ. IEEE Service Center, 1942-1948.

[2] Shi Y, Eberhart R C,(1998); A modified particle swarm optimizer. *IEEE Int. Conf. on Evolutionary Computation.* Piscataway, NJ. IEEE Service Center, 69-73.

[3] Naveed Iqbal, Azzedine Zerguine,Naofal Al-Dhahir,(2014); Decision Feedback Equalization using Particle Swarm Optimization. *Signal Processing*, 108:1-12, DOI: 10.1016/j.sigpro.2014.07.030

[4] Manish Mandloi, Vimal Bhatia,(2016); A low-complexity hybrid algorithm based on particle swarm and ant colony optimization for large-MIMO detection. *Signal Processing*, 50:66-74. DOI: 10.1016/j.eswa.2015.12.008

[5] Md Ashiqur Rahmana,Sohel Anwara,Afshin Izadian,(2016); Electrochemical model parameter identification of a lithium-ion battery using particle swarm optimization method.*Journal of Power Sources.*307, 86-97. DOI: 10.1016/j.jpowsour.2015.12.083

[6] Razieh Sheikhpour,Mehdi Agha Sarrama,Robab Sheikhpour,(2016); Particle swarm optimization for bandwidth determination and feature selection of kernel density estimation based classifiers in diagnosis of breast cancer.*Applied Soft Computing*, 40113-131. DOI: 10.1016/j.asoc.2015.10.005

[7] Yu-Shan Cheng,Man-Tsai Chuang,Yi-Hua Liu,Shun-Chung Wang,Zong-Zhen Yang,(2016); A particle swarm optimization based power dispatch algorithm with roulette wheel re-distribution mechanism for equality constraint.*Renewable Energ*, 88: 58-72, DOI: 10.1016/j.renene.2015.11.023.

[8] Migdat Hodzic, Li-Chou Tai (2016); Grey Predictor reference model for assisting particle swarm optimization for wind turbine contro.*Renewable Energ*, 86: 251-256, DOI: 10.1016/j.renene.2015.08.001.

[9] Tao Lin, Peng Wu, Fengmei Gao, Yi Yu, Linhong Wang (2016); Study on SVM temperature compensation of liquid ammonia volumetric flowmeter based on variable weight PSO, *International Journal of Heat and Technology* , 33(2):151-156, DOI: 10.18280/ijht.330224.

[10] Shi Y. and Eberhart, R.C. (2001); Fuzzy adaptive particle swarm optimization, *IEEE Int. Conf. on Evolutionary Computation.*Seoul, Korea, 101-106.

[11] Zhang L., Yu H., Hu S. (2003); A new approach to improve particle swarm optimization, *Genetic and Evolutionary Computation Conference 2003* .Chicago, IL, USA, 134-142.

[12] Jiang C. W., Etorre B. (2005); A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimization, *Mathematics and Computers in Simulation*, 68:57-65.

[13] Chen G.M., Huang X.B.,Jia J Y, Min Z.F.(2006); Natural exponential inertia weight strategy in particle swarm optimization. *World Congress on Intelligent Control & Automation*, 1: 3672-3675.

[14] Jiao B., Lian Z.G.,Gu X.S.(2008); A dynamic inertia weight particle swarm optimization algorithm.*Chaos Solitons & Fractals*, 37(3): 698-705.

[15] Zhang Dingxue, Liao Ruiquan,(2009); Adaptive particle swarm optimization algorithm based on population velocity.*Control and Decision*, 24(8): 1257-1265, DOI:10.13195/j.cd.2009.08.139.zhangdx.025

[16] Lv Zhensu, Hou Zhirong (2004); Particle Swarm Optimization with Adaptive Mutation.*Acta Electronica sinica*, 32(3):416-420.

[17] Zeng Jianchao, Cui Zhihua (2012); Nature Inspired Computation, *Beijing, National Defense Industry Press*, 252-256.

[18] Kazem A., Sharifi E., Hussain F.K., Saberi M., Hussain O.K. (2013); Support vector regression with chaos-based firefly algorithm for stock market price forecasting, *Applied Soft Computing*, 13: 947-958. DOI: http://dx.doi.org/10.1016/j.asoc.2012.09.024.

[19] LIU Huaying, LIN Yue (2006); A Hybrid Particle Swarm Optimization Based on Chaos Strategy to Handle Local Convergence, *Computer Engineering and Applications*, 42(13):77-79.

[20] Yang X.S.(2011); Chaos-enhanced firefly algorithm with automatic parameter tuning, *International Journal of Swarm Intelligence Research*, 2: 1-11.

[21] Lu Y., Liu X.(2011); A new population migration algorithm based on the chaos theory, *IEEE 2nd International Symposium on Intelligence Information Processing and Trusted Computing*, 147-150, DOI: 10.1109/IPTC.2011.44

[22] LV Jinhu, LU Junan, Chen Shihua,(2002); Chaotic time series analysis and its application, *Wuhan: Wuhan University press*, 87-89.

[23] R. Anderson (1996); Industrial Cryptography, *IEEE REV.*, 118-120.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(6):819-831, December 2016.

Delay-Sensitive Optimization Models and Distributed Routing Algorithms for Mobile Wireless Sensor Networks

G.A. Montoya, Y. Donoso

Germán A. Montoya, Yezid Donoso*
System and Computing Engineering Department
Universidad de Los Andes
Bogotá, Colombia, South America
ga.montoya44@uniandes.edu.co
*Corresponding author: ydonoso@uniandes.edu.co

Abstract: Communication disruptions caused by mobility in wireless sensor networks introduce undesired delays which affect the network performance in delay sensitive applications in MWSN. In order to study the negative effects caused by mobility, we propose two mathematical models to find the minimum cost path between a source node and a destination node considering the nodes position changes across time. Our mathematical models consider the usage of buffers in the nodes to represent the fact of storing a message if there is not an appropriate forwarding node for transmitting it. In order to contrast our mathematical models results we have designed two kinds of algorithms: the first one takes advantage of the closest neighbours to the destination node in order to reach it as fast as possible from the source node. The second one simply reaches the destination node if a neighbour node is precisely the destination node. Finally, we compare the delay performance of these algorithms against our mathematical models to show how efficient they are for reaching a destination node. This paper is an extension of [10].[a] The mathematical model proposed in [10] is improved by adding two new binary variables with the aim of make it more readable and compact mathematically. This means a post-processing algorithm is added only for evaluating if a solution is at the first network state.

Keywords: Mathematical model, Delays, MWSN.

[a] Reprinted (partial) and extended, with permission based on License Number 3961371369962 ©[2016] IEEE, from "Computers Communications and Control (ICCCC), 2016 6th International Conference on".

1 Introduction

The advances of Wireless Sensor Networks (WSN) have allowed attaching sensor devices to entities such as objects, animals or humans, for monitoring a physical variable presented in a particular environment. However, these sensors are equipped with limited batteries whereby it is required to implement energy efficient routing techniques to extend as much as possible the lifetime of these devices [1] [2]. Moreover, communication disruptions caused by mobility in wireless sensor networks introduce undesired delays which affect the network performance in delay sensitive applications, such as military or healthcare monitoring applications. For the latter, due to they deal with health states, illness and continuous medical supervision, a base station should no experiment delays from the information collected by the sensors [3] [4].

Given the scenario described above, novel routing algorithms are emerging for solving these delays problems [7] [8] [9]. Moreover, these algorithms need to be compared against a mathematical model for knowing their delay performance in terms of delay. With the purpose of analyse the negative effects caused by mobility, we propose a mathematical model to find the minimum cost path between a source node and a destination node considering the nodes position changes across time. Our mathematical model considers the usage of buffers in the nodes to represent the fact of storing a message if there is not an appropriate forwarding node for transmitting it. This paper is an extension of [10] (doi: 10.1109/ICCCC.2016.7496736). The mathematical model proposed in [10] is improved by adding two new binary variables with the aim of make it more readable and compact mathematically. This means a post-processing algorithm is added only for evaluating if a solution is at the first network state.

In order to contrast our mathematical model results we have designed two algorithms: the first one takes advantage of the closest neighbours to the destination node in order to reach it as fast as possible from the source node. The second one simply reaches the destination node if a neighbour node is precisely the destination node.

The remainder of the paper is organized as follows: Section 2 describes the problem statement in a general view. Section 3 presents the problem formulation, that is, how the problem is described as a modular problem. Section 4 shows the mathematical model proposed, the objective function and the constraints. Section 5 presents the proposed algorithms for comparing them against the optimal solution given by the mathematical optimization model. Finally, sections 6 and 6 show the results and conclusions respectively.

2 General problem statement

The Figure 1.a presents the problem we want to solve, where a rounded node is a sensor and a square node corresponds to a base station (destination node). Suppose we have a MWSN where at time t_1 there is a communication path between the source sensor node n_1 and the base station. However, at time t_2, the node n_2 moves away from the node n_3, causing a communication disruption for transmitting information from n_1 to the base station. Once n_3 has realized of this problem at time t_3, n_3 has to perform routing corrections in order to reestablish the communication path between n_1 and the base station. This reestablishment can be perfectly performed using routing techniques, but at the expense of introducing undesired delays for building again the communication path between n_1 and the base station. In some applications these delays can be omitted because do not affect the application purpose, but in other cases, such as delay sensitive applications like military or health applications, this disadvantage might mean a very low network performance in terms of delay.

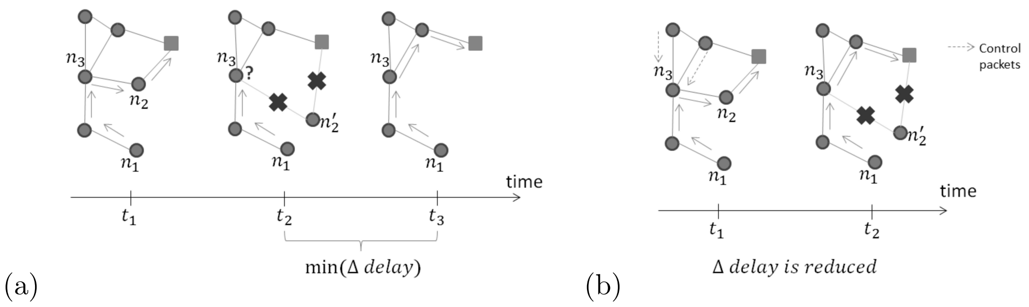


Figure 1: Problem definition: (a) Problem; (b) Solution

Given the problem above, our proposal consists to design a mathematical model for finding the minimum cost path between a source node and a destination node considering the network is moving across time [5] [6]. This formulation would be very important because it can give us optimal values for contrasting with algorithms results.

3 Problem formulation

In this section our problem is enunciated and described in detail, as well as some assumptions are shown in order to simplify our proposed mathematical models.

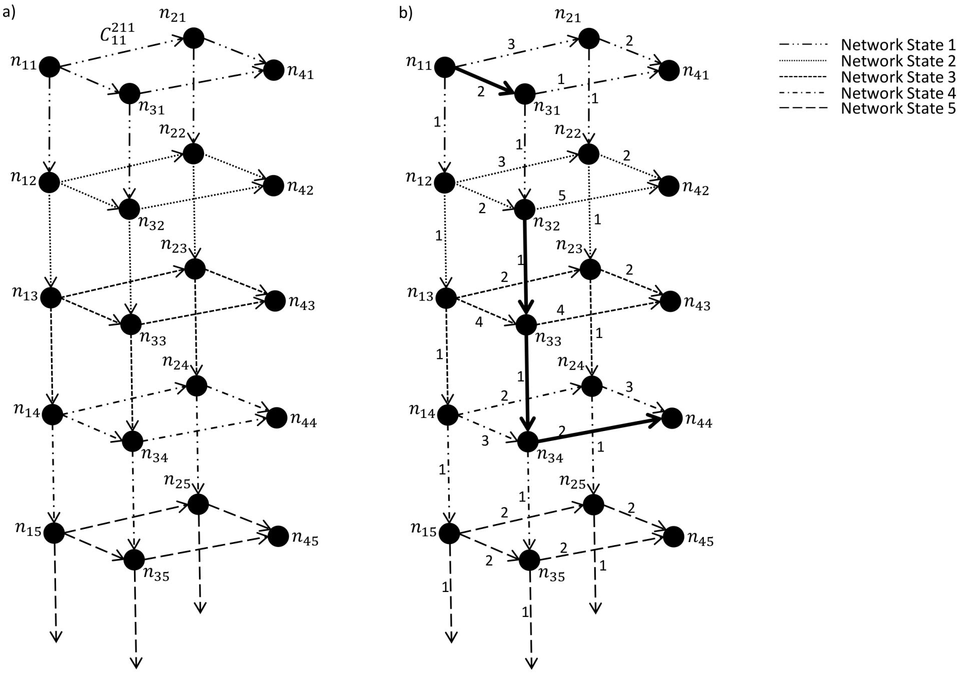


Figure 2: Problem scenario

Based on the Figure 2, we will describe our problem:

- Mobile Network: Assume we have a mobile network, at which the nodes position changes across time periods. For this reason, the links cost between the network nodes also changes across time periods. This means that at each time period the network has particular links cost, different from the links cost at other time period. Given that at each time period the network have different links cost, we could say this reflects the network state in a given time period. For this reason, each network at a given time period will be called *Network State*. For instance, the *Network State* at time period *1* is called *Network State 1*, the *Network State* at time period *2* is called *Network State 2*, and so on. In other words, according to the Figure 2.a we have an initial network (Network State 1) compound by 4 nodes. As these nodes conform a network, there are interrelations between them that we will call *Links*. These links have a cost, which can be represented, for example, by the distance, the signal to noise ratio or the RSSI measurement between the nodes. In the next time period, the network costs at the Network State 1 change and then these new interrelations between the nodes are now the *Network State 2*. As the next time period occurs, the network at the Network State 2 becomes the network at the Network State 3, and this network will be the network at the network State 4, and so on.

- Nodes: Each node is denoted as n_{it} where i is number of the node and t is the network state of the node. Depending on the communication range, a node can communicate with another node in the direction described by the Figure 2. For example, n_{11} can communicate with n_{21} and n_{31}.

- Buffer in each node: In telecommunication networks, a router or a sensor (a node) can decide not-sending its message, storing it in a buffer until it would be appropriated to send it. In our model, this situation is represented as a link between n_{11} and n_{12}, meaning that n_{11} can store its message in its buffer, that is, the node n_{12}.

- Costs: As it was mentioned previously, a link has a cost. Then, there is a cost for sending a message from n_{11} to n_{21} called C_{11}^{21l}, and denoted as C_{it}^{jul}, that is, this is the cost to carry a message form the node i at the state t to the node j at the state u at the Network State l.

- Directed graph: In this example our goal consists to carry a message from the node 1 to the node 4. Then, our *Source* node is the node 1, and our *Destination* node is the node 4. In this sense, a directed graph is constructed from the Source to the Destination. For this reason, the links direction points to the Destination.

- Goal: Our goal consists to carry a message from the Source node to the Destination node using the neighbours nodes as forwarding nodes for passing a message, and even using the buffers, if it is necessary, for waiting an appropriated situation for sending the message. In this sense, we have to find the minimum cost path between a Source node and a Destination node considering the network is changing across time, that is, through the Network States. Additionally, for simplicity we assume only one link can be selected for sending the message per each Network State. This means that if a message is at the node n_{11}, this node at this Network State 1 can send a message to only one neighbour, n_{21} or n_{11}, or storing it in its buffer, that is, n_{12}.

- Example Result: According to the example shown in the Figure 2.b and based on the links cost, the minimum cost path from the Source node, n_{11}, to the Destination node, node 4, is the path compounded by the highlighted links: n_{11} to n_{31}, n_{32} to n_{33}, n_{33} to n_{34} and n_{34} to n_{44}. As we will describe later in the mathematical formulation, this result can be also expressed in terms of X: $X_{11}^{311} = 1$, $X_{32}^{332} = 1$, $X_{33}^{343} = 1$ and $X_{34}^{444} = 1$. The rest of X_{it}^{jul} values are zero.

4 Mathematical model solution

Two mathematical models are proposed, which achieve same optimal solutions since they are equivalents, but the second one is mathematically more complete than the first one.

4.1 First approach

In this section is proposed a mathematical model for finding the minimum cost path between a source node and a destination node considering the network is changing across time. The Table 1 presents the sets, parameters and decision variables for building the mathematical model.

Next, our mathematical model is enunciated.

$$min \sum_{i\in N} \sum_{t\in S} \sum_{j\in N} \sum_{u\in S} \sum_{l\in S} C_{it}^{jul} X_{it}^{jul} \tag{1}$$

Subject to:

Table 1: Sets, parameters and variables description

Sets and Parameters	**Description**
N	Nodes set.
S	States set.
o	Source node.
d	Destination node.
st	State at which we want to obtain the minimum cost path from the *Source* to the *Destination*.
C_{it}^{jul}	Link cost from the node i at the state t to the node j at the state u at the network state l.

Variables	**Description**
X_{it}^{jul}	Determines if the link at the state l from the node i at the state t to the node j at the state u is selected for building the path towards the *Destination* (Binary variable).
$Y_{i,l}$	Determines if the node i at the state l is selected as a forwarding node for building the path towards the *Destination* (Binary variable).

$$\sum_{j\in N}\sum_{u\in S} X_{it}^{jul} Y_{jl} = 1 \qquad \forall i \in N \mid i = o, \quad \forall t, l \in S \mid t = 1, l = 1 \tag{2}$$

$$\sum_{i\in N}\sum_{t\in S}\sum_{j\in N}\sum_{u\in S} X_{it}^{jul} Y_{im} Y_{jl} = 1 \qquad \forall l, m \in S \mid l < st, m = l - 1 \tag{3}$$

$$\sum_{j\in N} Y_{jl} = 1 \qquad \forall l \in S \mid l \leq st \tag{4}$$

$$\sum_{i\in N}\sum_{t\in S} X_{it}^{jul} Y_{im} Y_{jl} = 1 \tag{5}$$
$$\forall j \in N \mid j = d, \quad \forall u, l, m \in S \mid l < st, u = l = st, m = l - 1$$

The equation 1 corresponds to the objective function, which attempts to find the X_{it}^{jul} variables at the minimum cost. The expression 2 establishes that once a X_{it}^{jul} is selected considering that $i = o$, $t = st1$ and $l = NetworkState1$, we can know the forwarding node j for sending the message. The equation 4.1 determines the predecessor node i at the network state m (previous to the network state l) required for building the path. The expression 3 allows to be coherent the forwarding and predecessor nodes in the intermediate states, that is, the network states different from the Network State 1 and the Network State indicated by the parameter st. Finally, the equation 4 assures that only one X_{it}^{jul} must be selected at each Network State from the Network State 1 up to the Network State indicated by the parameter st.

In summary, this mathematical model gives us the minimum cost path by introducing the following parameters: the Source node, the Destination node and the State (the Network State) at which we want to obtain the minimum cost path from the Source node to the Destination node. However, in this model we cannot prescind from the State parameter, that is, we cannot know the Network State (from all the Network States) at which we would obtain the minimum path cost. For this reason, it is required to use this mathematical model in an iterative way, that is, obtaining the minimum path cost for all the Network State and then, selecting the lowest of the minimum paths obtained. This process is represented in the Algorithm 1.

Algorithm 1 Solution pseudocode

```
$o = source;\ d = destination;\ k = number\ of\ states$
$minLocal = \infty; minState = 0$
for $i = 1\ to\ k$ do
    $minSolution = MathModel(o,\ d,\ i)$
    if $minSolution < minLocal$ then
        $minLocal = minSolution$
        $minState = i$
    end if
end for
```

In this algorithm, the mathematical model is used for each Network State, and once we have analysed all the Network States, we finally determine which one obtained the minimum cost path , denoted as *minSolution*, at which Network State, expressed as *minState.* In summary, with the mathematical model and this algorithm we can obtain the minimum cost path given a Source node and a Destination node.

4.2 Second approach

In this section is presented a second mathematical model proposed for constructing a minimal cost path from a source node to a destination node considering a mobile network. Notice that this mathematical model yields same results as the first one, but is mathematically better described.

The variables for this new approach are described in the Table 2. The sets and parameters are the same from the Table 1.

Table 2: Variables description for the Second Approach

Variables	**Description**
X_{it}^{jul}	Determines if the link at the state l from the node i at the state t to the node j at the state u is selected for building the path towards the *Destination* (Binary variable).
$Y_{i,l}$	Determines if the node i at the state l is selected as a forwarding node for building the path towards the *Destination* (Binary variable).
D_{jl}	Determines if the node j is selected at the destination state l (Binary variable).
DS_l	Determines if the state l is selected as a destination state (Binary variable).

Next, the second mathematical model is described.

$$min \sum_{itjul} C_{it}^{jul} X_{it}^{jul} \tag{6}$$

Subject to:

$$\sum_{l>1} D_{jl} = 1 \qquad \forall j \in N \tag{7}$$

$$\sum_{l} D_{jl} = 1 \qquad \forall j \in N \mid j \neq Destination \tag{8}$$

$$D_{jl} * DS_l = D_{jl} \qquad \forall j \in N; \quad \forall l \in S \tag{9}$$

$$\sum_{l} DS_l = 1 \tag{10}$$

$$DS_l = 0 \qquad \forall l \in S \mid l = 1 \tag{11}$$

$$DS_l \sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jul} Y_{im} D_{jl} = DS_l \qquad \forall l, m \in S \mid m = l - 1 \tag{12}$$

$$DS_l \sum_{i} Y_{im} = DS_l \qquad \forall l, m \in S \mid m \leqslant l \tag{13}$$

$$DS_l \sum_{i} Y_{im} = 0 \qquad \forall l, m \in S \mid m > l \tag{14}$$

$$\sum_{i \in N} \sum_{t \in S} \sum_{u \in S} \sum_{l \in S} X_{it}^{jul} = 1 \qquad \forall j \in N \mid j = Destination \tag{15}$$

$$DS_l \sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jum} Y_{in} Y_{jm} = DS_l \qquad \forall l, m, n \in S \mid m > 1 \wedge m = l \wedge n = m - 1 \tag{16}$$

$$DS_l \sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jum} = DS_l \qquad \forall l, m \in S \mid m \leqslant l \tag{17}$$

$$\sum_{i \in N} \sum_{t \in S} \sum_{j \in N} \sum_{u \in S} X_{it}^{jul} Y_{jl} = 1 \qquad \forall i \in N \mid i = Source \quad \forall l \in S \mid l = 1 \tag{18}$$

The equation 6 corresponds to the objective function, which will try to find the X_{it}^{jul} variables with the less possible cost C_{it}^{jul}. The previous expressions are explained in the following items:

- Destination State Constraints (from 7 to 15): The following expressions are referred to the Destination State, that is, the state at which the Destination node is found at the minimum possible cost.
  - Defining D_{jl}: D_{jl} allows to obtain the Destination State l at which the Destination node j is found at the minimum possible cost. The expression 7 avoids that D_{jl} will be one at the first state. The equation 8 avoids D_{jl} will be one for a node different from the destination node.
  - Defining DS_l: DS_l allows to extract only the Destination State l at which the Destination node is found at the minimum possible cost. The expression 9 allows to know the state l at which D_{jl} was selected. The equation 10 indicates that only one destination state is possible. In the expression 11 we assume it is not possible that the destination state will be the first state.

- Selecting the forwarding node: The forwarding node indicates the node selected at each state for constructing the minimum cost path. The expressions 12 and 13 restricts to one the number of Y_{jl} for each State less than the Destination State. The equation 14 restricts to zero the number of Y_{jl} for each State higher than the Destination State. The expression 15 indicates that it is possible only one link to the Destination node for all states, that is, only one state is selected, and for the rest of the states, the link must be zero.

- Intermediate State Constraints: These constraints allow selecting the predecessor node Y_{im} based on the current forwarding node Y_{jl}. In order to understand what these two types of nodes means, let's see an example. If we have a link between the nodes 1 and 2 in the direction from 1 to 2, the current forwarding node is 2 and the predecessor node is 1. The expression 16 allows to select the predecessor node at the intermediate states, where intermediate states refers to the states between the Destination and the Source States. The equation 17 restricts to one the number of X_{it}^{jul} for each state equal or less than the Destination State.

- Source State Constraint: The Source State indicates the State at which the Source node starts to construct the minimum cost path. The expression 16 restricts to one the number of X_{it}^{jul} for the Source state.

- Defining the First State Solution Constraint: All the constraints described above allow to find the minimum cost path between a Source node and a Destination node through several Network States. However, up to now our model does not consider the Destination State can be the first network state. For this reason it is necessary to apply the following post-processing pseudocode:

Algorithm 2 Post-processing pseudocode

```
parameters $Source,\ Destination$
$minSolution = MathModel(Source,\ Destination)$
$costFirstState = C_{it}^{jul} \quad | \quad i = Source, j = Destination, t = u = l = 1$
if $costFirstState < minSolution$ then
    $minSolution = costFirstState$
end if
```

This pseudocode basically indicates that if the cost between the Source and the Destination node is less than the solution found by the mathematical model, then the solution is at the first state, otherwise the solution is given by the mathematical model.

5 Algorithms proposals

To contrast the mathematical model results we have designed two kinds of algorithms, which will be described below:

5.1 Single path with connection to the sink (SIPCOS)

As we saw previously in the mathematical model, our goal consists to find the minimum cost path between a Source node and a Destination node. We are going to assume that a link cost represents the delay for sending the information from a node to another one. For simplicity, we assume that each link cost (delay) will be proportional to the distance between two nodes in the network. Therefore, our algorithm will try to reach the Destination node at the minimum delay possible from the Source node.

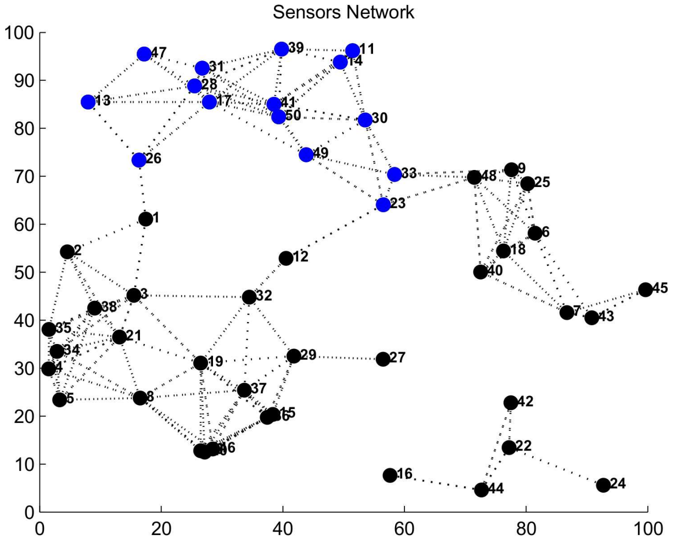


Figure 3: Problem scenario

In the Figure 3 is presented the following example. Suppose the source node is the node 3, and the Destination node is the node 50. Our proposal includes two methods. The first one consists to perform a partial broadcast from the Destination node to the closest neighbours. Now, these neighbours, the blue ones (called *connected nodes to the destination*) know how to reach quickly the Destination node in such case that a message arrives to them. The second method consists to send a message from the Source node to forwarding nodes in order to reach a connected node to the Destination. This method intends to reach as fast as possible the destination node, trying to obtain similar values respect to the mathematical model values. The pseudocodes of these two methods are described below.

Algorithm 3 Destination node algorithm

```
$d = destination$
$r = number\ of\ rounds$
$dn = neighbours(d)$                              ▷ $dn$: destination neighbours
while $r > 0$ do
    $dn = neighbours(dn)$
    $sendControlMessage(dn)$
    $r = r - 1$
end while
```

The Algorithm 2 defines a simple technique to select the neighbours nodes connected to the Destination node. This technique consists to send from the Destination node a control message to the Destination neighbours, and these ones send this control messages to its neighbours many times as the parameter *round* allows it.

The Algorithm 3 corresponds to the Forwarding Node Algorithm. This algorithm is initially performed in the Source node. Therefore, this node selects a neighbour node based on its low cost to transmit the message in order to reach a connected node to the Destination or the Destination itself. Once a neighbour node has received the message from the Source node, it has to find a new neighbour node for passing the message. This process is performed for the next selected neighbour nodes until a connected node or the destination node has been reached. Remember that a connected node will allow us to reach rapidly the Destination node in such case that a message arrives to them. This is possible since the connected node is linked directly to the Destination node or to other connected node which possibly will be connected directly to the Destination node. Once a neighbour node is selected for sending the message, this node is added to the array *path*. Then, when the Destination is reached, the array *path* contains the message's traceability from the Source node to the Destination node.

Algorithm 4 Forwarding node algorithm for SIPCOS

```
parameter $s$                                        ▷ Source Node
parameter $d$                                        ▷ Destination Node
parameter $dN$                                       ▷ Destination Neighbours Nodes
array $path = []$                                    ▷ The Building Communication Path
variable $fn = s$                                    ▷ $fn$: Forwarding Node
function $FN(i)$                                     ▷ $FN$: Forwarding Neighbours Function
$minCost = \infty$
$DestinationNeighbourFound = 0$
array $path = [path \quad fn]$
while $message \notin dN \vee d$ do
    $forwardingNeighbours = neighbours(fn)$
    for $i = 1$ to $forwardingNeighbours$ do
        if $FN(i) \in dN$ then
            $DestinationNeighbourFound = 1$
            $sendMessage(FN(i))$
            $path = [path \;\; FN(i)]$
        end if
    end for
    if $DestinationNeighbourFound == 0$ then
        for $i = 1$ to $forwardingNeighbours$ do
            $neighbourCost = cost(fn,\ FN(i))$
            if $neighbourCost < minCost$ then
                $fn = FN(i)$
                $sendMessage(fn)$
                $path = [path \;\; fn]$
            end if
        end for
    end if
end while
if $message \in dN$ then
    $dN = neighbours(dN)$
    if $Destination \in dN$ then
        $sendMessage(d)$
        $path = [path \;\; d]$
    else
        $sendMessage(dN)$
        $path = [path \;\; dN]$
    end if
end if
```

5.2 Single path without connection to the sink (SIP)

In contrast to the SIPCOS algorithm, the SIP algorithm does not take into account the connected nodes to the Destination. For this reason, the algorithm must find exactly the Destination. Therefore, it is too much difficult to find the Destination using this method. The pseudocode of this algorithm is described in the Algorithm 4.

Algorithm 5 Forwarding node algorithm for SIP

```
parameter $s$
parameter $d$
$array\ path = []$
variable $fn = s$
function $FN(i)$
$minCost = \infty$
array $path = [path\ \ fn]$
$DestinationFound = 0$
while $message\ \notin\ d$ do
    $forwardingNeighbours = neighbours(fn)$
    for $i = 1\ to\ forwardingNeighbours$ do
        if $forwardingNeighbours(i)\ ==\ d$ then
            $DestinationFound = 1$
            $sendMessage(d)$
            $path = [path\ \ FN(i)]$
        end if
    end for
    if $DestinationFound == 0$ then
        for $i = 1\ to\ forwardingNeighbours$ do
            $neighbourCost = cost(forwardingNode,\ FN(i))$
            if $neighbourCost < minCost$ then
                $forwardingNode = FN(i)$
                $sendMessage(fn)$
                $path = [path\ \ fn]$
            end if
        end for
    end if
end while
if $message\ \in\ d$ then
    $path = [path\ \ d]$
end if
```

6 Results

The graphs for the three scenarios, at its first Network State, are shown in the Figure 4.

There were chosen 3 basic scenarios in order to test in a basic way the behavior of the mathematical model, the SIPCOS and SIP algorithms. The results are shown in the following tables:

Table 3: Results for the mathematical model

Parameters and Variables	Scenario 1	Scenario 2	Scenario 3
Nodes	7	10	12
States	14	20	24
Source node	1	1	1
Destination node	7	10	12
Solution Path	1,3,7	1,2,8,7,10	1,8,12
Delay	2	4	2

According to the first scenario, the SIPCOS algorithm showed the same delay respect to the optimal value, while the SIP algorithm used an extra delay to reach the destination. This extra delay is a low value because this scenario corresponds to a small network. For this reason, if a network has few nodes, there is more probable the SIP algorithm reach the SIPCOS's performance. Once the network has increasing in size, such in case the scenario 2 or 3, the SIP's

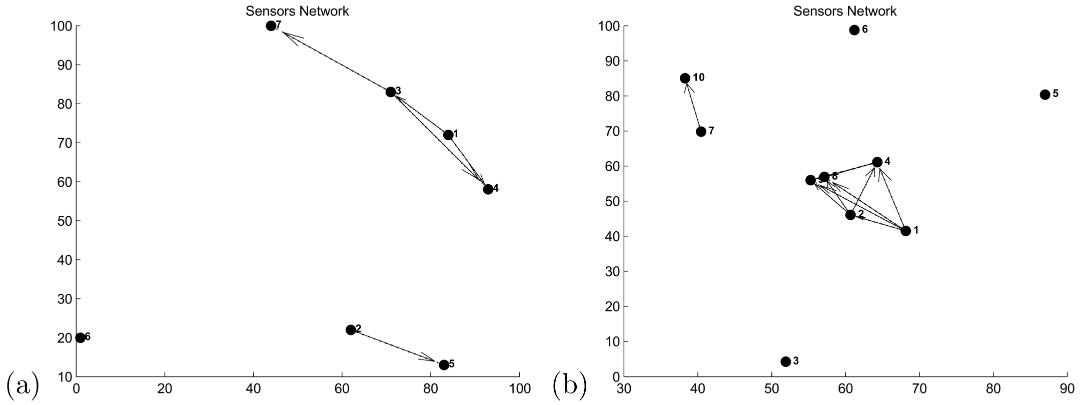


Figure 4: Problem definition: (a) Scenario 1; (b) Scenario 2

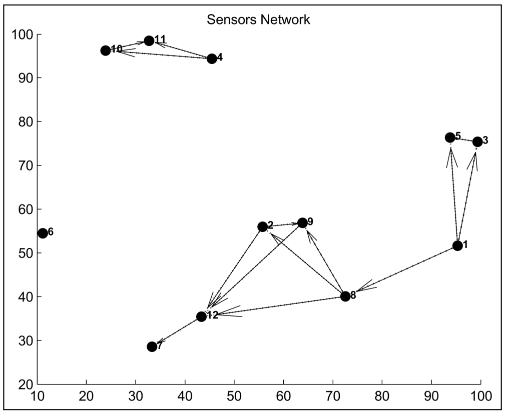


Figure 5: Scenario 3

Table 4: Results for SIPCOS algorithm

Parameters and variables	Scenario 1	Scenario 2	Scenario 3
Nodes	7	10	12
States	14	20	24
Source node	1	1	1
Destination node	7	10	12
Solution Path	1,3,7	1 2 9 7 10	1,8,2,12
Delay	2	4	3

Table 5: Results for SIP algorithm

Parameters and Variables	Scenario 1	Scenario 2	Scenario 3
Nodes	7	10	12
States	14	20	24
Source node	1	1	1
Destination node	7	10	12
Solution Path	1,4,3,7	1,2,8,9,7,10	1,3,5,9,2,12
Delay	3	5	5

performance decreases. Finally, for the third scenario the SIPCOS algorithm presented a better performance respect to the SIP algorithm, thanks to the connected nodes which allowed reach the destination node as fast as possible. In this scenario the SIP algorithm lose too much time in the nodes 3 and 5, because it had not a clear strategy to go out from ending nodes.

Conclusions

In this paper we propose a mathematical model which is able to find the minimum cost path between a source node and a destination node considering a mobile network. Additionally, there were proposed two algorithms, the SIPCOS and SIP algorithms, in order to compare their results against the mathematical model. The results showed the SIPCOS has a better performance compared against the SIP algorithm because it has efficient strategies for find quickly the destination node or destination neighbours, which facilitates the fact of building and finding the communication path from the source node to the destination node. For this reason, the SIPCOS's performance was very similar to the mathematical model and it could be a good option to be used in MWSN.

Bibliography

[1] I. F. Akyildiz and M. C. Vuran (2010); *Wireless Sensor Networks*, Wiley, 2010.

[2] J. Zheng and A. Jamalipour (2009); *Wireless Sensor Networks: A Networking Perspective*, Wiley-IEEE Press, 2009.

[3] A. A. Ahmed (2013); An enhanced real-time routing protocol with load distribution for mobile wireless sensor networks, *Computer Networks*, 57(6):1459-1473.

[4] A.A. Ahmed (2007); *Real-Time Wireless Sensor Networks*, University of Virginia, 2007.

[5] B. Buchli, F. Sutton, J. Beutel (2012); GPS-Equipped Wireless Sensor Network Node for High-Accuracy Positioning Applications, *Wireless Sensor Networks Lecture Notes in Computer Science*, Springer, 7158:179-195.

[6] S. Li, X. Ma, X. Wang, M. Tan (2011); Energy-efficient multipath routing in wireless sensor network considering wireless interference, *Journal of Control Theory and Applications*, 9(1):127-132.

[7] G. M. de Araujo, J. Kaiser and L. B.Becker (2014); Genetic Machine Learning Approach for Link Quality Prediction in Mobile Wireless Sensor Networks, *Cooperative Robots and Sensor Networks*, 1-14.

[8] G. M. de Araujo, J. Kaiser, L. B.Becker (2012); An Optimized Markov Model to Predict Link Quality in Mobile Wireless Sensor Networks, *Procedia Computer Science*, 10:1100-1105.

[9] J. A. Torkestani Young (2012); Mobility prediction in mobile wireless networks, *Journal of Network and Computer Applications*, 35(5):1633-1645.

[10] G. A. Montoya and Y. Donoso (2016); A Delay-Sensitive Mathematical Model Approach and a Distributed Algorithm for Mobile Wireless Sensor Networks, *Computers Communications and Control (ICCCC), 2016 6th International Conference on*, IEEE Xplore doi: 10.1109/ICCCC.2016.7496736, 45-50.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(6):832-844, December 2016.

General Model for Adequate Cloud Service Selection using Decision Making Methods

O. Pantelić, A. Pajić, A. Nikolić

Ognjen Pantelić*, Ana Pajić
Department of Information Systems
Faculty of Organizational Sciences, University of Belgrade
Jove Ilica 154, 11000 Belgrade, Serbia
*Corresponding author: pantelico@fon.bg.ac.rs
ana.pajic@fon.bg.ac.rs

Ana Nikolić
Novomatic Lottery Solutions
Djordja Stanojevica 12, 11070 Novi Beograd, Serbia
anikolic@novomaticls.com

Abstract: Cloud Computing (CC) is a technology that surely brings innovations in today's business world, more and more companies around the world are widely incorporating this technology into their businesses. From a technical, as well as organizational point of view transferring enterprise IT to the Cloud is a complex task. Various factors have to be taken into consideration in order to make a right choice when moving IT services to the Cloud.

The goal of this paper is to identify and to discuss in detail all factors that influence organization's decision to adopt Cloud. General model for Cloud adoption, introduced in Pantelić et al. [13][a], consists of the key factors driving the organizational benefits when moving to the Cloud. The purpose of the model is to support decision makers in evaluating the benefits, risks and costs of using Cloud Computing. In this paper the general model is extended with two new aggregation methods for harmonization of alternatives rankings in a group decision process. We present the results of two new methods using the method results from previous research [13], as rank inputs, into an aggregate (group) preference. The idea is to find consensus ranking that minimizes disagreement among previous methods results.

There were no strong differences between the results of performed methods. The results have shown that Software as a service model and Storage as a service model dominated according to not just arithmetic-mean method, but also to geometric-mean method.

Keywords: cloud adoption, general model, IT investments, cloud services, multi-criteria methods.

1 Introduction

During the past decade, many enterprises have shifted their focus towards the use of Cloud Computing (CC) as a viable option to reduce costs and to improve IT and business agility. The Cloud Computing model provides almost unlimited computing resources for users which are taken on demand and pay per use. It represents an extension of distributed computing and grid computing technology, which aims to deliver a variety of services over the Internet via Cloud providers. CC offers a lot of advantages to the businesses of all sizes. By moving business to the Cloud, IT decision makers can take advantage of highly efficient Cloud infrastructure

[a] Reprinted (partial) and extended, with permission based on License Number 3958430574606 ©[2016] IEEE, from "Computers Communications and Control (ICCCC), 2016 6th International Conference on".

which reduces energy consumption considerably and they are freed of problems related to the technological issues of installing and maintaining the IT. However, there are many uncertainties with regard to the actual realization of these benefits, particularly around the cost savings and the data security. Despite many Cloud applications are currently widely and successfully used, the growth of CC has slowed down. According to [6], the technical aspect of security in the Cloud is one of the main reasons for the slowing down in the growth of CC. Reference [15] underlined security, interoperability, and portability as major obstacles to broader Cloud adoption.

This research is an extension of [13] (doi: 10.1109/ICCCC.2016.7496751). The general model for Cloud adoption introduced in [13] is extended with two new methods for group decision making. Goal of the paper is to present approaches to creating the preference structure of alternatives by assigning ordinal ranks to the alternatives. Paper presents results of two new methods using the method results from our previous research, as rank inputs, into an aggregate (group) preference. The idea is to find consensus ranking that minimizes disagreement among previous methods results. In addition, we discussed in detail key factors driving Cloud adoption.

The paper is organized as follows. Section 2 points some background related to Cloud Computing and presents in detail advantages and disadvantages of different Cloud services. In section 3, we introduce a general model of Cloud adoption decision process with multi-criteria decision making methods as one of the key factors. Detailed analysis of relevant factors is reported. The purpose of the model is to support decision makers in evaluating the benefits, risks and costs of using CC. The model can be used as the guideline for enterprises that are considering moving their IT services to the Cloud. Our results from implementation of multi-criteria decision making methods, in particular Analytics Hierarchy Process (AHP), Iterative Compromise Ranking (ICOR) and Multi- Attribute Utility Theory (MAUT), are given in Section 4.1, and the results of new preference aggregation methods are given in Section 4.2. The last section contains our concluding remarks and ideas for future work.

2 Cloud Computing Service Models

There are many definitions of CC. A definition of CC by NIST (National Institute of Standards and Technology) is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [10]. Reference [3] characterize CC as a pay per use model which offering computing, storage and software "as a service". An important aim of this technology is delivering computing as a utility.

CC has always been divided into three broad service categories: Infrastructure as a Service (IaaS), offers virtualized computation, storage and communication resources to the customers on demand, Platform as a Service (PaaS), provides an environment for developing applications using languages, services and tools supported by provider, and finally Software as a Service (SaaS), provides to the consumer applications and services that run on the providers infrastructure. As new Cloud providers enter the market, even more Cloud service models are available today. Many additional models can be found in literature [16]. These service models are Network as a Service, Storage as a Service, Database as a Service, Security as a Service, Integration as a Service, Management as a Service, Testing as a Service, Information as a Service, Communication as a Service and Monitoring as a Service. They share many similarities but have their own distinct differences as well. We will analyze in more detail below the characteristics of these models.

Cloud service models describe how cloud services are made available to clients. The most widely used Cloud service model is SaaS that offers an on demand online software subscription. SaaS provides significant efficiencies in cost and delivery by removing the need for technical staff to install, manage and upgrade software. Moreover, it reduces the cost of licensing software. The consumer does not control the underlying Cloud infrastructure and IT operational functions, which cause important concern related to security, performance and service availability. Some software customization is often available, but this is more limited than with PaaS and IaaS models. With PaaS, consumers have control over the deployed applications while the service provider delivers the computing platform at predictable service level and cost. The platform consists of required software tools and it is hosted on the hardware infrastructure of the service provider. In this model, it is not necessary to hire people to maintain the platform and hardware infrastructure. Benefits for using PaaS include the ability to eliminate cost of upkeep for hardware and the consumers can concentrate on building software. The downside is that the developers must work within the constraints of the platform which may cause data portability and application interoperability issues. On the other side, consumers have more real control over their infrastructure with IaaS than with PaaS or SaaS model. IaaS model allows automated deployment of servers, processing power, networking and storage. Consumers have direct access to their infrastructure resources, just as they would with traditional infrastructure but optimize infrastructure spend. The downside is that some applications may not run on available infrastructure of service provider due to their very specific hardware requirements. "Compared to IaaS offerings, applications riding on PaaS deliver better performance due to the intrinsic cloud support for the programming platform" highlighted Buyya et al. in [3]. Core datacenter infrastructure consists of storage, servers, the network, and management tool for infrastructure maintenance and monitoring. Each of these components has created a separate market niche. It enables the design of efficient in-network services. Network as a Service is not a new concept, but a new business model for delivery efficient in-network services [5]. It enables the design of efficient in-network services. However, the same concern about the Cloud provider's ability to guarantee high service availability appears. Storage as a Service allows consumers to manage data storage with no need for IT staff and to align IT spends with data consumption. With Storage as a Service, disaster recovery of data in the Cloud is offered. The drawbacks are long latencies and limited bandwidth [14]. Database as a Service delivers database functionality as a service and consumers can deploy new databases quickly, securely and cheaply. The downside is a lack of control over network performance issues [16]. On the other side, Gartner is predicting a growth of the Cloud-based security service market [11]. Security as a Service is data and application protection solution, which is usually included in SaaS model. Businesses employ Integration as a Service in order to share data between systems as well as with one or more external applications in real-time. The ultimate goal is the process improvement, to gain agility and automate business processes. Besides, Management as a Service model offers a suit of SaaS based applications to ensure cost-effective IT management operations. Consumers can easily analyze, monitor and evaluate performance metrics to ensure the optimization of IT resources [18]. The Monitoring as a Service offerings are focused on monitoring the insights of IT infrastructure assets. It provides the complete overview of the infrastructure, servers, networks, storage as well as the applications. Privacy concerns are still the main obstacles in adapting this model. Communication as a Service is known as a collection of different vendors' services that facilitate enterprise communications. The core concept of this model is that using this service is very convenient. Furthermore, Information as a Service integrates the information to provide real-time and high quality information about a business entity [17]. The last model is Testing as a Service which delivers the on-demand test execution. It is most suitable for test executions that do not require in-depth knowledge of the system.

3 General Model Elements

There are many factors that determine which Cloud service model is right for the specific organizational needs. A review of the literature reveals that factors impacting Cloud adoption tend to be psychological as well as technical. The decision on migrating IT services to the Cloud should take into consideration a variety of socio-technical factors. It is not sufficient to examine simply the cost of deploying their IT systems on the Cloud.

In [13] the general model for Cloud adoption was briefly presented, aiming to support the Cloud adoption decision process. The model consists of seven key factors driving the organizational benefits when moving to the Cloud. Decision makers can use the model to evaluate the benefits, risks and costs of using Cloud. It is important that decision makers view Cloud adoption project from different perspectives in order to fully understand Cloud services and their various pros and cons. As most of the pros and cons are unquantifiable, making the decision about Cloud services today's biggest concern. It is important to notice a list of the major factors that weight into the decision. Once decision makers understand the key factors driving Cloud adoption, the next challenge is to select a most suitable Cloud service to migrate their IT services.

Therefore, in this section we thoroughly evaluate key factors that influence on organization decision to adopt Cloud Computing. The starting point must be to determine a company's objectives. It is important to consider what the company wants to achieve by migrating IT services to the Cloud. Ideally any company wants to increase profit, while decreasing cost. Cloud computing has the ability not only to decrease costs, but also to run the business more effectively. It can reduce the cost of equipment, energy, and infrastructure administration and management, needed to run the same business. Moreover, the business can remain current on technological advances. The company will be able to invest in new projects for direct business benefits, which eventually increases business value. Although Cloud Computing has many advantages over traditional techniques, it does not necessarily mean that Cloud will improve the business. Cloud migration can result in a considerable amount of organizational changes. For example, system administrators will no longer have complete control of IT infrastructure and services. In order to take advantages of the Cloud, the company should look to align Cloud service with business requirements and objectives. From an enterprise perspective, flexibility, scalability, and business continuity are also important besides cost. One of the most significant barriers to adoption is security, mainly the migration of critical applications and sensitive data to public cloud environment [8]. Perhaps organizations will not completely outsource their back-end computing resources and activities to Cloud provider. Rather, they will engage in a partnership with more than one Cloud provider and will establish heterogeneous computing environments.

The influence of Cloud services on global economy and environment is also significant. Cloud Computing seems to be a new model for doing business with a potential to change global industries' business models. Investing in Cloud globally will create new jobs and economic opportunities. It will allow developing economics to become more competitive driving more investments. Additionally, there is in fact a risk of IT sector job loss from investing in Cloud. For example, companies are more likely to move jobs related to maintaining infrastructure to the Cloud. IT departments will not need server engineers, systems administrators, or data center managers, however, they will still need good network engineers, IT managers and business analysts to support applications. Traditional roles will change and expand in development flexible application that suits Cloud environment. There will be new opportunities for IT staffs if they adjust their skills and focus more on systems integration, system security and data privacy. The organization will be more collaborative as employees can work from any location, any time. Furthermore, adopting cloud technologies requires changes in organizational culture and people's perception of new technologies. Employees may be fear on how new technologies will affect their work, which can result in employees resistance. Thus, employees' perception of Cloud benefits is possible barrier to Cloud adoption.

Cloud services also reducing IT sector's carbon footprint. Cloud providers operate in a massive scale, so energy consumption, the number of harmful substances and waste is significantly lower compared to the traditional in-house business. Since Cloud Computing paradigm is based on sharing configurable computer resources by multiple users, the number of equipment, facilities, and resources for business can be greatly reduced or completely eliminated. This way of doing business can certainly lead to a more sustainable IT infrastructure and processes. In July 2011, The Carbon Disclosure Project commissioned a study that examined financial and carbon benefits of Cloud [4]. A total of eleven organizations in the United States that have been using Cloud Computing for at least two years participated in the study. Detailed case studies revealed a 50 percentage reduction of carbon in 2020 with adopting Cloud technologies versus no Cloud at all. The potential carbon reductions of 85.7 million tons per year by 2020 for multi billion dollars United States companies is comparable to the annual emissions from 16.8 million passenger vehicles. The report shows that organizations can achieve an annual energy use reduction of 12.3 billion dollars by 2020.

The next step is CC evaluation based on financial metrics. As already mentioned, CC saves money but it might lead to financial loss due to security risks, possible data access problems, etc. Researchers recommend using several financial metrics to evaluate how well Cloud would be beneficial for enterprises. Decision makers should be able to use these metrics to assess what Cloud offering should best meet business and technical requirements. Metrics are used to understand the factors needed for clear distinction between two and more different Cloud offerings. These metrics are used for measurement of the business value of IT. There are direct and indirect financial metrics. Direct metrics refers to indicators that measure financial gain or loss at limited or no distance from the production functions, such as a metric used to measure investment performance. By contrast, indirect methods measure general business and IT performance. Most commonly used metrics are Cost Benefit Analysis (CBA), Return on investment (ROI), and Total cost of ownership (TCO) [9].

Cost Benefit Analysis is widely used direct financial metric which simply calculate the value of the benefits, and subtracts the financial cost associated with it. The payback method is built into the analysis. It measures the length of time required to regain the benefits of investment to repay its cost. The period of time could be one month, one year, or three years mostly. It lacks of the time value of money (TVM) function, which is considered reliable for investments whose benefits expected to be received over longer periods of time. On the other side, Net present value (NPV) analysis considers the time value of money through the use of discount rate. Discount rate is here the most critical decision variable used to define the present value of an investment. The present value of an investment is always less than or equal to the future value at the end of determined period of time. Academics' and practitioners' recommendations regarding discount rate are ranged from 1 to 15 percent. Net benefits are simple calculated by the sum of benefits discounted at the discount rate minus cost. If Cloud Computing project has a positive NVP value then it is expected to be a good decision. According to the Investopedia, Return on investments (ROI) is a cash flow metric used to compare the efficiency of a number of different investments. When risks and other factors between the choices are equal, the investment with the higher ROI is considered the better decision. Most forms of ROI analysis compare the benefits of investment with the cost of the investment. The result is represented as a ratio or percentage. A positive ROI result means that investment returns is higher than cost. If Cloud business case scenario has a positive ROI, it can be considered as investment with net gain. A simple ROI equation is not adjusted for the time value of money. Therefore, it cannot be trusted to measure returns over long periods of time. Investing in Cloud Computing can deliver significant long term benefits. It is good practice to use ROI metric discounted for present value. The result will be considerably less than the unadjusted ROI. Cloud Computing investment will be still worthwhile, if the discounted ROI result is higher than company's minimum acceptable rate of return.

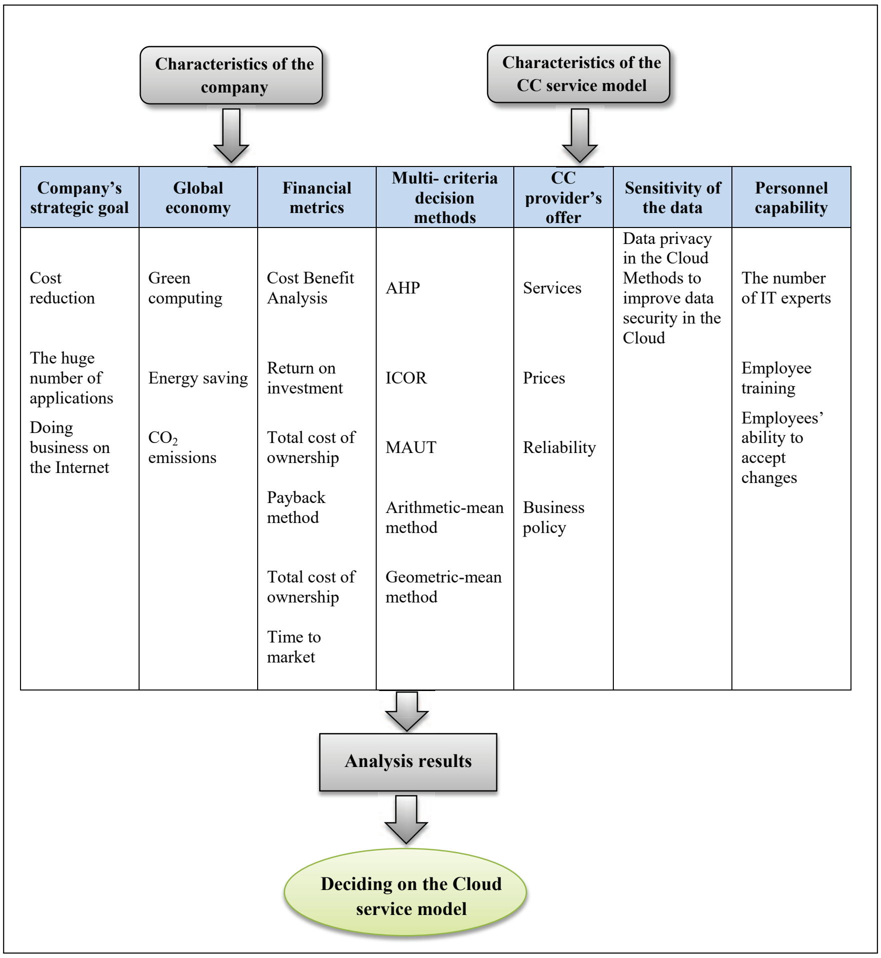
Up to this point, we have discussed the three most common direct metrics. In the following, we introduced Total cost of ownership (TCO) and Time to market as indirect metrics. Measuring the business impact of IT functions begins with comprehensive TCO methodology. TCO tries to quantify all tangible expenses, as well as some intangible ones that can be assigned a monetary value. Total costs are a critical component of the IT value equation, and TCO analysis includes the costs of deploying an IT product across the product's whole life cycle. For Cloud technologies, expenses may depend on selected Cloud service and selected Cloud provider. There are costs associated with Cloud license and maintenance fees, integration process, and employee training. At the same time, Cloud can reduce cost for software and hardware upgrades, infrastructure cost, energy cost, labour costs, etc. Many cloud providers will reveal a TCO analysis to show the overall costs and benefits associated with migrating to their Cloud platform. Hence, Cloud service evaluation can be done by comparing TCO of Cloud solution with TCO of traditional IT departments within enterprise [9]. For IT department, the agility to deploy new applications and enter new markets is important key performance indicator. It can be measured by Time to Market (TTM) metrics which measures the length of time to lunched new product or service. Lower TTM leads to an increase in the company's competitive advantage. As TTM increases, the risk of losing customers also increases. With Cloud Computing, speed to market can be reduced while significantly reducing the cost.

However, it is no longer sufficient to measure only financial performance [12]. Instead, it is necessary to use multi-criteria decision analysis methods to determine the value of intangible assets. By using these methods, decision makers can rate and apply weights to quality attributes of different Cloud service models, based on their preferences. The trend is to combine two or more methods to make up for shortcomings in any single particular method. Analytic Hierarchy Process (AHP), Iterative Compromise Ranking (ICOR) and Multi- Attribute Utility Theory (MAUT) are a well-known multi-criteria decision making methods [7]. MAUT approach uses a function called Utility. Decision maker assigns a utility value to each alternative. This utility reflects the alternatives importance to the decision. The utility values can range from 1 to 10, where 1 represents the lowest importance and 10 the highest. Decision maker gives a score to each of alternatives for each attribute. The final ranking value of the alternative is calculated as the sum of each alternative's score for each attribute multiplied by the weight of that attribute. The alternative with the best value of the aggregated function is considered as the best of the alternatives. MAUT method is applied in all types of problems which have significant amount of uncertainty, such as financial, energy management problems, etc. Reference [1] underlined AHP method as simpler form of MAUT. The utility function is calculated using the technique based on pairwise comparisons. The weights for the criteria and the relative performance scores for each of the alternatives on each criterion are determined by this technique. The relative scale from 1 to 9 is used to measure importance between two criteria. The higher number means the chosen criterion is considered more important than other criterion being compared with. The scores are assigned for each alternative option for each criterion using normalized pairwise comparisons matrix of identified criteria. ICOR is the third method described in this paper. It defines a narrow set of possible solutions closest to the ideal solution. The weight is assigned to each criterion and represents its relative importance. The compromise solution can be established by different strategies. The example is the strategy of "the majority of criteria". This strategy is demonstrated in our previous research [13].

We now propose to use two different group decision making methods to rank a finite set of alternatives. Arithmetic-mean and geometric-mean as aggregation methods are introduced. We want to present some approaches to creating the preference structure of alternatives by simply assigning ordinal ranks to the different alternatives. These methods will be further discussed throughout the following sections.

Choosing a Cloud provider is also a multiple attributes decision making problem. An increasing number of companies are offering variety of Cloud Computing services to enterprises. Amazon Web Services and Salesforce.com are regarded as the pioneers of Cloud Computing. Google, Microsoft, IBM, GoGrid, and Rackspace have also entered the Cloud market. So many Cloud providers create healthy competition among providers to meet the Service Level Agreement (SLA) requirements. The provider is obligated to meet performance objectives defined by the SLA metrics. Despite different protecting strategies and technologies for solving security and privacy problems, the service agreement signed with service provider becomes the only guarantee. It is necessary to assess security risks implied by the act of embedding resources within the CC environment. A major consideration should be data security and data availability of company's critical data in the Cloud. The location of the data centers and servers must be known to ensure the security regulations are followed. Cloud provider should have very limited access to sensitive and critical data. These added security features can result in higher providers' prices.

The extended general model for Cloud adoption is presented in Figure 1. The extended model is consist of two new aggregation methods. The Figure 1 shows seven key factors and brief explanation of each, with the aim to support appropriate analysis of different aspects.


Characteristics of the company
Characteristics of the CC service model

Company's strategic goal	Global economy	Financial metrics	Multi- criteria decision methods	CC provider's offer	Sensitivity of the data	Personnel capability
Cost reduction	Green computing	Cost Benefit Analysis	AHP	Services	Data privacy in the Cloud Methods to improve data security in the Cloud	The number of IT experts
The huge number of applications	Energy saving	Return on investment	ICOR	Prices		Employee training
Doing business on the Internet	CO_2 emissions	Total cost of ownership	MAUT	Reliability		Employees' ability to accept changes
		Payback method	Arithmetic-mean method	Business policy		
		Total cost of ownership	Geometric-mean method			
		Time to market				

Figure 1: Extended general model for Cloud adoption

4 Methods and Results

4.1 Decision- making using AHP, MAUT and ICOR Methods

In this section, we briefly present the results from methods that previously have been applied to ranking different Cloud services [13]. Experts from the academic and the business world provide criteria weights and take part in ranking different alternatives. Table 1 summarized the criteria considered in our research, while Table 2 summarized the alternatives.

Criteria
Data security
Data availability
The monthly average price
Ease of use
Flexibility
Consumer Scope of Control
The level of customization
The level of Cloud provider support
The risk of changing Cloud provider
Overall business growth
Company size
Number of IT experts

Table 1: Defined set of criteria

Alternatives
Software as a Service
Platform as a Service
Infrastructure as a Service
Network as a Service
Storage as a Service
Database as a Service
Security as a Service
Integration as a Service
Management as a Service
Testing as a Service
Information as a Service
Communication as a Service
Monitoring as a Service

Table 2: Defined set of alternatives

The methods results are presented in Table 3. From these results, we can conclude that fifth alternative, Storage as a Service model, is the alternative with the largest utility values. The flexibility and ease of use may be the reason for the highest score. On the other hand, it has a low impact on business's growth. The alternative with the lowest utility values is Platform as a Service. With PaaS, consumer needs to put in development effort to create and test applications and to have control over the deployed applications. The problem appears in understanding of who is responsible for which areas of CC architecture.

The resulting weights of the AHP method show Software as a Service model as the alternative with the highest weighted preference. Software as a Service model is the most suitable model because it provides significant efficiencies in cost and Cloud provider is fully responsible for all components of CC architecture. However, decision makers cannot neglect existing security challenges. The lowest weighted preference has Platform as a Service model.

Using IKOR method decision makers get the solution that is the closest to the ideal. The alternative option that is the closest to the ideal is Software as a Service model, while on the last position in ranking list is Network as a Service model. The disadvantage of creating the specific virtual network is the need to control the devices from the Cloud.

Alternatives	**MAUT method**	**AHP method**	**ICOR method**
Software as a Service	8.105	0.110	1
Platform as a Service	6.393	0.057	11
Infrastructure as a Service	6.662	0.062	10
Network as a Service	6.424	0.071	13
Storage as a Service	8.447	0.106	2
Database as a Service	7.328	0.065	6
Security as a Service	6.982	0.058	9
Integration as a Service	7.648	0.082	3
Management as a Service	6.994	0.058	7
Testing as a Service	8.152	0.088	4
Information as a Service	7.351	0.071	18
Communication as a Service	7.129	0.069	12
Monitoring as a Service	7.204	0.092	5

Table 3: The final ranking of the alternative options using MAUT, AHP and ICOR method

4.2 Group decision making using Aggregation Methods

In many group decision making problems, individual preferences on a given set of alternatives are obtained from several decision makers and then these individual preferences are aggregated to a single collective preference. For group problem solving, the individuals participating in making a group decision assign ranks to the same finite set of alternatives. The idea is to harmonize opinions of the participants in the group in order to select the most acceptable alternative for the whole group.

We will use in our approach two types of preference aggregation methods for harmonization of alternatives rankings [2]. These methods require, as the input data, individual preferences on a same set of alternatives and derive the consensus ranking that minimized disagreement among a group. They are considered as a single criterion analysis, since the individuals compare alternatives directly. Here we present the approach to implement these methods using alternatives rankings obtained by the previous demonstrated decision making methods in [13]. Each method's ranking is combined to produce an overall assessment. In this way we critically evaluate results of previous research. Moreover, it reduces subjectivity of assigning individual preference values to alternatives by decision makers.

First method is arithmetic-mean method based on arithmetic mean of the ranking made by all decision makers. The overall preference of an alternative is estimated by arithmetic mean.

The result is an aggregated value that is between maximum and minimum value. The most acceptable alternative has the lowest average overall rate. Geometric-mean method is similar to arithmetic-mean method. It uses the geometric mean of individual ranks instead of the simple arithmetic mean to determine the average rank. The alternative with the lowest value is the most acceptable. A geometric mean is always less than or equal to an arithmetic mean. It tends to reduce the effect of very high and very low values, which helps in eliminating extreme values (outliers).

First we define individual ordering matrix of alternatives. Rows of a matrix relate to the alternatives, columns relate to methods results. Elements of individual ordering matrix represent alternative assigned rank for a method. The elements are shown in Table 4.

Alternatives	**AHP method ranking**	**MAUT method ranking**	**ICOR method ranking**
Software as a Service	1	3	1
Platform as a Service	13	13	11
Infrastructure as a Service	10	11	10
Network as a Service	7	12	13
Storage as a Service	2	1	2
Database as a Service	9	6	6
Security as a Service	12	10	9
Integration as a Service	5	4	3
Management as a Service	11	9	7
Testing as a Service	4	2	4
Information as a Service	6	5	8
Communication as a Service	8	8	12
Monitoring as a Service	3	7	5

Table 4: Individual ordering matrix of alternatives

There were no strong differences between the results of arithmetic-mean and geometric-mean methods. Consensus is achieved for the most acceptable alternative and the next ranked alternative. Software as a service model and Storage as a service model is ranked at first and second place respectively. These alternatives dominated according to all three decision making methods. MAUT method ranked the Software as Service alternative in third place, while AHP and ICOR method elected this alternative as the most favorable Cloud service model. The reason for this result might be the significant reduction in IT cost and ultimately the provider is responsible for maintaining the software. We want to underline the slight difference in geometric mean value of these two alternatives. Moreover, two alternatives have the same average value of rankings. Table 5 reports arithmetic mean and geometric mean value of an alternative.

Furthermore, there is significant disagreement over the alternatives which are ranked the lowest. In these methods, Platform as a Service is ranked as the worst solution. The rank order of Security as a Service, Network as a Service, and Infrastructure as a Service, alternatives differs. It shows higher or lower score level of these alternatives in one of the previous decision making methods. Infrastructure as a Service alternative is valued considerably higher by MAUT method, while Security as a Service is valued considerably lower by AHP method.

Alternatives	**Arithmetic mean value**	**Geometric mean value**
Software as a Service	1.667	1.442
Platform as a Service	12.333	12.296
Infrastructure as a Service	10.333	10.323
Network as a Service	10.667	10.298
Storage as a Service	1.667	1.587
Database as a Service	7.000	6.868
Security as a Service	10.333	10.260
Integration as a Service	4.000	3.915
Management as a Service	9.000	8.849
Testing as a Service	3.333	3.175
Information as a Service	6.333	6.214
Communication as a Service	9.333	9.1588
Monitoring as a Service	5.000	4.718

Table 5: Arithmetic-mean method and geometric-mean method results

Conclusion

Cloud Computing is a technology that surely transforms the way corporate IT services are delivered and managed. Decision to migrate existing services to the Cloud can be complicated as selecting the most favorable Cloud service model is is a rather complex process. The paper described the characteristics of different Cloud services with the aim of helping decision makers to fully understand organizational benefits and drawbacks of CC. The focus was on developing general model for Cloud adoption with the aim to support decision makers to investigate Cloud adoption decisions and to make the right choice.

The paper is an extension of [13] where different Cloud service models are evaluated using three multi criteria decision making methods, in particular AHP, ICOR and MAUT methods. The criteria preference values have been assigned by the authors after discussions with experts from different stakeholder groups. The final ranking of the alternative options for all three methods were reported. In this paper we thoroughly evaluated seven key factors of general model for Cloud adoption. Because most of the pros and cons of Cloud are unquantifiable, it is important that decision makers view Cloud adoption project from different perspectives in order to fully understand Cloud services. Furthermore, decision makers have to compare the cost of different Cloud providers, deployment options and usage scenarios. By using decision making methods, decision makers can rate and apply weights to quality attributes of different Cloud service models, based on their preferences. In the present paper we have introduced some other decision making methods to show another view of ranking Cloud services. The general model for Cloud adoption is extended with two new aggregation ranking methods for harmonization of alternatives ranking for group problem solving. The feasibility and usefulness of the application of arithmetic-mean method, and geometric-mean method, have been demonstrated. The new approach used alternatives rankings obtained by the previous demonstrated decision making methods. The results have shown that Software as a service model and Storage as a service model dominated according to not just arithmetic-mean method, but also geometric-mean method confirmed this. Furthermore, results proved Platform as a Service model as alternative with the worst preferences.

Therefore, we can highlight that consumers prefer Cloud service models with low risk solution implementation. In all demonstrated methods, Platform as a Service is ranked as one of the worst solution. This is due to the fact that the developers must work within the constraints of the platform. We can conclude that factors like: cost, ease of use and disclosed scope of controls between provider and consumer, are the main factors which determine the prioritization of Cloud service models. Data security and privacy protection issues did not play a key role in selecting right Cloud service model as we expected. The findings of this study are a base for a future work consisting in the concrete case study implementations. Certainly there may be additional factors which may have not been considered within this paper. It is also possible to evaluate other group decision making methods and types of preference aggregation methods for harmonization of alternatives rankings.

Bibliography

[1] Alexander, M. (2012); Decision-Making Using the Analytic Hierarchy Process (AHP) and JMP ® Scripting Language, *2012 The SouthEast SAS Users Group Conference (SESUG)*, SESUG Inc.

[2] Alfares, H. K.; Duffuaa, S. O.(2008); Determining Aggregate Criteria Weights From Criteria Rankings By a Group of Decision Makers, *International Journal of Information Technology & Decision Making*, 7(4): 769-781.

[3] Buyya, R. et al (2008), *Cloud computing: Principles and paradigms*, 8: 3-121.

[4] Carbon Disclosure Project 2011, Carbon Disclosure Project Study 2011 Cloud Computing-The IT Solution for the 21st Century, Verdantix, White Paper.

[5] Costa, P.; Migliavacca, M.; Wolf, A. L. (2012); NaaS: Network-as-a-Service in the Cloud,*2nd USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Service*, San Jose, CA, USA.

[6] Cruz, Z.; Fernández-Alemán, J.; Toval A. (2015); Security in cloud computing: A mapping study, *Computer Science and Information Systems*, 12(1): 161-184.

[7] Figueira, J.; Greco, S.; Ehrogott, M. (2005),*Multiple Criteria Decision Analysis: State of the Art Surveys*, Springer Science+Business Media, Inc., 27-406.

[8] Hashizume, K. et al (2013); An analysis of security issues for cloud computing, *Journal of Internet Services and Application*, 4(5): 1-13.

[9] Kornevs, M.; Minkevica, V.; Holm, M. (2012); Cloud Computing Evaluation Based on Financial Metrics, *Information Technology and Management Science*, 15(1): 87-92.

[10] Mell, P.; Grance, T. 2011, The NIST Definition of Cloud Computing, NIST Special Publication 800-145, National Institute of Standards and Technology Gaithersburg.

[11] Messmer, E. (2013), Gartner: Cloud-based security as a service set to take off, http://www.networkworld.com/article/2171424/data-breach/gartner–cloud-based-security-as-a-service-set-to-take-off.html, *Information Technology and Management Science*

[12] Pajić, A.; Pantelić, O.; Stanojević B. (2014); Representing IT Performance Management as Metamodel, *International Journal of Computers Communications & Control*, 9(6): 758-767.

[13] Pantelić, O.; Pajić, A.; Nikolić, A.; (2016); Analysis of available cloud computing models to support cloud adoption decision process in an enterprise, *Computers Communications and Control (ICCCC), 2016 6th International Conference on*, IEEE Xplore, e-ISBN:978-1-5090-1735-5, doi: 10.1109/ICCCC.2016.7496751, 135-139.

[14] Patel, K. H. et al (2012); Tradeoffs between performance and security of cryptographic primitives used in Storage-as-a-Service for cloud computing, *Proceedings of the CUBE International Information Technology Conference*, New York, NY, USA.

[15] Pătraşcu, A.; Patriciu, V. V. (2015); Logging for Cloud Computing Forensic Systems Related work, *International Journal of Computers Communications & Control*, 10(2): 222-229.

[16] Sharma, S. (2015), Evolution of as-a-Service Era in Cloud, http://web.cs.iastate.edu/˜sugamsha/articles/Evolution%20of%20as-a-Service%20Era%20in%20Cloud.pdf

[17] Simmonds, M. (2009), Information as a service: A smarter way to SOA success, IBM Software Group, 1-19.

[18] Waschke, M. (2004), IT Management-As-A-Service, CA Technologies, Whitepaper, no. 451.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(6):845-859, December 2016.

Stochastic Stability Analysis of Power Control in Wireless Networks via a Norm-inequality-based Approach

R. Qian, Y. Qi

Rongrong Qian, Yuan Qi*
Beijing Univ. of Posts and Telecommunications (BUPT)
Beijing, 100876, China
rongrongqian@bupt.edu.cn
*Corresponding author: qiyuan@bupt.edu.cn

Abstract: Owing to the requirements from realistic wireless networks, the stochastic stability analysis for discrete-time power control, which concerns the randomness brought by the fading channels and noise of wireless systems, is of practical significance. By developing a norm-inequality-based framework of analyzing the stochastic stability of linear systems with random parameters, we show that a typical power-control law with linear system model is stable in the sense of the pth-moment stability. Several conditions of achieving the pth-moment stability for the considered power-control law are obtained, which can easily applied to realistic wireless networks. Besides, within this study, the stability analysis of power control for the first time takes into account the effect of multiple-access methods.

Keywords: wireless networks, power control, stochastic stability.

1 Introduction

There has been a great deal of research over the past several decades on the power control of wireless networks. These studies span multiple disciplines and include information theory [1, 8], communication [2, 6, 7, 9, 13], and control theory [3–5, 21]. It has already been recognized that the stability analysis of power-control laws can efficiently investigate the intrinsic properties of various power-control algorithms [3–5, 21].

The power-control laws with two-sided scalable iterative (interference) functions are convergent under the condition that an equilibrium exists [1], and the laws with contractive interference functions guarantee the existence and uniqueness of equilibriums along with linear convergence of iterates [2]. The global asymptotic stability of power control laws involving two-sided scalable interference functions and the exponentially scalability of laws with contractive interference functions are seen even under bounded time-varying delays [3]. A general class of power-control laws whose interference functions are monotonic and scalable are considered in [4, 5]. By employing appropriately constructed Lyapunov functions, [5] shows that any bounded power distribution obtained from these laws is uniformly asymptotically stable. Further, in [5] Lyapunov-Razumikhin functions are used to show that, even when the system incorporates time-varying delays, any solution along which the generalized system nonlinearity is bounded must also be uniformly asymptotically stable. In both of above cases the stability is shown to be global. Most of current wireless networks are digital communication systems, in which the link gains are random (stochastic) variables fluctuating as wireless channels of the underlying networks are experiencing fading at all time and the noises are also random variables. It is therefore important for the power-control laws to be designed and verified when considering the impact of randomness in discrete-time wireless systems with fading channels and random noise.

Stochastic power-control algorithm that uses noisy interference estimates (observations) is first studied in [6]. With conventional matched filter receivers, the stochastic power control is shown by [6] to converge to the optimal power vector in the mean square error sense. These results are later extended to the cases when a linear receiver or a decision feedback receiver is used [7, 22]. In [9], a stochastic-approximation based power-control algorithm is proposed to handle both measurement errors and randomness in the channel gain matrix, which is proved to converge to the optimal solution in the mean-squared sense.

Treading the elegant footsteps of recent works [3]- [5], one can gain a deep insight into the stability theory of typical power-control laws in wireless networks. Rather than being concerned with the stability analysis of power control, the studies [6]- [9] focus on developing extra techniques of reducing the impact of randomness encountered by power control in wireless networks, where these techniques includes matched filter receiver, decision feedback receiver, and stochastic approximation. To the best of our knowledge, it still lacks of systemic study on stability analysis of typical power-control laws taking into account the randomness existed in practical wireless networks while the power control does not use extra randomness reducing techniques proposed by [6]- [9]. We shall emphasize here that, this kind of study is indispensable, because, on the one hand, it could attentively reveal the inherent attributes of the typical power-control laws when the randomness exists (but no randomness reducing technique is involved) so as to make the stability theory of typical power-control laws complete, on the other hand, extra technique of reducing randomness may not necessarily be available in practical systems due to the objective factors such as realtime processing demands so that engineers have to be aware of the stability of power control in randomness environments without any randomness reducing technique. Hence, the aim of our work is to perform this study; specifically, we will develop a framework of stochastic stability analysis for discrete-time power control, which takes the randomness brought by the fading channels and noise of wireless systems into consideration.

Our main works are: (i) developing a norm-inequality-based framework of analyzing the stochastic stability (to be specific, the pth-moment stability) for linear systems with random parameters so as to investigate the stochastic stability of the power control in consideration of the randomness caused by the fading channels and noise; (ii) clarifying the conditions of achieving the stochastic stability for the considered linear systems and power-control law; and (iii) investigating the effect of multiple-access methods to stochastic stability of power control.

2 Notation and preliminaries

2.1 Notation

Throughout, the interval $[0\ +\infty)$ is denoted by $\mathbb{R}_+$, and the set of positive integers by $\mathbb{Z}_+$. The non-negative orthant of the N-dimensional real space is represented by $\mathbb{R}_+^N$. The vectors are written in **bold** lower case letters and matrices in **bold** capital letters, e.g., $\mathbf{a}$ and $\mathbf{A}$. The ith component of a vector $\mathbf{a}$ is denoted by a_i, and the ijth entry of a matrix $\mathbf{A}$ is denoted by A_{ij} such that $A_{ij} = [\mathbf{A}]_{ij}$ and $\mathbf{A} = [A_{ij}]$. The notation $\mathbf{A} \geq 0$ means that all of the components of $\mathbf{A}$ are greater than or equal to zero. The inequality $\mathbf{A} \geq \mathbf{B}$ implies that $a_{ij} \geq b_{ij}$ for all components ij. We let $(\cdot)^T$ denote the transpose of a vector or a matrix. If $\mathbf{a}$ is a vector with components $a_1, a_2, \cdots, a_N$, then its p-norm is defined by $\|\mathbf{a}\|_p = \left(\sum_{i=1}^N |a_i|^p\right)^{1/p}$ and its Euclidean norm by $\|\mathbf{a}\|_2 = \left(\sum_{i=1}^N |a_i|^2\right)^{1/2}$ that is actually p-norm with $p = 2$. For a square matrix $\mathbf{A}$, the induced norm corresponding to the p-norm of vectors is defined as

$$\|\mathbf{A}\|_p = \max_{\|\mathbf{x}\|_p \neq 0} \frac{\|\mathbf{A}\mathbf{x}\|_p}{\|\mathbf{x}\|_p} = \max_{\|\mathbf{x}\|_p = 1} \frac{\|\mathbf{A}\mathbf{x}\|_p}{\|\mathbf{x}\|_p}, \tag{1}$$

where $\|\mathbf{A}\|_1 = \max_j \sum_{i=1}^N |A_{ij}|$ is also known as the maximum column sum matrix norm, and $\|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^N |A_{ij}|$ is the maximum row sum matrix norm.

A probability space is a triple $(\Omega, \mathcal{F}, P)$ where Ω is a set of "outcomes", $\mathcal{F}$ is a set of "events", and $P : \mathcal{F} \to [0\ \ 1]$ is a function that assigns probabilities to events. If x is a random variable on $(\Omega, \mathcal{F}, P)$ then we define the expected value operator to be $Ex = \int xdP$. If $Ex^2 < +\infty$ then the variance of x is defined to be $\text{var}(x) = E(x - Ex)^2$. We let $\{\boldsymbol{x}[k], k \in \mathbb{Z}_+\}$ denote a stochastic process with random values in a set of vectors, and $\{\boldsymbol{X}[k], k \in \mathbb{Z}_+\}$ denote a stochastic process with random values in a set of matrices, by writing $\boldsymbol{x}[k]$ and $\boldsymbol{X}[k]$ in *italic* and **bold** letters. If $\boldsymbol{x}$ is a $N \times 1$ random vector then we define its expected value as $\bar{\mathbf{x}} = E\boldsymbol{x} \triangleq [Ex_1\ Ex_2 \cdots Ex_N]^T$. Analogously, for a $N \times N$ random matrix $\boldsymbol{X}$, we define its expected value as $\bar{\mathbf{X}} = E\boldsymbol{X} \triangleq [EX_{ij}]$ and its variance as $\text{var}(\boldsymbol{X}) \triangleq [\text{var}(X_{ij})]$.

Let $L^p(\Omega, \mathcal{F}, P)$ be the set of measurable function f on Ω such that $\int_\Omega |f|^p d\mu < +\infty$, we introduce an operator E_{L^p} as

$$E_{L^p} f \triangleq (E|f|^p)^{1/p}\,. \tag{2}$$

From [15, 2.2.5 Example], one shall find that $L^p(\Omega, \mathcal{F}, P)$ is a linear space and E_{L^p} is a semi-norm.

2.2 Preliminaries

In this part, we collect basic properties and definitions of matrix theory, algebra theory, probability theory, and stochastic stability theory, which will be used in the following analysis. For more details, see, e.g., [10, 11, 23–25].

Basic norm inequalities [23, 25]: The p-norm of vectors and the corresponding induced norm of square matrices are nonnegative numbers have the properties that

1. $\|\mathbf{x} + \mathbf{y}\|_p \le \|\mathbf{x}\|_p + \|\mathbf{y}\|_p$ and $\|\mathbf{A} + \mathbf{B}\|_p \le \|\mathbf{A}\|_p + \|\mathbf{B}\|_p$;
2. $\|\mathbf{A}\mathbf{x}\|_p \le \|\mathbf{A}\|_p \|\mathbf{x}\|_p$, which is derived from the definition of $\|\mathbf{A}\|_p$;
3. $\|\mathbf{A}\mathbf{B}\|_p \le \|\mathbf{A}\|_p \|\mathbf{B}\|_p$, since $\|\mathbf{A}\mathbf{B}\mathbf{x}\|_p \le \|\mathbf{A}\|_p \|\mathbf{B}\mathbf{x}\|_p \le \|\mathbf{A}\|_p \|\mathbf{B}\|_p \|\mathbf{x}\|_p$ and
$$\max_{\|\mathbf{x}\|_p = 1} \|\mathbf{A}\mathbf{B}\mathbf{x}\|_p = \|\mathbf{A}\mathbf{B}\|_p.$$

Upper bound of induced matrix norm [17]: For any $N \times N$ matrix $\mathbf{A}$, the induced norm $\|\mathbf{A}\|_p$ has no explicit representation unless $p = 1, 2$ or ∞. However, one can have the below inequalities [17, (1.8),(1.11)]

$$\|\mathbf{A}\|_p \le N^{1-1/p} \|\mathbf{A}\|_1, \tag{3}$$

and

$$\|\mathbf{A}\|_p \le \|\mathbf{A}\|_1^{1/p} \|\mathbf{A}\|_\infty^{1-1/p}, \tag{4}$$

provide two closed-form upper bounds of $\|\mathbf{A}\|_p$ with p other than $1, 2$ and ∞.

Cauchy-Schwarz-Buniakowsky inequality involving real numbers [25]: Let $a_1, a_2, \cdots, a_N$ and $b_1, b_2, \cdots, b_N$ be any two arbitrary sets of real numbers, then

$$\left(\sum_{i=1}^N a_i b_i \right)^2 \le \left(\sum_{i=1}^N a_i^2 \right) \left(\sum_{i=1}^N b_i^2 \right). \tag{5}$$

This inequality can be expressed in the vector form as $\mathbf{a}^T \mathbf{b} \le \|\mathbf{a}\|_2 \|\mathbf{b}\|_2$, where $\mathbf{a} = [a_1\ a_2\ \cdots\ a_N]^T$ and $\mathbf{b} = [b_1\ b_2\ \cdots\ b_N]^T$.

Definition 1 [24]: Let $(\mathcal{S},\mathcal{S})$ be a measurable space. A stochastic process $\{\mathbf{\Phi}[k], k \in \mathbb{Z}_+\}$ taking values in $\mathcal{S}$ is said to be a Markov chain with respect to a filtration $\mathcal{F}_k$, if $\mathbf{\Phi}[k] \in \mathcal{F}_k$ and for all $B \in \mathcal{S}$, $P(\mathbf{\Phi}[k+1] \in B|\mathcal{F}_k) = P(\mathbf{\Phi}[k+1] \in B|\mathbf{\Phi}[k])$. In words, given the present, the rest of the past is irrelevant for predicting the value of $\mathbf{\Phi}[k+1]$.

Stability properties of stochastic systems need to be established in the context of stochastic stability, in which a variety of inter-related definitions exist [10, 11]. This study concerns the so-called pth-moment stability which is borrowed from [11] with trivial differences and defined as follows.

Definition 2 [11]: The pth-moment stability can be stated as, for each initial distribution, there exists $\lim_{k\to+\infty} E\left(\|\boldsymbol{x}[k]\|_p^p\right) < +\infty$, where $p \in \mathbb{Z}_+$, and it shall hold that $\left(E\left(\|\boldsymbol{x}[k]\|_p^p\right)\right)^{1/p} = E_{L^p}\|\boldsymbol{x}[k]\|_p$.

This study will seek for an analytical framework of deriving the upper bound of $E\left(\|\boldsymbol{x}[k]\|_p^p\right)$ so as to prove the pth-moment stability for linear systems with random parameters.

3 System model and problem statement

3.1 System model of power control

We consider a wireless network with N wireless nodes, which employs a discrete-time power control algorithm given by $\mathbf{x}[k+1] = \boldsymbol{I}(\mathbf{x}[k])$, where $\mathbf{x}[k] = [x_1[k]\ x_2[k]\ \cdots\ x_N[k]]^T$ and $x_j[k] \in \mathbb{R}_+$ is the transmitted power of node j at the kth iteration, $\boldsymbol{I}(\mathbf{x}) = [I_1(\mathbf{x})\ I_2(\mathbf{x})\ \cdots\ I_N(\mathbf{x})]^T$ and $I_j : \mathbb{R}_+^N \to \mathbb{R}_+$ is the interference function modeling the interference together with noise measured at the receivers for node j that mainly comes from other nodes and local noise source. Denote the link gain between the transmitter of node j and the receiver for node i by G_{ij}.

To perform the study in a systematic fashion, we proceed from a simple but considerably typical law of power control that has a linear system model, helping us avoid any entanglement due to nonlinear effects. This power-control law is given by

$$\boldsymbol{x}[k+1] = \boldsymbol{I}(\mathbf{x}[k]), \tag{6}$$

where $\boldsymbol{I}(\mathbf{x}[k]) = \boldsymbol{D}[k]\left(\boldsymbol{C}[k]\boldsymbol{x}[k] + \boldsymbol{n}[k]\right) = \boldsymbol{D}[k]\boldsymbol{C}[k]\boldsymbol{x}[k] + \boldsymbol{D}[k]\boldsymbol{n}[k]$, $\boldsymbol{D}[k]$ is a $N \times N$ diagonal matrix whose diagonal elements are $\left\{\frac{\gamma_1[k]}{G_{11}[k]}, \frac{\gamma_2[k]}{G_{22}[k]}, \cdots, \frac{\gamma_N[k]}{G_{NN}[k]}\right\}$, in which $\gamma_j[k]$ is the target Signal-to-Interference-and-Noise Ratio (SINR) of node j at the kth iteration, and $G_{ij}[k]$ is the link gain G_{ij} at the kth iteration. In this study, we set $\gamma_j[k] = \gamma_j$ where γ_j is fixed target SINR value for node j. $\boldsymbol{C}[k] = [C_{ij}[k]]$ is a $N \times N$ matrix whose entries are either zero or positive depending on whether the entry is diagonal or off-diagonal, i.e.,

$$\boldsymbol{C}[k] = \begin{bmatrix} 0 & G_{12}[k] & \cdots & G_{1N}[k] \\ G_{21}[k] & 0 & \cdots & G_{2N}[k] \\ \vdots & \vdots & \ddots & \vdots \\ G_{N1}[k] & G_{N2}[k] & \cdots & 0 \end{bmatrix}. \tag{7}$$

$\boldsymbol{n}[k] = [n_1[k]\ n_1[k]\ \cdots\ n_N[k]]^T$ denotes the vector of noise power at the receivers for all N nodes. Here, note that $\boldsymbol{x}[k], \boldsymbol{n}[k] \geq 0$ and $\boldsymbol{D}[k], \boldsymbol{C}[k] \geq 0$ because the powers and link gains are all positive values.

The model (6) is thought to be typical because it covers the well-known Foschini-Miljanic algorithm [12] and can be extended (in future) to describe the power-control algorithms of *opportunistic* communications e.g., the utility-based power control (UBPC) algorithm [13].

In wireless channels, fading is deviation of the attenuation affecting a signal over certain propagation media. The fading can vary with time or geographical position, and is often modeled as a stochastic process. If let $\boldsymbol{G}[k] = [G_{ij}[k]]$, now one should bear in mind that, $\{\boldsymbol{G}[k], k \in \mathbb{Z}_+\}$ and $\{\boldsymbol{n}[k], k \in \mathbb{Z}_+\}$ are two stochastic processes when the fading channels and random noise appear in the wireless networks. As a consequences, $\{\boldsymbol{D}[k], k \in \mathbb{Z}_+\}$ and $\{\boldsymbol{C}[k], k \in \mathbb{Z}_+\}$ will also be stochastic processes. To define the randomness behavior of the wireless networks with fading channels and noises, the below assumptions are always employed.

Assumption 1 (Additive White Gaussian Noise). The noises existed in the wireless networks are i.i.d. additive white Gaussian noises with average power $\delta_n^2 > 0$ such that the noise power vector $\boldsymbol{n}[k]$ satisfies $E(\|\boldsymbol{n}[k]\|_1^1) = N\delta_n^2$. The noises are independent with the link gains. In words, if $n_q[k_2]$ has distribution μ^n_{q,k_2} and $G_{ij}[k_1]$ has μ^G_{i,j,k_1}, then $(n_q[k_2], G_{ij}[k_1])$ has distribution $\mu^n_{q,k_2} \times \mu^G_{i,j,k_1}$ [24], for $1 \le i, j, q \le N$ and $k_1, k_2 \in \mathbb{Z}_+$.

Assumption 2 (Temporal Independency of Link Gains). The link gains at different iterations are independent. In essence this implies, if $G_{ij}[k_1]$ has distribution μ^G_{i,j,k_1} and $G_{pq}[k_2]$ has μ^G_{p,q,k_2}, then $(G_{ij}[k_1], G_{pq}[k_2])$ has distribution $\mu^G_{i,j,k_1} \times \mu^G_{p,q,k_2}$ [24], for $1 \le i, j, p, q \le N$ whenever $k_1 \ne k_2$.

Assumption 3 (Stationarity). The distributions of $G_{ij}[k]$ and $n_q[k]$ are unrelated to k, i.e., whatever k is, $G_{ij}[k]$ and $n_q[k]$ has distribution $\mu^G_{i,j}$ and μ^n_q, respectively, for $1 \le i, j, q \le N$.

This assumption stipulates that $\{\boldsymbol{G}[k], k \in \mathbb{Z}_+\}$, $\{\boldsymbol{D}[k], k \in \mathbb{Z}_+\}$, $\{\boldsymbol{C}[k], k \in \mathbb{Z}_+\}$, and $\{\boldsymbol{n}[k], k \in \mathbb{Z}_+\}$ are *stationary* stochastic processes which do not change their statistical properties with k.

Assumption 4 (Deployment of Multiple Access Methods). In wireless networks, multiple access methods can suppress the leakage of signal power from one node to the receivers for other nodes under certain level such that $G_{ij}[k] \le \beta_{ij} G_{jj}[k]$ with constant values $\beta_{ij} \ll 1$ for any $i \ne j$. Moreover, $G_{jj}[k], 1 \le j \le N$ are N stationary and i.i.d. random variables with $E(G_{jj}[k]) = \mu_G$, $E(1/G_{jj}[k]) = \mu_{1/G}$, $E(G^2_{jj}[k]) = \mu_{G^2}$, and $E(1/G^2_{jj}[k]) = \mu_{1/G^2}$.

This assumption holds in case of the power control of wireless networks with multiple access methods, which implies

$$\boldsymbol{C}[k] \le \begin{bmatrix} 0 & \beta_{12}G_{22}[k] & \cdots & \beta_{1N}G_{NN}[k] \\ \beta_{21}G_{11}[k] & 0 & \cdots & \beta_{2N}G_{NN}[k] \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{N1}G_{11}[k] & \beta_{N2}G_{22}[k] & \cdots & 0 \end{bmatrix}.$$

In summary, the power-control system (6) has a linear system model with random parameters. This study will analyze the pth-moment stability for such a system.

3.2 Problem statement

In what follows, we begin by considering the linear systems with random parameters as

$$\boldsymbol{x}[k+1] = \boldsymbol{A}[k]\boldsymbol{x}[k] + \boldsymbol{B}[k]\boldsymbol{n}[k], \tag{8}$$

where $\{\boldsymbol{A}[k], k \in \mathbb{Z}_+\}$ and $\{\boldsymbol{B}[k], k \in \mathbb{Z}_+\}$ be two stationary stochastic processes. $\{\boldsymbol{n}[k], k \in \mathbb{Z}_+\}$ is a stationary stochastic process of additive white Gaussian noise, and independent with $\{\boldsymbol{A}[k], k \in \mathbb{Z}_+\}$ and $\{\boldsymbol{B}[k], k \in \mathbb{Z}_+\}$. Clearly, the model of system (8) is a generalization of the model (6), which is not same but closely related to the models appeared in several existed works [10] [14] [16].

Main problem of this work. How can we estimate whether the linear system (8) is pth-moment stable or not?

Theorem. The stochastic process $\{\boldsymbol{x}[k], k \in \mathbb{Z}_+\}$ that corresponds to the state vector $\boldsymbol{x}[k]$ in (8), is a Markov chain.

Proof The conclusion directly follows from the definition of Markov chain. □

The result above is a trivial but fundamental understanding of the system (8). We then turn to taking a closer look at this system. To state the further results, we need to rewrite (8) as

$$\boldsymbol{x}[k+1] \;=\; \underbrace{\left(\prod_{u=1}^{k}\boldsymbol{A}[u]\right)\boldsymbol{x}[1]}_{\text{non-noise term}} + \underbrace{\sum_{i=1}^{k}\left(\prod_{u=i+1}^{k}\boldsymbol{A}[u]\right)\boldsymbol{B}[i]\boldsymbol{n}[i]}_{\text{noise term}}. \tag{9}$$

If the noise term does not exist in (9), the Furstenberg-Kesten theorem [14] and the analytical framework developed by Feng *et al.* [10] would be available to study the stochastic stability properties for the associated system. However now, due to the existence of the noise term, we have to seek for a new proper framework to analyze such a system. In [16], Koning provided an analytical framework which is usable to investigate (8), but such a framework can only reflect the first- and second-order statistics of $\boldsymbol{x}[k]$. In this work, we will develop a norm-inequality-based framework that is capable of analyzing higher-order as well as the first- and second-order statistics of $\boldsymbol{x}[k]$.

4 Results

In this section, the pth-moment stability of the linear system (8) is analyzed through a norm-inequality-based approach, and then the analysis is applied to the power-control system (6). To attain the main results of this study, we need to derive and use several lemmas; however, we prefer not to introduce them in the main text but, rather, in Appendix, so as not to interrupt the presentation. For details, please refer to Lemmas A.1 to A.5.

4.1 A Norm-inequality-based approach of the pth-moment stability analysis

We are now ready to perform the pth-moment stability analysis of the linear system (8) via a norm-inequality-based approach.

Theorem 1. A sufficient condition for the first-moment stability of the system (8) is $\lim_{k\to+\infty}\bar{\mathbf{A}}^k = \mathbf{0}_{N\times N}$, where $\mathbf{0}_{N\times N}$ is the $N\times N$ zero matrix.

Proof Since $\boldsymbol{x}[k+1] \geq 0$,

$E\left(\|\boldsymbol{x}[k+1])\|_1^1\right) = \|E(\boldsymbol{x}[k+1])\|_1^1 = \|\bar{\mathbf{x}}[k+1]\|_1^1 = \left\|\bar{\mathbf{A}}^k\bar{\mathbf{x}}[1] + \sum_{i=1}^{k}\bar{\mathbf{A}}^{k-i}\,\bar{\mathbf{B}}\bar{\mathbf{n}}\right\|_1^1$. If the matrix $\bar{\mathbf{A}}$ has the property that $\lim_{k\to+\infty}\bar{\mathbf{A}}^k = \mathbf{0}_{N\times N}$, $\mathbf{I}-\bar{\mathbf{A}}$ will be nonsingular and its inverse can be expressed by [23, Corollary 5.6.16]: $(\mathbf{I}-\bar{\mathbf{A}})^{-1} = \sum_{k=0}^{+\infty}\bar{\mathbf{A}}^k$, and then we shall have $\lim_{k\to+\infty}E\left(\|\boldsymbol{x}[k+1])\|_1^1\right) = \left\|(\mathbf{I}-\bar{\mathbf{A}})^{-1}\bar{\mathbf{B}}\bar{\mathbf{n}}\right\|_1^1 < +\infty$ as long as $\lim_{k\to+\infty}\bar{\mathbf{A}}^k = \mathbf{0}_{N\times N}$ holds. □

The model (8) has an alternative formulation as

$$\boldsymbol{x}[k+1] = \boldsymbol{s}_x[k] + \sum_{i=1}^{k}\boldsymbol{s}_n^k[i], \tag{10}$$

where, for notational simplicity, $\boldsymbol{s}_x[k] = \left(\prod_{u=1}^{k}\boldsymbol{A}[u]\right)\boldsymbol{x}[1]$, $\boldsymbol{s}_n^k[i] = \left(\prod_{u=i+1}^{k}\boldsymbol{A}[u]\right)\boldsymbol{B}[i]\boldsymbol{n}[i]$.

The forthcoming analysis will involve applying the operator E_{L^p} to p-norm of some random vector $\boldsymbol{x}$, i.e., substituting $f = \|\boldsymbol{x}\|_p$ into $E_{L^p}f$, or to induced matrix norm of some random matrix $\boldsymbol{X}$, i.e., substituting $f = \|\boldsymbol{X}\|_p$ into $E_{L^p} f$. There is one important issue herein that must be mentioned:

Remark 2: Computing the induced matrix norm is a nonlinear optimization problem[1], and the induced norm $\|\boldsymbol{X}\|_p$ has no explicit representation unless $p = 1, 2$ or ∞. If $p \neq 1, 2$, or ∞, the integral $\int_\Omega \|\boldsymbol{X}\|_p^p d\mu < +\infty$ might not exist, in which case we can not take E_{L^p} to $\|\boldsymbol{X}\|_p$. Therefore, in this study when it is needed to apply the operator E_{L^p} to $\|\boldsymbol{X}\|_p$, we will seek for an integrable upper-bound $\varphi(\mathbf{X})$ of $\|\boldsymbol{X}\|_p$ and use $E_{L^p}\varphi(\mathbf{X})$ for theoretical analysis.

Theorem 2. Assume that $\|\mathbf{X}\|_p$ has an upper-bound $\varphi(\mathbf{X}) > 0$, i.e., $\|\mathbf{X}\|_p \leq \varphi(\mathbf{X})$, where $\int_\Omega |\varphi(\mathbf{X})|^p d\mu$ is integrable. A sufficient condition for the pth-moment stability of the system (8) is $E_{L^p}\varphi(\boldsymbol{A}) < 1$, $E_{L^p}\varphi(\boldsymbol{B}) < +\infty$, and $E_{L^p}\|\boldsymbol{n}\|_p < +\infty$.[2]

Proof By (10), we get

$$\begin{aligned} E_{L^p}\left\|\boldsymbol{x}[k+1])\right\|_p &= E_{L^p}\left\|\boldsymbol{s}_x[k] + \sum_{i=1}^{k} \boldsymbol{s}_n^k[i]\right\|_p \leq E_{L^p}\left\|\boldsymbol{s}_x[k]\right\|_p + \sum_{i=1}^{k} E_{L^p}\left\|\boldsymbol{s}_n^k[i]\right\|_p \\ &\leq \prod_{u=1}^{k} E_{L^p}\varphi(\boldsymbol{A}[u]) E_{L^p}\|\boldsymbol{x}[1]\|_p + \sum_{i=1}^{k}\prod_{u=i+1}^{k} E_{L^p}\varphi(\boldsymbol{A}[u]) E_{L^p}\varphi(\boldsymbol{B}[i]) E_{L^p}\|\boldsymbol{n}[i]\|_p, \end{aligned}$$

where the first inequality follows from Lemma A.4 and the second one is from Lemma A.5.

Under the assumption of stationarity, $E_{L^p}\varphi(\boldsymbol{A}[k])$, $E_{L^p}\varphi(\boldsymbol{B}[k])$, and $E_{L^p}\|\boldsymbol{n}[k]\|_p$ shall not change with k such that we can drop k for notational simplicity. Thus,

$$\begin{aligned} E_{L^p}\left\|\boldsymbol{x}[k+1])\right\|_p &\leq \left[E_{L^p}\varphi(\boldsymbol{A})\right]^k E_{L^p}\|\boldsymbol{x}[1]\|_p + \sum_{i=1}^{k}\left[E_{L^p}\varphi(\boldsymbol{A})\right]^{k-i} E_{L^p}\varphi(\boldsymbol{B}) E_{L^p}\|\boldsymbol{n}\|_p \\ &= \left[E_{L^p}\varphi(\boldsymbol{A})\right]^k E_{L^p}\|\boldsymbol{x}[1]\|_p + \frac{1-\left[E_{L^p}\varphi(\boldsymbol{A})\right]^k}{1-E_{L^p}\varphi(\boldsymbol{A})} E_{L^p}\varphi(\boldsymbol{B}) E_{L^p}\|\boldsymbol{n}\|_p. \end{aligned}$$

If $E_{L^p}\varphi(\boldsymbol{A}) < 1$, $E_{L^p}\left\|\boldsymbol{B}\right\|_p < +\infty$, and $E_{L^p}\|\boldsymbol{n}\|_p < +\infty$, then

$$\lim_{k\to+\infty} E\left(\|\boldsymbol{x}[k+1]\|_p^p\right) \leq \left(\frac{E_{L^p}\varphi(\boldsymbol{B}) E_{L^p}\|\boldsymbol{n}\|_p}{1-E_{L^p}\varphi(\boldsymbol{A})}\right)^p < +\infty.$$

Therefore, Theorem 2 is justified. □

Theorem 2 holds for all $p \in \mathbb{Z}_+$, thus it can reveal any pth-order statistics of $\boldsymbol{x}[k]$. It's a progress made by the proposed analytical framework of this study, compared to the framework developed by [16] that can only investigate the first- and second-order statistics of $\boldsymbol{x}[k]$. Since the derivation methods of Theorems 1 and 2 are not exactly same, the sufficient condition of the first-moment stability obtained by Theorem 1 is not necessarily identical as that by Theorem 2 with $p = 1$.

Theorem 3. If $E_{L^p}\varphi(\boldsymbol{A}) < 1$, $E_{L^p}\left\|\boldsymbol{B}\right\|_p < +\infty$, and $E_{L^p}\|\boldsymbol{n}\|_p < +\infty$, there exists $\alpha < +\infty$, such that $\lim_{k\to+\infty} E\left(\|\boldsymbol{x}[k])\|_p^p\right) = \alpha$.

Proof Let us begin by assuming that $k > j > J$. We get $E_{L^p}\left\|\boldsymbol{x}[j+1])\right\|_p \geq E_{L^p}\left\|\sum_{i=1}^{j} \boldsymbol{s}_n^j[i]\right\|_p$,

[1] An already known approach is to make use of the algorithm of estimating the induced matrix norm as well as the Matlab routines provided by Higham [17].

[2] Note that $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{n}$ are short for $\boldsymbol{A}[k]$, $\boldsymbol{B}[k]$, and $\boldsymbol{n}[k]$ by dropping the index k, since the statistics of $\boldsymbol{A}[k]$, $\boldsymbol{B}[k]$, and $\boldsymbol{n}[k]$ are irrelevant with k under the stationarity assumption.

because $\boldsymbol{s}_x[j]$ and $\boldsymbol{s}_n^j[i]$ $(i, j \in \mathbb{Z}_+,\ 1 \leq i \leq j)$ are all positive vectors. By (10), we can have

$$E_{L^p} \left\|\boldsymbol{x}[k+1])\right\|_p \leq E_{L^p} \left\|\boldsymbol{s}_x[k]\right\|_p + E_{L^p} \left\|\sum_{i=k-j+1}^{k} \boldsymbol{s}_n^k[i]\right\|_p + E_{L^p} \left\|\sum_{i=1}^{k-j} \boldsymbol{s}_n^k[i]\right\|_p . \qquad (11)$$

Due to the stationarity property, it holds true that $E_{L^p} \left\|\sum_{i=1}^{j} \boldsymbol{s}_n^j[i]\right\|_p = E_{L^p} \left\|\sum_{i=k-j+1}^{k} \boldsymbol{s}_n^k[i]\right\|_p$. Then, with $E_{L^p}\varphi(\boldsymbol{A}) < 1$, $E_{L^p} \left\|\boldsymbol{B}\right\|_p < +\infty$, and $E_{L^p}\|\boldsymbol{n}\|_p < +\infty$, it implies that

$$\begin{aligned} E_{L^p} \left\|\boldsymbol{x}[k+1])\right\|_p - E_{L^p} \left\|\boldsymbol{x}[j+1])\right\|_p &\leq E_{L^p} \left\|\boldsymbol{s}_x[k]\right\|_p + E_{L^p} \left\|\sum_{i=1}^{k-j} \boldsymbol{s}_n^k[i]\right\|_p \\ &\leq \left[E_{L^p}\varphi(\boldsymbol{A})\right]^k E_{L^p}\|\boldsymbol{x}[1]\|_p + \frac{[E_{L^p}\varphi(\boldsymbol{A})]^j}{1 - E_{L^p}\varphi(\boldsymbol{A})} E_{L^p}\varphi(\boldsymbol{B}) E_{L^p}\|\boldsymbol{n}\|_p. \end{aligned} \qquad (12)$$

Now we can conclude that, $\forall \varepsilon > 0$, $\exists J > 0$, such that $E_{L^p} \left\|\boldsymbol{x}[k+1])\right\|_p - E_{L^p} \left\|\boldsymbol{x}[j+1])\right\|_p < \varepsilon$ as long as $k, j \geq J$. It states that $E_{L^p} \left\|\boldsymbol{x}[j+1])\right\|_p$ has a limit value as $k \to +\infty$, and thus finishes the proof. □

Theorem 4. For the system (8), if let $\varphi(\boldsymbol{A}) = N^{1-1/p} \left\|\boldsymbol{A}\right\|_1$ or $\varphi(\boldsymbol{A}) = \left\|\boldsymbol{A}\right\|_1^{1/p} \left\|\boldsymbol{A}\right\|_\infty^{1-1/p}$, where $\boldsymbol{A}$ is short for $\boldsymbol{A}[k]$, $\int_\Omega |\varphi(\boldsymbol{A})|^p d\mu$ would exist which means $E_{L^p}\varphi(\boldsymbol{A})$ exists.

Proof Both $\left\|\boldsymbol{A}\right\|_1$ and $\left\|\boldsymbol{A}\right\|_\infty$ are continues measurable functions. Then $\int_\Omega N^{p-1} \left\|\boldsymbol{A}\right\|_1^p d\mu$ and $\int_\Omega \left\|\boldsymbol{A}\right\|_1 \left\|\boldsymbol{A}\right\|_\infty^{p-1} d\mu$ exist. This leads to the results of Theorem 4. □

Remark 3: Theorem 4 yields two sufficient conditions for the pth-moment stability of the system (8), i.e., $E_{L^p}(N^{1-1/p} \left\|\boldsymbol{A}\right\|_1) < 1$ and $E_{L^p}(\left\|\boldsymbol{A}\right\|_1^{1/p} \left\|\boldsymbol{A}\right\|_\infty^{1-1/p}) < 1$. Although there might exist certain conservation, these two conditions are convenient for practical operations, because both $\left\|\boldsymbol{A}\right\|_1$ and $\left\|\boldsymbol{A}\right\|_\infty$ have the explicit representations.

Remark 4: Taniguchi [18] provided stochastic stability theorems of the nonlinear difference equations through using norm inequalities; however, the theorems obtained in [18] can not assist us to achieve the results with practical significance for the system (8). While by employing the norm-inequality-based framework, this study dedicates to derive the results for the system (8). One could also find the moment stability studies attract many interests recently, e.g., the pth-moment exponential ultimate boundedness is investigated for impulsive stochastic differential systems [19], and the pth-moment asymptotic stability is analyzed for stochastic delayed hybrid systems with Levy noise [20].

4.2 The pth-moment stability of power control

Going back to the power-control system (6), we can obtain many useful results without too much efforts based on the previous analysis.

Remark 5: By letting $\boldsymbol{A}[k] = \boldsymbol{D}[k]\boldsymbol{C}[k]$ and $\boldsymbol{B}[k] = \boldsymbol{D}[k]$, one can directly apply Theorems 1 to 4 to the power-control system (6).

One important novelty of this study is not only to assess the stability of power-control system (6) but also to acquire more knowledge of relations between the stochastic stability and power control together with other wireless communication technologies. We will show that the proposed norm-inequality-based approach allows us to recognize the effect of multiple-access methods to the pth-moment stability of power control.

The sufficient conditions for the pth-moment stability given by Theorems 1, 2, 3, and 4 are only related to $\boldsymbol{A}[k]$ $(= \boldsymbol{D}[k]\boldsymbol{C}[k])$, while $\boldsymbol{D}[k]\boldsymbol{C}[k]$ is partly determined by the target SINRs and link gains according to (6)-(7). This fact inspires us to investigate the pth-moment stability by thinking over the power control together with the effect of multiple access technique.

Consider the power-control system (6) with a multiple access method, under Assumption 4, we have an upper bound of $\boldsymbol{A}[k]$ as

$$\boldsymbol{A}[k] = \boldsymbol{D}[k]\boldsymbol{C}[k] \leq \begin{bmatrix} 0 & \gamma_1\beta_{12}\frac{G_{22}[k]}{G_{11}[k]} & \cdots & \gamma_1\beta_{1N}\frac{G_{NN}[k]}{G_{11}[k]} \\ \gamma_2\beta_{21}\frac{G_{11}[k]}{G_{22}[k]} & 0 & \cdots & \gamma_2\beta_{2N}\frac{G_{NN}[k]}{G_{22}[k]} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_N\beta_{N1}\frac{G_{11}[k]}{G_{NN}[k]} & \gamma_N\beta_{N2}\frac{G_{22}[k]}{G_{NN}[k]} & \cdots & 0 \end{bmatrix}.$$

In the reminder of this section, the index k will be dropped from $\boldsymbol{A}[k]$, $\boldsymbol{D}[k]$, $\boldsymbol{C}[k]$, and $G_{ii}[k]$ such that $\boldsymbol{A}$, $\boldsymbol{D}$, $\boldsymbol{C}$, and G_{ii} are used.

From above, we get the following results.

Theorem 5. If the power-control system (6) employs a multiple access method so that Assumptions 1 to 4 are satisfied, it will hold that $\bar{\mathbf{A}} = E\boldsymbol{A} \leq \mu_G \cdot \mu_{1/G} \cdot \Theta_{\bar{\gamma},\beta}$, where

$$\Theta_{\gamma,\beta} = \begin{bmatrix} 0 & \gamma_1\beta_{12} & \cdots & \gamma_1\beta_{1N} \\ \gamma_2\beta_{21} & 0 & \cdots & \gamma_2\beta_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_N\beta_{N1} & \gamma_N\beta_{N2} & \cdots & 0 \end{bmatrix}.$$

Then if the values of γ_i $(1 \leq i \leq N)$ and β_{ij} $(1 \leq i, j \leq N)$ are properly chosen such that

$$\min\left\{\max_i \sum_{j\neq i}^{N} \gamma_i\beta_{ij}, \max_j \sum_{i\neq j}^{N} \gamma_i\beta_{ij}\right\} < \frac{1}{\mu_G \cdot \mu_{1/G}}, \tag{13}$$

the system will be first-moment stable.

Proof The upper bound of $\bar{\mathbf{A}}$ can be obtained by taking expectation to the upper bound of $\boldsymbol{A}$ given above.

Let $\rho(\cdot)$ denote the spectral radius. Using [23, Theorem 8.1.22] to show that

$$\max\left\{\min_i \sum_{j=1}^{N} \bar{A}_{ij}, \min_j \sum_{i=1}^{N} \bar{A}_{ij}\right\} \leq \rho\left(\bar{\mathbf{A}}\right) \leq \min\left\{\max_i \sum_{j=1}^{N} \bar{A}_{ij}, \max_j \sum_{i=1}^{N} \bar{A}_{ij}\right\}.$$

i.e., the smallest row sum of a nonnegative matrix is a lower bound on its spectral radius, and the largest row sum is an upper bound. Then, by applying $\bar{A}_{ij} \leq \mu_G \cdot \mu_{1/G} \cdot \gamma_i\beta_{ij}$, we have

$$\rho\left(\bar{\mathbf{A}}\right) \leq \mu_G \cdot \mu_{1/G} \cdot \min\left\{\max_i \sum_{j\neq i}^{N} \gamma_i\beta_{ij}, \max_j \sum_{i\neq j}^{N} \gamma_i\beta_{ij}\right\}. \tag{14}$$

Combing this result with [23, Theorem 5.6.12] which says $\lim_{k\to+\infty} \bar{\mathbf{A}}^k = \mathbf{0}$ if and only if $\rho(\bar{\mathbf{A}}) < 1$, we see that letting the right-hand side of (14) be less than 1 is a sufficient condition for $\lim_{k\to+\infty} \bar{\mathbf{A}}^k = \mathbf{0}$, which therefore makes the power-control system to be first-moment stable (see Theorem 1 for reference). □

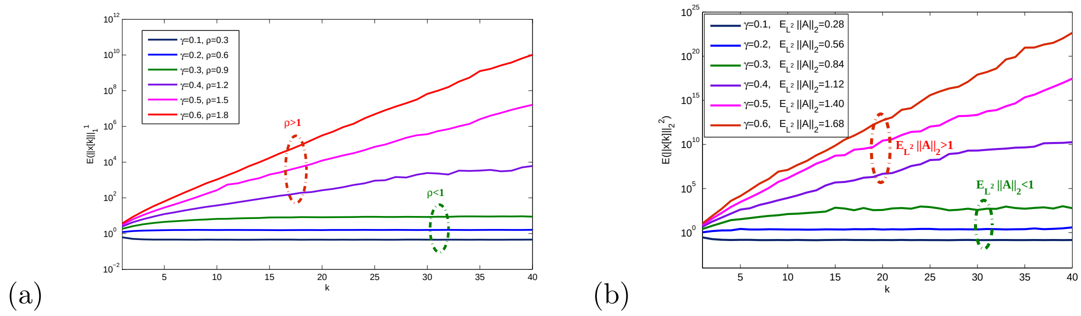


Figure 1: (a): $E(\|\boldsymbol{x}[k]\|_1^{\,1})$ versus k, where ρ is short for $\rho\left(\bar{\mathbf{A}}\right)$. The curves with $\rho < 1$ tend to finite values, while those with $\rho > 1$ increase ceaselessly; (b): $E(\|\boldsymbol{x}[k]\|_2^{\,2})$ versus k. The curves with $E_{L^2}\,\|\boldsymbol{A}\|_2 < 1$ tend to finite values, while those with $E_{L^2}\,\|\boldsymbol{A}\|_2 > 1$ are progressively growing

Theorem 6. Consider the power-control system (6) with a multiple access method such that Assumptions 1 to 4 are satisfied, if the values of γ_i $(1 \le i \le N)$ and β_{ij} $(1 \le i, j \le N)$ are properly chosen such that

$$\sum_{j=1}^{N} \sum_{i \ne j}^{N} \gamma_j^2 \beta_{ij}^2 < \frac{1}{\mu_{G^2} \cdot \mu_{1/G^2}}, \tag{15}$$

the system will be second-moment stable.

Proof Since $\|\boldsymbol{A}\|_2 = \sqrt{\mathrm{tr}(\boldsymbol{A}^T\boldsymbol{A})}$ [23], where $\mathrm{tr}(\cdot)$ is the trace operation, by setting $\varphi(\boldsymbol{A}) = \sqrt{\mu_{1/G^2} \cdot \mathrm{tr}\left(\Theta_{\gamma,\beta}^T \Theta_{\gamma,\beta}\right)}$, it follows that $E_{L^2}\varphi(\boldsymbol{A}) = \mu_{G^2} \cdot \mu_{1/G^2} \cdot \mathrm{tr}\left(\Theta_{\gamma,\beta}^T \Theta_{\gamma,\beta}\right) = \mu_{G^2} \cdot \mu_{1/G^2} \cdot \left(\sum_{j=1}^{N} \sum_{i \ne j}^{N} \gamma_j^2 \beta_{ij}^2\right)$. Observe that $E_{L^2}\varphi(\boldsymbol{A}) < 1$ as long as (15) holds. Then, recalling Theorem 2 completes the proof. □

Remark 6: The Cauchy-Schwarz inequality [24] leads to $\frac{1}{\mu_G \cdot \mu_{1/G}} < 1$ and $\frac{1}{\mu_{G^2} \cdot \mu_{1/G^2}} < 1$. So we can have more conservative but simpler conditions than (13) and (15) to achieve the first- and second-moment stability, respectively, which are $\min\left\{\max_i \sum_{j \ne i}^{N} \gamma_i \beta_{ij}, \max_j \sum_{i \ne j}^{N} \gamma_i \beta_{ij}\right\} < 1$, and $\sum_{j=1}^{N} \sum_{i \ne j}^{N} \gamma_j^2 \beta_{ij}^2 < 1$.

Furthermore, let us extend Theorems 5 and 6 to a generalized case, i.e., the pth-moment stability with any $p \in \mathbb{Z}_+$.

Theorem 7. Suppose that the power-control system (6) employs a multiple access method so that Assumptions 1 to 4 are established, the system will be pth-moment stable if $E_{L^p}\varphi(\boldsymbol{A}) < 1$ with $\varphi(\boldsymbol{A}) = N^{1-1/p}\left(\max_j \sum_{i \ne j}^{N} \gamma_i \beta_{ij} \frac{G_{jj}}{G_{ii}}\right)$, or

$$\varphi(\boldsymbol{A}) = \left(\max_j \sum_{i \ne j}^{N} \gamma_i \beta_{ij} \frac{G_{jj}}{G_{ii}}\right)^{1/p} \left(\max_i \sum_{j \ne i}^{N} \gamma_i \beta_{ij} \frac{G_{jj}}{G_{ii}}\right)^{1-1/p}.$$

Proof Through replacing $\|\boldsymbol{A}\|_1$ and $\|\boldsymbol{A}\|_\infty$ in Theorem 4 with the maximum column sum and maximum row sum of $\boldsymbol{A}[k]$, respectively, Theorem 7 can be validated. □

The importance of Theorems 5, 6, 7, and Remark 6 lies in that they can guide system designers to assess and select suitable target SINR schemes and multiple access methods for wireless-network systems and also to pick out the proper system parameters for them, from the perspective of power-control stability.

Table 1: Numerical Values of Examples 2 and 3.

γ	73.7	86.0	98.3	110.6
$\min\left\{\max_i \sum_{j\neq i}^N \gamma\beta_{ij}, \max_j \sum_{i\neq j}^N \gamma\beta_{ij}\right\}$	0.90	1.05	1.20	1.35
$\rho\left(\bar{\mathbf{A}}\right)$	0.66	0.77	0.87	1.01
$\sum_{j=1}^N \sum_{i\neq j}^N \gamma^2\beta_{ij}^2$	0.85	1.16	1.51	1.91
$E_{L^2}\left\Vert\boldsymbol{A}\right\Vert_2$	0.53	0.72	0.94	1.19
$\left(\int_\Omega \left\Vert\boldsymbol{A}\right\Vert_5^5 d\mu\right)^{1/5}$	2.05	2.40	2.70	3.13
$E_{L^5}\left(\left\Vert\boldsymbol{A}\right\Vert_1^{1/5}\left\Vert\boldsymbol{A}\right\Vert_\infty^{4/5}\right)$	2.26	2.64	2.98	3.45

5 Numerical examples

Example 1: We consider the power-control system (8) in main text with i.i.d. Rayleigh fading link gains (that is, all G_{ij} are Rayleigh distributed with unit variance) and fixed target SINRs $\gamma_1[k] = \gamma_2[k] = \cdots = \gamma_N[k] = \gamma$. There are four nodes in the network.

We let $\boldsymbol{n}[k]$ be the power vector of Gaussian noise with unit variance, and initially set $\boldsymbol{x}[1] = [1\ 0\ 0\ 0]^T$. In case of $\gamma = 0.1, 0.2, \cdots, 0.6$, Figs. 1(a) and 1(b) illustrate how $E(\left\Vert\boldsymbol{x}[k]\right\Vert_1^1)$ and $E(\left\Vert\boldsymbol{x}[k]\right\Vert_2^2)$ grow with k, where $\rho\left(\bar{\mathbf{A}}\right)$ and $E_{L^2}\left\Vert\boldsymbol{A}\right\Vert_2$ are estimated during the simulations. Fig. 1(a) shows that the curves of $E(\left\Vert\boldsymbol{x}[k]\right\Vert_1^1)$ with $\gamma = 0.3, 0.6, 0.9$ tend to finite values (in other words, the system is first-moment stable), while others increase ceaselessly. This result is in accordance with Theorem 1 because $\lim_{k\to+\infty} \bar{\mathbf{A}}^k = \mathbf{0}$ if and only if $\rho(\bar{\mathbf{A}}) < 1$ [23, Theorem 5.6.12]. From Fig. 1(b), it is observed that the curves of $E(\left\Vert\boldsymbol{x}[k]\right\Vert_2^2)$ with $\gamma = 0.28, 0.56, 0.84$ tend to finite values (or, equivalently, the system is second-moment stable), while others are progressively growing. This observation is a consequence of Theorems 2, 3.

Example 2: We consider the power-control system (8) in main text with multiple access method and fixed target SINRs $\gamma_1[k] = \gamma_2[k] = \cdots = \gamma_N[k] = \gamma$. All $G_{jj} = 1 + G\prime_j$ where $G\prime_j$ is the Rayleigh distributed with unit variance. Let $G_{ij}[k] = \beta_{ij}G_{jj}[k]$ without loss of generality. It can be obtained that $\mu_G \cdot \mu_{1/G} = 1.20$ and $\mu_{G^2} \cdot \mu_{1/G^2} = 2.03$. There are four nodes in the network which employs a multiple access method such that

$$\begin{bmatrix} 0 & \beta_{12} & \beta_{13} & \beta_{14} \\ \beta_{21} & 0 & \beta_{23} & \beta_{24} \\ \beta_{31} & \beta_{32} & 0 & \beta_{34} \\ \beta_{41} & \beta_{42} & \beta_{43} & 0 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{100} & \frac{1}{200} & \frac{1}{300} \\ \frac{1}{400} & 0 & \frac{1}{500} & \frac{1}{600} \\ \frac{1}{700} & \frac{1}{800} & 0 & \frac{1}{900} \\ \frac{1}{1000} & \frac{1}{1100} & \frac{1}{1200} & 0 \end{bmatrix}.$$

For above,

$$\min\left\{\max_i \sum_{j\neq i}^N \beta_{ij}, \max_j \sum_{i\neq j}^N \beta_{ij}\right\} = 1.22\times 10^{-2}, \qquad \sum_{j=1}^N \sum_{i\neq j}^N \beta_{ij}^2 = 1.565\times 10^{-4}. \tag{16}$$

Again, let $\boldsymbol{n}[k]$ be the power vector of Gaussian noise with unit variance, and initially set $\boldsymbol{x}[1] = [1\ 0\ 0\ 0]^T$. In case of $\gamma = 73.7, 86.0, 98.3, 110.6$, we build Table 1 by recalling two previous equalities, i.e., (16), and performing simulations. From Table 1, it is clear that $\min\left\{\max_i \sum_{j\neq i}^N \gamma\beta_{ij}, \max_j \sum_{i\neq j}^N \gamma\beta_{ij}\right\}$ is less than $\rho\left(\bar{\mathbf{A}}\right)$, and $\sum_{j=1}^N \sum_{i\neq j}^N \gamma^2\beta_{ij}^2$ is less than $E_{L^2}\left\Vert\boldsymbol{A}\right\Vert_2$, as we could expect from the relations between Theorems 1, 2, 5, and 6. The numerical results of Figs. 2(a) and 2(b) are in agreement with Theorems 1, 2, 5, and 6.

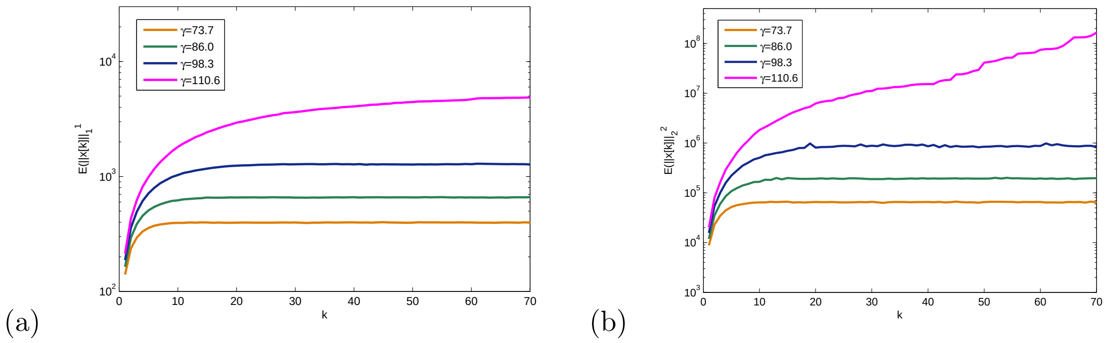


Figure 2: (a): $E(\|\boldsymbol{x}[k]\|_1^1)$ versus k; (b): $E(\|\boldsymbol{x}[k]\|_2^2)$ versus k.

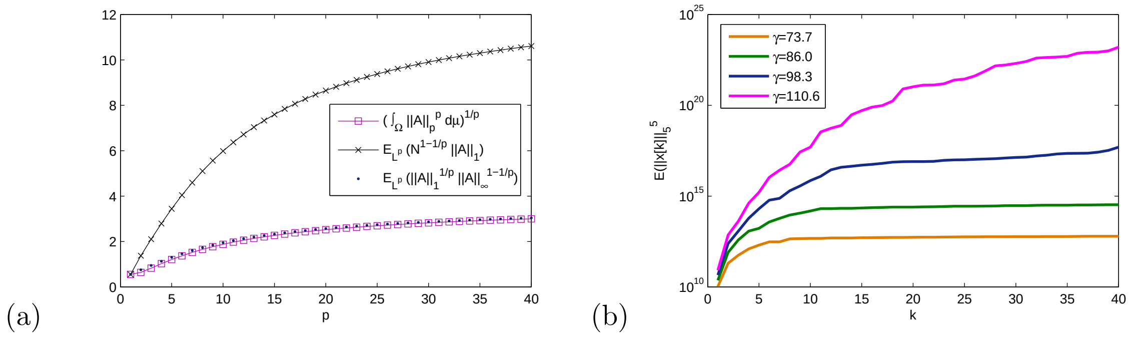


Figure 3: (a): Comparison among $\left(\int_\Omega \|\boldsymbol{A}\|_p^p \, d\mu\right)^{1/p}$, $E_{L^p}(N^{1-1/p}\|\boldsymbol{A}\|_1)$, and $E_{L^p}\left(\|\boldsymbol{A}\|_1^{1/p}\|\boldsymbol{A}\|_\infty^{1-1/p}\right)$ for different p. Note that, the Matlab routines provided by Higham [17] which can directly estimate $\|\boldsymbol{A}\|_p$ is used for computing $\left(\int_\Omega \|\boldsymbol{A}\|_p^p \, d\mu\right)^{1/p}$; (b): $E(\|\boldsymbol{x}[k]\|_5^5)$ versus k

Example 3: The same system model and parameters as Example 2 are used.

In Fig. 3(a), we have compared $\left(\int_\Omega \|\boldsymbol{A}\|_p^p \, d\mu\right)^{1/p}$ with its two upper bounds as given by Remark 3, i.e., $E_{L^p}(N^{1-1/p}\|\boldsymbol{A}\|_1)$ and $E_{L^p}\left(\|\boldsymbol{A}\|_1^{1/p}\|\boldsymbol{A}\|_\infty^{1-1/p}\right)$, which can also be refereed to Theorem 4. It is seen that, $E_{L^p}\left(\|\boldsymbol{A}\|_1^{1/p}\|\boldsymbol{A}\|_\infty^{1-1/p}\right)$ stays quite close to $\left(\int_\Omega \|\boldsymbol{A}\|_p^p \, d\mu\right)^{1/p}$, however, there is an evident gap between $E_{L^p}(N^{1-1/p}\|\boldsymbol{A}\|_1)$ and $\left(\int_\Omega \|\boldsymbol{A}\|_p^p \, d\mu\right)^{1/p}$, which is amplified as p increases. Therefore, being the upper bound of $\left(\int_\Omega \|\boldsymbol{A}\|_p^p \, d\mu\right)^{1/p}$, $E_{L^p}\left(\|\boldsymbol{A}\|_1^{1/p}\|\boldsymbol{A}\|_\infty^{1-1/p}\right)$ is more tight than $E_{L^p}(N^{1-1/p}\|\boldsymbol{A}\|_1)$. As a consequence, we propose to use $E_{L^p}\left(\|\boldsymbol{A}\|_1^{1/p}\|\boldsymbol{A}\|_\infty^{1-1/p}\right)$ when the upper bound of $\left(\int_\Omega \|\boldsymbol{A}\|_p^p \, d\mu\right)^{1/p}$ is needed. For $\gamma = 73.7, 86.0, 98.3, 110.6$, Fig. 3(b) illustrates how $E(\|\boldsymbol{x}[k]\|_5^5)$ evolves with k, and Table 1 presents the data of $\left(\int_\Omega \|\boldsymbol{A}\|_5^5 \, d\mu\right)^{1/5}$ and $E_{L^5}\left(\|\boldsymbol{A}\|_1^{1/5}\|\boldsymbol{A}\|_\infty^{4/5}\right)$. When $\gamma = 73.7, 86.0$, $E(\|\boldsymbol{x}[k]\|_5^5)$ tends to finite values as long as k is sufficiently large. Then if $\gamma = 98.3, 110.6$, $E(\|\boldsymbol{x}[k]\|_5^5)$ will be found to grow infinitely. The numerical result is in accordance with Theorem 3.

Conclusion

This study develops a norm-inequality-based framework of analyzing the pth-moment stability of linear systems with random parameters, so as to show that a typical power control law with linear system model is stable in the sense of the pth-moment stability. It is the first time to recognize the effect of multiple-access methods to stability analysis of power control.

Acknowledgment

The authors are grateful to the anonymous reviewers for many valuable comments concerning the presentation. This work was supported by the National Natural Science Foundation of China under Grant 61501043. We would like to thank Dr. Miao Diao and Prof. Duan Zhisheng for all their help and support while writing this paper.

Bibliography

[1] C.W. Sung, K.K. Leung (2005); A Generalized framework for distributed power control in wireless networks, *IEEE Trans. Inf. Theory*, 51(7): 2625-2635.

[2] H. Feyzmahdavian, M. Johansson,T. Charalambous (2012); Contractive interference functions and rates of convergence of distributed power control laws, *IEEE Trans. Wireless Commun.*, 11(12):4494-4502.

[3] H. Feyzmahdavian, T. Charalambous, and M. Johansson (2014); Stability and performance of continuous-time power control in wireless networks, *IEEE Trans. Automat. Contr.*, 59(8):2012-2023.

[4] I. Lestas (2012); Power control in wireless networks: Stability and delay independence for a general class of distributed algorithms, *IEEE Trans. Automat. Contr.*, 57(5):1253-1258.

[5] E. Devane and I. Lestas (2014); Stability of a general class of distributed algorithms for power control in time-varying wireless networks, *IEEE Trans. Automat. Contr.*, 59(8):1999-2011.

[6] S. Ulukus, R. Yates (1998); Stochastic power control for cellular radio systems, *IEEE Trans. Commun.*, 46(6):784-798.

[7] M. Varanasi, D. Das (2002); Fast stochastic power control algorithms for nonlinear multiuser receivers, *IEEE Trans. Commun.*, 50(11):1817-1827.

[8] J. Lu, S. Ulukus,A. Ephremides (2005); Standard and quasi-standard stochastic power control algorithms, *IEEE Trans. Inf. Theory*, 51(7):2612-2624.

[9] H. Zhang, W. S. Wong, W. Ge, P. E. Caines (2007); A stochastic approximation approach to the power-control problem, *IEEE Trans. Commun.*, 55(8): 878-886.

[10] X. Feng, K A. Loparo, Y. Ji, H. J. Chizeck (1992); Stochastic stability properties of jump linear systems, *IEEE Trans. Automat. Contr.*, 37(1):38-53.

[11] J. Leth, H. Schioler, M. Gholami, V. Cocquempot (2013); Stochastic stability of Markovianly switched systems, *IEEE Trans. Automat. Contr.*, 58(8): 2048-2054.

[12] G. J. Foschini and Z. Miljanic (1993); A simple distributed autonomous power control algorithm and its convergence, *IEEE Trans. Vehic. Technol.*, 42: 641-646.

[13] M. Xiao, N. Shroff, and E. Chong (2003); A utility-based power-control scheme in wireless cellular systems, *IEEE J. Sel. Areas Commun.*, 11(2):210-221.

[14] H. Furstenberg, H. Kesten (1960); Products of random matrices, *Annals of mathematical statistics*, 31(2):457-469.

[15] A. Bobrowski (2005); *Functional analysis for probability and stochastic processes*, Cambridge Universtity Press, 2005.

[16] W. L. D. Koning (1984); Optimal estimation of linear discrete-time systems with stochastic parameters, *Automatica*, 20(1):113-115.

[17] N. J. Higham (1992); Estimating the matrix p-norm, *Numer. Math*, 62:511-538.

[18] T. Taniguchi (1990); Stability theorems of stochastic difference equations, *Journal of Mathematical Analysis and Applications*, 147(1):81-96.

[19] L. Xua, S. S. Geb (2015); The pth moment exponential ultimate boundedness of impulsive stochastic differential systems, *Applied Mathematics Letters*, 42:22-29.

[20] J. Yang, W. Zhou, X. Yang, X. Hu, L. Xie (2015); pth moment asymptotic stability of stochastic delayed hybrid systems with Levy noise, *International Journal of Control*, 88(9):1726-1734.

[21] T. Charalambous, I. Lestas, G. Vinnicombe (2008); On the stability of the Foschini-Miljanic algorithm with time-delays, in *Proc. 47th IEEE Conf. on Decision Control*, 2991-2996.

[22] M. Varanasi (1999); Nonlinear multiuser receivers with distributed power control in cellular radio networks, in *Proc. 37th Annu. Allerton Conf. Communication, Control and Computers*, 820-830.

[23] R. A. Horn and C. R. Johnson (1985); *Matrix analysis*, Cambridge University Press, Second Edition, 1985.

[24] R. Durrett (2005); *Probability: Theory and examples*, Cengage Learning, Third Edition, 2005.

[25] I. S. Gradshteyn, I. M. Ryzhik (2007); *Table of integrals, series, and products, 7th edition*, Academic, 2007.

Appendix: Several lemmas

This Appendix is devoted to present the lemmas (and their proofs) which are required to derive the main results of our work.

Lemma A.1: Let the nonnegative numbers $a_1, a_2, \cdots, a_N$ and the positive $p_1, p_2, \cdots, p_N$ be given. Set $\sum_{j=1}^{N} \frac{1}{p_j} = 1$ then the inequality $\prod_{j=1}^{N} a_j \leq \sum_{j=1}^{N} \frac{1}{p_j} a_j^{p_j}$ holds with equality if and only if all a_k with $p_k > 0$ are equal.

Proof: $\prod_{j=1}^{N} a_j \leq \sum_{j=1}^{N} \frac{1}{p_j} a_j^{p_j}$ in Lemma A.1 is an inequality of the weighted arithmetic mean and geometric mean, which can be proved by using the finite form of Jensen's inequality [24] for the natural logarithm. □

Lemma A.2: Let $x_1, x_2, \cdots, x_N$ be N random variables and $p_1, p_2, \cdots, p_N$ be nonnegative numbers numbers. If $\sum_{j=1}^{N} \frac{1}{p_j} = 1$ and $E|x_j|^{p_j} < +\infty$ for $1 \leq j \leq N$, then $E\left(\prod_{j=1}^{N} |x_j|\right) \leq \prod_{j=1}^{N} (E|x_j|^{p_j})^{\frac{1}{p_j}}$, where $(E|x_j|^{p_j})^{\frac{1}{p_j}} = E_{L^p}||x_j||$ with $p = p_j$.

Proof: By using Lemma A.1, we get $\frac{\prod_{j=1}^{N} |x_j|}{\prod_{j=1}^{N} \left(E|x_j|^{p_j}\right)^{\frac{1}{p_j}}} = \prod_{j=1}^{N} \frac{|x_j|}{\left(E|x_j|^{p_j}\right)^{\frac{1}{p_j}}} \leq \sum_{j=1}^{N} \frac{|x_j|^{p_j}}{p_j E|x_j|^{p_j}}$. Then applying the expectation to above inequality, $\frac{E\left(\prod_{j=1}^{N} |x_j|\right)}{\prod_{j=1}^{N} \left(E|x_j|^{r_j}\right)^{\frac{1}{p_j}}} \leq \sum_{j=1}^{N} \frac{E|x_j|^{p_j}}{p_j E|x_j|^{p_j}} = \sum_{j=1}^{N} \frac{1}{p_j} = 1$. Thus, Lemma A.2 is verified. □

Lemma A.3: Suppose $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_K$ are random matrices with size $S_1 \times S_2$, $S_2 \times S_3$, $\cdots$, $S_K \times S_{K+1}$, where $S_1, S_2, \cdots, S_K, S_{K+1}$ are all simply positive integers and the subscripts are labels corresponding to the matrices. If the entries of $\boldsymbol{X}_k$ are independent with those of $\boldsymbol{X}_l$ for any $k \neq l$ then $E\left(\prod_{k=1}^{K} \boldsymbol{X}_k\right) = \prod_{k=1}^{K} E(\boldsymbol{X}_k)$.

Proof: The product of K matrices can be expressed in the index notation as

$$\left[\prod_{k=1}^{K} \boldsymbol{X}_k\right]_{ij} = \sum_{i_1=1}^{S_1} \sum_{i_2=1}^{S_2} \cdots \sum_{i_{K-1}=1}^{S_{K-1}} [\boldsymbol{X}_1]_{ii_1} [\boldsymbol{X}_2]_{i_1 i_2} [\boldsymbol{X}_3]_{i_2 i_3} \cdots [\boldsymbol{X}_{n-1}]_{i_{n-2} i_{n-1}} [\boldsymbol{X}_n]_{i_{n-1} j}.$$

This implies every entry of the resultant matrix after matrix product is a linear function of the entries of all $\boldsymbol{A}_k$ matrices. The independence condition can further yield [24]

$$E\left(\left[\prod_{k=1}^{K} \boldsymbol{X}_k\right]_{ij}\right) = \sum_{i_1=1}^{S_1} \sum_{i_2=1}^{S_2} \cdots \sum_{i_{K-1}=1}^{S_{K-1}} E([\boldsymbol{X}_1]_{ii_1}) E([\boldsymbol{X}_2]_{i_1 i_2}) \cdots E([\boldsymbol{X}_{n-1}]_{i_{n-2} i_{n-1}}) E([\boldsymbol{X}_n]_{i_{n-1} j}).$$

Therefore, Lemma A.3 is proved. □

Lemma A.4: Suppose that $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_I$ are random matrices with size $S \times 1$, one can have $E_{L^p} \left\|\sum_{i=1}^{I} \boldsymbol{x}_i\right\|_p \leq \sum_{i=1}^{I} E_{L^p} \left\|\boldsymbol{x}_i\right\|_p$, where $p \in \mathbb{Z}_+$.

Proof: The norm inequalities and Lemma A.2 combine to provide

$$\begin{aligned}
\left(E_{L^p} \left\|\sum_{i=1}^{I} \boldsymbol{x}_i\right\|_p\right)^p &= E\left(\left\|\sum_{i=1}^{I} \boldsymbol{x}_i\right\|_p^p\right) \leq E\left(\left(\sum_{i=1}^{I} \left\|\boldsymbol{x}_i\right\|_p\right)^p\right) \\
&= \sum_{i_1=1}^{I} \sum_{i_2=1}^{I} \cdots \sum_{i_p=1}^{I} E\left(\left\|\boldsymbol{x}_{i_1}\right\|_p \left\|\boldsymbol{x}_{i_2}\right\|_r \cdots \left\|\boldsymbol{x}_{i_p}\right\|_p\right) \\
&\leq \sum_{i_1=1}^{I} \cdots \sum_{i_p=1}^{I} \left(E\left(\left\|\boldsymbol{x}_{i_1}\right\|_p^p\right) \cdots E\left(\left\|\boldsymbol{x}_{i_p}\right\|_p^p\right)\right)^{1/p} = \left(\sum_{i=1}^{I} E_{L^p} \left\|\boldsymbol{x}_i\right\|_p\right)^p.
\end{aligned}$$

This gives the desired result. □

Lemma A.5: Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_K$ be random matrices with size $S_1 \times S_2$, $S_2 \times S_3$, $\cdots$, $S_K \times S_{K+1}$ and $\boldsymbol{y}$ be a $S_{K+1} \times 1$ random vector. If the entries of $\boldsymbol{X}_k$ are independent with those of $\boldsymbol{X}_l$ for any $k \neq l$ and $\boldsymbol{y}$ for any $1 \leq k \neq K$, then $E_{L^p} \left\|\left(\prod_{k=1}^{K} \boldsymbol{X}_k\right) \boldsymbol{y}\right\|_p \leq \left(\prod_{k=1}^{K} E_{L^p} \varphi(\boldsymbol{X}_k)\right) \left(E_{L^p} \left\|\boldsymbol{y}\right\|_p\right)$, where $p \in \mathbb{Z}_+$.

Proof: The norm inequalities implies $\left\|\left(\prod_{k=1}^{K} \boldsymbol{X}_k\right) \boldsymbol{y}\right\|_p \leq \left(\prod_{k=1}^{K} \left\|\boldsymbol{X}_k\right\|_p\right) \left\|\boldsymbol{y}\right\|_p$, and thus we get $E\left(\left\|\left(\prod_{k=1}^{K} \boldsymbol{X}_k\right) \boldsymbol{y}\right\|_p^p\right) \leq \left(\prod_{k=1}^{K} E(|\varphi(\boldsymbol{X}_k)|^{\,p})\right) E(\left\|\boldsymbol{y}\right\|_p^p)$. Such an inequality directly yields the result. □

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(6):860-876, December 2016.

A New Method for Colour Image Segmentation

G. Schleyer, G. Lefranc, C. Cubillos, G. Millán, R. Osorio-Comparán

Gustavo Schleyer, Gastón Lefranc*, Claudio Cubillos
Pontificia Universidad Católica de Valpaíso
Escuela de Ingeniería Eléctrica
Avda. Brasil #2147
Valparaíso, Chile
*Corresponding author: glefranc@pucv.cl

Ginno Millán
Universidad Católica del Norte
Escuela de Ingeniería
Larrondo #1281 Coquimbo, Chile
E-mail: gmillan@ucn.cl

Román Osorio-Comparán
Universidad Nacional Autónoma de México
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
Ciudad Universitaria
Coyoacán, México D.F., México
E-mail: roman@unam.mx

Abstract: This paper presents an unsupervised algorithm of colour image segmentation and is an extension of [1][a] . This method combines the advantages of the approaches based on split and merge and region growing, and the use of the RGB and HSV colour representation models. The effectiveness of the proposed method has been verified by the implementation of the algorithm using three different testing images with homogeneous regions, spatially compact and continuous. It was observed that the proposed algorithm outperforms the other analysed techniques requiring shorter processing time when compared with the other methods.

Keywords: Computer vision, Image processing, Image segmentation.

[a] Reprinted (partial) and extended, with permission based on License Number 3930961102612 [2016] ©IEEE, from "Computers Communications and Control (ICCCC), 2016 6th International Conference on".

1 Introduction

Image segmentation involves the identification of regions of interest, which generally are an object or a part in a digital image. Each region must maximise the homogeneity of its pixels features (colour, texture) and simultaneously maximise the differences with neighbouring regions; moreover, each region must be spatially compact. In general, image segmentation considers an important initial step in most image processing applications, which use the segmentation information in order to perform upper level tasks, such as object tracking or identification, and scenes interpretation.

This paper is an extension of paper presented in ICCCC2016 [1]. Two approaches widely used in colour image segmentation are split and merge [2]-[4] and region growing [5]-[9]. In general, the split and merge method begins with an initial and no homogeneous image partition, then keeps on splitting it until homogeneous partitions are obtained. A common data structure used in the implementation of this procedure is the quad tree representation [10]. After a division step, usually, many small and fragmented regions appear connected in some way during the merging phase. On the other hand, the method based on region growing consists in obtaining a homogeneous image region through a growing process that, beginning from a preselected seed, progressively crowd pixels around it fulfilling a determined homogeneity criterion. The growing process stops when it is not possible to add new pixels to the region. A common post-processing step consists in a union phase that eliminates small regions or neighbouring regions with similar attributes, making larger regions.

In this paper proposes an unsupervised colour image segmentation algorithm that combines the advantages of the split and merge and region growing approach using model features in the RGB and HSV colour representation. The effectiveness of the proposed method is compared with a parameter less quadrilateral-based image segmentation method [10] with a fuzzy segmentation HCI [11], and with a image segmentation using situational descriptors DCT [12]. The proposed algorithm performs the best segmentation and requires the least processing time compared with other methods used in its evaluation. It resulted to be the best approach for real applications that provides regions of homogeneous color, spatially compact and continuous, in a shorter processing time and more robustly against over-segmentation.

2 Colour images models

A digital image is modelled as three monochrome images, where each group is related to different colour. The RGB model is one of the most used methods where the resulting colour is formed by combining red, green and blue. The RGB model representation is a cube, using RGB coordinates instead of (x, y, z). The RGB complex algorithm establishes a different colour in a cube vertex. It is not easy to determine if these coordinates represent a particular tonality. In order to alleviate this problem a cylindrical coordinate model (HSV) is used.

In the HSV model (Hue Saturation Value), the colour represent by three fundamental quantities: tonality, saturation and the intensity value. The tonality identifies the colour, the saturation establishes the colour quantity represented in each pixel and finally its value. When a pixel has low saturation it is grey, and a pixel total saturated is a pure colour. The pixel intensity corresponds to the light quantity of the pixel. HSV model permits colour selection in a better way than RGB model. Colour space in HSV model is represented as a cone, where an angle in the circumference base is the tonality, the saturation is measured outwards from the origin and the vertical axis is the intensity value.

3 Colour image segmentation method

3.1 Technical methods of image segmentation

The Technical methods of image segmentation are classified in image segmentation techniques based colour image features, techniques based colour image, and image segmentation techniques based on physics [13]-[17].

Segmentation techniques based on colour image features can classified in clustering, adaptive clustering and thresholding histogram.

Image segmentation techniques based on colour image are: split and merge, techniques based on growing regions, graph theory, contours based, and neural networks.

- Split & merge technique considers spatial information and it has good results in images with homogeneous regions. However, the definition of uniformity of color can be difficult, and the tree can generate quadrangles not present objects in the image.

- Growing regions technique generates compact and spatially connected regions and some approaches produce very fast algorithms. Nevertheless, they are expensive in computational time and memory, and they have difficulties in establishing the seeds and the appropriate standard of homogeneity, because the algorithms are sequential.

Image segmentation techniques based on physics are ssupposedly robust against excessive lighting, and segmented surfaces depending on the composition of the material. The technique is restricted to one or two types of material. It has difficulties in identifying real images when are excessive lighting and shadows.

The segmentation method proposed in this paper can be classify into image-based approaches and as a mixture between two categories of segmentation techniques: split & merge and growth regions. The proposed approach divides the image in many regions (split), some of which subsequently merges based on a predetermined criterion (merge), and then algorithm can be classified into the first category. However, if the initial regions generated by the proposed algorithm considers the seeds of a region growing process, then the proposed method can be classify in the second category. The fact that the synergy produced between the two approaches can avoid disadvantages that typically find in the independent performance.

3.2 Proposed colour image segmentation method

As the original image I_0 is 24 bits/pixel RGB format and is very large, the segmentation reduces information. Then, the characteristics of the pixels of the image segment is analysed to decide whether to segment them based on their hue, saturation or intensity, using representation in the HSV model, or the segmentation must be done based on the general characteristics of colour using RGB model.

The proposed image segmentation method is in Fig. 1. This method consists on several algorithm to have an efficient and better segmentation of an image that it can use in real time applications.

First step is the RGB in HSV image conversion, reducing to 26 values. It is following with tonality segmentation including tonality label algorithm and extreme value of intensity mask application. After this segmentation, applies a merge of tonality of small and great group algorithm. Then, it continues a pixel segmentation with low saturation algorithm, with mask application low saturation and union of low saturation neighbors groups.

The next two algorithm are pixel segmentation of extreme values of intensity algorithm and remaining pixels segmentation algorithm. To finish, applies the final merge of small groups algorithm.

Because of an excessive amount of information produces in excessive processing times a reduction, gives the possibility of practical application of the algorithm, and it is much simpler to identify a colour from HSV model than RGB, the algorithm is to convert RGB image to a HSV image. The reduction is taking 26 different values for each band, which represents by 5 bits/pixel, and as there are, three bands need 3 x 5 = 15 bits per pixel.

Special pixels those image pixels having extreme values of intensity or low saturation values. The segmentation of pixels with extreme intensity values are white dot (high intensity value) or a black dot (low intensity value) in the digital image. Segmentation pixel Io with low saturation values is more complex, they appear as grey dots on the image and it is not possible segmentation by hue (tonality), so it is necessary to identify them in the first instance and subsequently dedicate a different processing to receive those pixels segmented by colour.

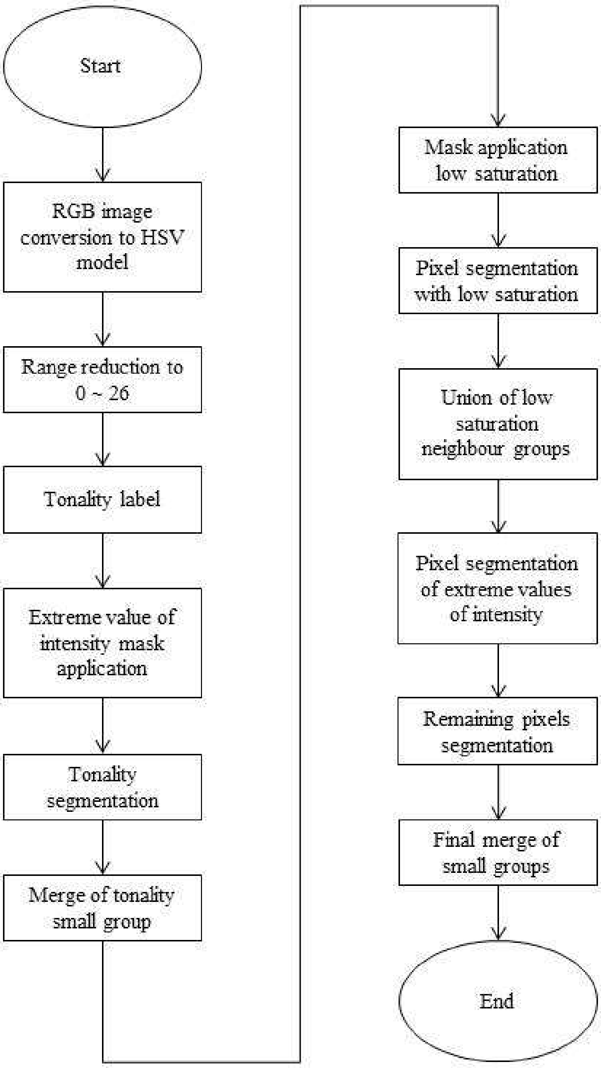

Start
RGB image conversion to HSV model
Range reduction to 0 ~ 26
Tonality label
Extreme value of intensity mask application
Tonality segmentation
Merge of tonality small group
Mask application low saturation
Pixel segmentation with low saturation
Union of low saturation neighbour groups
Pixel segmentation of extreme values of intensity
Remaining pixels segmentation
Final merge of small groups
End

Figure 1: The proposed image segmentation method

Reduction of possible combinations

First step is the RGB in HSV image conversion, reducing to 26 values, to have an acceptable detail level on image segmentation, reducing the information in the image, reducing processing time. It is following with tonality segmentation including tonality label algorithm and extreme value of intensity mask application. After this segmentation, applies a merge of tonality of small and great group algorithm. Then, it continues a pixel segmentation with low saturation algorithm, with mask application low saturation and union of low saturation neighbors groups.

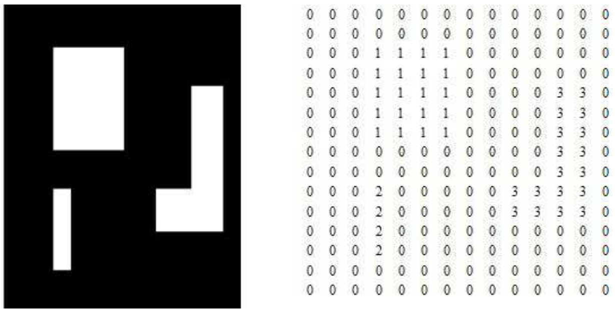
This step diminishes the quantity of information provided by the HSV model, so they can adopt integer values between 1 and 26, reducing the number of possible combinations to the inferior integer.

Segmentation by tonality

In the segmentation by tonality, special pixels (image pixels having extreme values of intensity or low saturation values) are separated from the image.

Once identified the pixels to segment according to their tonality, the neighbouring pixels that have the same tonality are merged and those ones with a different tonality are split. Then, 26 matrixes are created, one for each possible tonality value, which have been called H matrixes.

The segmentation of the no neighbouring pixels of the same tonality is carried out labelling each H matrix. The labelling consists in assigning a number to each component found in the image. So, the image is scanned from left to right and from top to bottom searching for cell sets with value "1", the first set found keeps its original value, but in the next set, the values “1" are replaced by “2", and so on, until the value n is reached, where n is the number of components in the scene. A result of the application of the previous procedure to an image is presented in Fig. 2.



0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	3	3	0
0	0	0	1	1	1	1	0	0	0	0	3	3	0
0	0	0	1	1	1	1	0	0	0	0	3	3	0
0	0	0	0	0	0	0	0	0	0	0	3	3	0
0	0	0	0	0	0	0	0	0	0	0	3	3	0
0	0	0	2	0	0	0	0	0	3	3	3	3	0
0	0	0	2	0	0	0	0	0	3	3	3	3	0
0	0	0	2	0	0	0	0	0	0	0	0	0	0
0	0	0	2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2: Labelling of components

The H matrixes store information of tonality, which is lost when they are merged in order to form H_{seg}.

From this point, the information of tonality of each region is stored in a column vector, called *color_reg*. The label value of each region, designated by the *num_etiqueta* variable, indicates the row of *color_reg* in, which must be, stored the tonality information.

Tonality small groups merging

Analyzing H_{seg} matrix, it shows an over segmentation image, because within the images are small groups of pixels that can correspond to shadows, grooves, reliefs, etc. These details do not represent relevant information, with less than 25 pixels. If H matrices are filtered to remove these small groups, tonality information could lost. Instead of filtering these arrays, it creates an algorithm capable of merging these small groups small with larger groups.

Then, all the groups that have less than a certain quantity of pixels, determined by an algorithm parameter called *par_area* must be merged with some neighbouring group. The *par_area* parameter, establishes a pixel minimum area of each segmented group.

To find the group with which each component has the biggest quantity of neighbouring pixels, it must be counted each kind of elements, inside the vector of the *vecinos* structure associated, with this component and to identify which one appears more times. Then, in order to identify small groups the *areas* vector is analysed searching values less than *par_area*, more formally a small group.

The group to merge with G_p, with its neighbouring group G_{fusion_p} with which exists the least tonality difference. In the case that there are more than one group with such difference, that is, if the set has more than one element, the group choose is a set, which has the bigger quantity of element.

Tonality big groups merging

A big group G_g is defined as all those groups which pixels quantity is bigger than *par_area*.

The big groups G_g are merged with those neighbouring groups with which it have a tonality difference less than or equal to one. The condition of the areas establishes that the prevailing label value is the one belonging to the group that have the biggest quantity of pixels, considering this, the label value of all the pixels belonging to G_g must be replaced by the label value of G_{fusion_g}.

Mask application low saturation. Re-labeling of groups

The merger of groups leads to the elimination of some regions and increased the size of others. The formation of groups is constantly changing when the segmentation algorithm iterates. This information must be recorded in the matrix which contains the location and H_{seg} tag value of all groups. However, update this matrix every time a group merges with another would require much processing time, thus new label values are only updated once known all new groups in H_{seg}.

As the H_{seg} matrix is scanned, each time that a pair of group, which must be merged, is found, the label values of both groups are stored in a table called T_{fusion}.

An algorithm that searches for all the groups of the table T_{fusion}, which are related with each other, is re-labelling of the c matrix. As a result, k groups that cannot be merged are obtained, which are stored in the $E(k)$ variable, finally, the H_{seg} values are refreshed.

With this step, it concludes the first iteration of the algorithm of segmentation by tonality. In the next iterations the definition of *color_reg* cannot be used, in the future it will be considered that *color_reg* for each group; will correspond to the mean of the I_0 values belonging to the respective group. The iterations of the algorithm of segmentation by tonality continue until no new groups can be merged, that is, when $T_{fusion} = \phi$. Once concluded this part of the segmentation process, it continue with that described in the next sections.

Low saturation pixels segmentation. Pixel segmentation of extreme values of intensity

The low saturation pixel segmentation algorithm is in charge for segmenting these pixels that have not been segmented by tonality segmentation algorithm. The first thing to consider is exist pixels with low saturation, but at the same time presenting intensity extreme values. This type of pixels will not be segmented by the segmentation algorithm pixels with low saturation. Therefore, first is to remove in the matrix those pixels with extreme values of intensity.

During the design segmentation algorithm pixels with low saturation various criteria such as segmentation were tested for strength value gray, HSV values and RGB values, obtaining better results with RGB. To identify the RGB value of each group present in the matrix, considers the average RGB values of each band. However, the tonality segmentation algorithm, has not identified pixels with low saturation were, they separated only those with extreme values of intensity, which has been done intentionally. As a result some groups of the matrix may be formed in part by low saturation pixel by pixel and partly medium or high saturation. It obtains two averages of the RGB bands, one for pixels with low saturation called P_{sat}, named and defined for each group and band RGB with other pixels with medium and high saturation called P_{no_sat}.

The Euclidean distance between the vectors that contains the mean values of the RGB bands coordinates, corresponding to the low saturation pixels of a group, and the vector that contains the same information in relation with the medium and high saturation pixels of the same group.

Each element of the vector $D_{sat\_no\_sat}$ is analysed in order to confirm if they are greater than the δ parameter, if the condition is fulfilled the group associated with this element must be divided, if the area of the two resultant groups is bigger than *par_area*. δ and *par_area* are algorithm parameters entered by the user. Each time a group that must be divided is identified, a register stored in the variable *separa*.

Once all the groups that must be divided are identified, the H_{seg} matrix is labelled again in order to register these changes. In this case, the labelling of the H_{seg} matrix is performed using the *separa* vector.

Low saturation neighboring groups merging

Neighbouring groups with low saturation merge. First, the areas of all the groups are calculated, refreshing the *areas* vector, then the areas of the low saturation groups are obtained refreshing the $areas_{sat}$ vector, through the multiplication of t per $M_{S\_Bajo}(fil, col)$. Second, the neighbouring pixels of the low saturation groups are identified, considering that a low saturation group is the one in which more than 50% of its pixels have low saturation values. The neighbouring pixels of each one of these groups are found.

Another important quantity that must be calculated is the mean value of each one of the coordinates of the RGB model, called P_R, P_G, and P_B, for each segmented group, not considering their saturation level.

In order to decide which low saturation groups will be merged, the Euclidean distance between the mean of each RGB band corresponding to each group, and all their neighbouring groups is calculated. If this distance is lesser than a certain value established by a new parameter called δ_2, then the groups are merged. In other words, each group will be merged with all its neighbouring groups with which the Euclidean distance between their mean RGB values is lesser than δ_2. The temporal information of the group that must be merged is stored in the *agrupa* variable. Once again, only when all the groups that will be merged are known, the labelling of the H_{seg} matrix proceeds.

If there are more than a value stored in $agrupa(k)$, the one that has the minimum Euclidean distance with the RGB mean values of the k group is chosen, and the remaining values are eliminated. During the new H_{seg} labelling the merge and labelling algorithm is utilized, but replacing the T_{fusion} variable, by *agrupa*. Once the building of the E matrix is finished, the labelling of the H_{seg} matrix. With this step the low saturation pixels segmentation concludes.

Segmentation of pixels with extreme brightness values

So far they have been segmented groups of pixels with high saturation, which allow good segmentation, then the pixel groups with low saturation have been segmented using averages of the RGB coordinates. It yet to analyze what happens with pixels with intensity extreme values, corresponding to high and low intensity values. The pixels with high intensity values appears as a white dot on the image regardless of the value of hue and saturation, so does the pixels with low intensity values, with the difference that they appear as black dots in the digital image. Pixels with intensity high and low brightness have been identified by M_{V_alto} and M_{V_bajo} matrixes, respectively. The first step of the algorithm is to label these matrices as described by M_{V_alto} and M_{V_bajo} matrixes. But due to the way they were built, in each pixels are black in color which indicates the presence points with extreme brightness intensities, so that are these pixels should re labels. As the labeling was defined to identify color pixel.

To finish the segmentation algorithm with pixels intensity extreme values, each label value of H_{seg} corresponds to a different region in the image. It adds *color_reg*, P_R, P_G, P_G and *areas* variables, and the values for the new groups.

Segmenting remaining pixels

When finish all segmentation procedures, and still exist pixels unsorted, means there are groups consisting of fewer pixels than the established by the *par_area* parameter. During arrays mask filters applied tto M_{V_alto} and M_{V_bajo} matrixes eliminate such groups. Pixels belonging to the edge of the image have not analysed by windows 3x3 and therefore not classified or incorrect. This corresponds to 0.43% of the total number of pixels of the image. To finish segmentation algorithm. To identify all remaining pixels, H_{seg} is traversed looking for pixels with value 0. The information regarding the position of these pixels within is stored in a new array called R. This demands significant processing time compared with the rest of the algorithm.

Final merge of small groups

All pixels in the digital image I_0 are been classified. The next step is merge of small groups or eliminate these groups. As previously, small group has fewer pixels than the parameter set by the *par_area*. Small groups merge with the most appropriate neighbouring group. Merger criterion is the smaller Euclidean distance between RGB averages of two neighbouring groups.

Before deleting or merging small groups identified within the H_{seg} matrix, it is not necessary to traverse the array again. Simply analysing the vector and find the smaller elements that *par_area*, the position vector, of these elements indicates the value of the label of these groups within the matrix H_{seg}, stored in vector *unir*.

The next step is to find neighbouring groups of each small group, with more *par_area* pixels. Using the vecinos expression of all neighbouring groups of small groups of the image, with the exception of those whose pixels are on the edge. In this case, it uses the w 3x3 window to search for group neighbours. To find neighbouring groups of edge pixels of the image must modified according to w window edge where they are looking for these groups.

Thus, the pixel position to which is seeking its neighbouring groups determines the w window. The next task is then to traverse H_{seg} the edges of the matrix and complete the neighbours variable using the w window corresponding pixel according to the position analysed. Once the above process, in the event that any neighbouring row of elements having varying yet, it must filled with a zero.

Then, calculate the Euclidean distance between the RGB averages for each group belonging to the group and all its neighbours groups. For each k element of *unir* vector, it chooses the minimum distance, after the computation of Euclidean distances between the RGB average and all its neighbours. The neighbouring group that has minimum distance is the group that merge k group. This information is stored in a second column of the unir vector created especially for this purpose.

The small groups are identified and what group is merged, the changes is put in the H_{seg} label matrix. Labelling of the H_{seg} matrix is the last step of image segmentation algorithm: The result are image region, identified with a different label value in the HSEG matrix. Segmentation has been used HSV and RGB colour model, identifying areas by tonality, saturation and intensity brightness level, as well as the average of the RGB pixel coordinates of each region.

3.3 Algorithm implementation

It utilizes the filtering images method included in the MATLAB platform. The goal is to designate the minimum image size that can be each targeted group, so that all groups with a smaller size are included in a larger group. The *par_area* parameter of the algorithm implies a minimum group size. The default value of this parameter is set to 25, that is to say, once the image segmentation no groups that contain less than 25 pixels over. This prevents over segmentation, which means that many groups do not constitute an object or a major part, these groups could have a digital area (defined as the number of pixels) as small as a single pixel is created.

The *par_area* parameter, establishes a pixel minimum area of each segmented group. A small value means a high resolution in the segmentation algorithm, but it could have an over segmented image. If the algorithm estimates that *par_area* parameter is greater, the algorithm-processing time is high. A recommendation to adjust this parameter depends on the expected object size in the image.

In an extreme case, the image may be segmented into as many groups as the form pixels, which makes no sense. The filtering process is performed automatically by the "bwareaopen" function of Matlab, which eliminates an original binary image all the components that are under *par_area* pixels with value 0, producing a new binary image without deleted items. To remove pixels with value 1 must be obtained negative image, filtering and then get the negative of the filtered image. The bwareaopen function has three steps: determine the connected components; compute the area of each component and eliminate small objects. It has been used connectivity 4 to set the connected components, considering that a couple of pixels with value 0 are neighbours if they are attached horizontally or vertically, at this level of connectivity if only they are attached diagonally, they are neighbours. See Fig. 3.

When shapes with holes exist, they are removed if less than the minimum number of pixels required to be considered as a region within the segmentation.

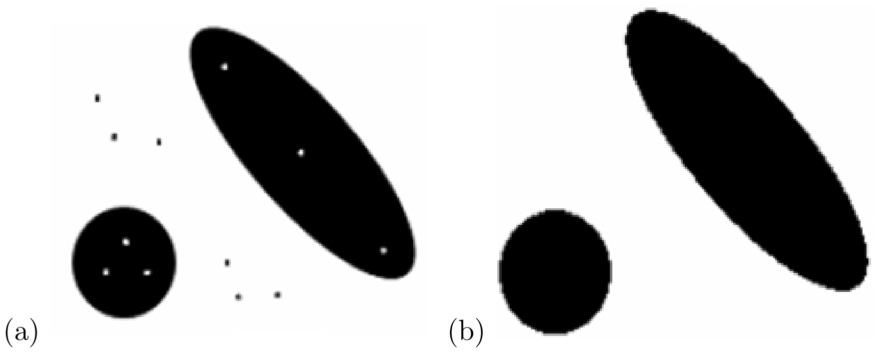


Figure 3: (a) Original image; (b) Filtered image

Once identified the pixels, they are segmented according to their tonality. The neighbouring pixels with the same tonality are merged and those with a different tonality are split. It creates 26 H matrixes, one for each possible tonality value.

The segmentation of the no neighbouring pixels of the same tonality is carried out labelling each H matrix. The labelling consists in assigning a number to each component found in the image. So, the image is scanned from left to right and from top to bottom searching for cell sets with value 1, the first set found keeps its original value, but in the next set, the values “1" are replaced by “2", and so on, until the value “n" is reached, where "n" is the number of components in the scene. A result of the application of the previous procedure to an image is represented in Fig. 4.

The results shows that there is an over segmentation image, because there are small groups of pixels may correspond to shadows, grooves, reliefs, etc. constituting the image details but alone do not represent relevant information. It observes many groups with less than 25 pixels. If they had leaked the H matrices would have removed these small groups, but had lost the information tonality of these pixels. It creates an algorithm capable of merging these small groups with larger groups. This merging algorithm determines the neighbours groups for each component.

The merger of groups lead to the elimination of some regions and increased the size of others, the formation of groups is constantly changing as the segmentation algorithm iterates. Once merging clusters and updating values are done, the first iteration of the tonality segmentation algorithm concludes until no new groups can be fused.

Pixels with low saturation, that have not been segmented by the tonality segmentation algorithm, are segmented by other algorithm using RGB intensity value grey segmentation, which gives better results.

After the segmentation of the groups of pixels with low saturation, there are two possible scenarios for each group. One is that no segmentation is performed and the group remains united and the other is that the pixels with low saturation of the group are separated from their pixels with medium or high saturation, dividing the group into two.

To decide if the group remains united or divided the δ parameter is used, with a default value of this parameter 25. The criteria states that for a group to be divided, the Euclidean distance between two vectors, one containing the coordinates of the mean values of the bands RGB pixels with low saturation of a group, and with other similar pixels with medium and high saturation of the same group must be greater than . In addition the area of the two groups resulting after the division must be greater than the *par_area* division parameter to be performed. The Euclidean distance between two vectors is described by

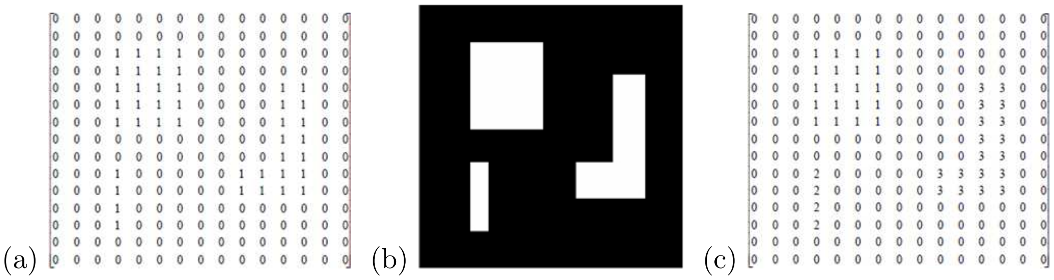
$$D_{sat\_no\_sat}(k) = 0.2264\sqrt{R^2 - G^2 - B^2}, \tag{1}$$

where $R = P_{sat\_R}(k) - P_{no\_sat\_R}(k)$, $G = P_{sat\_G}(k) - P_{no\_sat\_G}(k)$, $B = P_{sat\_B}(k) - P_{no\_sat\_B}(k)$, and k is the label number.

Each element of the $D_{sat\_no\_sat}$ is vector check if greater than δ parameter. If it is greater than δ means that the group associated with this element has to always be divided. When the area of the resulting two groups is greater than *par_area* parameter, the group to divide, a record is stored in the *separa* variable is created, which is defined by the expression

$$separa(k) = \begin{cases} \max(separa) + 1, & \text{if } D_{sat\_no\_sat}(k) > \delta \wedge area_{sat}(k) > par\_area \\ 0 & , \text{ if } D_{sat\_no\_sat}(k) \leq \delta \vee area_{sat}(k) \leq par\_area \end{cases}, \tag{2}$$

where k is the label number an the initial value of *separa* is zero.



(a)

$$\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&1&1&1&1&0&0&0&0&1&1&0&0\\
0&0&0&1&1&1&1&0&0&0&0&1&1&0&0\\
0&0&0&1&1&1&1&0&0&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&0&0&1&0&0&0&0&0&1&1&1&1&0&0\\
0&0&0&1&0&0&0&0&0&1&1&1&1&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

(b)

(c)

$$\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&1&1&1&1&0&0&0&0&3&3&0&0\\
0&0&0&1&1&1&1&0&0&0&0&3&3&0&0\\
0&0&0&1&1&1&1&0&0&0&0&3&3&0&0\\
0&0&0&0&0&0&0&0&0&0&0&3&3&0&0\\
0&0&0&0&0&0&0&0&0&0&0&3&3&0&0\\
0&0&0&2&0&0&0&0&0&3&3&3&3&0&0\\
0&0&0&2&0&0&0&0&0&3&3&3&3&0&0\\
0&0&0&2&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&2&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

Figure 4: (a) Original matrix binary; (b): Binary matrix image; (c): Matrix labeled

Once the labelling, all new groups are identified. The δ parameter is the maximum Euclidean distance between the mean of each low saturation pixels RGB average in each group, and all their neighbouring groups.

To decide which groups with low saturation will be merged, the Euclidean distance between the averages of each RGB band of each group with respect to all its neighbouring groups is calculated, if the distance is less than a certain set δ_2 value, then it proceeds to the merger groups. In other words, each group will be merged with its neighbours groups with which Euclidean difference between the averages of RGB values is less than the parameter δ_2. The newly described Euclidean distance is defined by following expression with its mixture of pixels with different levels of saturation,

$$D_{P\_RGB}(k)_i = 0.2264\sqrt{R_v^2 - G_v^2 - B_v^2}, \tag{3}$$

where $R_v = P_R(k) - vecinos(k)$, $G_v = P_G(k) - vecinos(k)$, $B_v = P_B(k) - vecinos(k)$, $k = 1$, 2, ..., $\max(H_{seg})$, and $i = 1$,2,..., quantity of neighbours groups to k group.

The temporal information groups to be merged is stored in the *agrupa* variable, expression

$$agrupa(k) = agrupa(k) \cup v(k), \tag{4}$$

where $v(k) = \left\{vecinos(k)_i | D_{P\_RGB}(k)_i \leq \delta_2\right\}$.

Then, only once all the groups will be merged known proceed to label matrix H_{seg}. In the case, that there is more than one stored in *agrupa*(k) value, the minimum distance of RGB average with group k is chosen, and the remaining values are removed. A new labelling of H_{seg} and labelling groups, is used the fusion algorithm.

Once all kinds of segmentation described above, it is possible that still exist unclassified pixels. These correspond to those groups formed by a smaller number of pixels than that set by the *par_area* parameter, which is set to 25.

To identify all the remaining pixels, it has to traverse the array in searching of pixels with value 0. The information regarding the position of these pixels within H_{seg} is stored in a new array called R.

When all pixels in the digital image are classified, the next step is to remove all small groups. As previously stated, a small group is one that is composed of a smaller number of pixels than that set by the parameter *par_area*. Small groups merge with the most appropriate neighbouring group.

Once the small groups are identified and known to be merged with that group, you must register the respective label changes in the matrix. For this, we have created an algorithm.

Summarizing, the image segmentation method has three parameters: *par_area*, δ and δ_2.

The *par_area* parameter, establishes a pixel minimum area of each segmented group. A small value means a high resolution in the segmentation algorithm, but it could have an over segmented image. If the algorithm estimates that *par_area* parameter is greater, the algorithm-processing time is high. A recommendation to adjust this parameter depends on the expected object size in the image.

The δ parameter is the maximum Euclidean distance between the mean of each low saturation pixels RGB average in each group, and all their neighbouring groups. If this distance is less than a certain established value, the parameter is called δ_2; in this case, the groups are merged. In other words, each group will be merged with all neighbouring groups with which the Euclidean distance between their mean RGB values is less than δ_2.

The δ and δ_2 work with low saturation pixels, controlling the pixel tendencies to be in a higher saturation group or to merge to neighbouring groups with low saturation pixels. The image segmented groups probably grows when these parameters values diminishes. Their configuration has to consider the different tonalities. For example, in the image with several objects of similar tonalities, it has to separate groups with low saturation and to avoid those neighbouring groups with the same saturation that merge. This is achieved reducing δ and δ_2 values, changing the variable value in the proposed algorithm.

The *par_area*, δ, and δ_2 parameters should have to adjust to allow the proposed segmentation algorithm to adapt to images with different colour characteristic. During evaluations of this proposed point of view, the parameters values remain constant in all image tested, obtaining good segmentation. However, the results estimated can be improved choosing optimal parameters values in each digital image to be segmented. In the algorithm, the initials parameters are $par\_area = 25$, $\delta = 5$ y $\delta_2 = 2$.

The HSV possible values are reduced from 256 to 26, in an arbitrary way, to have good results in images with small tonality, instead of reducing all tonalities spectrum presented.

4 Evaluations of proposed segmentation method

4.1 Proposed segmentation method test

To test the proposed segmentation method, it utilize a House image and Lego image.

Figure 5a shows the original House test image, having neighbouring regions with similar tonal values inside; different objects with complex shapes and different textures; and shadows in some regions of the image. Figures 5b and 5c show a good performance segmentation, despite some shadow areas segmented as a region. Objects with complex shapes are segmented, without an over-segmentation or excessive union of regions.

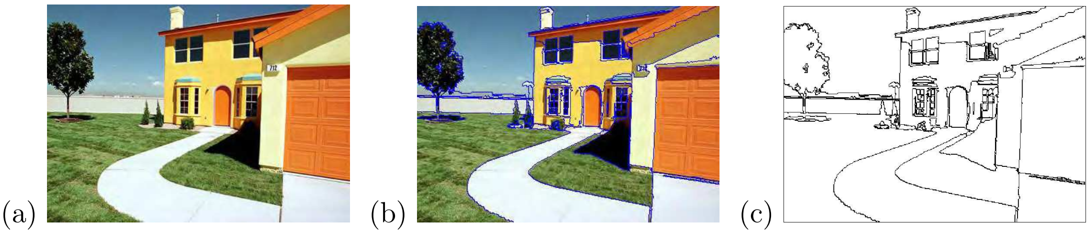


Figure 5: (a) Original matriz binary; (b): Segmented image; (c): Segmented regions

The house image test has a resolution of 449x322. Proposed algorithm obtains a segmented image with 120 groups and it lasts 35.19 seconds. Proposed segmentation method has good performance.

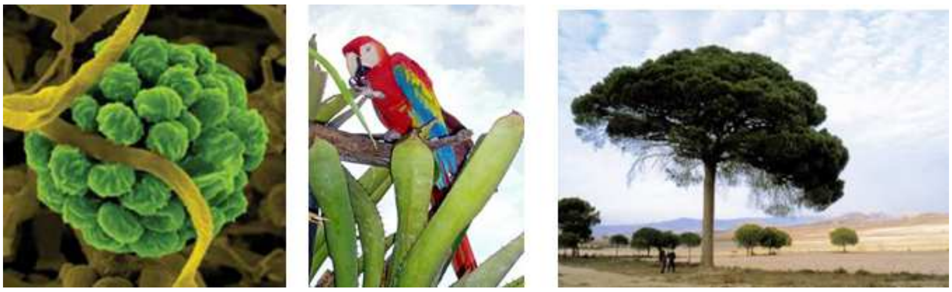


Figure 6: Original images for the evaluation

To calculate the processing time, it is necessary to consider that the algorithm has been programmed in version 7.0 (release 14) MATLAB and has been run on a computer with Intel Centrino processor only 1.86 GHz and 1.5 GB of RAM. The processing time is linked to the number of regions in the image. An image so large and few regions may require less processing time than and image of smaller size and more regions to segment. Furthermore, for images with the same number of regions and different resolution, it requires a longer processing time. In summary, the processing time required by the algorithm depends on both the image size and the number of regions present therein, the latter being unknown amount before execution algorithm.

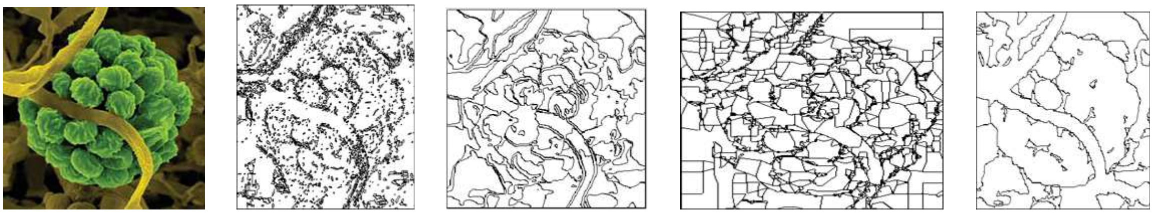
To compare the results with other methods, it is used the objective evaluation method proposed in [18], since it does not require the setting of parameters or a reference image. The $L(I)$ evaluation function is defined through equation (5).

$$L(I) = \sqrt{N_r}[1000(MN)]^{-1} \sum\nolimits_{i=1}^{N_r} (e_i^2 A_i^{-1/2}), \tag{5}$$

where I is the original image to be segmented N_r is the number of segmented regions in I, A_i is the number of pixels in the region R_i, M and N are the height and the width of I, and e_i^2 is the colour error in the R_i region, which is defined as the sum of the Euclidean distance of the RGB colour vector between I and the corresponding segmented image for each pixel in the region.

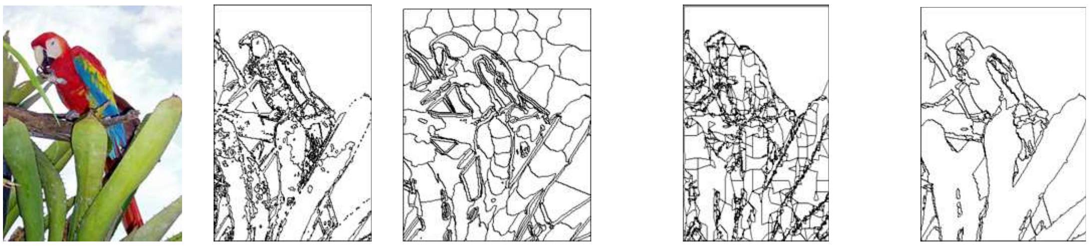
Smaller L value means a better performance. The evaluation is performed comparing the obtained results by means of the proposed algorithm with results obtained by other three methods: HCI fuzzy segmentation [11], segmentation using situational DCT (Discrete Cosine Transform) descriptors [12] and parameterless quadrilateral-based image segmentation method [10].

Figures 7, 8, and 9 show image segmentation using the different methods. Where (a) is the original image, (b) the HCI fuzzy segmentation, (c) the segmentation using situational DCT descriptors, (d) is the parameter less quadrilateral-based image segmentation, and (e) the proposed image segmentation.



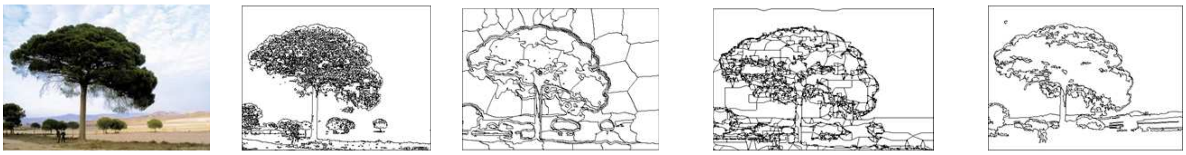
(a) Original (b): HCI (c): DCT (d): Less quad-based (e): Proposed

Figure 7



(a) Original (b): HCI (c): DCT (d): Less quad-based (e): Proposed

Figure 8



(a) Original (b): HCI (c): DCT (d): Less quad-based (e): Proposed

Figure 9

Parameter less quadrilateral-based image segmentation is constructed from a map edge (generated from a segmentation algorithm edges), neighbours quadrangles which have similar characteristics are fused to generate quadrilaterals regions allow local variations eliminating unnecessary details, so each region is full and accurately described by a set of quadrilaterals having neighbouring pixels between them. These areas have been separated into different regions, generating more groups. The amount of resulting groups has been called actual or real number of groups, while the original amount of groups calculated by the respective method is called simply number of groups.

Since (1) penalizes the creation of new groups, which increase the error in those segmentation methods that tend to do over segmentation. The $L(I)$ error is always less than the real $L(I)$ error. When the HCI fuzzy and the situational DCT descriptors segmentation methods are compared with the proposed method, the real $L(I)$ error must be considered. The objective analysis of the results summarized in the Table 1 demonstrates that the proposed method presents a smaller error and it requires less processing time than the methods used during its evaluation when they are applied to the three test images presented in Figs. 6, 7, 8.

Table 1: Comparison of results with other methods

Image	**Stachybotrys Chartarum**	**Parrot**	**Tree**
Resolution	300x300	256x340	450x315
Method 1: HCI Fuzzy Segmentation			
Number of Groups	6	8	5
L(I) Error	0.0731	0.0869	0.0275
Real Number of Groups	1731	1029	1990
Real Error L(I)	1.3967	1.5517	0.8052
Time (sec)	381.47	604.66	641.58
Method 2: Parameterless Quadrilateral-based Image Segmentation			
Number of Groups	62	106	105
L(I) Error	0.3511	0.8985	0.41
Real Number of Groups	252	448	390
Real Error L(I)	0.9105	2.54	1.0628
Time (sec)	1706.20	1622.06	4287.42
Method 3: Segmentation Using Situational DCT Descriptors			
Number of Groups	347	357	446
L(I) Error	4.1938	10.282	12.3995
Time (sec)	4742.25	7551.27	12071.94
Method 4: Proposed Segmentation Method			
Number of Groups	62	106	105
L(I) Error	0.199	0.4654	0.2781
Time (sec)	24.75	30.11	36.08

The region quantity is also smaller than in the other methods. It is compact and contents precise regions. Hence, the proposed method outperforms the other segmentation algorithms especially for real time applications. All experiments are carried out using the same computer.

The HCI Fuzzy segmentation method resulted to have a good segmentation in high value zones and small intensity, but it can over segment in other zones. Its processing times are large. In comparison, the DCT method has good segmentation in some regions, but over segmentation in others. The parameter less quadrilateral-based image segmentation method has the same problem. In this latter case, the processing time was very large.

The obtained results indicate that the proposed method is fast without over segmentation. It has good segmentation capability compared to the other methods. Few regions were not segmented, but with overall better performance delivering more homogeneous regions of colour, spatially compact and continuous.

The processing time required by the proposed algorithm in order to perform the segmentation mainly depends of the image size as much as the initial quantity of regions contained in the image. The evaluation of the proposed method indicates that this requires less than 7% of the processing time required by the approaches involved in the comparison in order to perform the segmentation over the test images.

Therefore, it is estimated that the proposed approach is suitable in order to be implemented in real applications with small processing times. The parameters par_area, δ, and δ_2 must be adjusted to allow the segmentation with different colour features. During the evaluation, the value of these parameters was maintained constant for all testing images, obtaining good results in the segmentation. In the proposed algorithm, the possible values of each HSV channel that the pixel image can take have been arbitrarily reduced from 256 to 26. This reduction has provided good results. It was considered that a scarce variation of tonality was preferable to reduce the complete HSV spectrum. All corresponding algorithms to each method run on the same computer (Intel Centrino processor 1.86 Ghz, 4 GB of RAM). As noted in Table 1, the proposed method requires less than 7% of the processing time involved employing approaches in comparison to the segmentation of the test images. However, the computer processing time required can decrease further proposed implemented in C++ programming language.

Conclusions

A new unsupervised colour image segmentation algorithm is presented. This method combines the advantages of the approaches based on split and merge and region growing, and the use of the RGB and HSV colour representation. The effectiveness of the proposed method has been checked by means of its implementation using an algorithm, which supplies homogeneous regions, spatially compacts and continuous. It has been observed that the proposed algorithm outperforms other segmentation algorithms and requires shorter processing time.

The combinations of both segmentation techniques results in an increment of the rate of the division in the split and merge method. The determining the initial number in the growing region is solved. The proposed algorithm is compared with the results obtained by other three methods: HCI fuzzy segmentation, segmentation using situational DCT (Discrete Cosine Transform) descriptors and parameter less quadrilateral-based image segmentation method.

The obtained result indicates that the proposed method is fast, does not have over segmentation, it has good segmentation and is better than the other methods. A few regions are not segmented. The method has better performance, delivering more homogeneous regions of colour, spatially compact and continuous.

The processing time required by the proposed algorithm in order to perform the segmentation mainly depends on the image size and the amount of regions contained within the image. The evaluation of the presented method indicates that this requires less than 7% of the processing time required by the compared approaches making it a good candidate where fast image processing is required.

The proposed method can make good segmentation without previous information. It is estimated that it can perform better if par_area, δ, and δ_2 parameters are adjusted automatically and to design a method to reduce the HSV space automatically according to the images colour characteristics.

Bibliography

[1] Scheleyer, G.; Cubillos, G.; Millán G., Osorio-Comparán, R.; Lefranc G. (2016); A New Colour Image Segmentation, *Computers Communications and Control (ICCCC), 2016 6th International Conference on*, IEEE Xplore, DOI: 10.1109/ICCCC.2016.7496766, ISBN 978-1-5090-1735-5, 232-239.

[2] Sanmati S. Kamath; Jackson, J.(2006); Color Image Segmentation in RGB Using Vector Angle and Absolute Difference Measures, *Proc. 14th European Signal Processing Conference (EUSIPCO 2006)*, 1-5.

[3] Navkirat Kaur, V. K.; Banga, Avneet Kaur (2013); Image Segmentation Based On Color, *Proc. IJRET: International Journal of Research in Engineering and Technology*, 495-497.

[4] Shamik Sural; Gang Qian; Sakti Pramanik (2002); Segmentation and Histogram Generation Using the HSV Color Space for Image Retrieval, *Proc. 2002 International Conf. on year 2002* 2(2):589-592.

[5] Zhi-Kai Huang; De-Hui Liu (2007); Segmentation of Color Image Using EM algorithm in HSV Color Space, *Information Acquisition Intl Conf.*, 8(11): 316-319.

[6] Kanchan Deshmukh; IAENG, B.; Ganesh Shinde (2006); Adaptive Color Image Segmentation Using Fuzzy Min-Max Clustering, *Engineering Letters*, 13(2): 1-8.

[7] Krishna Kant Singh; Akansha Singh (2010); A Study of Image Segmentation Algorithms For Different Types Of Image, *IJCSI International Journal of Computer Science*, 7: 414-417.

[8] Dibya Jyoti Bora; Anil Kumar Gupta (2014); Clustering Approach towards Image Segmentation: An Analytical Stud, *IJRCAR*, 12(7): 115-124.

[9] A. Jurio, M. Pagola, M. Galar, C. Lopez-Molina (2010). A Comparison Study of Different Color Spaces in Clustering Based Image Segmentation, *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications*, 81: 532-541.

[10] Chung, R., Yung N., Cheung P. (2005); An Efficient Parameterless Quadrilateral-Based Image Segmentation Method, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(9): 1446-1458.

[11] Carron, T; Lambert, P. (1996); Symbolic Fusion of Hue-Chroma-Intensity Features for Region Segmentation, *Proc.1996 IEEE Intl Conf. on Image Processing*, 1: 971-974.

[12] Jie Wei (2001); Image Segmentation Using Situational DCT Descriptors, *Proc.2001 IEEE Intl Conf. on Image Processing*, 1: 738-741.

[13] Osorio-Comparán, R.; Olivera, M.; Pe na, M.; López-Juárez, I., Savage, J.; Lefranc, G. (2013); Using Background and Segmentation Algorithms Applied in Mobile Robots, *6th IFAC Int. Conf. on Management and Control of Production and Logistics*, 6(1): 135-140.

[14] Rajeshwar Dass; Priyanka, Swapna Devi (2012); Image Segmentation Techniques, *Intl Journal of Electronics & Communication Technology*, 3(1): 66-70.

[15] MD Hedayetul Islam Shovon (2012); *Clustering and Image Segmentation*, LAP Lambert Academic Publishing, 2012.

[16] Nida M. Zaitoun; Musbah J. Aqel (2015); Survey on Image Segmentation Techniques. *Procedia Computer Science*, 65: 797-806.

[17] Szeliski, R. (2010); *Computer Vision: Algorithms and Applications*, Springer.

[18] Liu, J., Yang, Y. (1994); Multiresolution Color Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7): 689-700.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(6):877-888, December 2016.

Coverage Optimization Strategy for WSN based on Energy-aware

L. Zhu, C. Fan, H. Wu, Z. Wen

Li Zhu, Chunxiao Fan*, Zhigang Wen
Beijing University of Posts and Telecommunications,
Beijing Key Laboratory of Work Safety Intelligent Monitoring,
School of Electronic Engineering, No.10 Xitucheng Road, Beijing, P.R.China, 100876
jolie.zhl@hotmail.com
*Corresponding author: cxfan@bupt.edu.cn
zwen@bupt.edu.cn

Huarun Wu
National Engineering Research Center for Information Technology in Agriculture,
Key Laboratory for Information Technologies in Agriculture, Ministry of Agriculture,
Room 316, Beijing Agriculture Science and Technology Building A,
No. 11 Beijing Shuguang Garden Middle Road,
Haidian District West Suburb, Beijing,China, 100097
wuhr@nercita.org.cn

Abstract: In order to optimize the wireless sensor network coverage, this paper designs a coverage optimization strategy for wireless sensor network (EACS) based on energy-aware. Under the assumption that the geographic positions of sensor nodes are available, the proposed strategy consists of energy-aware and network coverage adjustment. It is restricted to conditions such as path loss, residual capacity and monitored area and according to awareness ability of sensors, it would adjust the monitored area, repair network hole and kick out the redundant coverage. The purpose is to balance the energy distribution of working nodes, reduce the number of "dead" nodes and balance network energy consumption. As a result, the network lifetime is expanded. Simulation results show that: EACS effectively reduces the number of working nodes, improves network coverage, lowers network energy consumption while ensuring the wireless sensor network coverage and connectivity, so as to balance network energy consumption.

Keywords: WSN, coverage optimization, energy-aware, hole repair, sensing radius.

1 Introduction

Wireless sensor network (WSN) consists of many sensor nodes which draw on self-organization to transmit data between nodes. These sensor nodes are small in size with limited energy and certain awareness, making them widely applied to many fields, such as military, transportation, environment protection, medical care, disaster relief and agriculture. Currently, sensor nodes are distributed manually or randomly. Manual distribution is executed in certain circumstances requires high on environment. So in most cases, random distribution [1–3] is the primary choice as it is easy to operate and able to reduce the interference on human activities. However, under circumstances of high density and large area, wireless sensor network is weak in expansion, which results in that the transmission signal is easy to be disturbed, the network energy is unevenly distributed and network become unstable.

It is urgent to solve problems of increasing the effective coverage in monitored areas as much as possible, lowering network energy consumption, extending network lifetime, and improving network performance under the condition of limited energy and using the least sensor nodes. And network coverage is one of the most important indicators of network performance measurement.

Among existing studies, literature [4] proposes a method to calculate redundancy of probability node. With this method, nodes in the network can acquire information about their own, such as redundancy, without knowing the geographic position. It also proposes a node adjustment strategy without knowing node position. But this strategy neglects the awareness redundancy area within two-hop neighbor nodes, making many redundant working nodes exist during node adjustment. Literature [5] proposes a node distribution strategy based on wireless beacon self-adaption. This strategy repairs network coverage holes by adding wireless beacon to the network but fails to consider the cost of adding the beacon and the influence on monitored area. Literature [6] proposes to use a circle to solve node adjustment coverage. It calculates the angle between the awareness circle and the monitored area to get the minimum number of working nodes and reduces redundant working nodes. But under high coverage, the connectivity is weak. Literature [7] acquires information of null nodes and introduces mobile nodes. To be specific, it replaces null nodes with mobile nodes to fill the network holes. The network coverage is increased, but the energy of mobile nodes is limited and there are too many null nodes in the network. Literature [8] proposes a coverage strategy based on genetic algorithm. It selects the maximum solution set of coverage nodes. With genetic algorithm, it uses evaluation function to optimize random samples and reach a balance of network coverage. But the algorithm is relatively complicated and requires much calculate. To some extent, it increases network energy consumption.

In order to improve effective coverage, this paper proposes a coverage optimization strategy for WSN based on energy aware, establishes a network coverage model. Network nodes are distributed randomly; a network coverage model is established the relationship between node residual capacity and sensing range are examined to set reasonable sensing range of nodes, balance network energy consumption, repair network coverage holes, and extend network lifetime. Finally, a simulation experiment is conducted to prove the relationship between network energy consumption and coverage of the proposed algorithm under different circumstances. The proposed algorithm aims at balancing network energy consumption, improving network coverage, and minimizing node energy consumption. This paper establishes a non-linear model of coverage optimization strategy for wireless sensor network (EACS) and exercise stricter constraints to turn it to linear-constraints and get the second-best solution.

2 Network model and problem description

2.1 Analysis model of energy-aware node

Suppose the wireless sensor network consists of N randomly distributed nodes, during node awareness, the energy of physical signal changes inversely with the distance between the signal and the awareness target. Such tendency is mainly due to path attenuation during transmission, Select any source node s_i within the scope of monitoring, when target node s_j is anywhere in the plane, the awareness intensity $\Psi_i(j)$ of souse node to target node is expressed as:

$$\Psi_i(j) = \begin{cases} 0 & R_s < d(i,j) \\ \lambda e^{-kd(i,j)} & 0 < d(i,j) \leq R_s(1) \end{cases} \tag{1}$$

where k is the indicator of signal attenuation. R_s is the maximum effective sensing radius of the node. $d(i,j)$ is the Euclidean distance between node i and j. λ is a constant value. The awareness intensity $\Phi(j)$ of node j is expressed as:

$$\Phi(j) = 1 - (1 - \Psi_1(j))(1 - \Psi_2(j)) \cdots (1 - \Psi_i(j)) \cdots (1 - \Psi_n(j)) = 1 - \prod_{i=1}^{i=n}(1 - \Psi_1(j)) \tag{2}$$

When $\Phi(j) > \epsilon$, target node j is sensed. When $\Phi(j) < \epsilon$, target node j is not sensed, and at this time, j is the coverage blind-point (ε target node j is not sensed, and at this time, j is the coverage blind-point

$$\begin{array}{ll} H_0 : g(i) = \vartheta_i & i = 1, 2, \ldots, N \\ H_1 : g(i) = \vartheta_i + \psi_i(j) & i = 1, 2, \ldots, N \end{array} \tag{3}$$

Where ϑ_i is the background noise signal following the normal distribution $\vartheta_i \sim N(\mu, \sigma^2)$, and $\vartheta_i(j)$ is useful signal. The signal detected by sensor nodes is $g(i)$. The target that exits is expressed by H_0 and the target that does not exist is expressed by H_1.

2.2 Network model

N non-overlapped wireless sensor nodes, $S = \{s_i | 1 < i < n\}$, whose sensing radius is R_s are randomly distributed in a two-dimensional monitored area A. Suppose the network has the following features:

(1) The node position is permanent. There is no inherent relationship between the communication radius R_c and sensing radius R_s.

(2) All nodes have isomorphism. The wireless self-organized network is constructed in the monitored area.

(3) All nodes adopt the node probability awareness model.

(4) The sensing radius of sensor node can be adjusted according to node residual capacity.

(5) The initial energy of sensor nodes is W, and they have synchronous clock.

(6) Sensor nodes use the location technology in literature to acquire their own locations. At the same time nodes can acquire the information of neighbor nodes, such as residual capacity, node position within effective communication distance.

2.3 Problem description

A number of sensor nodes are distributed randomly within the monitored area A. Probabilistic sensing model is deployed, and network nodes are adjusted according to data from 2.2network model; a wireless sensor network is established to balance network energy consumption and network coverage, adjust sensing ranges based on residual capacities, reduce the possibility of "death node" and redundant network coverage, and extend network lifetime. This paper expects to reach a balance between residual capacity of any sensor node and sensing range in the monitored area (where W_i is the residual capacity, $\Im_i$ is the sensing range and A is the monitored area),

$$W_i < W_j \quad \&\& \quad \Im_i < \Im_j \tag{4}$$

$$A = \Im_1 \bigcup \Im_2 \cdots \bigcup \Im_i \cdots \bigcup \Im_n = \bigcup_{i=1}^{i=n} \Im_i \tag{5}$$

In the monitored area A, nodes should satisfy (3)(4) to reach the requirement of coverage. This paper also turns energy consumption and coverage balance strategy to a non-linear and multi-target optimization problem:

$$\begin{cases} 0 < Cov_{\max} = f(\sum_{i=1}^{i=n} cov_i) \leq A \\ 0 \leq W_{\min} = \varphi(\sum_{i=1}^{i=n} \Delta w_i) \leq nW_0 \\ T_{\max} = \max(\min(T_1, T_2 \cdots T_n)) \\ \zeta_{\min} = \Theta(\sum_{i=1}^{i=n} \zeta_i) \end{cases} \tag{6}$$

Given that path loss, node residual capacity [9] and monitored area are key factors [10–12] to realize network performance optimization and extend network lifetime, the constraint conditions are dealt with according to mathematical programming so that the optimized network has high accuracy. Construct the optimized model with maximizing the network lifetime as the target. The optimized model is restricted to constraint conditions. Adjust the monitored area, and optimize the network lifetime and the overall network overhead. In the non-linear and multi-target optimization, optimized factors are not in the optimal state, but the network overhead is almost the minimum and the convergence has high accuracy.

3 Energy coverage strategy

Coverage strategy of the wireless sensor network optimize network energy consumption while improving network coverage [13–15]; in addition, they should also balance energy consumption of single node and overall network energy consumption. If nodes with lower energy are assigned the same work load as nodes with higher energy, the former will end in "premature death" and the transmission and reliability of the entire network will be undermined. To reach the balance of two optimization coverage strategies and solve the contradiction between individual nodes and the overall performance, this paper proposes a coverage optimization strategy for WSN based on energy-aware, namely Energy-aware Coverage Strategy (EACS). The strategy is mainly divided into two phases: one is the energy-aware phase in which the sensing field of each node is confirmed by probability according to residual capacity of sensor node in the monitored area. The second is the network coverage adjustment phase where the sensing radius of node is adjusted effectively according to the sensing field of each node and the overall network coverage to lower the redundant coverage, reduce redundant coverage and unnecessary energy consumption, so as to extend network lifetime.

3.1 Energy-aware phase

In the wireless sensor network, sensor nodes in the monitored area transmit information through awareness coordination. Given that sensor nodes have limited energy, this strategy concerns not only about energy consumption of any node in the area, but also the equilibrium of the whole network energy consumption. As time goes by, survived nodes may suffer from breaking the equilibrium of energy consumption due to signal interference, resulting in the change of node residual capacity. Considering that node residual capacity is related to node sensing range, as shown in Fig.1, reasonable sensing range is set up for each node to balance the network energy consumption and finally, to extend network lifetime.

In the energy-aware phase, after time t of working, the relationship between the electric quantity W_i consumed by node si and the sensing radius R_i of sensing field A_i is (k is a constant value):

$$W_i = kR_i^2 \tag{7}$$

For any two neighboring nodes s_i and s_j, their residual capacity is Q_i and Q_j respectively. After time t, their electric quantity is consumed simultaneously. The sensing radius between two nodes fits expression (2):

$$R_i = d(i,j) \cdot \frac{\sqrt{Q_i}}{\sqrt{Q_i} + \sqrt{Q_j}} \tag{8}$$

Where d (i, j) is the Euclidean distance between nodes. According to expression (8), the sensing radius and residual capacity of s_i and s_j have the following relationship:

$$R_i : R_j = \sqrt{Q_i} : \sqrt{Q_j} \tag{9}$$

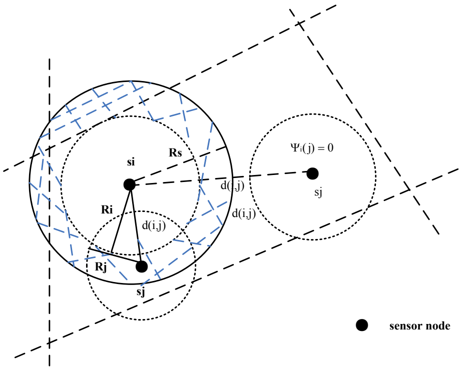


Figure 1: Node sensing range

Theory 1 The sensing filed of node S_i is Ω, its awareness is radius R_i. Ω falls in the set $\gamma(\gamma_1, \gamma_2, \cdots \gamma_i)$ and any γ_k satisfies $d(si, \gamma_k) < Ri$. If the probability of $X(X \in \Omega)$ sensed by any node in γ is p, there are Pxsensing fields in Ω on average that are sensed by γ.

Suppose the sensed set in the sensing field Ω is $\Pi\{x_1, x_2 \cdots x_n\}$, and these nodes are independent from each other following even distribution. If $k(k < n)$ nodes in being sensed by γ is called event X, so X follows binomial distribution, namely $X \sim B(n,p)$.

$$P\{X = k\} = C_n^k p^k (1-p)^{n-k}$$

where $E[X]$ is the expectation of X. When $n \to +\infty$, the average awareness in Ω is:

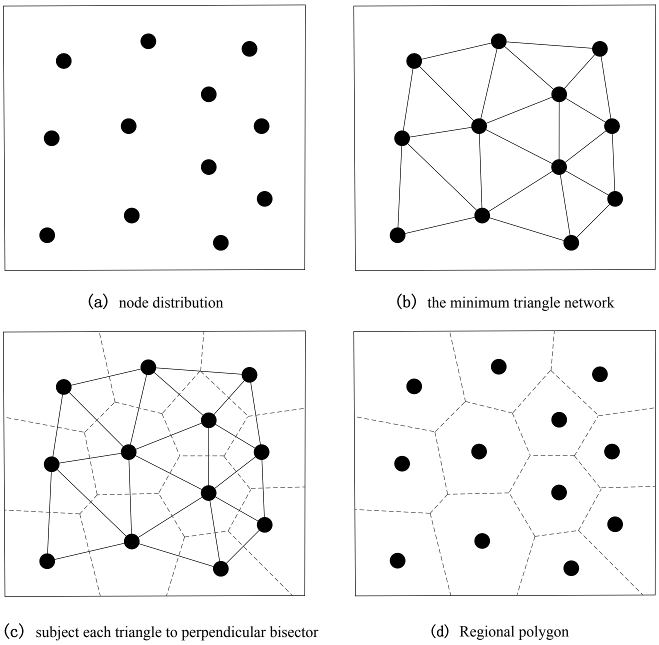
$$\lim_{n \to +\infty} \frac{E[X]}{n} = p$$

So there are psensing fields on average that are sensed by γ in Ω.

3.2 Network coverage adjustment

Select any node Si in the monitored area, according to the energy-aware strategy, acquire the energy-aware range and node residual capacity of last phase and send the information such as sensing radius and energy consumption to the neighboring node. According to awareness intensity, confirm the sensing range and monitored area of each node. When there produce holes in the monitored area, re-distribute the nodes. Given that the sensing field may overlap, when no new holes are produced, adjust the sensing radius based on node residual capacity to wipe out redundant monitored area, so as to reduce unnecessary energy consumption. Detailed steps are described below:

Step 1ŁşDivide the monitored area. For any node in the area, connect it with neighboring nodes and form the minimum triangle network as shown in 2(b). Subject each triangle to perpendicular bisector and connect the lines, as shown in 2(c). Finally, monitored area A consists of many regional polygons, as shown in 2(d).


(a) node distribution

(b) the minimum triangle network

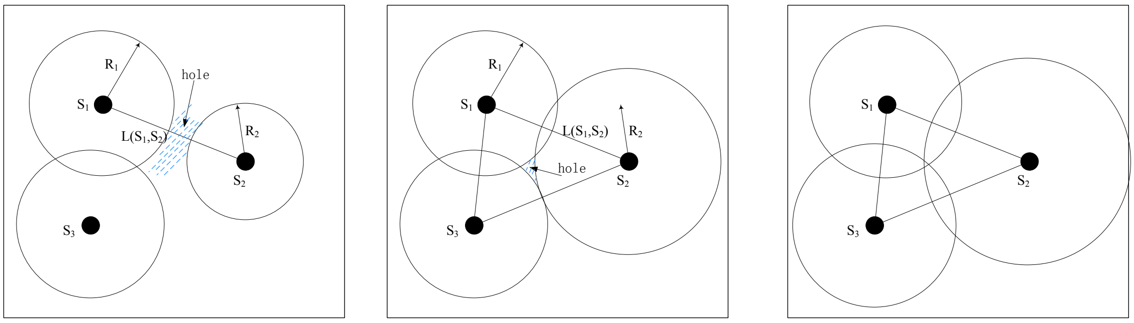
(c) subject each triangle to perpendicular bisector

(d) Regional polygon

Figure 2: Distribution of monitored area

Step 2: detect irreparable holes and nodes filling. Firstly, detect holes on the edge of monitored area A. The polygon which cannot be fully covered by the minimum triangle network is called the edge monitoring polygon. Take the vertex of the polygon at the edge of the monitored area as the circle center and twice the maximum sensing radius of nodes R_{max} as the radius to draw a circle. Places that cannot be covered by the circle are irreparable holes and need to be filled. New nodes should fill in the middle of the vertex and original nodes. For any side of the triangle in the network $< S_1, S_2 >$, if the side is twice more than the maximum sensing radius of nodes, namely $L(S_1, S_2) >$ 2Rmax, it is also considered that there are irreparable holes. Thus, we should fill node in the middle of the triangle side. After nodes are filled, repeat step 1 to construct the new triangle network and the monitored polygon. Go on to step 3 until no new nodes should be filled.

Step 3: detect regional holes. Confirm the initial sensing radius of nodes according to expression (9). Construct the network model with the minimum triangle and detect the triangle network. To be specific, judge whether a minimum triangle $S_1S_2S_3$ has sensing holes. If one side of the triangle $< S_1, S_2 >$ is longer than the total sensing radius of two tip nodes, namely $L(S_1, S_2) > R(S_1) + R(S_2)$, as shown in 3(a), then $\triangle S_1S_2S_3$ has holes. If three sides fit $L < R_i + R_j$, in other word, two circles intersect, the premise for no holes is that any circle intersects with the overlapping part of the other two circles. That is to say, the intersection of any two circles should fall within the sensing radius of the third one. Otherwise, $\triangle S_1S_2S_3$ has holes, as shown in 3(b) and 3(c). When all minimum triangles fit this rule, a set with holes is formed. Calculate the number of nodes that have appeared, and form the hole-node set $\{S_i, \ldots, S_j\}$.



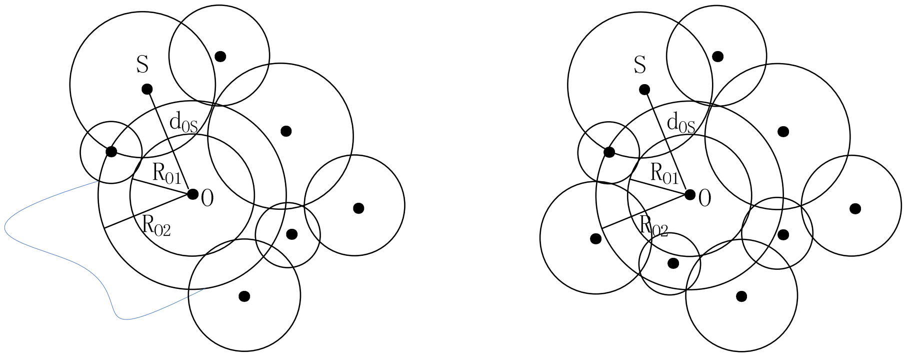
(a) the total sensing radius < the triangle side

(b) the intersection circles with hole

(c) the intersection circles without hole

Figure 3: Network hole detection

Step 4: hole repair. Start repairing from the nodes that appeared the most times in the statistics in step 2. If two nodes appear the same times, the one with higher residual capacity is adjusted primarily. For example, S_i is the first one to be filled. Increase the transmitting power of S_i, by adding its sensing radius, until all triangles with S_i as the tip node do not have holes. Repeat step 4 until all holes are filled, as shown in Fig.4



(a) WSN with coverage hole

(b) WSN without coverage hole

Figure 4: Coverage hole repair

Step 5: wipe out the redundant holes. Get the complementary set of the triangle set with holes in step 2, namely the minimum triangle set without holes. As the initial sensing range of nodes is big, or two neighboring nodes are close to each other, there may be redundant coverage. Related nodes with relatively large sensing radius may consume unnecessary energy. Thus, it is necessary to adjust the sensing range of nodes in the complementary set. Firstly, calculate the number of nodes that appeared and rank them from the most appeared to the least appeared. Priority is given to nodes with less residual capacity. Reduce the sensing radius of nodes gradually until no new holes appear in the triangle area. Repeat step 5 to all nodes in the complementary set.

4 Analysis of algorithm performance

4.1 Analysis of coverage quality

Conclusion 1 If the awareness intensity probability of j by node S in monitored area A is $P\{\tau < \Phi(j) < 1\}$, where $Ss_1, s_2 \ldots s_n$ is the node set and τ is the threshold of awareness intensity. When and only when the expectation value of network coverage γ in A is bigger than γ_0 (where γ_0 is the coverage threshold), namely $E[\gamma] > \gamma_0$, the coverage in A meets the requirement of the network coverage.

Coverage connectivity is one of the important indicators to measure the service quality of wireless sensor network. According to the model, the probability of awareness intensity for node q in monitored area A is $P\{\tau < \Phi(j) < 1\}$. Elements in the node set S form a connected graph $G(V, E)$, where V is the node set, and E is the side set. Nodes are independent from each other. The connectivity degree of G is $\Xi(G)$ (n is the number of selected nodes).

$$\lim_{n \to +\infty} P\{\Gamma(Sn, V, \Xi(G)) \geq k\} = 1$$

When n approaches infinity, the awareness probability of node in the connected graph increases. So does the connectivity degree. According to the definition, the expectation value of network coverage γ is:

$$E[\gamma] = E[\iint_A Cov(i)dA/\|A\|] = P\{\tau < \Phi(i) < 1\} > \gamma_0$$

Select the sub-areas in the monitored area randomly. According to the probability event, when the connectivity degree has a high probability, the node sets in the area are all connectivity set and the network coverage would also meet the expectation.

4.2 Analysis of network lifetime

Conclusion 2 Distribute n sensor nodes in monitored area A. The node set is $S = \{s_1, s_2 \ldots s_i \ldots s_n\}$ and s_i and s_j are neighboring nodes. The neighboring node set of s_i is N$_i$. When the network parameters are set the same as that of the network model, the network lifetime is $Max\ t_i$ (t_i is the lifetime of node, $i = 1, 2 \ldots\ldots n$).

Node s_i can be any node in the network. Its neighboring node set is $N_i = \{s_j, s_k \ldots\ldots s_m\}$. Select one neighboring node s$_j$. At this moment, the communication distance of two nodes is within the effective sensing range of s_i. So the energy consumption between nodes is:

$$Q = \sum_{j \in N_i} \tau_{ij}\eta_{ij} + \sum_{j \in N_i} \nu\eta_{ji} \tag{10}$$

where η_{ij} is the information transmitted by s_i to s_j. τ_{ij} and ν_{ji} are energy loss factor when node receiving and transmitting information. Under normal condition, the lifetime of s_i is:

$$t_i = \frac{\varpi_i}{Q_i} \tag{11}$$

where ϖ_i is the residual capacity of s_i. According to expression (10) and (11), the network lifetime T_{life} is:

$$T_{life} = Max\{t_1 \ldots t_i \ldots t_n\}$$

5 Simulation experiment analysis

Through simulation experiment, this paper makes a comparative analysis on algorithm performance. The setting is described as follows: place 120 sensor nodes in the monitored area 100 × 100m^2. The sensing radius is 5-20m and the initial electric quantity is 200J. The ratio of energy consumption of working, leisure and dormant state is 24:5:0.02. EACS algorithm is compared with the distributed random algorithm and PAYY algorithm (Proposed Approach of yourim y) proposed by literature [5] to assess the performance of EACS algorithm. Network coverage, network overhead and network lifetime are important parameters in the comparison.

Table 1: Experiment parameter

Parameter	Value
M	100m*100m
N	120
To	120s
W	0.2J
Node Sampling Frequency	1Hz
R	10m
Initial Energy	200J
Minimum Energy Limit	0.02J
Ro	5m
Rm	20m

In the initial state of network operation, the performance parameters of all nodes in the monitored area are set the same, as shown in Fig.5(a). After working for some time, there produce difference on nodes. Adjust sensing radius of nodes according to their energy consumption. Nodes adjust themselves according to residual capacity and relationship with neighboring nodes, as shown in Fig.5(b). In the whole operation, due to differences between nodes and the change of sensing range, there may be redundant areas or coverage holes. Hole detection is conducted followed by hole repair and redundancy elimination in other to enhance the coverage, as shown in Fig.5(c).

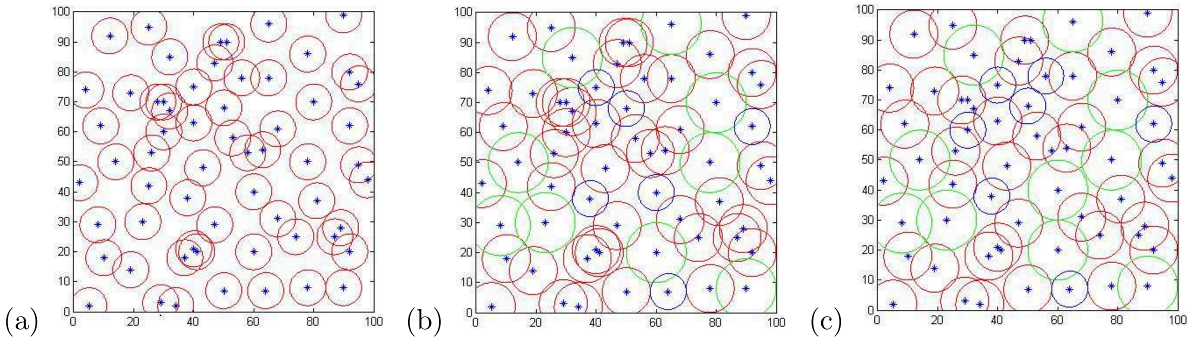


Figure 5: Network coverage

Network coverage ratio refers to the ratio of the effective coverage area by sensor nodes against the overall monitored area. For Random algorithm, operate it for 50 times and get the average coverage. Fig.1 shows the network coverage under different network scales of different algorithm. From Fig.6, it is known that with the increasing of the number of sensor nodes, the network coverage also increases. Under the same number of working nodes, the coverage of Random algorithm is relatively low. When 100 nodes are started, EACS algorithm can reach 90% coverage, 15% higher than Random algorithm. EACS algorithm is better that the other two algorithms in terms of coverage, because such algorithm can adjust the sensing radius according to residual capacity and reduces redundant monitored area. When sensor nodes increase in number, the coverage would increase rapidly.

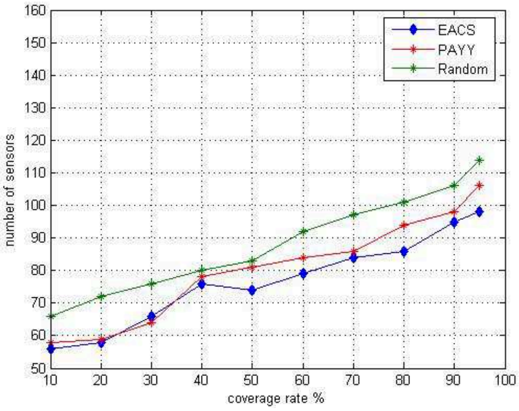


Figure 6: Working nodes and network coverage

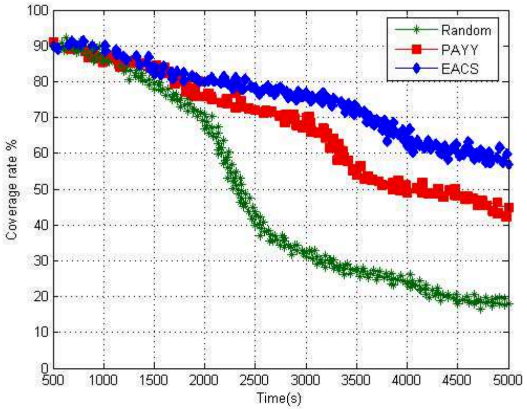


Figure 7: Coverage changes over time

Fig.7 shows network coverage of three algorithms changing over time. In the initial operation of network, the network coverage changes relatively slowly with little divergence. As time goes by, EACS algorithm witnesses the increase of network coverage, though the energy-aware and hole detection require some energy consumption. From Fig.7, it can be told that the coverage curve of EACS algorithm drops steadily, reflecting that the network energy consumption is distributed evenly.

Fig.8 compares nodes location of PAYY and EACS algorithm under the same initial settings and after the network operates for some time. Working nodes of EACS algorithm is more evenly distributed than those of PAYY algorithm. This is mainly because EACS algorithm adopts energy-aware model. The sensing radius of nodes is adjusted according to real situation of the network, so that the energy can be evenly distributed and the network lifetime can be extended.

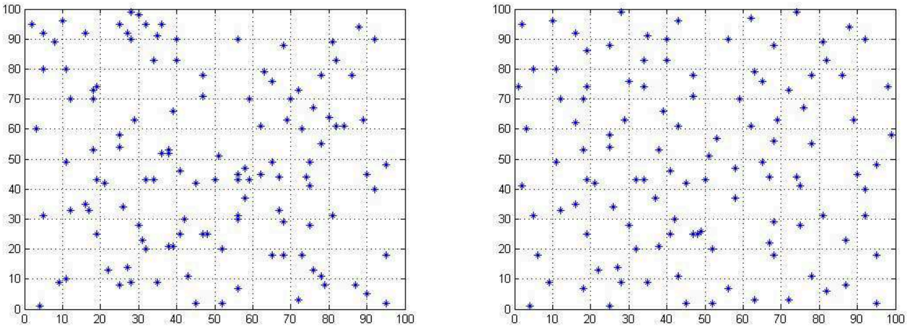


Figure 8: Working nodes distribution

Conclusion

This paper proposes a coverage optimization strategy for WSN based on energy-aware, namely EACS algorithm. It can reach energy balance of nodes in the wireless sensor network. Re-assign tasks of awareness according to working nodes residual capacity, and adjust the sensing radius of nodes according to probability awareness in order to repair the network holes and redundant areas, improve network coverage and reduce redundant coverage and overall network consumption. Results show that the method proposed in this paper takes node residual capacity as an important factor of adjusting sensing field to reduce the burden of nodes with little energy and save them from being "null" too early. Otherwise, it may affect the connectivity and the network lifetime. At the same time, when the coverage holes are under repair, adjust the redundant coverage so that the balance of coverage reaches a reasonable range and the network coverage satisfies actual need. As a result, the unnecessary network energy consumption can be reduced effectively, a balance between the network coverage and energy consumption can be reached and the network lifetime can be improved.

Acknowledgments

The work presented in this paper was supported by the National Natural Science Foundation of China (Grants No. NSFC-61471067) and the National Natural Science Foundation of China (Grants No. NSFC-61271257). Fund for the Doctoral Program of Higher Education of China (Grants No.20120005110002).The work presented in this paper was supported by National Great Science Specific Project(Grants No. 2012ZX03005008)

Bibliography

[1] Mohamadi H., Ismail A.S., Salleh S. (2014); Solving target coverage problem using cover sets in wireless sensor networks based on leraning automata. *Wireless Personal Communications*, 75(1): 447-463

[2] He S.B. et al. (2013); Barrier coverage in wireless sensor networks: From lined-based to curve-based deployment, *Marco A, ed. Proc. of the 32nd IEEE Int'l Conf. on Computer Communications* , 10(9): 470-474

[3] Chen J.M., Li J.K., Lai T.H. (2013): Energy-Efficient intrusion detection with a barrier of probabilistic sensors: Global and local; *IEEE Trans. on Wireless Communications*, 12(9): 4742-4755

[4] Wu K., Gao Y., Li F., Xiao Y. (2005); Light weight deployment aware scheduling for wireless sensor network, *ACM/Kluwer Mobile networks and Applications(MONET)*, 10(6):837-852.

[5] Bulusu N., Heidemann J., Estrin D., Tran T. (2004); Self configuring Localization Systems: Design and Expermiental Evaluation, *ACM Transac-tions on Embedded Computing Systems*, 3(1): 24-60.

[6] Khedr A.M., Osamy W. (2011); Minimum perimeter coverage ofquery regions in a heterogeneous wireless sensor network, *Information Sciences*, 181(15):3130-3142.

[7] La GuilingWang, Guohong Cao T.P.(2006); Movement-assisted SensorDeployment, *IEEE Transactions onMobile Computing*, 5(6): 640-652.

[8] Y. Yoon (2013); An efficient genetic algorithm for maximum coverage deployment in wireless sensor network, *IEEE Transactions on Cybernerics*, 45(5):1473-1483.

[9] Giuseppe Anastasi et al(2009); Energy Conservation in Wireless Sensor Networks: A Survey, *Ad Hoc Networks*, 7(3) : 537-568.

[10] Martins F.V.C. et al. (2011). A hybrid multiobjective evolutionary approach for improvingthe performance of wireless sensor networks, *IEEE Sensors Journal*, 11(3): 545-554.

[11] Wang D, Xie B, Agrawal DP (2008); Coverage and lifetime optimization of wireless sensor networks with Gaussian distribution, *IEEE Trans. on Mobile Computing*, 7(12): 1444-1458

[12] Amato G., Chessa S., Gennaro C., Vairo, C. (2011); Efficient detection of composite events in wireless sensor networks: Design and evaluation, *Proc. of the IEEE Symp. on Computers and Communications*, 10(11):821-823.

[13] Yourim Y., Yong H.K. (2013); An efficient genetic algorithm for maximum coverage deployment in wireless sensor network, *IEEE Transactions on Cybernetics*, 45(5): 1473-1483.

[14] Hossain A., Chakrabarti S., Biswas P.K. (2012); Impact of sensing model on wireless network coverage, *IET Wireless Sensor Systems*, 2(3): 272-281.

[15] Mini S., Udgata S.K., Sabat S.L. (2014); Sensor deployment and scheduling for target coverage problem in wireless sensor networks, *IEEE Sensors Journal*, 14(3): 636-644.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(6):889-901, December 2016.

Learning Bayesian Networks in the Space of Structures by a Hybrid Optimization Algorithm

M. Zhu, S. Liu, J. Jiang

Mingmin Zhu*, Sanyang Liu
School of Mathematics and Statistics, Xidian University
2 South Taibai Road, Xi'an, China, 710071
*Corresponding author: zmmzhu2010@126.com
liusanyang@126.com

Jiewei Jiang
School of Computer Science and Technology, Xidian University
2 South Taibai Road, Xi'an, China, 710071
jiangjw924@126.com

Abstract: Bayesian networks (BNs) are one of the most widely used class for machine learning and decision making tasks especially in uncertain domains. However, learning BN structure from data is a typical NP-hard problem. In this paper, we present a novel hybrid algorithm for BN structure learning, called MMABC. It's based on a recently introduced meta-heuristic, which has been successfully applied to solve a variety of optimization problems: Artificial Bee Colony (ABC). MMABC algorithm consists of three phases: (i) obtain an initial undirected graph by the subroutine MMPC. (ii) Generate the initial population of solutions based on the undirected graph and (iii) perform the ABC algorithm to orient the edges. We describe all the elements necessary to tackle our learning problem, and experimentally compare the performance of our algorithm with two state-of-the-art algorithms reported in the literature. Computational results demonstrate that our algorithm achieves better performance than other two related algorithms.

Keywords: Bayesian network, Artificial Bee Colony, Structural learning, Meta-heuristics, Scoring function.

1 Introduction

Bayesian networks (BNs), also known as belief networks, are becoming a popular tool for representing uncertainty in artificial intelligence. They have been applied to a wide range of tasks such as natural spoken dialog systems, vision recognition, expert systems, medical diagnosis, and genetic regulatory network inference [1-6]. With a network at hand, probabilistic inference can be performed to predict the outcome of some variables based on the observations of others [7,8]. There has been a lot of work in the last ten years on the learning of BNs for both graph structure and probability parameters. However, learning the structure is harder and, arguably, more critical. Most of these algorithms can be grouped into two different categories: constraint-based methods [9,10,11] and search-and-score methods [12,13,14]. The first one poses the learning process as a constraint satisfaction problem, and then constructs a network structure by testing the conditional independence relations. The second one poses the learning problem as a structure optimization problem. Namely, it uses a score metric to evaluate every candidate network structure, and then, finds a network structure with the best score. Though the implement of the former approach is relatively simple, the computation for high-order conditional independence tests is complex and irresponsible. Thus, the score-and-search approach gradually becomes a popular approach for learning BNs [15,16,17].

Recently, a hybrid algorithm that combines the conditional independence methodology with scoring metric optimization, max-min hill climbing (MMHC)[18], has been introduced and proven to be very competitive. In MMHC approach, after reconstructing an undirected graph encoding parent-child relationships by using the max-min parents and children (MMPC) algorithm, a greedy Bayesian scoring hill-climbing search is used to orientate the edges.

In this paper, the idea is further explored, by using the artificial bee colony (ABC)[19,20] algorithm to perform the edge orientation. We call the new hybrid algorithm max-min artificial bee colony (MMABC) algorithm. The ABC algorithm can be used to search for the near-optimal graph in the space of directed acyclic graphs (DAGs). We describe all the elements necessary to tackle our learning problem using this meta-heuristic, and experimentally compare the performance of our ABC-based algorithm with other algorithms used in the literature. The remainder of this paper is organized as follows. Section 2 describes the concepts and methods related to BNs. We introduce the ABC algorithm and our main algorithm in Section 3. Simulation studies are conducted to demonstrate the performance of our algorithm and existing algorithms in Section 4. Finally, in Section 5, we conclude and outline our future work.

2 Preliminaries

In this section we briefly review some basic concepts related to BNs and how to learn them.

A BN is a graphical representation of a joint probability distribution that includes two components. One is a directed acyclic graph (DAG) $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$, the set of nodes, represents the random variables, and E is the set of edges, represents direct dependency relationships between variables. The other is a conditional probability table that quantifies the effects $(p(v_i|pa(v_i)))$ that the parent set of v_i have on the variable v_i in G, where $pa(v_i)$ is the parent set of v_i in G, i.e., $pa(v_i) = \{v_j|v_j \to v_i \in E, v_j \in V\backslash v_i\}$. If v_j is the parent of v_i, then v_i is the child of v_j in G. Learning the structure of BN is equivalent to identifying the structure of the DAG G that best matches the data set D. Robinson [21] showed that the number $f(n)$ of possible structures for BN having n nodes is given by the recurrence formula:

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \frac{n!}{i!(n-i)!} 2^{i(n-1)} f(n-i). \tag{1}$$

Searching for the best structure is difficult because the search space increase exponentially with the number of variables. It has been proven to be an NP-hard problem [22].

Recently, a new and quite competitive method, the MMHC algorithm [18], that combines both scoring-based and conditional independence test approaches has been introduced. It first learns the undirected graph (i.e., the edges without their orientation) of a BN using a local discovery algorithm called the max-min parents and children (MMPC) algorithm. Then, it orients the undirected graph using a greedy hill climbing search. The subroutine MMPC is sound in the sample limit. It attempts to quickly identify the set of parents and children $pc(v_i)$ of a variable v_i in two phases. In the first phase, variables that are conditionally independent on v_i can enter the set of candidate parents and children according to a heuristic function. In each iteration, variable v_j enters the candidate set that maximizes the minimum association to v_i given the current candidate set. In the second phase, the false-positive variables that possibly entered the candidate set in the first phase are removed. More detailed explanation can be found in [18].

In this paper, we first use the MMPC algorithm to learn the undirected graph of a BN, and then orientate the edges by applying the artificial bee colony (ABC) algorithm. In the following, we briefly introduce the ABC algorithm, which is a new meta-heuristic approach inspired by the intelligent foraging behavior of honeybee swarm. After that, we give the main algorithm of this paper.

3 Learning Bayesian networks using the ABC algorithm

3.1 The Artificial Bee Colony algorithm

The ABC [20] algorithm is a swarm based meta-heuristic that simulates foraging behavior of honey bees. This algorithm includes three kinds of bees considering the division of labor: employed bees, onlooker bees and scout bees. A bee that is currently exploiting a food source is called an employed bee. A bee waiting in the hive for making decision to choose a food source is named as an onlooker. A bee carrying out a random search for a new food source is called a scout. An employed bee keeps a food source in her mind when she leaves from the hive and she shares the information about her food source with onlookers on dance area. Onlookers select a food source by watching the dances of the employed bees and try to improve this source. If a food source is abandoned, its employed bee becomes a scout to explore new food sources randomly. Scouts either randomly search the environment in order to find a new food source depending on an internal motivation or based on possible external clues.

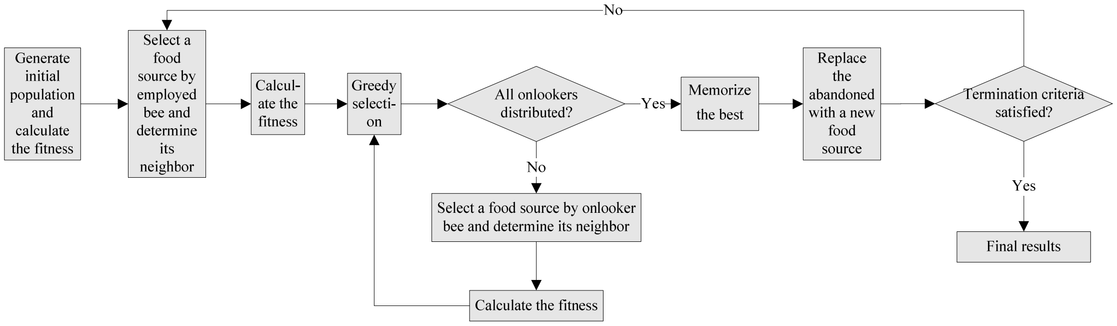

Generate initial population and calculate the fitness
Select a food source by employed bee and determine its neighbor
Calcul-ate the fitness
Greedy selecti-on
All onlookers distributed?
Yes
Memorize the best
Replace the abandoned with a new food source
Termination criteria satisfied?
No
No
Select a food source by onlooker bee and determine its neighbor
Calculate the fitness
Yes
Final results

Figure 1: Flowchart of the ABC algorithm

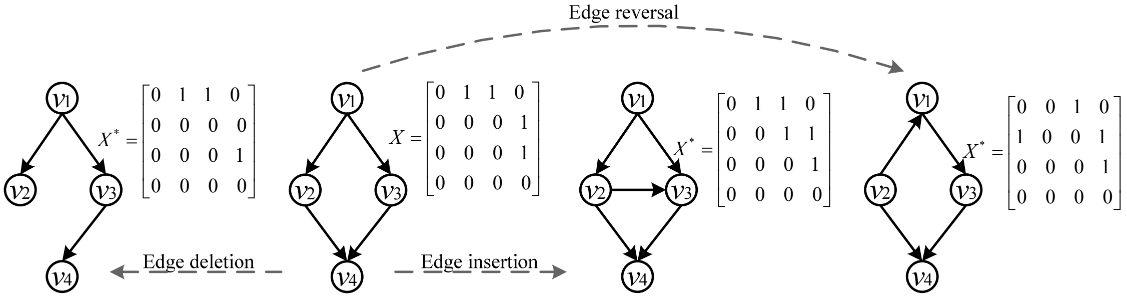


Figure 2: Three edge operators used in the MMABC algorithm

In ABC algorithm each food source represents a possible solution for the problem and the fitness value of the solution corresponds to the nectar amount of this food source. Each food source is exploited by only one employed bee. In other words, the number of employed bees is equal to the number of candidate solutions. Similar to the other swarm intelligence based approaches, the ABC algorithm is also an iterative process. It repeats a series of steps until a termination criterion is met. We show all these steps and interactions between them as a flowchart on Figure 1.

3.2 A hybrid algorithm for learning BNs using ABC optimization

The above basic ABC algorithm, originally designed for the continuous nature of optimization problems, can not directly be used for discrete problems. In this section, we develop a new ABC-based algorithm for learning BNs that performs the search in the space of directed acyclic graphs. The new algorithm is called MMABC for short. We first use the MMPC algorithm to learn the undirected graph of a BN. Based on this undirected graph UG, the initial population for MMABC is produced randomly. More specifically, the edges of the initial graph corresponding to each individual are randomly selected from the edge set of UG, which is the candidate edge set of the optimal BN structure. This greatly reduces the randomness of the initial population and improve the convergence speed. The main components of our MMABC algorithm are described below.

a. Initialization.

Let m and n denote the total number of food sources and the number of variables in a BN, respectively. Since the structure of a BN is a directed graph $G = (V, E)$, we represent each individual by a binary node-node adjacency matrix $X = (x_{ij})_{n\times n}, i, j = 1, 2, \ldots, n$. Entry (i, j) is 1 if there is a directed arc from node i to node j, and 0 otherwise. That is,

$$x_{ij} = \begin{cases} 1, & \text{if } V_i \to V_j \in E, \\ 0, & \text{if } V_i \to V_j \notin E. \end{cases} \tag{2}$$

Initial food sources (solutions) $X_k = (x_{ij})_{n\times n}, k = 1, 2, \ldots, m$ are generated randomly by selecting edges from an undirected graph $UG = (V, \overline{E})$, which is obtained by MMPC. Counters which store the numbers of trials of solutions are reset to 0 in this phase.

b. Employed Bees Phase.

Each employed bee is associated with only one food source site. An employed bee generates a neighboring solution by three operators: edge insertion, edge deletion and edge reversal. We show these three operators in Figure 2. Let X_k and $X_k{}^*$ denote the current solution and the generated one, respectively. The quality of the graph G corresponding to $X_k{}^*$ is measured by the K2 score function [23] as follows:

$$f_{K2}(G, D) = \sum_{i=1}^{n} f_{K2}(v_i, pa(v_i)|D_{v_i,pa(v_i)}), \tag{3}$$

$$f_{K2}(v_i, pa(v_i)|D_{v_i,pa(v_i)}) = \sum_{j=1}^{q_i} \left[\log\left(\frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \right) + \sum_{k=1}^{r_i} \log(N_{ijk}!) \right], \tag{4}$$

where D is the domain data set. $D_{v_i,pa(v_i)}$ are the statistics of the variable v_i and $pa(v_i)$ in D, i.e., the number of instances in D that match each possible instantiation of v_i and $pa(v_i)$. N_{ijk} is the number of cases in D where v_i is in its kth state and its parents are in their jth state. $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$ denotes the number of cases in D where $pa(v_i)$ is in its jth state and r_i denotes the number of states of variable v_i, q_i is the number of parent configurations of v_i. Because the K2 metric is decomposable and the graphs corresponding to $X_k{}^*$ and X_k are different from one edge by each one of three operators, we can reuse most of the computations made in the food source X_k. In other words, we only need to update the statistics corresponding to the variables whose parent sets have been modified. Thus, the time performance can be greatly enhanced.

As our problem is a maximization problem and the value of $f_{K2}(\cdot)$ is negative, therefore, the fitness value assigned to the solution $X_k{}^*$ is determined using the following expression:

$$fit_k = \frac{1}{1 + abs(f_k)}, \tag{5}$$

where f_k is the K2 value of the graph G corresponding to $X_k{}^*$. A greedy selection is employed between X_k and $X_k{}^*$ such that the better one is selected depending on fitness values representing the nectar amount of the food sources at X_k and $X_k{}^*$. If X_k cannot be improved, its counter is incremented by 1, otherwise, the counter is reset to 0.

c. Onlooker Bees Phase.

An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source site with a probability related to its nectar amount. The roulette wheel selection scheme is used here in which each slice is proportional in size to the fitness value as follows:

$$p_k = \frac{fit_k}{\sum_{j=1}^{m} fit_j}. \tag{6}$$

By doing so, the onlooker bees are encouraged to visit the food sources that have higher nectar amounts. Briefly, food source selection by onlookers is based on the information provided by employed bees. A uniform random number within the range $[0, 1]$ is obtained for each source. If the probability value p_k is greater than this random number, then the onlooker bee generates a neighboring food source again by using three operators as in the case of the employed bee. After the source is evaluated, greedy selection is applied. If solution X_k cannot be improved, its counter is incremented by 1, otherwise, the counter is reset to 0. This process is repeated until all onlookers are distributed onto food source sites.

d. Scout Bees Phase.

If the value of the counter is greater than the control parameter "limit", then the food source associated with this counter is abandoned. Assume that the abandoned source is X_k, then the scout randomly discovers a new food source $X_k{}^*$ which can be achieved by randomly generating any directed acyclic graph. This procedure can increase solutions diversity and avoid falling into the local optimum. In our algorithm, there is only a single food source which can be abandoned in each cycle, and only one employed bee can be a scout.

e. Termination.

If a termination is not satisfied, go to Employed Bees Phase; otherwise stop the procedure and output the best food source found so far.

As shown in Algorithm 1, the MMABC algorithm first constructs the undirected graph UG of the underlying BN structure by the MMPC algorithm, which obtained the parents and children of each variable. It gives the best-so-far undirected graph, from which the initial population can be generated randomly by selecting $n - 1$ edges from UG (assumed the underlying BN has n variables). Figure 3 shows the process of generating initial population by a simple example. Consider the second graph in Figure 2. Assumed that its undirected graph UG obtained by MMPC is shown in Figure 3 (a). By randomly selecting 3 directed edges from UG, we can get an initial solution. Figure 3(b), Figure 3(c) and Figure 3(d) show 3 solutions of the initial population generated from Figure 3(a).

After producing and evaluating the initial population, the main loop of the MMABC algorithm then begins for $Iter$ iterations. Step 7 and 15 show one of the edge operators can be chosen to obtain the neighbor of the current solution in Employed Bees Phase and Onlooker Bees Phase, respectively. Each of the operators can be chosen randomly with equal probability. If there is a cycle in the generated graph, another new graph can be achieved again.

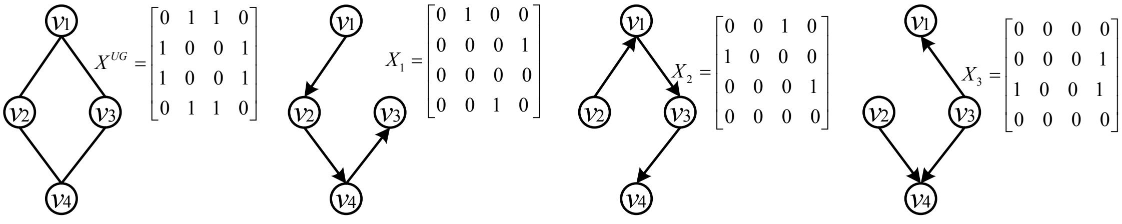
4 Experimental results

In this section, we present the experimental results carried out with our algorithm and compare MMABC with two hybrid heuristic methods: MMHC [18] and MMACO [24]. In all the

Algorithm 1 The MMABC algorithm.

```
Obtain the undirected graph $UG$ by MMPC;
Generate the initial population of solutions $X_k, k = 1,2,\dots,m$, $tr_k = 0$. $m$ is the total number of food sources. $tr_k$ is the non-improvement number of the solution $X_k$, used for abandonment;
Evaluate the population;
Iter=1;
repeat
    for $k = 1$ to $m$ do
        Produce a new food source $X_k{}^*$ for the employed bee of the food source $X_k$ by using one of the three edge operators and evaluate its quality according to equation (3);
        Apply a greedy selection process between $X_k{}^*$ and $X_k$ and select the better one;
        If solution $X_k$ does not improve $tr_k = tr_k + 1$, otherwise $tr_k = 0$;
    end for
    Calculate the probability value $p_k$ by (6), $k = 1,2,\dots,m$;
    $t = 0, k = 1$;
    repeat
        if $random < p_k$ then
            Produce a new food source $X_k{}^*$ for the onlooker bee of the food source $X_k$ by using one of the three edge operators and evaluate its quality by (3);
            Apply a greedy selection process between $X_k{}^*$ and $X_k$ and select the better one;
            If solution $X_k$ does not improve $tr_k = tr_k + 1$, otherwise $tr_k = 0$;
            $t = t + 1$;
        end if
    until $t = m$;
    if $max(tr_k) > limit$ then
        Replace $X_k$ with a new randomly produced solution;
    end if
    Memorize the best solution achieved so far;
    Iter=Iter+1;
until Iter=Maximum Iteration.
```

Figure 3: The MMABC algorithm


v1
v2
v3
v4
$X^{UG} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$
$X_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
$X_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
$X_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

(a) *UG* (b) Solution 1 (c) Solution 2 (d) Solution 3

Figure 4: The process of generating initial population

cases the scoring metric used to guide the search is the K2 metric, so, the difference among implementations is minimized. All these experiments are based on the BN toolbox written by Murphy [25] and the Causal Explorer System developed by Aliferis et al.[26]. The experimental platform is a personal computer with Pentium 4, 3.06 GHz CPU, 1.0 GB memory, and Windows XP.

The standard way of assessing the accuracy of a learning method is to draw samples from a known BN, apply structure learning on the artificial data, and to compare the learned structure with the original one. To benchmark our method, we investigate its performance on artificial data with real-world characteristics focusing on the well-known Asia [27], Insurance [28], and Alarm [29] networks. The first one is a small BN using a fictitious medical example that considers the dependence of the health status of a patient (having tuberculosis, lung cancer, or bronchitis) on risk factors (visit to Asia and smoking) and clinical observations (Dyspnea and X-ray). It consists of eight nodes with eight edges connecting them. Each random variable is discrete in nature and can take two states. The second example is a network for estimating the expected claim costs for a car insurance policyholder. It consists of 27 nodes and 52 edges. The last one is a medical diagnostic system for patient monitoring. It consists of 37 nodes and 46 edges connecting them. The random variables in the ALARM network are discrete in nature and can take two, three, or four states. The characteristics of the networks are summarized in Table 1. For these three networks, data sets with 200, 500, 1000, 3000, and 5000 samples are drawn. For each of the sample sizes, ten independent data sets are generated by applying Gibbs sampling [30] on the benchmark networks, resulting in a set of 50 data sets for each network. 10 replicates are done for each of different network parameters and sample size. We will discuss our experimental results in more detail in the following two subsections.

Table 1: Bayesian networks

Network	Variables	Edges	Max In/Out-degree	Domain range
Asia	8	8	2/2	2
Insurance	27	52	3/7	2-5
Alarm	37	46	4/5	2-4

Table 2: The K2 values for the true networks with different samples

Network	$n = 200$	$n = 500$	$n = 1000$	$n = 3000$	$n = 5000$
Asia	-501.79	-1194.9	-2350.1	-6881.9	-11458
Insurance	-5291.2	-9851.6	-17105	-45406	-73717
Alarm	-3259.2	-6364.4	-11162	-29894	-48597

4.1 Comparison of the solution quality

In this subsection, we compare the performance of the different algorithms on all of the data sets. Our main objective in these experiments is to determine whether MMABC is more efficient and effective than the other two hybrid algorithms. For MMABC and MMACO, the population size is set to 20, and the maximum number of iterations is set to 100. The significant level used in these algorithms is set to 0.05. Other parameter settings of the MMACO algorithm can be found in [24]. The parameter "limit" for MMABC is set to 6, namely, if the solution associated with an employed bee does not improve for 6 times, then that solution is replaced by some randomly generated solution. In order to provide an objective measure of network structure reconstruction behavior, we compare the underlying DAG with the learned DAG and record the structural hamming distance (SHD), where SHD is defined as the total number of operations to verify the learned DAG to the underlying DAG. The K2 scoring function was also presented to measure the learned network structure, which could be used to compare the quality of the structures obtained by different algorithms. The K2 values for the true network structures with different data sets can be seen in Table 2.

Table 3: The results for three algorithms with different datasets

Network	Algorithm	$n = 200$	$n = 500$	$n = 1000$	$n = 3000$	$n = 5000$
Asia	MMABC	3.0	2.1	1.0	0.7	0.3
		-500.53±0.7	-1191.6±1.1	-2351.6±3.1	-6877.0±3.5	-11454.0±5.2
	MMACO	4.8	2.9	1.2	0.6	0.3
		-488.8±1.3	-1179.3±3.2	-2346.5±7.1	-6868.0±9.1	-11478.0±11.9
	MMHC	6.1	3.3	2.0	1.1	0.9
		-507.6±1.1	-1209.2±1.8	-2351.9±3.7	-6896.3±5.7	-11593.0±6.8
Insurance	MMABC	15.8	11.6	8.7	6.6	4.1
		-5292.1±1.2	-9853.8±1.8	-17116.1±2.6	-45408.3±4.7	-73716.9±5.6
	MMACO	16.9	12.4	9.4	6.7	5.3
		-5295.8±3.1	-9857.6±5.6	-17115.3±6.9	-45412.1±8.2	-73716.7±10.1
	MMHC	36.2	27.9	19.3	12.4	8.9
		-5361.3±1.5	-9911.5±3.2	-17190.4±3.4	-45815.6±5.3	-73850.3±6.8
Alarm	MMABC	18.4	11.3	10.0	6.9	3.9
		-3198.3±1.2	-6363.5±1.7	-11149.9±2.8	-29883.9±4.1	-48600.9±5.0
	MMACO	19.1	12.1	11.2	7.2	4.4
		-3072.2±3.1	-6344.3±5.5	-11085.7±6.9	-29891.1±9.2	-48605.9±11.1
	MMHC	35.8	27.6	19.2	13.8	9.6
		-3085.8±1.4	-6435.1±3.1	-11210.2±3.3	-29849.3±5.3	-48694.0±6.7

Table 3 provides the mean and standard deviation of the K2 score and the mean SHD values obtained for the learned networks with different data sets. $\mu \pm \sigma$ indicates the mean and the standard deviation over the executions carried out. We have carried out 10 executions of each algorithm and for each domain considered. In Table 3, the first six rows display the results of three algorithms on Asia data sets, and the rest twelve rows is the results on Insurance and Alarm data sets. For example, figures in the second row are the SHD values obtained by the MMABC algorithm on Asia data sets. Figures in the third row are their corresponding mean and standard deviation of the K2 scores. On the whole, we can see that MMACO algorithm is a little better than MMHC algorithm on the Asia network, and our algorithm is best. However, in the case of Insurance and Alarm networks, the advantages of MMABC and MMACO are more evident, and MMABC performs better than MMACO, which means our algorithm has better performance with more complicated networks. The K2 score of our algorithm is always closer to that of the original network. The reason is that the construction process of a BN for an ant is dependent much too on the pheromone and heuristic information of candidate arcs, which is prone to overfitting. Although it can be seen that the K2 score of MMABC and MMACO becomes similar as the sample size increase, MMABC has a lower SHD value. We also note that in some cases, the score obtained by MMACO is better than that of MMABC, but MMABC has lower SHD values. This indicates that a better score does not necessarily mean a better SHD value and vice-versa. It can occur because of small size and because of the parameters given to the scoring function, which have been shown to produce differences in scoring function behavior. It is worth mentioning that the standard deviation of the MMABC is lower than that of the other two algorithms, which means that the MMABC guarantees good quality networks with less fluctuation than other algorithms.

4.2 Comparison of the convergence and time performance

In order to measure the convergence speed and the runtime complexity to learn the structure, we compared with the typical runs of the MMABC and MMACO algorithms on the same data set by large number of experiments.

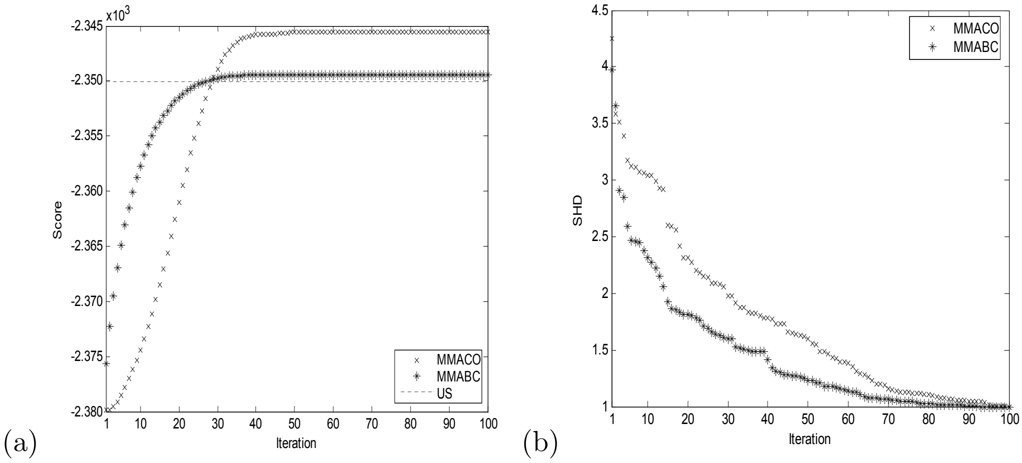


Figure 5: Comparison of the solution quality for MMACO and MMABC on Asia data sets: (a) Scores for Asia; (b) SHD for Asia

Figure 4 and Figure 5 show the curves of the convergence performance on two data sets, Asia-1000 and Alarm-5000, respectively. In these figures, the abscissas depict the number of iterations, and the Y-coordinates depict the K2 score or SHD value of the learned graph. *US* specifies the underlying score of the original network. From a general point of view, MMABC achieve better quality networks in the initial stages of the optimization and has a rapidly converging learning curve, which means that MMABC require a significantly lower number of iterations than MMACO to achieve networks of better quality. Although we are testing on two very different data sets, we obtain similar observation that as the score is improving over iterations, the SHD value is deteriorating. It provides strong evidence that MMABC is performing well against the MMACO algorithm.

In Figure 6, we test MMABC with 4 different population sizes on Asia-1000: 10 (MMABC1), 20 (MMABC2), 30 (MM ABC3), 40 (MMABC4). From the curves in Figure 6(a) and Figure 6(b), it can be found that the population size m can have an effect on the convergence speed of the algorithm. In other words, MMABC takes a comparatively large number of iterations to converge to an acceptable solution with a small number of bees. Because a large population

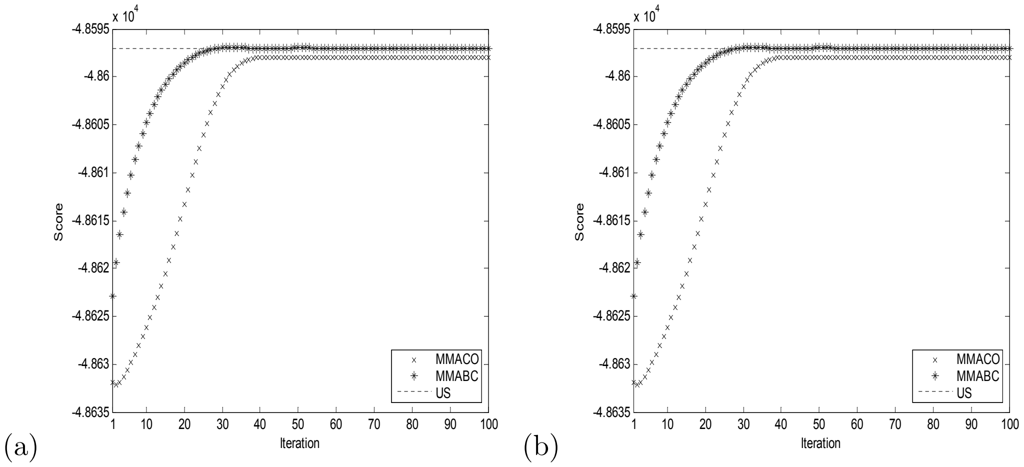


Figure 6: Comparison of the solution quality for MMACO and MMABC on Alarm data sets: (a) Scores for Alarm; (b) SHD for Alarm

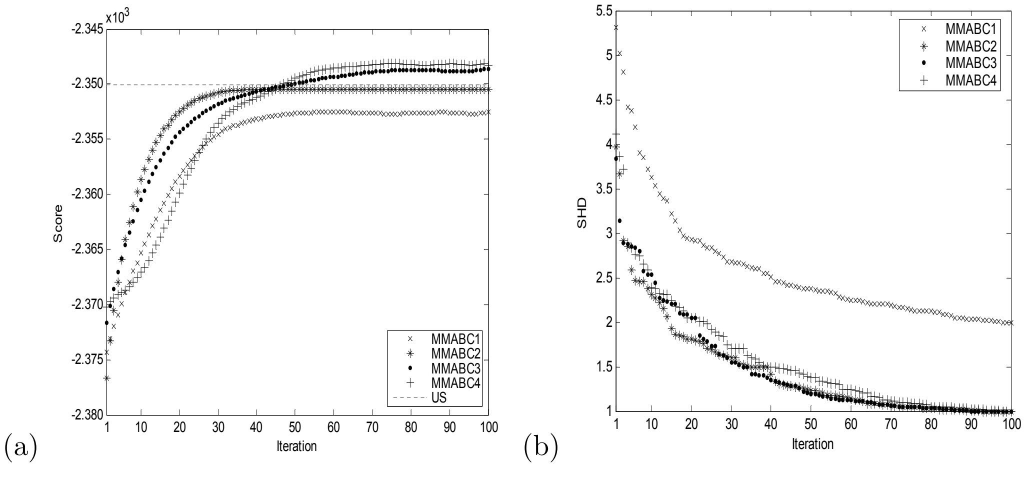


Figure 7: Comparison of the solution quality for different population sizes on Asia data sets: (a) Scores for Asia; (b) SHD for Asia

means that more food sources are employed, there are more different solutions examined than in a smaller population. On the other hand, the time consumed also increased proportionally with increasing population size. It should be noted that tiny changes to the score value can lead to large structural changes as an algorithm converges towards the optimum network.

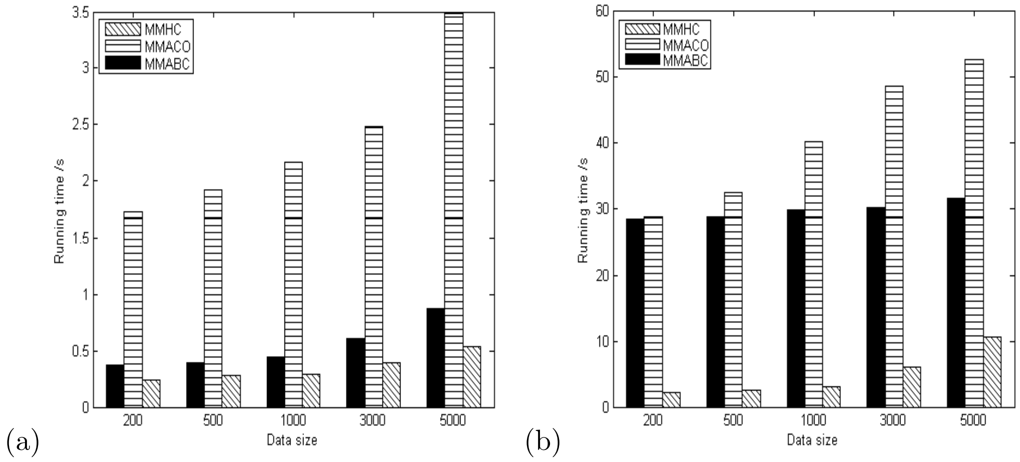


Figure 8: Comparison of the time performance on two networks: (a) Running time for Asia data sets; (b) Running time for Alarm data sets

Figure 7 gives the average running time to reach the best solution for three algorithms on two networks. Based on Figure 7, we can see that the proposed MMABC has a considerable better time performance than MMACO, and MMHC has the best time performance. However, from the results of Table 3, we can see that MMHC obtains worse K2 score and SHD values. Moreover, the advantage is very obvious when the data set is large on the same network, namely, the bigger the sample size, the more obvious the improvement. We also noted that the running time of MMABC is affected less by the size of data but more by the problem dimension. It means that MMABC is not sensitive to the increase of the sample capacity and able to handle very large data sets.

Conclusions

In this paper we have designed and implemented a hybrid ABC-based algorithm for the BN structure learning problem. Ideas from constraint based local learning and score-based ABC were combined into effective learning processes. Its application to several benchmark networks yielded better results than the previously used MMHC and MMACO algorithms.

This paper demonstrates the capability of ABC algorithm in learning a probability graphical model. Ideas presented in this paper can be applied to many other probability graphical problems also, such as dynamic BN, chain graph, etc. Moreover, we have applied the basic three edge operators for determining the neighboring solutions in MMABC. It will be interesting to see whether there exists some alternative approach to serve the same purpose here. We also intend to expand this analysis to larger benchmark networks and make further conclusions on the usefulness of the constraint-based and score-based learning approaches. Finally, although we assume in this paper that the data are completely observed, missing data or data with latent variables may arise in practice. Generalization of the proposed algorithm to incomplete data is of great research interest.

Acknowledgment

The authors would like to thank the editor and the anonymous reviewers for their insightful comments and suggestions. This work has been partially supported by the National Natural Science Foundation of China (Grant No. 61373174, 11401454), the Natural Science Foundation of Shannxi Province, China (Grant No. 2014JQ1031) and the Fundamental Research Funds for the Central Universities (Grant No. JB140711).

Bibliography

[1] Z. Cai, S. Sun, S. Si, B. Yannou, Identifying product failure rate based on a conditional Bayesian network classifier, *Expert Systems with Applications*, 38(5): 5036-5043.

[2] Y. Sun, Y.Y. Tang, S.X. Ding, S.P. Lv, Y.F. Cui, Diagnose the mild cognitive impairment by constructing Bayesian network with missing data,*Expert Systems with Applications*, 38(1): 442-449.

[3] V. Aquaroa, M. Bardoscia, R. Bellotti, A. Consiglio, F.D. Carlo, G. Ferri, A Bayesian Networks approach to Operational Risk, *Physica A*, 389(8): 1721-1728.

[4] D.C. Kim, X. Wang, C.R. Yang, J. Gao, Learning biological network using mutual information and conditional independence, *BMC Bioinformatics*, 11(3): S9.

[5] S. Nikolajewa, R. Pudimat, M. Hiller, M. Platzer, R. Backofen, BioBayesNet: a web server for feature extraction and Bayesian network modeling of biological sequence data, *Nucleic Acids Research*, 35: 688-693.

[6] R. Wei, J. Zhen, L. Bao, Study on mining big users data in the development of HuBei auto-parts enterprise, *Mathematical modelling of engineering problems*, 2(4): 1-6.

[7] K. Burns, Bayesian inference in disputed authorship: a case study of cognitive errors and a new system for decision support, *Information Science*, 176(11): 1570-1589.

[8] S. Lee, Y. Son, J. Jin, Decision fild theory extentions for behavior modelling in dynamic environment using Bayesian belief network, *Information Science*, 178(10) 2297-2314.

[9] J. Cheng, R. Greiner, J. Kelly, D. Bell, W. Liu, Learning Bayesian networks from data: an information theory based approach, *Artificial Intelligence*, 137: 43-90.

[10] J.P. Pellet, A. Elisseef, Using Markov Blankets for Causal Structure learning, *Journal of Machine Learning Research*, 9: 1295-1342.

[11] C. Borgelt, A conditional independence algorithm for learning undirected graphical models, *Journal of Computer and System Sciences*, 76: 21-33.

[12] E. Perrier, S. Imoto, S. Miyano, Finding Optimal Bayesian Network Given a Super-Structure, *Journal of Machine Learning Research*, 9: 2251-2286.

[13] L.M. de Campos, A Scoring Function for Learning Bayesian Networks based on Mutual Information and Conditional Independence Tests, *Journal of Machine Learning Research*, 7: 2149-2187.

[14] L.M. de Campos, J.M. Fernández-Luna, J.A. Gámez, J.M. Puerta, Ant colony opyimization for learning Bayesian networks, *International Journal of Approximate Reasoning*, 31: 291-311.

[15] R. Daly, Q. Shen, Learning Bayesian Network Equivalence Classes with Ant Conoly Optimization, *Journal of Artificial Intelligence Research*, 35: 391-447.

[16] L. Bouchaala, A. Masmoudi, F. Gargouri, A. Rebai,Improving algorithms for structure learning in Bayesian Networks using a new implicit score, *Expert Systems with Applications*, 37: 5470-5475.

[17] J. Ji, H. Wei, C. Liu, An artificial bee colony algorithm for learning Bayesian networks, *Soft Computing*, 17: 983¨C994.

[18] I. Tsamardinos, L.F. Brown, C.F. Aliferis, The max-min hillclimbing BN structure learning algorithm, *Machine Learning*, 65: 31-78.

[19] D. Karaboga, An idea based on honey bee swarm for numerical optimization, *Technical Report-TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department.

[20] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing*, 8: 687-697.

[21] R.W. Robinson, Counting unlabeled acyclic digraphs, *Combinational Mathematics*, 622: 28-43.

[22] D. Chickering, D. Heckerman, C. Meek, Large-Sample Learning of Bayesian Networks is NP-hard, *Journal of Machine Learning Research*, 5: 1287-1330.

[23] G. Cooper, E. Hersovits, A bayesian method for the induction of probabilistic networks from data, *Machine Learning*, 9: 309-347.

[24] P.C. Pinto, A. Nägele, M. Dejori, T.A. Runkler, J.M.C. Sousa, Using a Local Discovery Ant Algorithm for Bayesian Network Structure Learning, *IEEE transactions on evolutionary computation*, 13(4): 767-779.

[25] K. Murphy, The Bayes net toolbox for Matlab, *Computing Science and Statistics*, 33: 331-350.

[26] C.F. Aliferis, I. Tsamardinos, A. statnikov, L.E. Brown, Causal Explorer: A causal probabilistic network learning tooltik for biomedical discovery, *In Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, 371-376.

[27] S.Lauritzen, D.Spiegelhalter, Local Computations with Probabilities on Graphical Structures and Their Application on Expert Systems, *J.Royal Statistical Soc.*, 50: 157-224.

[28] I. Beinlich, G. Suermondt, R. Chavez, G.Cooper, The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks, *Proc. Second European Conf. Artificial Intelligence in Medicine.*

[29] J. Binder, D. Koller, S.J. Ruddell, K. Kanazawa, Adaptive probabilistic networks with hidden variables, *Machine Learning*, 29: 213-244.

[30] T. Hrycej, Gibbs sampling in BNs, *Artificial Intelligence*, 46(3): 351-363.

Author index