

Volume 44 Number 3 September 2020

ISSN 0350-5596

Informatica

**An International Journal of Computing
and Informatics**



1977

Editorial Boards

Informatika is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatika is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatika is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Executive Editor – Editor in Chief

Matjaž Gams
Jamova 39, 1000 Ljubljana, Slovenia
Phone: +386 1 4773 900, Fax: +386 1 251 93 85
matjaz.gams@ijs.si
<http://dis.ijs.si/mezi/matjaz.html>

Editor Emeritus

Anton P. Železnikar
Volaričeva 8, Ljubljana, Slovenia
s51em@lea.hamradio.si
<http://lea.hamradio.si/~s51em/>

Executive Associate Editor - Deputy Managing Editor

Mitja Luštrek, Jožef Stefan Institute
mitja.lustrek@ijs.si

Executive Associate Editor - Technical Editor

Drago Torkar, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Phone: +386 1 4773 900, Fax: +386 1 251 93 85
drago.torkar@ijs.si

Executive Associate Editor - Deputy Technical Editor

Tine Kolenik, Jožef Stefan Institute
tine.kolenik@ijs.si

Editorial Board

Juan Carlos Augusto (Argentina)
Vladimir Batagelj (Slovenia)
Francesco Bergadano (Italy)
Marco Botta (Italy)
Pavel Brazdil (Portugal)
Andrej Brodnik (Slovenia)
Ivan Bruha (Canada)
Wray Buntine (Finland)
Zhihua Cui (China)
Aleksander Denisiuk (Poland)
Hubert L. Dreyfus (USA)
Jozo Dujmović (USA)
Johann Eder (Austria)
George Eleftherakis (Greece)
Ling Feng (China)
Vladimir A. Fomichov (Russia)
Maria Ganzha (Poland)
Sumit Goyal (India)
Marjan Gušev (Macedonia)
N. Jaisankar (India)
Dariusz Jacek Jakóbczak (Poland)
Dimitris Kanellopoulos (Greece)
Samee Ullah Khan (USA)
Hiroaki Kitano (Japan)
Igor Kononenko (Slovenia)
Miroslav Kubat (USA)
Ante Lauc (Croatia)
Jadran Lenarčič (Slovenia)
Shiguo Lian (China)
Suzana Loskovska (Macedonia)
Ramon L. de Mantaras (Spain)
Natividad Martínez Madrid (Germany)
Sando Martinčić-Ipišić (Croatia)
Angelo Montanari (Italy)
Pavol Návrat (Slovakia)
Jerzy R. Nawrocki (Poland)
Nadia Nedjah (Brasil)
Franc Novak (Slovenia)
Marcin Paprzycki (USA/Poland)
Wiesław Pawłowski (Poland)
Ivana Podnar Žarko (Croatia)
Karl H. Pribram (USA)
Luc De Raedt (Belgium)
Shahram Rahimi (USA)
Dejan Raković (Serbia)
Jean Ramaekers (Belgium)
Wilhelm Rossak (Germany)
Ivan Rozman (Slovenia)
Sugata Sanyal (India)
Walter Schempp (Germany)
Johannes Schwinn (Germany)
Zhongzhi Shi (China)
Oliviero Stock (Italy)
Robert Trappl (Austria)
Terry Winograd (USA)
Stefan Wrobel (Germany)
Konrad Wrona (France)
Xindong Wu (USA)
Yudong Zhang (China)
Rushan Ziatdinov (Russia & Turkey)

Reminder of the First Paper on Transfer Learning in Neural Networks, 1976

Stevo Bozinovski
South Carolina State University, Orangeburg, SC, USA
E-mail: sbozinovski@scsu.edu

Overview paper

Keywords: transfer learning, neural networks

Received: June 10, 2019

This paper describes a work on transfer learning in neural networks carried out in 1970s and early 1980s, which produced its first publication in 1976. In the contemporary research on transfer learning there is a belief that pioneering work on transfer learning took place in early 1990s, and this paper updates that knowledge, pointing out that the transfer learning research started more than a decade earlier. This paper reviews the pioneering 1970s research, and addresses important issues relevant for the current transfer learning research. It gives a mathematical model and geometric interpretation of transfer learning, and a measure of transfer learning indicating positive, negative, and no transfer learning. It presents experimental investigation in the mentioned types of transfer learning. And it gives an application of transfer learning in pattern recognition using datasets of images.

Povzetek: Ta članek opisuje delo na področju prenosa učenja v nevronskih omrežjih, opravljeno v sedemdesetih in zgodnjih osemdesetih letih prejšnjega stoletja, ki je prvo publikacijo izdalo leta 1976. V sodobni raziskavi o transfernem učenju obstaja prepričanje, da je pionirsko delo na področju transfernega učenja potekalo v začetku devetdesetih let, in ta članek to znanje posodablja. Poudarja, da so se raziskave o transfernem učenju začele 15 let prej. Ta članek pregleduje raziskave in obravnava pomembna vprašanja za sedanje raziskave o transfernem učenju. Daje matematični model in geometrijsko razlago transfernega učenja. Daje merilo transfernega učenja, vključno s pozitivnim, negativnim in tabula rasa prenosnim učenjem. Predstavlja eksperimentalno raziskovanje omenjenih vrst transfernega učenja. Uporablja prenosno učenje pri prepoznavanju nabora podatkov.

1 Introduction

Transfer learning is a machine learning method where a learning model developed for a first learning task is reused as the starting point for a learning model in a second learning task (Tan et al. 2018). It is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem (Wikipedia > Transfer Learning, October 2020). Often previous learning is referred to as source and the next learning as target (Pratt 1993, Pan and Yang 2010, Weiss et al. 2016). Basically it is using a pre-trained neural network (trained for Task1) for achieving shorter training time (positive transfer learning) in learning Task2. Transfer learning is an emphasized way of learning in contemporary multistage neural networks named deep neural networks (e.g., Goodfellow et al. 2016).

According to (Wikipedia > Transfer Learning, October 2020), the earliest work on transfer in machine learning is attributed to Lorien Pratt (1993). That work points out the earlier work on the subject (Pratt et al. 1991). After 1993, as pointed in Pan and Yang (2010) the fundamental motivation for transfer learning in the field

of machine learning was discussed at a NIPS-95 workshop on “Learning to Learn” (Baxter et al. 1995).

In the context described above, this paper informs on an explicit work on transfer learning which took place fifteen years before the Pratt et al. (1991) work. That research, reviewed here, started 1972 producing some unpublished reports (Bozinovski 1972, 1974) and a published report in 1976 (Bozinovski and Fulgosi, 1976) which explicitly in the title addressed the transfer learning concept. Research continued after that, and reports were given in (Bozinovski et al. 1977, Bozinovski 1978, 1981, 1985a, 1985b, 1995).

That initial research on transfer learning is important to the current effort in transfer learning, because in addition of presenting initial concept of transfer learning in neural networks, it describes an early approach of defining a measure of transfer learning which is of interest to current efforts in transfer learning (Tan et al. 2018). The review presented here, in addition to mathematical treatment of transfer learning, describes the experimental investigation on transfer learning which took place during 1976-1981. This paper also gives an

application of transfer learning, in obtaining shorter training sequences in learning a dataset of images representing letters.

In the sequel the paper first reviews the neural network used in early research on transfer learning, during 1972-1981. Then it gives a mathematical model of supervised learning, in which it explicitly introduces transfer learning. Then it gives a geometrical model of transfer learning, including positive, negative, and no transfer learning. Then, in Section 5, it defines a mathematical index, a measure of transfer learning. In Section 6 the paper discusses a search for a solution of pattern classification problem in case of negative transfer learning. In Section 7 the paper discusses the multi-class multi-template problem of transfer learning. Section 8 shows results of experimental investigation in transfer learning. It first shows experiments with small set of low resolution images representing letters, demonstrating experimentally the effect of tabula rasa, positive, and negative transfer. The paper then extends to an application of transfer learning in case of learning a dataset of three sets each containing 26 images representing letters. The section 9 reviews the related work by other authors which appeared after 1986, influenced by the renewed interest in neural networks due to the book of Rumelhart, McClelland, and the PDP Group (1986), including the work of Pratt et al. (1991) and Pratt (1993). The paper ends with a discussion and conclusion section.

2 The neural network

The neural network used in our study (Bozinovski 1972, 1974, 1995) is shown in Figure 1.

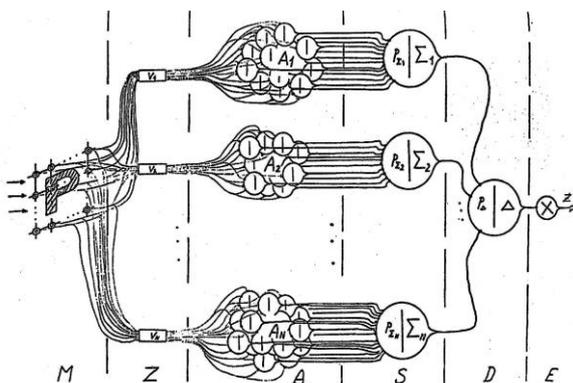


Figure 1: A 5-layer neural network used in supervised learning for pattern recognition in the research described here (Bozinovski 1974, 1995).

The network contains 5 computational stages (layers). The first one, M, is the sensor layer, with sensors arranged according to a need, for example as a matrix retina. Sensors are binary giving values 0 or 1.

Second layer, Z, is a *feature extraction layer*. Feature is a pattern which is used as input in recognition of a higher level pattern. Examples of features might be "horizontal line", "circle", "upper left corner", or a rather complex feature. Important is that the feature is a first

stage in recognizing a pattern, which is a set of features. One way of defining a feature is to pre-wire all sensors in a horizontal line and to create an output from Z layer, with interpretation "horizontal line". The other way is to add a Z-element with trainable weights and produce an output with interpretation "horizontal line". The number of outputs from Z-layer is often larger than the number of input sensors. For example each sensor can be considered a feature, plus some needed features such as "middle horizontal line", "left corner" or "square".

The outputs of the layer Z are inputs to the third layer, the A-layer. It contains A-elements, or associative units, as named originally by Rosenblatt (1958, 1962), and used in early neural learning research (e.g. Glushkov, 1967). We will use that term, but we will also use the term associative weights. A weight represents the *relevance* of the feature in creating the concept of a pattern. They are divided into subsets A_1, A_2, \dots, A_n , each subset having inputs from the feature layer Z. The subsets are associated to a cognitive concept, a *class* to which input patterns are classified. If there are n possible classes and N_s possible features, then each A element can be represented by values $w_{is}, i=1, \dots, n; j=1, \dots, N_s$. They are in general real numbers. Each class of A elements represent a concept, a cognitive class, that will be learned in the pattern classification process. For example, if a task is to classify images, then one set of A elements will be devoted to recognize image "E", another to recognize the image "F" etc.

Next layer, S, are elements that perform some computation over the subsets of A elements representing cognitive classes. An S element s_i computes some function $y(w_{ij})$ over the elements $w_{ij}, i=1, \dots, n; j=1, \dots, N_s$. Most often these elements compute a weighted and thresholded sum $y_i = \sum_j w_{ij} x_j - p_{\Sigma i}$ where $p_{\Sigma i}$ is named threshold of the element s_i . Further in the text we will use the θ -notation for threshold, i.e. $p_{\Sigma i} = \theta_i$. There are n S-elements in this layer, s_1, s_2, \dots, s_n . A subset of A elements and the corresponding S element is named a neuron of the neural network.

The next layer, D, is an *arbiter layer*, which chooses an S-element out of n alternative S-elements. Usual way is computing a maximum function. This layer can be composed by set of neurons which have a common threshold. Such an Isothreshold Neural Network (e.g. Bozinovski 1985a) has a common threshold value equal to maximal value of the individual neuron thresholds. Such a network provides a mechanism for computing maximal value in neural networks. In addition, the maximal value might be normalized to 1, and the maximum computing network can be viewed as computing fuzzy union if the input values are also normalized between 0 and 1. The output of this layer is an integer from 0 to n . For example, output $d = 2$ means that the observed pattern belongs to class 2 out of the considered n classes. The output $d = 0$ means that the classification is undecided, possibly there are two S-elements computing the same largest value, so there is no single maximal value.

The next layer E, is output interface layer. It activates some device that is controlled by this neural

network. For example if $d=2$ is computed, then this layer may activate a speech device telling the sound representation of the class 2.

The neural network presented in Figure 1 was the one we started our research in neural networks with. The first task we considered was distinguishing a horizontal vs vertical line on a matrix retina (Bozinovski 1972). That is not reviewed here. This paper is focused on modeling transfer learning.

3 Mathematical modeling of transfer learning in a neural network

For purpose of presenting the concept of transfer learning, here we use a simplified version of the 5-layer network on Figure 1. Let the layer M consists of m synapses or sensors. Let layer Z does not compute any additional feature besides the sensor inputs, so it just represents connections from sensors to A-elements. Let each subset of A elements has the same connections to the sensors as the other subset of A-elements. The A-elements will be named synaptic weights, such that the weight w_{is} represents the s -th synapse element in the i -th class of A elements. Then the S-element s_i computes the function $y_i = \sum_s w_{is}x_s - \theta_i$. Let the layer D is represented by a maximum selector function: ($d_i = 1$ if $y_i = \max_i\{y_i\}$ otherwise $d_i = 0$). Other way of denoting a maximum selector is $d = \text{indmax}\{y_i\}$ where $\text{indmax}\{ \}$ returns the index of the maximal element in the considered set. In the literature this function is usually written as $d = \text{argmax}\{ \}$, but we use our original notation (Bozinovski and Fulgosi, 1976).

3.1 An approach toward modeling supervised learning in neural networks

The principal learning concept of the neural network approach toward machine learning is the concept of (synaptic) weights (e.g. Rumelhart et al. 1986, Goodfellow et al. 2016). In pattern classification with neural networks the principal representation spaces are the *pattern feature space* and *weights space*. However, it should be noted that while in artificial neural nets synaptic weights are observable, in real biological systems they are not observable. So it is interesting to use a representation of the supervised learning problem which will not deal with synaptic weights as primary representation concept. Here we will describe such a representation which is a *weights-free* and we call it *teaching space* (Bozinovski 1981, 1985b).

Let us note that in a supervised learning there is a system named teacher who has a reference model of the knowledge to be transferred in the other system named learner or student. The teaching space approach is based on the following notation:

Let x be a body of knowledge to be learned by the student. For example x might be a visual pattern to be classified in a class. The supervised learning procedure (training) contains both teaching trials (where the teacher presents the knowledge about x), and examination trials (where the student presents its knowledge about x). After

the training is completed there will be many exploitation trials, where the learner will show its knowledge in an application.

Let $!(x)$ denotes a teaching (or advising) trial, representing the teacher's reference model knowledge about x . Let $?(x)$ denotes a test (or examination) trial, representing the current learner's knowledge about x . Then, the goal of the teaching process becomes

$$?(x) = !(x) \text{ for all considered } x. \quad (1)$$

The learner we use is a maximum selector classifier (Figure 1). For each input pattern x in an test trial, the learner computes n alternatives, i.e., computes n functions $y_1(x), \dots, y_n(x)$, and chooses the one with maximal value. If there is no maximal value the learner gives special answer meaning "undecided", for example value 0.

Lets define a set X of N objects (patterns), $X = \{x_1, \dots, x_i, \dots, x_j, \dots, x_N\}$, to be classified into n classes, $C_1, \dots, C_k, \dots, C_q, \dots, C_n$, where $N \geq n$. Let, by teachers reference model, the i -th pattern belongs to the k -th class and j -th pattern belongs to the q -th class. That can be written as

$$!(x_i) = C_k; \quad i=1, \dots, N; k=1, \dots, n; \quad (2.1)$$

$$!(x_j) = C_q; \quad j = 1, \dots, N, q=1, \dots, n; j \neq i, q \neq k; \quad (2.2)$$

In an examination trial it is computed the maximum value, which means that the correct classification is achieved if the following pair of inequalities holds

$$?(x_i) = !(x_i) = C_k \Leftrightarrow y_k(x_i) > y_q(x_i) \quad (3.1)$$

$$?(x_j) = !(x_j) = C_q \Leftrightarrow y_q(x_j) > y_k(x_j) \quad (3.2)$$

Further, we assume that the patterns are represented as feature vectors x_1, \dots, x_N and that the weight vectors are represented with w_1, \dots, w_n , where w_k is associated with the class C_k .

The learning process is governed by a consequence driven teaching process with an *error correction learning rule*

$$\text{if } ?(x_i) \text{ is different than } !(x_i) = C_k$$

$$\text{then correct } w_k \text{ toward } x_i: w_k = w_k + c x_i \quad (4)$$

where c is a constant. In words, if the classifier erroneously classifies the pattern x_i in an test trial, a teaching trial is introduced in which the pattern x_i is added to the weight w_k , lecturing that x_i belongs to C_k .

Here c is a learning rate which is a constant and we use the value $c=1$.

3.2 Introducing transfer learning

Consider neural network from Figure 1 which has capability to classify N patterns into n classes, $N \geq n$. Consider the simplest task, two patterns x_i and x_j to be classified into two classes C_k and C_q . The problem is stated with relations (3). However let us emphasize that k and q are arbitrary in the set of $\{1, \dots, n | k \neq q\}$, and also i and j are arbitrary in the set $\{1, \dots, N | i \neq j\}$.

Now we introduce transfer learning. Let assume the considered neural network has been subject of a learning task which we call *first learning task*. After that first learning task the neural network learner is now subject to

a *second learning task*. The second learning task will be carried out by a supervised learning (or teaching) process represented by a teaching sequence L. The teaching sequence contains both teaching and test (examination) trials. However, the memory of the learner is updated only during the teaching trials. The test trials demonstrate the knowledge already stored in the memory of the neural network learner.

Let $y_k(\mathbf{x}_i)$ be the output of S_k element of the neural network at the completion of the first learning task. It is the initial knowledge as demonstrated by this neural network before the second learning task. We emphasize that with notation $y_k^0(\mathbf{x}_i) := y_k(\mathbf{x}_i)$, pointing out with a superscript 0 that it is initial knowledge for the second learning task. So the output $y_k^0(\mathbf{x}_i)$ manifests *the transfer learning from the first teaching task* about the concept class k, before the second teaching task with teaching sequence L is applied.

Let $y_k(\mathbf{x}_i/L)$ be the output of S_k element representing class k when shown pattern \mathbf{x}_i after the *learning in the second task* with the learning sequence L. So the second learning task will be modeled with the following outputs from elements S_k and S_q

$$y_k(\mathbf{x}_i/L) = y_k^0(\mathbf{x}_i) + a_{ij}p_i \tag{5.1}$$

$$y_q(\mathbf{x}_j/L) = y_q^0(\mathbf{x}_j) + a_{ji}p_j \tag{5.2}$$

where p_i is the number of appearance of pattern \mathbf{x} in a teaching trial of the teaching sequence L, i.e. the number of application of the learning rule (4), and a_{ij} is the inner product between patterns, $a_{ij} = \mathbf{x}_i^T \mathbf{x}_j$.

So, in order a correct pattern classification to be achieved in the second task, by a neural network with maximum selector as in Figure 1, it is necessary and sufficient that the following system of inequalities holds

$$y_k(\mathbf{x}_i/L) > y_q(\mathbf{x}_i/L) \tag{6.1}$$

$$y_q(\mathbf{x}_j/L) > y_k(\mathbf{x}_j/L) \tag{6.2}$$

which leads to

$$a_{ii} p_i - a_{ij} p_j > -y_k^0(\mathbf{x}_i) + y_q^0(\mathbf{x}_i) \tag{7.1}$$

$$-a_{ji} p_i + a_{jj} p_j > y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j) \tag{7.2}$$

That reasoning leads to the following Theorem:

Theorem 1. (Transfer learning in case of learning arbitrary two patterns from a set of patterns) Let \mathbf{x}_i and \mathbf{x}_j be arbitrary patterns from a set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_j, \dots, \mathbf{x}_N\}$ of N patterns, which a maximum selecting neural classifier should learn to classify into given two classes C_k and C_m respectively, from a set $C = \{C_1, \dots, C_k, \dots, C_m, \dots, C_n\}$ of n classes. Let $a_{ij} = \mathbf{x}_i^T \mathbf{x}_j$. Let the lecture (teaching trial) $!x_i = C_k$ is presented p_i times, and let $!x_j = C_m$ is presented p_j times in the teaching sequence L. Then, the problem of correct classification learning is equivalent to the problem of finding p_i and p_j which satisfy the pair of inequalities

$$\begin{pmatrix} a_{ii} & -a_{ji} \\ a_{ij} & a_{jj} \end{pmatrix} \begin{pmatrix} p_i \\ p_j \end{pmatrix} > \begin{pmatrix} \tau_{qk}(\mathbf{x}_i) \\ \tau_{kq}(\mathbf{x}_j) \end{pmatrix} \tag{8}$$

which in compact form can be written as

$$\mathbf{A} \mathbf{p} > \boldsymbol{\tau} \tag{9}$$

where

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_{qk}(\mathbf{x}_i) \\ \tau_{kq}(\mathbf{x}_j) \end{pmatrix} = \begin{pmatrix} y_q^0(\mathbf{x}_i) - y_k^0(\mathbf{x}_i) \\ y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j) \end{pmatrix} \tag{10}$$

is named transfer learning vector.

Before we present the proof of the Theorem 1 we will give interpretation of the variables which appear in the theorem.

First we point out that the left side of the inequalities (8) contain a matrix of all inner products between patterns. The inner product a_{ij} between two patterns \mathbf{x}_i and \mathbf{x}_j shows how much their features overlap. It can be viewed as covariance, a *manifestation of pattern similarity*. We denote that matrix $\mathbf{A} = [a_{ij}]$, and name it a *matrix of mutual similarity between patterns*. Note that this matrix is invariant to the teaching process, it simply describes relation between the given patterns.

The vector $\mathbf{p} = (p_i \ p_j)^T$ shows how many times patterns were shown in a teaching trial in the teaching sequence L. It is a *training vector of the second learning task*.

The right side of inequalities contain the variables are due to *transfer learning* from a learning task prior to this considered task of training using curriculum L. It contains differences of outputs of S-elements for each pattern shown in the teaching process, i.e. $y_{qk}^0(\mathbf{x}_i) = y_q^0(\mathbf{x}_i) - y_k^0(\mathbf{x}_i)$ for shown pattern \mathbf{x}_i , and $y_{kq}^0(\mathbf{x}_j) = y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j)$ for shown pattern \mathbf{x}_j .

So the left side of matrix inequalities, $\mathbf{A} \mathbf{p}$, contains all controllable and observable parameters of the teaching process. If patterns are known, the matrix \mathbf{A} is known. The teaching sequence L is the one it is looked for, and after it is found, the vector \mathbf{p} will be known. However, the right side of inequalities, vector $\boldsymbol{\tau}$, which represents *transfer learning, is in general case not known*. Teaching of a biological brain does not assume that initial values of weights are known. Often the task is to teach a learner regardless the transfer learning. However, because of unknown transfer learning teaching process might converge in a longer time.

The proof of the Theorem can be expressed using a reasoning flow diagram as shown in Figure 2. The equations and inequalities used have been already described in the text.

Note that if all thresholds in the network are equal, then the transfer learning can be expressed as

$$\tau_{kq}(\mathbf{x}_j) = (\mathbf{w}_k^0 - \mathbf{w}_q^0) \mathbf{x}_j \tag{11}$$

3.3 Modeling positive and negative transfer learning

In this section we will address formally the following questions.

Given a neural network that has been subject to a learning Task1, is it possible to find a teaching sequence L which will solve the teaching Task2 regardless the transfer learning from Task1?

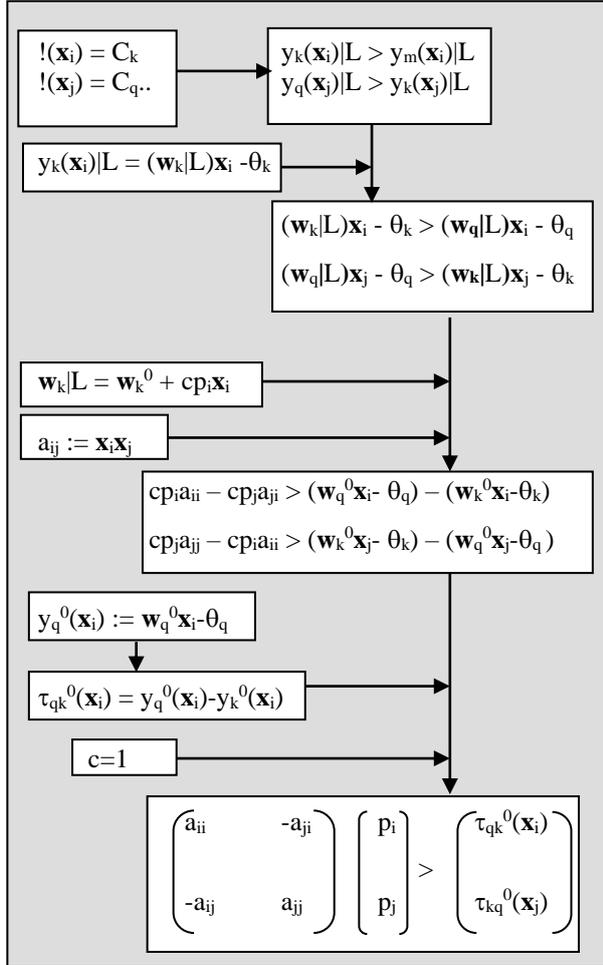


Figure 2: Proof of the Theorem 1 in a reasoning flow representation.

Case of positive transfer of learning. Is it possible that Task1 helps achieving shorter sequence L in Task2, than if starting from no previous transfer of learning?

Case of negative transfer of learning. Is it possible that Task1 will produce a longer sequence L in Task2, than if starting from no previous transfer of learning?

In order to answer those questions we will further elaborate on the inequalities (7). We repeat them here for clarity and renumber them (12) for keeping the sequence:

$$a_{ii} p_i - a_{ij} p_j > -y_k^0(\mathbf{x}_i) + y_q^0(\mathbf{x}_i) \quad (12.1)$$

$$-a_{ji} p_i + a_{ij} p_j > y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j) \quad (12.2)$$

The inequalities (12) can be rewritten to see explicitly how p_j depends on p_i . To see that, we move terms with p_i on the right side of the inequalities (12) and we obtain the following system of inequalities:

$$-a_{ij} p_j > -a_{ii} p_i - y_k^0(\mathbf{x}_i) + y_q^0(\mathbf{x}_i) \quad (13.1)$$

$$a_{ij} p_j > a_{ji} p_i + y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j) \quad (13.2)$$

Now we multiply equation (13.1) with -1 , which changes the inequality sign from $>$ to $<$. We obtain the following system of inequalities:

$$a_{ij} p_j < a_{ii} p_i + y_k^0(\mathbf{x}_i) - y_q^0(\mathbf{x}_i) \quad (14.1)$$

$$a_{ij} p_j > a_{ji} p_i + y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j) \quad (14.2)$$

where from

$$p_j < (a_{ii} / a_{ij}) p_i + (y_k^0(\mathbf{x}_i) - y_q^0(\mathbf{x}_i)) / a_{ij} \quad (15.1)$$

$$p_j > (a_{ji} / a_{ij}) p_i + (y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j)) / a_{ij} \quad (15.2)$$

and finally

$$p_j < (a_{ii} / a_{ij}) p_i + \tau_{kq}(\mathbf{x}_i) / a_{ij} \quad (16.1)$$

$$p_j > (a_{ji} / a_{ij}) p_i + \tau_{kq}(\mathbf{x}_j) / a_{ij} \quad (16.2)$$

These inequalities can be observed geometrically as in Figure 3.

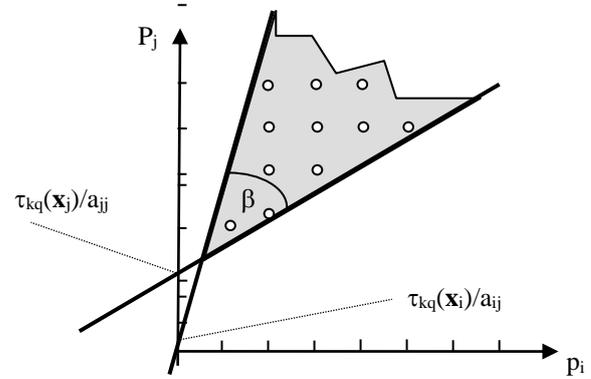


Figure 3: Geometric interpretation of Theorem 1.

Note that because $a_{ij} = \mathbf{x}_i^T \mathbf{x}_j$, the coefficient $a_{ii} / a_{ij} \geq 1$, and the coefficient $a_{ji} / a_{ij} \leq 1$. Because $\mathbf{x}_i \neq \mathbf{x}_j$ those coefficients are never at the same time equal 1. Because coefficients $a_{ii} / a_{ij} \geq 1$ and $a_{ji} / a_{ij} \leq 1$ are slopes of the boundaries of the solution region, it means because patterns are different, $\mathbf{x}_i \neq \mathbf{x}_j$, the angle β on Figure 3 always exists, and the solution points for (p_i, p_j) inside the shaded region defined by the angle β always exist.

So we can formulate the following statement.

Theorem 2. It is always possible to chose a teaching sequence L which will contains patterns \mathbf{x}_i and \mathbf{x}_j ($\mathbf{x}_i \neq \mathbf{x}_j$), such that after training with L the learner is able to correctly classify the patterns regardless transfer of learning from a previous learning task.

The proof is given in the previous reasoning using equations (12)-(16).

The teaching space in which we observe transfer training is an *integer space*. The components p_i and p_j are non-negative integers. In Figure 3 it is shown that only the integer points are solutions for correct classification of \mathbf{x}_i and \mathbf{x}_j .

4 Geometric interpretation of transfer learning: positive, negative, and tabula rasa

From Figure 3 we can see that the solution region of the teaching process is a convex cone defined by two parameters: 1) the position of the coordinate origin relative to the vertex of the cone, and 2) the angle of the convex cone. The orientation of the cone is always such that most of it lies within the first quadrant, although the vertex may be in any quadrant. We call this a positive convex cone.

The angle of the convex cone is determined solely by inner products between patterns. The angle represents the similarity between patterns in a sense of overlapping features.

Transfer learning is geometrically represented by the position of origin of the coordinate space (p_i, p_j) relative to the convex cone. That is illustrated in Figure 4.

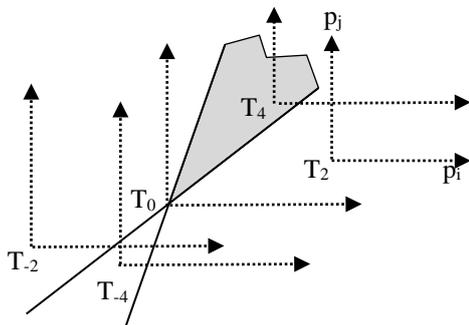


Figure 4: A geometric interpretation of transfer learning. The plane (p_i, p_j) , the convex cone, and various coordinate origins representing transfer learning from a previous learning task.

As Figure 4 shows, if the peak of the convex cone is in the coordinate origin (coordinate system T_0), then there is no transfer learning. The learner starts from tabula rasa initial conditions. It means that the memory values are all equal, for example all zero. However they are not necessary zero, they need only to be all equal (homogenous initial conditions). In this condition a learning process must take place for both patterns (or lessons) x_i and x_j in order the learner to correctly recognize those patterns.

If the coordinate origin is inside the solution region (coordinate system T_4) the learner has positive transfer learning. There is no need of additional learning. The previous learning is enough for the correct recognition of the patterns.

If the coordinate origin is in region symmetrically opposite the solution region, (negative convex cone), it is an example of negative transfer learning. Coordinate system T_{-4} is such a case. Both patterns x_i and x_j have been previously, in Task1, classified into classes which are incorrect according to the new Task2. So the new learning process must include both patterns. The learning process will be longer than in case of tabula rasa condition.

If the coordinate system is in the area outside the positive and negative convex cones (examples T_2 and T_{-2} coordinate systems), then there are situations in which for one pattern there is positive transfer learning and for the other is negative.

5 Defining an index of transfer learning in a neural network

Based on the geometrical interpretation of transfer learning in Figure 4 we will now define an index of transfer learning, a numerical representation of transfer learning. Measure of negative transfer as well as transferability measure are emphasized in contemporary transfer learning research (Tan et al. 2018). The index which we will discuss here is proposed in (Bozinovski and Fulgosi 1976).

The mathematical measure of transfer learning was introduced using the following reasoning. Observe the segments the lines in Figure 4 define intercepting with ordinate p_j . For T_0 coordinate system both lines have intercept 0. For coordinate system T_4 , one intercept is positive (for the boundary line $p_j > p_i$) and the other is negative (for the boundary line $p_j < p_i$). For coordinate system T_2 one intercept is positive and the other is negative. For T_1 both intercepts are negative. For T_3 both intercepts are positive. Note that also in Figure 3 above, it is shown a case of both positive intercepts. So we will only observe the sign of the intercepts, positive, negative, or zero, and we will define index of transfer learning.

Note that the intercepts are defined as $\tau_{kq}(\mathbf{x}_i)/a_{ij}$ and $\tau_{kq}(\mathbf{x}_j)/a_{ij}$ and consequently their signs are defined as $\text{sign}(\tau_{kq}(\mathbf{x}_i)/a_{ij})$ and $\text{sign}(\tau_{kq}(\mathbf{x}_j)/a_{ij})$ where $\text{sign}(\cdot)$ is a function that gives 1 for positive, 0 for zero, and -1 for negative argument. Now we can define an index, a measure of transfer learning on the basis of signs of intercepts of the boundary lines for patterns \mathbf{x}_i and \mathbf{x}_j .

$$TL(\tau_{kq}(\mathbf{x}_i), \tau_{kq}(\mathbf{x}_j)) = 3\text{sign}(\tau_{kq}(\mathbf{x}_i)) - \text{sign}(\tau_{kq}(\mathbf{x}_j)) \quad (17)$$

According to this index, if both signs are positive then $TL = 2$. That corresponds to a coordinate system T_2 in Fig. 4. If both are negative, then $TL = -2$ and that corresponds to the coordinate system T_{-2} in Fig 4. If both are zero, then $TL=0$, which corresponds to coordinate system T_0 in Fig. 4. If $\text{sign}(\tau_{kq}(\mathbf{x}_i)) = 1$ and $\text{sign}(\tau_{kq}(\mathbf{x}_j)) = -1$, then $TL = 4$ which corresponds to coordinate system T_4 in Fig. 4.

Note that the index TL considers all the integer values in the interval $[-4,+4]$. Figure 5 shows the TL values and their geometric interpretation.

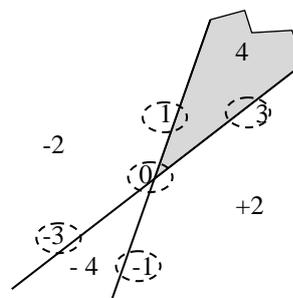


Figure 5: A geometric interpretation of index TL , a numerical index of transfer learning. It shows the values of the regions of the (p_i, p_j) plane where a learner finds itself after the first learning task Task1, and facing the second learning task Task2.

The introduced index of transfer learning shows position of the coordinate origin in the plane (p_i, p_j) relative to the peak of the vertex inside which is a solution of the pattern recognition problem. It shows where in the (p_i, p_j) plane is the starting point to learn Task2 by a learner with transfer learning from previous Task1. From Fig. 5 we can give following interpretations for transfer learning index TL :

If $TL = 4$ the learner correctly classifies both patterns, without need for additional learning. It is a positive transfer learning from a previous Task1.

columns) equal to number of templates for that class. In case of inequalities (20), the first class contains two templates and the second contains one template pattern. From this case study we can generalize the transfer learning model for a multi-class and multi-template per class case as

$$(X/L): \mathbf{A}p > \mathbf{W}^0\mathbf{X} + \boldsymbol{\theta} \quad (21)$$

where $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ is the set of patterns which should be learned in the second task with the curriculum sequence L .

Note that the mathematical model of transfer learning (21) divides the left side of relation to be a *teacher's side*, and right side a *learner's side*.

At the teacher side are similarity matrix A and distribution vector p showing how many times each pattern appeared in a teaching trial of the curriculum L . Matrix A shows that what matters in the teaching process are not the patterns themselves but rather their correlations, inner products, which can be interpreted as similarities.

At the learner side, \mathbf{W}^0 represents difference of initial conditions of the memory due to transfer learning, \mathbf{X} is the vector of template vectors, a matrix containing patterns to be classified, and $\boldsymbol{\theta}$ represents difference between thresholds of neurons representing classes. Note that the matrix \mathbf{W}^0 contains blocks showing which template is assigned to which class.

As pointed before, the space $p = (p_1, \dots, p_N)$ is an integer space. Dealing with neural network learning is actually an *integer programming problem*. We are interested in the most efficient training, and we are looking for a training sequence L of the minimal length. So we look for a criterion

$$p_1 + p_2 + \dots + p_N = \min \quad (22)$$

Such a criterion will observe the appearance of patterns only in a teaching trial. If we are interested in minimal sequence that includes test trials, then the optimality criterion is

$$(p_1 + q_1) + (p_2 + q_2) + \dots + (p_N + q_N) = \min \quad (23)$$

where q_i is number of appearances of the pattern \mathbf{x}_i in a test trial, which does not change the memory of the learner, but affects the length of the training sequence L .

8 Experimental investigation on transfer learning

Experimental investigation on transfer learning was carried out in the period 1976-1981. Initial experiments was with a dataset containing images of letters A, B, E, F, and T taken from the terminal IBM29 card puncher. Those experiments were carried out on the computer IBM 1130. Later experiments were carried out with two datasets. One dataset contained 40 images, consisting of 26 letters, 10 numbers, and 4 special symbols from the terminal IBM29. The other dataset can be described as Computer Terminals dataset, consisting of $3 \times 26 = 78$ images, taken from three computer terminals: IBM29 card puncher, VR14 video screen, and VT50 video

screen. The experiments were carried out on a computer VAX/VMS. Figure 7 shows the Computer Terminals dataset. As can be seen, the letters of the three terminals are mostly identical on an image with resolution 7×5 , with differences in letters A, B, D, G, J, M, N, O, V, and W.

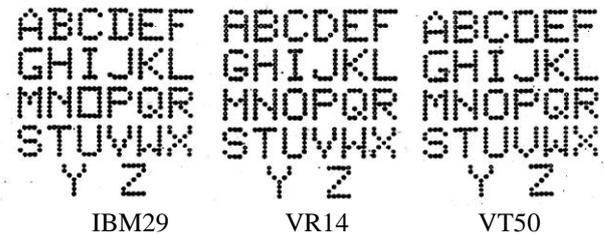


Figure 7: The dataset Computer Terminals used in experimental investigation.

8.1 An experiment in tabula rasa condition, showing influence of pattern similarity

Here we will show an experiment in tabula rasa learning, to see the influence of similarity (overlapping pattern features) on the learning process. Consider the patterns E, T, F, shown in Fig. 7. They are the same for all considered terminals. Figure 8 shows the search through the (p_E, p_T, p_F) space that the learning process performs.

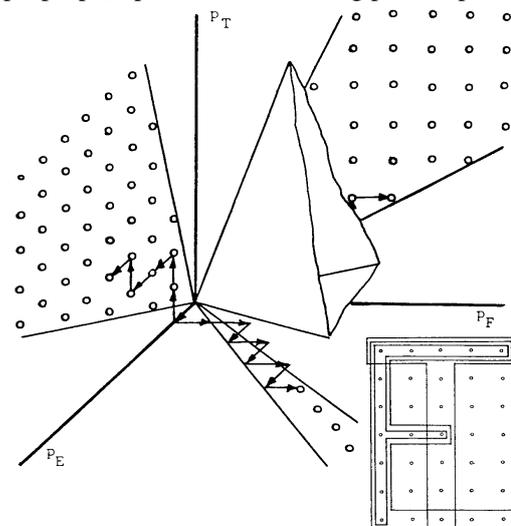


Figure 8: Learning trajectory in case of tabula rasa learner, learning similar patterns E and F, together with the pattern T (Bozinovski 1981, 1985b).

As Fig. 8 shows, the problem is the distinction between the patterns F and E. The convex cone angle is narrow, and it is possible that in some search steps the cone does not contain an integer point. The search for an integer solution is what makes necessary to repeat images E and F several times until they are distinguished by the learner.

This experiment emphasizes the problem of feature overlapping and the problem of one image included in another image. To emphasize the image-subimage relation, a measure of similarity between patterns is introduced in (Bozinovski and Fulgosi 1976). The following index

$$SL(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j / \min\{\mathbf{x}_i^T \mathbf{x}_i, \mathbf{x}_j^T \mathbf{x}_j\} \quad (24)$$

has values between 0 and 1. If $SL = 0$ the solution convex cone covers the entire first quadrant of the teaching space (p_i, p_j) . If $0 < SL < 1$, the convex cone includes the line $p_j = p_i$. If $SL = 1$, one of the cone boundaries is the line $p_j = p_i$.

Such a measure is used to predict the length of the teaching sequence L , and with that the efficiency of the training.

8.2 Experimental investigation in positive and negative transfer learning

Experiments shown here are carried out during 1976-1978 on a IBM1130 computer. Table I shows the results of the transfer learning experiments which show both positive and negative transfer. (Bozinovski et al. 1977, Bozinovski 1978).

Task 1 Images	Task 2 Images	Task 2 Teaching sequence
No transfer learning, tabula rasa		
	A, B	AB.
	A, B, T	ABT.
	E, F	EFFEFEFEF.
	E, F, T	EFFEFEFEFTTT.
Negative transfer learning		
E, F	A, B	ABABABAB.
Positive transfer learning		
A, B, T	E, F, T	EFT.

Table 1: Experiments in transfer learning. Cases of tabula rasa, negative, and positive transfer learning.

In presenting the results of the experiments with transfer learning here we introduce the notation $L_{D2/D1}$, meaning training sequence of Task2, trained with a set of patterns D2, after the Task1 in which the learner is trained with a set of patterns D1. For a tabula rasa training, we use notation $L_{D2/\emptyset}$.

Experiment with no transfer learning. As can be seen from the presented experiments, learning the patterns E, T, and F with no transfer learning needs the teaching sequence $L_{ETF/\emptyset} = EFFEFEFEFTTT$. The length of the sequence is due to similarity between E and F.

Experiment showing positive transfer learning. If the neural network is previously exposed to the Task1 where it learned to recognize A and B, and after that is exposed to Task2 to learn E, T, and F, then the teaching sequence for Task2 is $L_{EFT/ABT} = EFT$. The teaching sequence for learning E, T, F in this case is *shorter than in case of tabula rasa*. That is experimental evidence of positive transfer learning.

Experiment showing negative transfer learning. If the neural network is previously exposed to a Task1 to learn E and F, and after that in Task2 to learn A and B, the teaching sequence for learning A and B is $L_{AB/EF} = ABABABAB$. It is *longer than in case of learning A and B in tabula rasa condition*, $L_{AB/\emptyset} = AB$. That is an experimental evidence of negative transfer learning.

8.3 Application of transfer learning

Here we show results of experiments carried out during 1980-1981 on a VAX/VMS computer (Bozinovski 1981). The experiments consider real application, learning to recognize letters from computer terminals.

Consider the dataset Computer Terminals from Figure 7. The question we would like to answer experimentally is: If in the Task1 we teach a learner to recognize the letters from the terminal VR14, how faster the learner will be able to learn in Task2 to recognize the letters from the terminal IBM29, comparatively to learning from tabula rasa condition.

In these experiments we used the following teaching strategy (Bozinovski 1981) named perceptron teaching strategy:

```

Procedure PerceptronTeachingStrategy
iteration: teachflag = 0;
i:=0; n=26;
while i < n do
i:=i+1
grade = test(xi);
if grade = 'incorrect'
then teach(xi), teachflag=1;
endwhile;
if teachflag = 1 goto iteration;
end.
    
```

This strategy performs test trials on all $n=26$ images, and only when needed, a teaching trial is applied for a particular image. After such an iteration (or epoch), another iteration takes place, and so on, until no teaching trial appeared in an iteration ($teachflag=0$). That means there were only test trials in the last iteration and the learner now recognizes all the patterns correctly.

Using this strategy applied to the set of letters IBM29, in case of tabula rasa, it gives the 9 iterations as shown in Figure 9.

```

T* = ABCDEFGHIJKLMNOPQRSTUVWXYZ
      CEF GHIJKLMNOPQRSTUVWXYZ
      ABCDFGJLOPQRSUWZ
      ACDEFHIJKLMNOPRSTUVXY
      BEFGHIJKLMNOPQRUWZ
      ACJPRTVWXY
      BDEFHIKLMNPQVZ
      FOU
      CEGJLS
    
```

Figure 9: Teaching sequence of learning the set of letters IBM29 with no transfer learning.

With T^* we denote the solution teaching sequence in which only the teaching trials appear. With $|T^*|$ we denote its length, in trials. With C^* we denote teaching sequence containing both teaching and test trials, and with $|C^*|$ its length. For the experiment on Fig. 9 we obtained

$$|T^*|_{IBM29/\emptyset} = 135 \text{ and } |C^*|_{IBM29/\emptyset} = 395.$$

If before learning the set IBM29 in Task2, the set VR14 was learned in Task1, then in Task2 the teaching process completes in 4 iterations, with the teaching sequence shown in Fig. 10.

T* = A DGIMNOQTUW
 AC DGH JKNOQRSUWX
 C DGH I JMNPS VW YZ
 ABEF KLMX

Figure 10: The teaching sequence in case when set IBM29 is leaned, providing that previously was learned the set VR14.

In the experiment shown in Fig. 10 we obtained $|T^*|_{IBM29/VR14} = 48$ and $|C^*|_{IBM29/VR14} = 178$.

The experiment shown in Figure 9 and 10 shows an application of positive transfer learning. We obtained shorter training sequence

$|C^*|_{IBM29/VR14} = 178 < |C^*|_{IBM29/\emptyset} = 395$.

The teaching time is $178/395 = 0.45$ of the tabula rasa teaching time, and the speed of learning increases $1/0.45 = 2.2$ times.

When we carried out an experiment of learning the set VT50 if previously learned the set VR14, the result was

$|T^*|_{VT50/\emptyset} = 207$, $|T^*|_{VT50/VR14} = 43$,
 $|C^*|_{VT50/VR14} = 199 < |C^*|_{VT50/\emptyset} = 545$

The transfer learning teaching time is $199/545 = 0.36$ of the tabula rasa teaching time, and the speed of learning increases $1/0.36 = 2.8$ times.

This application shows the reason of use of transfer learning. If you have a knowledge of a dataset classification stored in a neural network in Task1, then transfer that knowledge to a different task which learns classification of a similar dataset. The training time will be shorter.

Here in this application subsection we give also the result of learning a dataset IBM29(40) of 40 images, defined as

$IBM29(40) = IBM29 \cup \{+, -, =, /\} \cup \{0,1,\dots,9\}$
 starting with tabula rasa condition.

The result we obtained is: 10 iterations,

$|T^*|_{IBM29(40)/\emptyset} = 204$ and $|C^*|_{IBM29(40)/\emptyset} = 604$.

This is an example of a 1981 machine learning experiment with 40 patterns (Bozinovski 1981).

9 Transfer learning research after 1986

The main focus of this paper is to give a review of the initial work on transfer learning in neural networks which took place between 1972 (Bozinovski 1972) and 1985 (Bozinovski 1985a, 1985b). To the best of our knowledge during that time period, there was no other work on transfer learning in neural networks. That was the period when neural networks were not the main topic in Artificial Intelligence, due to the book of Minsky and Papert (1969) which pointed out some limitations of perceptron type neural networks. Although during 1970's and 1980's there were works on multilayered neural networks (e.g. Fukushima, 1975, 1980), the interest in multilayered neural networks significantly increased after 1986, due to appearance of the book by the Parallel Distributed Processing (PDP) Group (Rumelhart et al. 1986). That book reignited the interest in neural networks, and after some time, the interest in transfer

learning in neural networks. Here we will give a short review on the works on transfer learning after 1986.

Early works after 1986 used other terms to describe transfer learning. One such term was "sequential learning", where negative transfer learning was covered with the term "interference" (McCloskey and Cohen, 1989). Other terms used were "adaptive generalization" (Sharkey and Sharkey, 1992), "learning by learning" (Naik and Mammone, 1993), and "lifelong learning" (Thrun and Mitchell, 1993).

In 1991 the term transfer learning related to neural networks reappeared in literature. That was the work of Pratt, Mostow, and Kamm (1991). That paper introduced a framework of transfer learning, pointing out various types of transfer learning. That framework was also described in the work of Pratt (1993). The framework is shown in Figure 11.

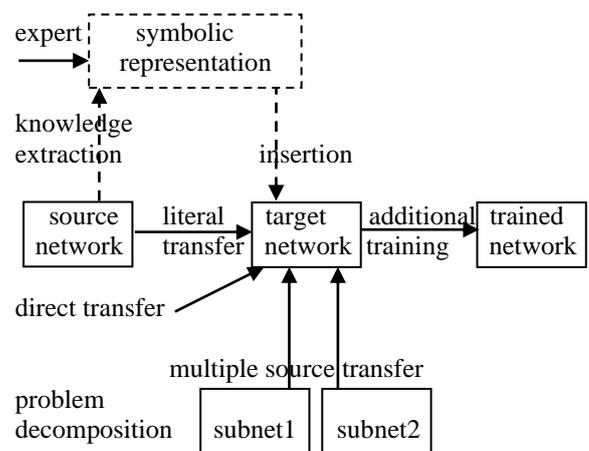


Figure 11. A general framework for transfer learning (adopted from Pratt et al., 1991).

As can be seen from Fig. 11, the general framework for transfer learning proposed in 1991 includes four types of transfer.

One is named literal transfer learning, and it is the transfer learning we used in our work (Bozinovski and Fulgosi, 1976), and is reviewed in this paper.

The second type is a transfer learning which uses direct intervention in the weights of a neural network. We call this direct memory access (DMA) type transfer of knowledge. It is an intervention in a neural network knowledge without a process of incremental learning. The weights change is named weights perturbation. An example of direct weight change described in (Pratt et al. 1991) is $w = w + rw$, where r is a random number between $(-0.6, 0.6)$. Weights perturbation method was also used in the work (Agarwal et al. 1992)

The third type uses problem decomposition into subproblems, represented by subnetworks, and training the subnetworks for the subproblems, and then insert the subproblem knowledge into the target network.

The fourth type of transfer is indirect transfer, where the weight-based knowledge is extracted, then it is represented as a rule-based knowledge, then it is updated using rule based representation, and then it is inserted in a target neural network as weights-based knowledge.

A review of transfer learning in the early 1990's is given by Pratt and Jennings (1996). A review by Pan and Yang (2010) covers the period after that. Tan et al. (2018) review the deep transfer learning.

10 Discussion and conclusion

The contribution of this paper is a review of an early period of transfer learning research, a period which was not known to the current researchers in transfer learning. In current history part of transfer learning, as covered by Wikipedia >Transfer Learning >History (2020) there is information which suggests that the beginning of transfer learning research is in 1993. This paper gives information on the transfer learning research during 1970's and early 1980's.

In this discussion let us mention that the original 1976 paper was published in the Proceedings of the symposium Informatica 1976, which took place in Bled, Slovenia, one year before appearance of the first issue of the journal Informatica, in 1977. The paper was published in Croatian, not in English, which is the main reason why the paper was not known for a rather long time.

In the review of the period 1990 - 2000 given in this paper, we can notice that the research during that period was focused on forms that transfer learning can take, and directions it can go. The fundamental concepts like a measure of transfer learning was not covered. The interest of fundamental notions was pointed out again in 2000s (Tan et al, 2018). That relates the research in 1970's to the contemporary research in transfer learning.

Let us mention that the application of transfer learning with real datasets of images described here, such as Computer Terminals dataset containing 3x26 letters and the IBM29(40) containing 40 characters on a matrix 7x5 is an early use of datasets of characters in machine learning. An example of a character dataset used in contemporary research (e.g. Wang et al. 2019) contains 9 characters (digits 0 to 9) on a matrix 28x28, with variety of templates.

In conclusion, this paper extends the knowledge in transfer learning with a relation between the pioneering work (in 1970's and early 1980's) and the current research on transfer learning, giving also a review of the period in early 1990's. Important part of that relation is the reminder of the theoretical 1976 paper, which presented the first mathematical and geometrical modeling, and a measure of transfer learning. The experimental work during 1976-1981 with datasets representing images of characters also relates to the contemporary research in machine learning.

11 Acknowledgement

The author wishes to thank professor Gjorgi Cupona from Mathematical Institute with Numeric Center of the University of Skopje, who in 1968 encouraged the author to prepare a maturnal thesis from the Glushkov's book. He also wishes to thank professor Ante Santic from Electrical Engineering Department of University of

Zagreb, who in 1972 allowed author's work on perceptrons. Professor Ante Fulgosi from Psychology department of University of Zagreb was coauthor of the first published paper on transfer learning in 1976. The author also wishes to thank professors Michael Arbib and Nico Spinelli who enabled him to work on the research related to this paper at the Computer and Information Sciences (COINS) Department of the University of Massachusetts at Amherst for the period 1980-81, and to thank professor Andrew Barto for the period 1995-96. The work at University of Massachusetts in both periods was supported by Fulbright grants. Part of the research reported here was supported by a Sigma Xi grant-in-aid in 1981. The author also wishes to thank the reviewers of this paper for their valuable comments.

12 References

- [1] A. Agarwal, R. Mammone, and D. Naik (1992) An on-line training algorithm to overcome catastrophic forgetting. In *Intelligence Engineering Systems through Artificial Neural Networks*. volume 2, pages 239-244. The American Society of Mechanical Engineers, AS-IE Press.
- [2] J. Baxter, R. Caruana, T. Mitchell, L. Pratt, D. Silver, S. Thrun (organizers) *Learning to Learn: Knowledge Consolidation and Transfer in Inductive Systems*, NIPS*95 Post-conference workshop, Vail, Colorado <http://socrates.acadiau.ca/courses/comp/dsilver/NIPS95ltnips95.workshop.pdf>
- [3] S. Bozinovski (1972) Perceptrons: Training in pattern recognition. (original in Croatian: Perceptroni i obucavanje u prepoznavanju oblika) unpublished student scientific competition paper, University of Zagreb
- [4] S. Bozinovski (1974). Perceptrons and possibility of simulation of a teaching process (original in Croatian: Perceptroni i mogucnost simuliranja procesa obucavanja), unpublished M.Sc. thesis, Electrical Engineering Department, University of Zagreb
- [5] S. Bozinovski, A. Fulgosi (1976). The influence of pattern similarity and transfer of learning upon training of a base perceptron B2. (original in Croatian: Utjecaj slicnosti likova i transfera ucenja na obucavanje baznog perceptrona B2), Proc. Symp. Informatica 3-121-5, Bled.
- [6] S. Bozinovski, A. Santic, A. Fulgosi (1977). Normal teaching strategy in pair-association in the case teacher:human-learner:machine. (original in Croatian: Normalna strategija obicavanja u obucanju asocojacije parova u slucaju ucitelj:čovjek-ucenik:masina), Proc. Conf. ETAN, 21:IV-341-346, Banja Luka, [available online].
- [7] S. Bozinovski (1978). Experiments with non-biological systems teaching. (original in Macedonian: Eksperimenti na obucavanje na nebioloski sistemi) Proc. Conf ETAN, 22:IV-371-379, Zadar [available online].
- [8] S. Bozinovski (1981). Teaching space: A representation concept for adaptive pattern

- classification. COINS Technical Report, University of Massachusetts at Amherst, No 81-28 [available online: [UM-CS-1981-028.pdf](#)]
- [9] S. Bozinovski (1985a). Adaptation and training: A viewpoint. *Automatika* 26 (3-4) 137-144
- [10] S. Bozinovski (1985b). A representation theorem for linear pattern classifier training. *IEEE Transactions on Systems, Man, and Cybernetics* 15(1): 159-161
- [11] S. Bozinovski (1995). Neuro-genetic agents and a structural theory of self-reinforcement learning systems. *CMPSCI Technical Report 95-107*, University of Massachusetts at Amherst [available online: [UM-CS-1995-107.pdf](#)].
- [12] K. Fukushima (1975) Cognitron: A self organizing multilayered neural network. *Biological Cybernetics* 20: 121-136
<https://doi.org/10.1007/BF00342633>
- [13] K. Fukushima (1980) Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* 36: 193-202, doi: 10.1007/BF00344251.
- [14] V. Glushkov (1967) Introduction to Cybernetics (original in Serbian: Uvod u Kibernetiku, translated from Russian, published by Zavod za izdavanje udzbenika Srbije)
- [15] I. Goodfellow, Y. Bengio, A. Courville (2016) *Deep Learning*, MIT Press, DOI:10.1007/s10710-017-9314-z
- [16] M. McCloskey, N. Cohen (1989) Catastrophic interference in connectionist networks: the sequential learning problem. *The Psychology of Learning and Motivation*, 24 DOI:10.1016/S0079-7421(08)60536-8
- [17] M. Minsky, S. Papert (1969) *Perceptrons*. The MIT Press, 1969
- [18] D. Naik, R. Mammone (1993) Learning by learning in neural networks, In R. Mammone (ed.) *Artificial Neural Networks for Speech and Vision*, Chapman and Hall, London.
- [19] S. Pan, Q. Yang (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345– 1359 DOI:10.1109/TKDE.2009.191
- [20] L. Pratt, J. Mostow, C. Kamm (1991). Direct transfer of learned information among neural networks. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, p. 584-589, Anaheim, CA.
- [21] L. Pratt (1993). Discriminability-based transfer between neural networks. In *NIPS Conference: Advances in Neural Information Processing Systems 5* Morgan Kaufmann Publishers. pp. 204-211
- [22] L. Pratt, B. Jennings (1996) A survey of transfer between connectionist networks, *Connection Science* 8(2) 163-184.
<https://doi.org/10.1080/095400996116866>
- [23] F. Rosenblatt (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review* 65: 386-408. DOI:10.1037/h0042519
- [24] F. Rosenblatt (1962). *Principles of Neurodynamics*. Spartan Books. DOI:10.2307/1419730
- [25] D. Rumelhart, J. McClelland, and the PDP Group (1986). *Parallel Distributed Processing*. MIT Press.
- [26] N. Sharkey and A. Sharkey (1992) Adaptive generalisation and the transfer of knowledge, *Proceedings of the Second Irish Neural Networks Conference*, Belfast.
- [27] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu (2018). A Survey on Deep Transfer Learning, arXiv:1808.01974v1 [cs.LG] 6 Aug 2018. DOI:10.1007/978-3-030-01424-7_27
- [28] S. Thrun, T. Mitchell (1993) Lifelong robot learning, Technical Report IAI-TR-93-7, Institute for Informatics III, University of Bonn.
[https://doi.org/10.1016/0921-8890\(95\)00004-Y](https://doi.org/10.1016/0921-8890(95)00004-Y)
- [29] H. Wang, C. Li, X. Zhen, W. Yang, B. Zhang (2019) Gaussian Transfer Convolutional Neural Networks, *IEEE Transactions on Emerging Topics in Computational Intelligence* 3 (5) 360-368. DOI:10.1109/TETCI.2018.2881225
- [30] K. Weiss, T. Khoshgoftaar, D. Wang (2016) A survey of transfer learning. *Journal of Big Data* 3:9. DOI:10.1186/s40537-016-0043-6
- [31] Wikipedia > Transfer Learning (October 2020)
https://en.wikipedia.org/wiki/Transfer_learning

Minimum Flows in Parametric Dynamic Networks - the Static Approach

Grigoraş (Avesalon) Nicoleta

Transilvania University of Braşov, 50091 Braşov, Iuliu Maniu, 50, Romania

E-mail: nicole.grigoras@gmail.com

Keywords: dynamic network, parametric network, minimum flow

Received: October 15, 2018

The problems of flows in parametric networks extend the classical problems of optimal flow to some special kind of networks where capacities of certain arcs are not constants but depending on several parameters. Consequently, these problems consist of solving a range of ordinary (nonparametric) optimal flow problems for all the parameter values within certain sub-intervals of the parameter values. Although classical network flow models have been widely used as valuable tools for many applications [1], they fail to capture the essential property of the dynamic aspect of many real-life problems, such as traffic planning, production and distribution systems, communication systems, evacuation planning, etc. In all these cases, time is an essential component, either because the flows take time to pass from one location to another, or because the structure of the network changes over time. Accordingly, the dynamic flow models seem suited to catch and describe different real-life dynamic problems such as network-structure changing over time or timely decision-making, but, because of their complexity, these models have not been as thoroughly investigated as those of classical flows.

This article presents and solves the problem of the minimum flows in a parametric dynamic network. The proposed approach consists in applying a parametric flow algorithm in the reduced expanded network which is obtained by expanding the original dynamic network. A numerical example is also presented for a better understanding of the used approach.

Povzetek: V članku je predstavljena metoda za rešitev problema najmanjšega pretoka v parametričnih dinamičnih mrežah.

1 Introduction

The parametric maximum flow problem, as well as that of the related minimum flow one represent generalizations of ordinary problems for the maximum, respectively minimum flow in which the upper/lower bounds of some arcs depend on a single parameter, being monotonically increasing (or decreasing) functions of the parameter. For the parametric maximum flow problem with zero lower bounds and linear capacity functions of a single parameter, G.Ruhe proposed [16] an original 'piece-by-piece' approach while papers [11] and [13] solves the problems of parametric minimum, respectively maximum flows via an partitioning approach. Finally, the same partitioning approach is extended to a discrete time dynamic network in paper [3]. This class of problems is known as the flow in parametric networks problem. Beside the applications of the ordinary, nonparametric flow problems, the applications of those of flows in parametric networks may include: multiprocessor scheduling with release times and deadlines, integer programming problems, computing sub-graph density and network vulnerability and partitioning a data base between fast and slow memory, product selection, flow sharing, database record segmentation in large shared databases, optimizing field repair kits, etc.[15]

Besides this, the static network flow models arise in a num-

ber of combinatorial applications that on the surface might not appear to be maximum flow problems at all. The problem also arises directly in problems as far reaching as machine scheduling, the assignment of computer modules to computer processors, tanker scheduling etc. [1]. However, in some other applications, time is an essential ingredient, whether it involves problems like maximum or minimum flows in time-varying networks [2], [3], dynamic flows of minimum cost [12], general problems of temporally repeated flows [6] such as earliest arrival flow [17] or dynamic resource allocation problem for large-scale transportation network evacuation [7]. Starting from the complexities of the classic algorithms, an important series of other papers concern with the complexities of dynamic network algorithms. Cai et al. [4] proved that the complexity of finding a shortest (quickest) dynamic flow augmenting path, by exploring the forward and reverse arcs successively, is $O(nmT^2)$. For algorithms which explores the two sub-networks (the forward sub-network, consisting of the set of direct arcs and the reverse sub-network consisting of the set of reverse arcs simultaneously, Miller-Hooks and Patterson [8] also reported a complexity of $O(n^2T^2)$. By using special node addition and selection procedures, Nasrabadi and Hashemi [9] succeeded to reduce significantly the number of node time pair that needs to be visited. The worst-case complexity of their algorithm is

$O(nT(n + T))$. Finally, Orlyn reported [10] an extremely good running time of $O(nm)$ for the problem of maximum flow. In this case we need to use dynamic network flow models. Due to the powerful versatility of dynamic flow algorithms, it is not surprising that these algorithms are often more difficult to design than their static counterparts [10] but still they are also very challenging problems. The approaches for solving the minimum parametric flow over time problem via applying classical algorithms can be grouped in two main categories:

i) by applying a non-parametric minimum dynamic flow algorithm [12] in dynamic residual networks generated by partitioning the interval of the parameter values [9], [14]. This first category approach was also used [3] for finding a maximum parametric flow in discrete-time dynamic networks;

ii) by applying a static (classical sequential or parallel [5]) parametric flow algorithm for the maximum [13], [16] or for the minimum [11] flow in a (static network) reduced expanded network, which is obtained by expanding the original dynamic network.

In this paper, the case of the minimum flows in parametric dynamic networks is considered. The proposed approach consists in transformation of the problem of minimum flow in parametric dynamic network into that of the minimum flow in parametric static network. This problem generalizes the problems of flow in dynamic network and of flow in parametric static networks through the following assumptions: (1) the dynamic network and corresponding expanded static network are with lower bounds, (2) we address the minimum flow problem on a dynamic network with time varying transit times as well as time varying lower and upper bounds on arc i.e. the dynamic network is not stationary.

The remainder of this paper is organized as follows. In Section 2 some dynamic network notations and terminology are presented. Then, in Section 3 we expose, the minimum flow in parametric static network, while in Section 4 the minimum flow in parametric dynamic network problem is presented. In Section 5 an example is given. In the presentation to follow, some familiarity with flow problems is assumed and many details are omitted.

2 The minimum flows in dynamic networks.

Let $G = (N, A, l, u)$ be a static network with the nod set $N = \{1, \dots, i, \dots, j, \dots, n\}$, the arc set $A = \{a_1, \dots, a_k, \dots, a_m\}$, $a_k = (i, j)$, the lower bound function $l : A \rightarrow \mathbf{R}^+$ and the upper bound (capacity) function $u : A \rightarrow \mathbf{R}^+$, where \mathbf{R} is the real number set. To define the minimal static flow problem, we distinguish two special nodes in the static network $G = (N, A, l, u)$: a source node 1 and a sink node n .

Let \mathbf{N} be the natural number set and let $H = \{0, 1, \dots, T\}$ be the set of periods, where T is a finite time horizon, $T \in$

\mathbf{N} . Let $h : A \times H \rightarrow \mathbf{N}$ be the transit time function, $l_h : A \times H \rightarrow \mathbf{R}^+$ the time lower bound function and $u_h : A \times H \rightarrow \mathbf{R}^+$ the time upper bound function. For each arc $(i, j) \in A$ the $h(i, j; t)$ represents the transit time of arc (i, j) at time t , $t \in H$.

A dynamic flow from source node 1 to sink node n is any flow from 1 to n in which not less than $l_h(i, j; t)$ and not more than $u_h(i, j; t)$ flow units starting from node i at time t and arriving at node j at time $\theta = t + h(i, j; t)$ for all arcs (i, j) and all t . The minimal dynamic flow problem for T time periods is to determine a flow function $f_h : A \times H \rightarrow \mathbf{N}$, which should satisfy the following conditions in dynamic network $G_h = (N, A, h, l_h, u_h)$:

$$\sum_{i=1}^T (\sum_j f_h(i, j; t) - \sum_k \sum_\tau f_h(k, i; \tau)) = v_H, \tag{2.1.a}$$

$$\sum_j f_h(i, j; t) - \sum_k \sum_\tau f_h(k, i; \tau) = 0, \tag{2.1.b}$$

$i \neq 1, n, t \in H,$

$$\sum_{i=n}^T (\sum_j f_h(i, j; t) - \sum_k \sum_\tau f_h(k, i; \tau)) = -v_H, \tag{2.1.c}$$

$$l_h(i, j; t) \leq f_h(i, j; t) \leq u_h(i, j; t), \tag{2.2}$$

$\forall (i, j) \in A \text{ and } \forall t \in H$

$$\min v_H, \tag{2.3}$$

where $\tau = t - h(k, i; \tau)$, $v_H = \sum_{t=0}^T v(t)$, $v(t)$ is the flow value at time t and $f_h(i, j; t) = 0$, $(i, j) \in A$, $t \in \{T - h(i, j; t) + 1, \dots, T\}$.

Obviously, the problem of finding a minimum dynamic flow is more complex than the problem of finding a minimum static flow. Fortunately, this complication can be solved by rephrasing the dynamic flow problem into a static flow problem in a static network $G' = (N', A', l', u')$ called reduced expanded network. First, we form the expanded network $G_H = (N_H, A_H, l_H, u_H)$ with $N_H = \{i_t | i \in N, t \in H\}$, $A_H = \{(i_t, j_\theta) | (i, j) \in A, t = 0, 1, \dots, T - h(i, j; t)\}$, $l_H(i_t, j_\theta) = l_h(i, j; t)$, $u_H(i_t, j_\theta) = u_h(i, j; t)$, $(i_t, j_\theta) \in A_H$.

We have $|N_H| = n(T + 1)$ and $|A_H| \leq m(T + 1) - \sum_A \bar{h}(i, j)$, where $\bar{h}(i, j) = \min\{h(i, j; 0), \dots, h(i, j; T)\}$. Clearly, any dynamic flow from the source node 1 to the sink node n in dynamic network G_h is equivalent to a static flow from the source nodes $1_0, 1_1, \dots, 1_T$ to the sink nodes n_0, n_1, \dots, n_T in static network G_H and vice versa [2]. We can further reduce the multiple source, multiple sink problem in network G_H to the single source, single sink problem by introducing a supernode 1^* and a supersink node n^* building superexpanded network $G_H^* = (N_H^*, A_H^*, l_H^*, u_H^*)$, where $N_H^* = N_H \cup \{1^*, n^*\}$, $A_H^* = A_H \cup \{(1^*, 1_t) | t \in H\} \cup \{(n_t, n^*) | t \in H\}$, $l_H^*(i_t, j_\theta) = l_H(i_t, j_\theta)$, $u_H^*(i_t, j_\theta) = u_H(i_t, j_\theta)$, $(i_t, j_\theta) \in A_H$, $l_H^*(1^*, 1_t) = l_H^*(n_t, n^*) = 0$, $u_H^*(1^*, 1_t) = u_H^*(n_t, n^*) = \infty$, $t \in H$.

Next, we build the reduced expanded network $G' = (N', A', l', u')$ as follows. We define the function $h^* : A_H^* \rightarrow \mathbf{N}$, $h^*(1^*, 1_t) = h^*(n_t, n^*) = 0$, $t \in H$, $h^*(i_t, j_\theta) = h(i, j; t)$, $(i_t, j_\theta) \in A_H$. Let $d^*(1^*, i_t)$ be the length of the shortest path from the source node 1^* to the node i_t and $d^*(i_t, n^*)$ the length of the shortest path from node i_t to the sink node n^* with respect to h^* in network G_H^* .

The computation of $d^*(1^*, i_t)$ and $d^*(i_t, n^*)$, $i_t \in N_H^*$ is performed by means of the usual shortest path algorithms [1]. In network G' we rewrite the nodes $1^*, n^*$ by $1'$ respectively n' .

We obtain $N' = \{1', n'\} \cup \{i_t | i_t \in N_H, d^*(1^*, i_t) + d^*(i_t, n^*) \leq T\}$, $A' = \{(1', 1_t) | 1_t \in N_H, d^*(1_t, n^*) \leq T\} \cup \{(n_t, n') | n_t \in N_H, d^*(1^*, n_t) \leq T\} \cup \{(i_t, j_\theta) | (i_t, j_\theta) \in A_H, d^*(1^*, i_t) + h^*(i_t, j_\theta) + d^*(j_\theta, n^*) \leq T\}$ and l', u' are restriction of l_H^*, u_H^* at A' .

It is easy to see that the network G' is always a partial sub-network of G_H^* . Since an item released from a node at a specific time does not return to that location at the same or an earlier time, the networks G_H, G_H^*, G' cannot contain any circuit, and are therefore always acyclic.

In the most general dynamic model, the parameter $h(i) = 1$ is the waiting time at node i , and the parameters $l_h(i, t), u_h(i, t)$ are defined as lower bound and upper bound, which represents the minimum respectively the maximum amount of flow that can wait at node i from time t to $t+1$. This most general dynamic model is not discussed in this paper.

The maximum dynamic flow problem for T time periods in dynamic network G_h formulated in conditions (2.1), (2.2), (2.3) is equivalent with the maximum static flow problem in static network G' as follows:

$$\sum_{j_\theta} f'(i_t, j_\theta) - \sum f'(k_\tau, i_t) = \begin{cases} v', & \text{if } i_t = 1', & (2.4.a) \\ 0, & \text{if } i_t \neq 1', n', & (2.4.b) \\ -v', & \text{if } i_t = n', & (2.4.c) \end{cases}$$

$$l'(i_t, j_\theta) \leq f'(i_t, j_\theta) \leq u'(i_t, j_\theta), (i_t, j_\theta) \in A', (2.5)$$

$$\min v', (2.6)$$

where by convention $i_t = 1'$ for $t = -1$ and $i_t = n'$ for $t = T + 1$. For further details we recommend the works [2], [3], [4], [6], [7], [17].

3 The minimum flow in parametric static networks

A natural generalization of the minimum flow problem in static networks can be obtained by making the lower bounds of some arcs function of a single parameter. Since the minimum flow value function in a parametric network is a continuous piecewise linear function of the parameter, the parametric minimum flow problem can alternately be

defined as to find all the breakpoints and their corresponding minimum flow and maximum cuts. The approach presented in this section is presented in [11], the first approach for minimum flow in parametric static network.

A static network $G = (N, A, l, u)$ with the lower bounds $l(i, j)$ of some arcs $(i, j) \in A$, functions of a real parameter λ is referred to as a parametric static network and is denoted by $\bar{G} = (N, A, \bar{l}, u)$. The upper bound function $\bar{l} : A \times \mathbf{R}^+ \rightarrow \mathbf{R}^+$ is defined by the relation:

$$\bar{l}(i, j; \lambda) = l_0(i, j) + \lambda \cdot L(i, j), \lambda \in [0, \Lambda] = I (3.1)$$

where $L : A \rightarrow \mathbf{R}$ is the parametric part of the upper bound function \bar{l} and $l_0 : A \rightarrow \mathbf{R}^+$ is the non parametric part of the function \bar{l} with $\bar{l}(i, j; 0) = l_0(i, j)$, $(i, j) \in A$. The $L(i, j)$ and $l_0(i, j)$ must satisfy $-l_0(i, j)/\Lambda \leq L(i, j) \leq (u(i, j) - l_0(i, j))/\Lambda$ and $0 \leq l_0(i, j) \leq u(i, j)$, $(i, j) \in A$.

The minimum flow problem in parametric static network $\bar{G} = (N, A, \bar{l}, u)$ is to compute all minimum flows for every possible value of λ in I :

$$\sum_j \bar{f}(i, j; \lambda) - \sum_k \bar{f}(k, i; \lambda) = \begin{cases} \bar{v}(\lambda), & \text{if } i=1 & (3.2.a) \\ 0, & \text{if } i \neq 1, n & (3.2.b) \\ -\bar{v}(\lambda), & \text{if } i=n & (3.2.c) \end{cases}$$

$$\bar{l}(i, j; \lambda) \leq \bar{f}(i, j; \lambda) \leq u(i, j), (i, j) \in A, (3.3)$$

$$\min \bar{v}(\lambda) (3.4)$$

For the minimum flow problem in the parametric static network $\bar{G} = (N, A, \bar{l}, u)$, the sub-intervals $I_k = [\lambda_k, \lambda_{k+1}]$, $k = 0, 1, \dots, K$ of the parameter λ values can be determined such as a maximum $1 - n$ cut in the nonparametric static network $G_k = (N, A, l_k, u)$, $l_k(i, j) = \bar{l}(i, j; \lambda_k)$, remains the maximum $1 - n$ cut for $\lambda \in I_k$. A parametric $1 - n$ cut in parametric static network $\bar{G} = (N, A, \bar{l}, u)$ can be defined as a finite set of cuts $[S_k, T_k]$, $k = 0, 1, \dots, K$ together with a partitioning of the interval I in disjoint subintervals I_k , $k = 0, 1, \dots, K$, such that $I = I_0 \cup I_1 \cup \dots \cup I_K$. The $[S_k, T_k]$ is denoted by $[S_k; I_k]$ for each $k, k = 0, 1, \dots, K$. The capacity of $[S_k; I_k]$ is defined as:

$$\bar{c}[S_k; I_k] = \sum_{(S_k, T_k)} \bar{l}(i, j; \lambda) - \sum_{(T_k, S_k)} u(i, j), k = 0, 1, \dots, K (3.5)$$

A parametric maximum $1 - n$ cut in network G_k is denoted by $[S_k^*; I_k]$, $k = 0, 1, \dots, K$. For \bar{f} in parametric network $\bar{G} = (N, A, \bar{l}, u)$ the parametric residual capacity $\bar{r}(i, j; \lambda)$, $(i, j) \in A$ is given by:

$$\bar{r}(i, j; \lambda) = u(j, i) - \bar{f}(j, i; \lambda) + \bar{f}(i, j; \lambda) - \bar{l}(i, j; \lambda), \lambda \in I_k, k = 0, 1, \dots, K (3.6)$$

For a flow \bar{f} in parametric static network \bar{G} , we define the set $\bar{s}(i, j) = \{\lambda | \bar{r}(i, j; \lambda) > 0\}$, $(i, j) \in A$. The

static network $\tilde{G} = (N, \tilde{A}, \bar{r})$, with $\tilde{A} = \{(i, j) | (i, j) \in A, \bar{s}(i, j) \neq \phi\}$ is named the parametric residual static network. If $(i, j) \in A$ and $(i, j) \notin \tilde{A}$, then $\bar{s}(i, j) = \phi$.

Let \tilde{P} be a directed path from the source node 1 to the sink node n in the parametric residual static network \tilde{G} . If \tilde{P} verifies the restriction:

$$\bar{s}(\tilde{P}) = \bigcap_{\tilde{P}} \bar{s}(i, j) \neq \phi \tag{3.7}$$

then \tilde{P} is named conditional decreasing directed path. The parametric residual capacity of a conditional decreasing directed path \tilde{P} is $\bar{r}(\tilde{P}; \lambda) = \min\{\bar{r}(i, j; \lambda) | (i, j) \in \tilde{P}, \lambda \in \bar{s}(\tilde{P})\}$. From paper [11] we have the theorem:

Theorem 1 (11). *A flow \bar{f} is a minimum flow in parametric static network \tilde{G} if and only if the parametric residual static network \tilde{G} contains no conditional decreasing directed path \tilde{P} .*

If residual static network \tilde{G} contains no conditional decreasing path \tilde{P} , then the minimum flow in parametric network G is computed as:

$$\bar{f}(i, j; \lambda) = \bar{l}(i, j; \lambda) + \max\{\bar{r}(i, j; \lambda) - u(j, i) + \bar{l}(j, i; \lambda), 0\} \tag{3.8}$$

The first phase of finding a minimum flow in network \tilde{G} consists in establishing a feasible flow, if one exists, in non-parametric network $\hat{G} = (N, A, \hat{l}, u)$ with $\hat{l}(i, j) = l_0(i, j)$ for $L(i, j) \leq 0$ and $\hat{l}(i, j) = l_0(i, j) + \Lambda \cdot L(i, j)$ for $L(i, j) > 0$. After a nonparametric feasible flow \hat{f} (see [1]) we compute the parametric residual network \tilde{G}_0 for this flow \hat{f} .

The parametric residual capacities in \tilde{G}_0 can be written as $\bar{r}_0(i, j; \lambda) = \alpha_0(i, j) + \lambda\beta_0(i, j)$, where $\alpha_0(i, j) = u(j, i) - \hat{f}(j, i) + \hat{f}(i, j) - l_0(i, j)$ and $\beta_0(i, j) = L(i, j), \lambda \in I_0 = [0, \lambda_1]$.

The second phase of the algorithm starts with the parametric residual network $\tilde{G}_0, \lambda_0 = 0$ and $I_0 = [0, \lambda_1]$.

The algorithm for minimum flow in a parametric static network (the algorithm MFPSN) is presented in Figure 3.1.

```
(01) Algorithm MFPSN;
(02) BEGIN
(03)   compute a feasible flow  $f_0$  in network  $G_0$ ;
(04)   compute parametric residual network  $\tilde{G}_0$ ;
(05)   B:={0}; k := 0;  $\lambda_k := 0$ ;
(06)   REPEAT
(07)     SDDP( k,  $\lambda_k$ , B);
(08)     k:=k+1;
(09)   UNTIL ( $\lambda_k = \Lambda$ );
(10) END.
```

Figure 3.1.a. The algorithm for the minimum flow in parametric static network

```
(01) PROCEDURE SDDP( k,  $\lambda_k$ , B);
(02) BEGIN
(03)   compute the network  $\tilde{G}_k$ ;
(04)   compute exact distance labels  $\tilde{d}(i)$  in  $\tilde{G}_k$ ;
(05)   p:=( n+1, ..., n+1);  $\alpha_k(\tilde{P}) := 0$ ;  $\beta_k(\tilde{P}) := 0$ ;
(06)    $\lambda_{k+1} := \Lambda$ ; j:=n;
(07)   WHILE  $\tilde{d}(n) < n$  DO
(08)     BEGIN
(09)       IF(exists an admissible arc (i, j))
(10)         THEN BEGIN
(11)           p(i):=j;
(12)           j:=i;
(13)           IF (j=1)
(14)             THEN BEGIN
(15)               RCDADP(p,  $\lambda_{k+1}$ , B,  $\alpha_k(\tilde{P})$ ,  $\beta_k(\tilde{P})$ );
(16)               j:=n;
(17)             END;
(18)           END
(19)         ELSE BEGIN
(20)            $\tilde{d}(j) := \min\{\tilde{d}(i) + 1 | (i, j) \in \tilde{A}_k\}$ ;
(21)           IF j  $\neq$  n
(22)             THEN j:= p(j);
(23)           END;
(24)         END;
(25)       compute the flow  $\bar{f}_k$ ;
(26)       add  $\lambda_{k+1}$  to the list B;
(27)     END;
```

Figure 3.1.b. The Procedure Shortest Decreasing Directed Path (SDDP)

```
(01) PROCEDURE RCDADP(p,  $\lambda_{k+1}$ , B,  $\alpha_k(\tilde{P}), \beta_k(\tilde{P})$ );
(02) BEGIN
(03)   compute  $\tilde{P}$  based on p;
(04)    $\alpha_k(\tilde{P}) := \min\{\alpha_k(i, j) | (i, j) \in \tilde{P}\}$ ;
(05)    $\beta_k(\tilde{P}) := \min\{\beta_k(i, j) | (i, j) \in \tilde{P}\}$ ;
(06)   i:=1;
(07)   WHILE i  $\neq$  n DO;
(08)     BEGIN
(09)       IF( $\beta_k(p(i), i) < \beta_k(\tilde{P})$ )
(10)         THEN BEGIN
(11)            $\lambda' := \lambda_k + (\alpha_k(i, p(i)) - \alpha_k(\tilde{P})) / (\beta_k(\tilde{P}) - \beta_k(i, p(i)))$ ;
(12)           IF ( $\lambda' < \lambda_{k+1}$ )
(13)             THEN  $\lambda_{k+1} := \lambda'$ ;
(14)           END;
(15)            $\alpha_k(i, p(i)) := \alpha_k(i, p(i)) - \alpha_k(\tilde{P})$ ;
(16)            $\beta_k(i, p(i)) := \beta_k(i, p(i)) - \beta_k(\tilde{P})$ ;
(17)            $\alpha_k(p(i), i) := \alpha_k(p(i), i) + \alpha_k(\tilde{P})$ ;
(18)            $\beta_k(p(i), i) := \beta_k(p(i), i) + \beta_k(\tilde{P})$ ;
(19)         END;
(20)       i:=p(i);
(21)     END;
```

Figure 3.1.c. The Procedure Residual Capacity of Conditional Decreasing Directed Path(RCDADP).

In order to avoid working with piecewise linear functions, the algorithm MFPSN works in parametric residual static networks defined for sub-intervals of the parameter

values where the parametric residual capacities of all arcs remain linear functions. The parametric residual network \tilde{G} for a sub-interval $I_k = [\lambda_k, \lambda_{k+1}]$ is denoted by \tilde{G}_k . In k -th step of the algorithm MFPSN, the SDDP procedure computes the parametric residual static network \tilde{G}_k where $\bar{r}_k(i, j; \lambda) = \alpha_k(i, j) + (\lambda - \lambda_k) \cdot \beta_k(i, j)$, with $\alpha_k(i, j) = \alpha_0(i, j) + \lambda_k \beta_0(i, j)$, $\beta_k(i, j) = \beta_0(i, j)$ and computes the shortest decreasing directed path \tilde{P} in network \tilde{G}_k . The RCDADP procedure computes $\bar{r}_k(\tilde{P}, \lambda), \lambda_{k+1}$ and alters $\alpha_k(i, p(i)), \beta_k(i, p(i)), \alpha_k(p(i), i), \beta_k(p(i), i)$. Obviously in \tilde{G}_k we have $\bar{r}_k(\tilde{P}, \lambda) = \alpha_k(\tilde{P}) + (\lambda - \lambda_k) \cdot \beta_k(\tilde{P})$.

In paper [11] the following two theorems are presented.

Theorem 2 (Theorem of Correctness). *The algorithm MFPSN computes correctly a minimum flow in parametric static network $G = (N, A, \bar{l}, u)$ and $\lambda \in I = [0, \Lambda]$.*

Theorem 3 (Theorem of Complexity). *The algorithm MFPSN runs in $O(Kn^2m)$ time, where $K+1$ is the number of λ value in the set B at the end of the algorithm.*

For further details of this section we recommend the work [11].

We remark the fact that the maximum flow problem in the nonparametric static network can be solved in $O(nm)$ time (see [10]). In this case the algorithm MFPSN runs in $O(Knm)$ time.

4 The minimum flow in parametric dynamic networks

A dynamic network $G_h = (N, A, h, l_h, u_h)$ for which the lower bounds $l_h(i, j; t)$ of some arcs $(i, j) \in A$ are functions of a real parameter $\lambda \in I = [0, \Lambda]$ is referred to as a parametric dynamic network and is denoted by $\tilde{G}_h = (N, A, h, \bar{l}_h, u_h)$. The parametric lower bound function $\bar{l}_h : A \times H \times I \rightarrow \mathbf{R}^+$ is defined by the relation:

$$\bar{l}_h(i, j; t; \lambda) = l_{0h}(i, j; t) + \lambda \cdot L_h(i, j; t), \quad (i, j) \in A, t \in H, \lambda \in I, \tag{4.1}$$

The parametric part $L_h(i, j; t)$ and the $l_{0h}(i, j; t)$, must satisfy the constraints:

$$\begin{aligned} -l_{0h}(i, j; t)/\Lambda &\leq L_h(i, j; t) \\ L_h(i, j; t) &\leq (u_h(i, j; t) - l_{0h}(i, j; t))/\Lambda, \text{ and} \\ 0 &\leq l_{0h}(i, j; t) \leq u_h(i, j; t), \\ (i, j) &\in A, t \in H. \end{aligned}$$

The minimum flow problem in parametric dynamic network $G_h = (N, A, h, \bar{l}_h, u_h)$ is to compute flow function $f_h : A \times H \times I \rightarrow \mathbf{R}^+$ that satisfies the following constraints:

$$\sum_{i=0}^T (\sum_j \bar{f}(i, j; t; \lambda) - \sum_k \sum_\tau \bar{f}_h(k, i; \tau; \lambda)) = \bar{v}_H(\lambda), \quad (4.4.a)$$

$$\sum_{i \neq 1, n, t \in H} \bar{f}(i, j; t; \lambda) - \sum_k \sum_\tau \bar{f}_h(k, i; \tau; \lambda) = 0, \tag{4.2.b}$$

$$\sum_{i=n}^T (\sum_j \bar{f}(i, j; t; \lambda) - \sum_k \sum_\tau \bar{f}_h(k, i; \tau; \lambda)) = -\bar{v}_H(\lambda), \tag{4.2.c}$$

$$\bar{l}_h(i, j; t; \lambda) \leq \bar{f}_h(i, j; t; \lambda) \leq u_h(i, j; t), \quad (i, j) \in A, t \in H, \lambda \in I \tag{4.3}$$

$$\min \bar{v}_H(\lambda), \lambda \in I \tag{4.4}$$

where $\bar{f}_h(i, j; t; \lambda) = 0, (i, j) \in A, t \in \{T - h(i, j; t) + 1, \dots, T\}, \lambda \in I$.

In network $\tilde{G}_h = (N, A, h, \bar{l}_h, u_h)$ we consider the following assumption: if $(i, j) \in A$ then $(j, i) \in A$. This assumption is non-restrictive because if $(i, j) \in A$ and $(j, i) \notin A$ we consider that $(j, i) \in A$ with $\bar{l}_h(j, i; \theta; \lambda) = u_h(j, i; \theta) = 0, \theta = t + h(i, j; t), t \in H, \lambda \in I, h(j, i; \theta) = -h(i, j; t)$, if $0 \leq t \leq T - h(i, j; t)$ and $h(j, i; \theta) = \infty$, if $T - h(i, j; t) + 1 \leq t \leq T$.

The parametric dynamic residual capacities with respect to given flow f_h are defined as follow:

$$\begin{aligned} \bar{r}_h(i, j; t; \lambda) &= u_h(j, i; t) - \bar{f}_h(j, i; t; \lambda) + \bar{f}_h(i, j; t; \lambda) - \bar{l}_h(i, j; t; \lambda), \\ (i, j) &\in A, t \in H, \lambda \in I \end{aligned} \tag{4.5}$$

For the minimum flow in parametric dynamic network $\tilde{G}_h = (N, A, h, \bar{l}_h, u_h)$ with respect to given flow f_h is defined as $\tilde{G}_h = (N, \tilde{A}, \tilde{r}_h)$, where $\tilde{A} = \{(i, j) | (i, j) \in A, \bar{r}_h(i, j; t; \lambda) > 0, t \in H, \lambda \in I\}$.

In this paper, the proposed approach consists in applying the algorithm MFPSN presented in Section 3 in parametric static reduced expanded network $\tilde{G}' = (N', A', \bar{l}', u')$ which is constructed similar with construction of the network $G' = (N', A', l', u')$ presented in Section 2.

The algorithm for the minimum flow in parametric dynamic network (the algorithm MFDPN) is presented in Figure 4.1.

- (1) ALGORITHM MFDPN;
- (2) BEGIN
- (3) construct the network \tilde{G}' ;
- (4) apply the algorithm MFPSN in network \tilde{G}' ;
- (5) END.

Figure 4.1. The algorithm for minimum flow in parametric dynamic network.

Theorem 4 (Theorem of Correctness). *The algorithm MFDPN computes correctly a minimum flow in parametric dynamic network $G_h = (N, A, h, \bar{l}_h, u_h)$ and $\lambda \in I$.*

Proof. This theorem results from the fact that the minimum flow in parametric dynamic network with lower bounds $\tilde{G}_h = (N, A, h, \bar{l}_h, u_h)$ is equivalent with the maximum flow in parametric static network $\tilde{G}' = (N', A', l', u')$ and Theorem 2. □

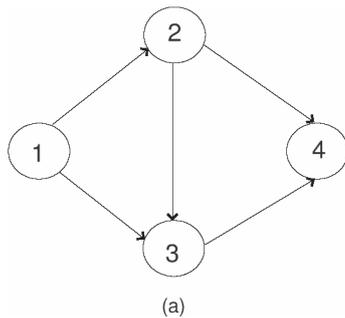
Theorem 5 (Theorem of Complexity). *The algorithm MF-PDN runs in $O(KT^3n^2m)$ time, where $K+1$ is the number for λ values in the set B at the end of the algorithm.*

Proof. From Theorem 3 results that the algorithm MFPSN runs in $O(K(n'_H)^2m'_H)$ time. From Section 2 we obtain $n' = O(nT)$ and $m' = O(mT)$. Therefore we obtain that algorithm MF-PDN runs in $O(KT^3n^2m)$ time. \square

In accordance with the remark in Theorem 3, we note that the algorithm MF-PDN runs in $O(KnmT^2)$ time.

5 Example

The support parametric dynamic network is presented in Figure 5.1.(a) and the time horizon is set to $T = 3$, therefore $H=\{0, 1, 2, 3\}$. The transit times $h(i, j; t)$ and the dynamic upper bounds (capacities) $u_h(i, j; t)$, as well as the parametric dynamic lower bounds $l_h(i, j; t; \lambda) = l_{0h}(i, j; t) + \lambda \cdot L_h(i, j; t)$ the for all arcs in \tilde{G}_h are indicated in the two tables in Figure 5.1.b. The interval of parameter λ values is set to $[0, 1]$, i.e., $\Lambda=1$.



(i, j)	$h(i, j; t)$	$U_h(i, j; t)$
(1, 2)	$1, t = 0$	5
	$2, t = 1, 2, 3$	
(1, 3)	$1, t = 0, 1$	5
	$2, t = 2, 3$	
(2, 3)	$1, t = 0, 1, 2, 3$	5
(2, 4)	$1, t = 0, 1$	5
	$2, t = 2, 3$	
(3, 4)	$2, t = 0, 1$	5
	$1, t = 2, 3$	

(i, j)	$l_{0h}(i, j; t)$	$L_h(i, j; t)$
(1, 2)	$3, t = 0$	$-2, t = 0$
	$0, t = 1, 2, 3$	$0, t = 1, 2, 3$
(1, 3)	$1, t = 0, 1$	$4, t = 0$
	$0, t = 2, 3$	$1, t = 1$
		$0, t = 2, 3$
(2, 3)	$0, t = 0, 1, 2, 3$	$3, t = 1$
(2, 4)	$0, t = 0, 1, 2, 3$	$0, t = 0, 2, 3$
		$0, t = 0, 1, 2, 3$
(3, 4)	$0, t = 0$	$-2, t = 2$
	$2, t = 1, 2$	$0, t = 0, 1, 3$
	$0, t = 3$	

(b)

Figure 5.1. The parametric dynamic network \tilde{G}_h .

The support graph for parametric super-extended network \tilde{G}_H^* is presented in Figure 5.2.

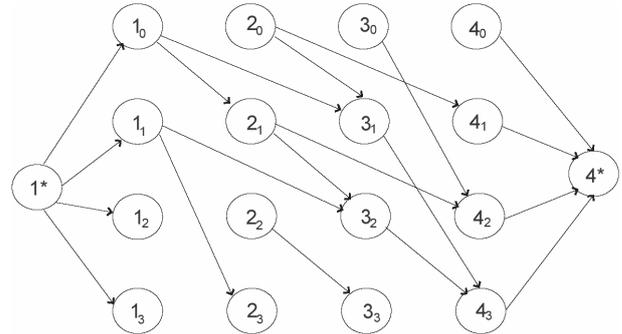


Figure 5.2. The support graph for parametric superextended network \tilde{G}_H^* .

The support graph for parametric reduced expanded network \tilde{G}' is showed in Figure 5.3.

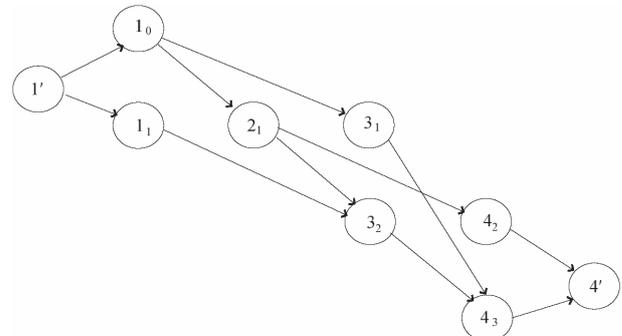


Figure 5.3. The support graph for parametric reduced expanded network \tilde{G}' .

The lower bounds $\bar{l}'(i_t, j_\theta; \lambda) = l'(i_t, j_\theta) + \lambda \cdot L'(i_t, j_\theta)$ and the parametric upper bounds $u'(i_t, j_\theta)$ for all arcs in \tilde{G}' are indicated in table from Figure 5.4.

(i_t, j_θ)	$l'(i_t, j_\theta)$	$L'(i_t, j_\theta)$	$u'(i_t, j_\theta)$	$\hat{f}'(i_t, j_\theta)$
$(1', 1_0)$	0	0	∞	10
$(1', 1_1)$	0	0	∞	2
$(1_0, 2_1)$	3	-2	5	5
$(1_0, 3_1)$	1	3	5	5
$(1_1, 3_2)$	1	1	5	2
$(2_1, 3_2)$	0	3	5	3
$(2_1, 4_2)$	0	0	5	2
$(3_1, 4_3)$	2	0	5	5
$(3_2, 4_3)$	2	-2	5	5
$(4_2, 4')$	0	0	∞	2
$(4_3, 4')$	0	0	∞	10

Figure 5.4. The $\bar{l}'(i_t, j_\theta; \lambda) = l'(i_t, j_\theta) + \lambda L'(i_t, j_\theta)$, $u'(i_t, j_\theta)$ and $\hat{f}'(i_t, j_\theta)$ in network \tilde{G}' .

In the first phase we determine in static residual network \tilde{G}' : $\tilde{P}'_1 = (1', 1_0, 2_1, 4_2, 4')$, $\hat{r}'(\tilde{P}'_1) = 2$; $\tilde{P}'_2 = (1', 1_0, 3_1, 4_3, 4')$, $\hat{r}'(\tilde{P}'_2) = 5$; $\tilde{P}'_3 = (1', 1_1, 3_2, 4_3, 4')$, $\hat{r}'(\tilde{P}'_3) = 2$; $\tilde{P}'_4 = (1', 1_0, 2_1, 3_2, 4_3, 4')$, $\hat{r}'(\tilde{P}'_4) = 3$. The feasible flow \hat{f}' in network \tilde{G}' is presented in the table from Figure 5.4.

In the second phase by applying algorithm MF-PSN in network \tilde{G}' we obtain in the parametric static residual network \tilde{G}' the following directed path: $\tilde{P}'_1 = (1', 1_1, 3_2, 2_1, 4_2, 4')$, $\tilde{P}'_2 = (1', 1_0, 2_1, 4_2, 4')$, $\tilde{P}'_3 = (1', 1_0, 3_1, 4_3, 4')$, $\tilde{P}'_4 = (1', 1_0, 2_1, 3_2, 4_3, 4')$ in $\tilde{G}'_0, \tilde{G}'_1, \tilde{G}'_2$. The results of this example are synthetically indicated in the table from Figure 5.5. The graphic of $\bar{v}'(\lambda)$ is presented in Figure 5.6.

k	λ_k	\tilde{P}'_i	$\bar{r}_k(\tilde{P}'_i)$	λ_{k+1}	$\bar{v}'(\lambda)$
0	0	\tilde{P}'_1	$1 - \lambda$	1	$6 - \lambda$
		\tilde{P}'_2	$1 + \lambda$	1	
		\tilde{P}'_3	3	1/4	
		\tilde{P}'_4	$1 + \lambda$	1/4	
1	1/4	\tilde{P}'_1	$1 - \lambda$	1	$5 + 3\lambda$
		\tilde{P}'_2	$1 + \lambda$	1	
		\tilde{P}'_3	$4 - 4\lambda$	1	
		\tilde{P}'_4	$1 + \lambda$	3/5	
2	3/5	\tilde{P}'_1	$1 + \lambda$	1	$2 + 8\lambda$
		\tilde{P}'_2	$1 + \lambda$	1	
		\tilde{P}'_3	$4 - 4\lambda$	1	
		\tilde{P}'_4	$4 - 4\lambda$	1	

Figure 5.5. Results of applying the algorithm in \tilde{G}' .

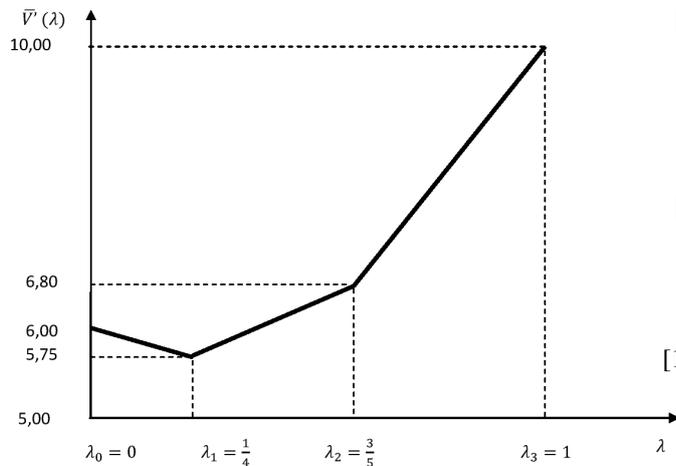


Figure 5.6. The graphic of $\bar{v}'(\lambda)$

References

[1] Ahuja R., Magnanti T., Orlin J. (1993) *Network Flows. Theory, algorithms and applications*, Prentice Hall, Inc., Englewood Cliffs, New Jersey.

[2] Aronson J.A. (1989) A survey of dynamic networks flows, *Annals of Operations Research*, **20** (5), pp.1–66. <https://doi.org/10.1007/bf02216922>

[3] Avesalon N., Ciurea E., Parpalea M. (2017), Maximum parametric flow in discrete-time dynamic networks, *Fundamenta Informaticae*, **156** (2), pp.125–139. <https://doi.org/10.1007/bf02216922>

[4] Cai X., Sha D., Wong C. (2007) *Time-varying Network Optimization*, Springer.

[5] Ciurea E., Ciupală L. (2004) Sequential and parallel algorithms for minimum flow, *Journal of Applied Mathematics and Computing*, **15**(1-2), pp.53–75. <https://doi.org/10.1007/bf02935746>

[6] Ciurea E. (2002) Second best temporally repeated flows, *The Korean Journal of Computational and Applied Mathematics*, **9**(1), pp.77–86. <https://doi.org/10.1007/bf03012341>

[7] He X., Zheng H., Peeta S. (2015) Model and solution algorithm for the dynamic resource allocation problem for large-scale transportation network evacuation, *Transportation Research Part C.: Emerging Technologies*, **59**, pp.233–247. <https://doi.org/10.1016/j.trc.2015.05.005>

[8] Miller-Hooks E., Patterson S.S. (2004) On solving quickest time problems in time-dependent dynamic networks, *Journal of Mathematical Modelling and Algorithms*, **3**, pp.39–71. <https://doi.org/10.1023/b:jmma.0000026708.57419.6d>

[9] Nasrabadi E., Hashemi S.M. (2010) Minimum cost time-varying network flow problems, *Optimization Methods and Software*, **25** (3), pp.429–447. <https://doi.org/10.1080/10556780903239121>

[10] Orlin J. (2013) Max Flows in $O(nm)$ time, or better, *Proceeding of the forty-fifth Annual ACM Symposium on Theory of Computing*, ACM Press, New York, pp.765–774. <https://doi.org/10.1145/2488608.2488705>

[11] Parpalea M., Ciurea E. (2016) Minimum parametric flow. A partitioning approach, *British Journal of Applied Science and Technology*, **13** (6), pp.1–8. <https://doi.org/10.9734/bjast/2016/22636>

[12] Parpalea M., Ciurea E. (2011) The quickest maximum dynamic flow of minimum cost, *International Journal of Applied Mathematics and Informatics*, **3** (5), pp.266–274.

[13] Parpalea M., Ciurea E. (2013), Partitioning algorithm for the parametric maximum flow, *Applied Mathematics*, **4** (10A), pp.3–10. <https://doi.org/10.4236/am.2013.410a1002>

- [14] Parpalea M., Avesalon N., Ciurea E. (2018) Minimum parametric flow in time dependent dynamic networks, *Revue d'Automatique, d'Informatique et de Recherche Opérationnelle - Theoretical Informatics and Applications (RAIRO: ITA)*, **52** (1), pp.43–53.<https://doi.org/10.1051/ita/2018002>
- [15] Rashidi H., Tsang E. (2015) *Vehicle Scheduling in Port Automation: Advanced Algorithms for Minimum Cost Flow Problems*, CRC Press, 2 edition, Boca Raton, London, New York.<https://doi.org/10.1201/b18984>
- [16] Ruhe G. (1991) *Algorithmic Aspects of Flows in Networks*, Kluwer Academic Publisher, Dordrecht, The Netherlands.<https://doi.org/10.1007/978-94-011-3444-6>
- [17] Zheng H., Chiu Y.C., Mirchandani P.B. (2015) On the System Optimum Dynamic Traffic Assignment and Earliest Arrival Flow Problems, *Transportation Science*, **49**, pp.13–27.<https://doi.org/10.1287/trsc.2013.0485>

Investigating Algorithmic Stock Market Trading Using Ensemble Machine Learning Methods

Ramzi Saifan, Khaled Sharif, Mohammad Abu-Ghazaleh, and Mohammad Abdel-Majeed

Computer Engineering Department, School of Engineering, University of Jordan, Queen Rania Street, Amman, Jordan

E-mail: r.saifan@ju.edu.jo, kldsr@gmail.com, mohd.ag@live.com, m.abdel-majeed@ju.edu.jo

Keywords: machine learning, stock price prediction, ensemble methods, gradient boosting, extremely randomized trees, random forest, stock market simulation, algorithmic trading, financial forecast, forecasting returns, risk analysis, volatility forecasting

Received: July 21, 2019

Recent advances in the machine learning field have given rise to efficient ensemble methods that accurately forecast time-series. In this paper, we use the Quantopian algorithmic stock market trading simulator to assess ensemble methods performance in daily prediction and trading. The ensemble methods used are Extremely Randomized Trees, Random Forest, and Gradient Boosting. All methods are trained using multiple technical indicators and automatic stock selection is used. Simulation results show significant returns relative to the benchmark and large values of alpha are produced from all methods. These results strengthen the role of ensemble method based machine learning in automated stock market trading.

Povzetek: Razvit je nov algoritem za napovedovanje delnic s pomočjo ansambla programov za strojno učenje.

1 Introduction

Predicting the stock market has been the ultimate goal of stock investors since its existence. Everyday billions of dollars are traded in stock markets around the world, and behind each dollar is an investor hoping to profit by correctly forecasting the rise or fall of the associated stock price.

If an investor somehow predicts that a stock price will rise, he will buy a certain amount of that stock, wait for a specified period of time, and then sell those stocks at their increased price; this method of trading is referred to as *longing*. It is also possible for the investor to profit from the decrease of a stock through a different process called *shorting*; this is when the investor predicts that a stock will fall, borrows a certain amount of that stock and sells them, buys the same amount of stocks after their price has decreased, then returns the stocks he has borrowed to the lender.

Longing and shorting stocks combined with an accurate way of stock market price forecasting makes it possible for an investor to profit from any change in the stock market. This creates a dire need for strong prediction methods. There are various ways for stock price prediction; they basically fall into two categories, either Fundamental Analysis (FA) or Technical Analysis (TA). Many experts use a combination of the two for finer predictions.

For decades, investors have been using a human-based prediction method called *fundamental analysis* (FA); this technique involves acquiring all the relevant information that a person can collect about a certain stock in order to determine its “*true value*”. It goes into the economics of the company itself, such as sales and

profit data. External factors are also taken into consideration, such as politics, regulations, and industry trends [1]. Methods that aid an investor in FA include financial statements, asset ratios ^[1], liquidity ratios ^[2], debt ratios ^[3], market value ratios, and portfolio management [2]. Based on the determined true value, the investor will decide what sort of position to take with the stock; if it is overpriced the investor will *short* the stock, or *long* the stock if it is underpriced, under the belief of the investor that the price will eventually fall or rise respectively to meet its *true value*). One of the limitations of FA is that it has been practiced for decades without any unifying theoretical framework [3]. Since it lacks a solid mathematical foundation, there is an emotional factor that may cause the investor to make the wrong decisions.

The second method used is called *technical analysis* (TA). It is a method that does not take into account anything about the company, because the investor is interested only in short term movements in the stock price. It concentrates on the movement of stock prices; by examining past stock price movements, future stock price can be accurately predicted. Investors that use TA believe that all the information you need to know about a stock, and the stock prices future movement, is embedded in its historical data. Based on visual examination of the historical data such as price changes and volume of transactions, usually in graphical form and charts, trading advice can be provided [4]. The volume of a stock is the total number of shares that are traded in a security during a certain period of time, and a security with higher volume means that it is more active. TA can use the fluctuations in a stock’s volume and price over a certain period of time to try to determine the future movement of the price.

With new advances in technology, and the emergence of high speed computing, computer programs can

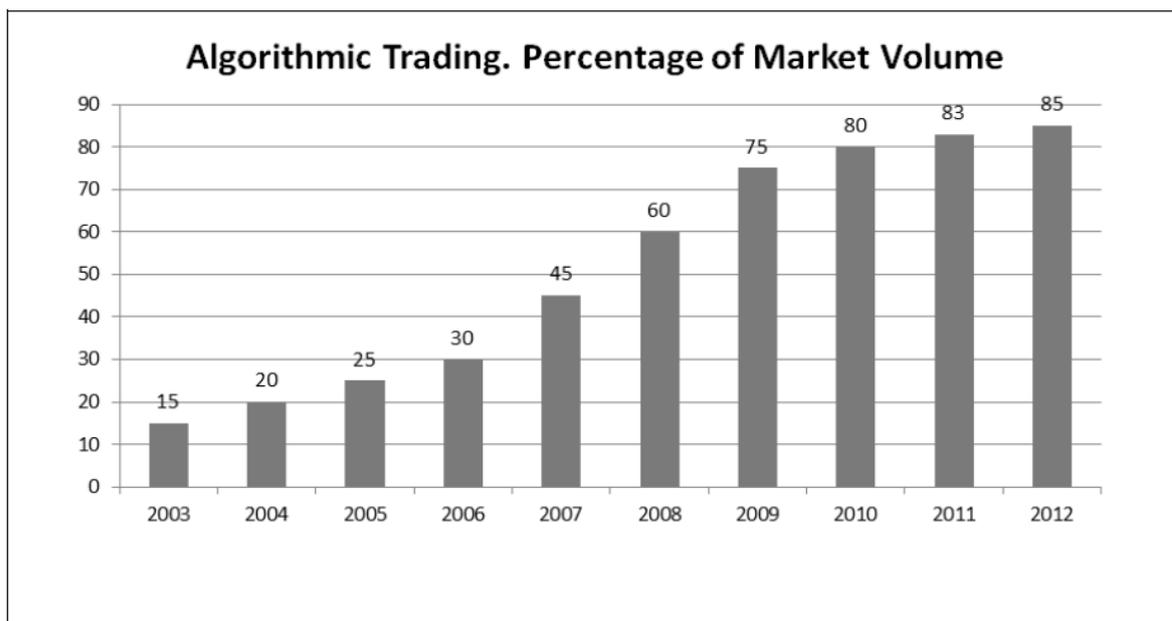


Figure 1: Increase of algorithmic trading as a percentage of the New York Stock Market's volume [6].

automatically run complex TA methods on big amounts of historical data and automatically trade stocks based on the program's inferred predictions. This entire workflow is known in the financial industry as *algorithmic trading* (AT) [5]. AT has revolutionized the market and the way financial assets are traded after it became popular in the early 2000s. Investors wanted to make sure to use all the tools that can be offered from the increasing technological advancements, which will place them in a better position to address the changing market environment [6]. Figure 1 shows the trend of using AT through years 2003 to 2012. These estimates include even investors that do not directly deal with the AT program, but deal with a stock broker who eventually will use an AT program to place the order on the stocks required.

The concept of automated prediction is known in the world of computer science as *machine learning* (ML), and is a term that relates to the construction of algorithms that can learn from and make predictions on data. The emergence of strong machine learning methods that can accurately identify stock market patterns and predict the future movement of a stock has led to a surge in research in AT based on ML methods.

The increasing usage of AT makes perfect sense, and this can be accredited to multiple reasons: firstly, the use of AT completely eliminates any emotional and psychological factors that might affect any trade undertaken by an investor. Secondly, placing orders through an AT system occurs instantly with precision and accuracy. Thirdly, AT allows the investor to monitor huge amounts of stock market and financial data in real-time, without the risk of manual human errors. Lastly, because of the programmatic nature of AT systems, simulating an algorithm on large amounts of historical data provides

relatively accurate ¹ indication of portfolio performance. A paper published by Hendershott et. al. (2007) studied the effect of AT on the New York Stock Exchange (NYSE). In it they concluded that AT most likely causes an improvement in market liquidity [7]. Another study on the foreign exchange market concluded through evidence that due to AT programs being highly correlated to each other, the use of AT had reduced volatility in the stock market [8].

The use of ML in AT has been met with resistance by economists due to three main reasons: firstly, the complexity of ML methods from the perspective of fields other than computer science; secondly, the random nature of a machine learning method and the inconsistency in its prediction results; thirdly, the insufficient amount of published academic work (in the area of stock market prediction) that include AT simulations showing the predictions being undertaken in live trading.

However, the ML methods currently being investigated rarely perform well enough (i.e.: make enough accurate predictions to be considered profitable) for them to be used in real trading situations. Existing methods also suffer from low returns over long trading periods, making them less attractive to traders when compared to existing algorithms reliant on human predictions.

The problem with currently published research that attempts to investigate AT that uses ML for prediction is either the results are undesirable, or that no simulation is included in the results, or both. This lack of research is, in the opinion of the authors, the main reason stopping the widespread use of machine learning prediction in stock market trading today. This paper will thoroughly investigate using efficient ML techniques to accurately

¹ When compared to manual investing approaches, algorithmic trading is more likely to produce a similarly performing result given the same data, and this is because

it depends on a series of steps rather than an investor's intuition.

predict the future movement of a stock, taking into account the three reasons for resistance mentioned above. It is therefore the chief goal of this paper to encourage the economic world to undertake AT using new ML methods, by providing them with solid, consistent, and repeatable simulations.

In this paper, we will focus on three new ML methods, namely Gradient Boosting, Random Forest, and Extremely Randomized Trees; we have chosen these methods because they have all been published recently in the ML world. Moreover, these methods have been tested before on time series prediction and have shown accurate prediction results even with noisy data (i.e.: data that fluctuates randomly) and very large datasets (i.e.: datasets that are too large for weaker ML methods to work on in sufficient time).

To simulate these ML methods in AT, we will use Quantopian, a browser-based AT platform that can be used to write trading strategies in Python [9] and back-test them against 13 years of minute-level US stock price and fundamental data. In each simulation, the returns of the algorithm ⁽⁴⁾ are compared with a suitable benchmark, and performance is evaluated according to eight evaluation methods. Our simulation results will prove to the readers that using our suggested ML methods in AT will consistently provide better revenue than the benchmark.

In the Literature Review section, we will review the state of the art literature and academic research that revolves around AT, and the application of ML methods into AT. In the Trading Strategy section, we will discuss how the ML model is created and trained, how stocks are automatically selected during the AT process, and briefly go over some simulator settings. In the Methodology section, we will go over the performance indicators that will be used to judge how well the ML methods perform relative to the performance of known financial benchmarks. Finally, in the Results section we will compare and comment on the simulation results of the ML methods when using Quantopian.

2 Literature review

While academic journals are filled with projects discussing stock trading techniques [10] [11] [12], the world of algorithmic trading is relatively new, and therefore the application of machine learning to algorithmic trading is the new trend of academic research [13] [5]. The three machine learning techniques, interchangeably referred to as classifiers ⁽⁵⁾, we will be using are the Gradient Boosting [14], Random Forests [15], and Extremely Randomized Trees algorithms [16].

The Gradient Boosting algorithm produces a prediction model that is in the form of an ensemble ⁽⁶⁾ of weak decision tree prediction models, also known as *estimators* [17]. The Random Forest and Gradient Boosting algorithms are much related, because both of the algorithms are techniques for regression and classification problems by constructing a multitude of decision trees [18]. The Random Forest algorithm is easier to tune than the Gradient Boosting algorithm, although the Gradient Boosting algorithm will, in general, outperform Random

Forests with proper tuning. This is because the Gradient Boosting algorithm attempts to add new trees that complement the already built trees, and usually this produces better accuracy with fewer trees. The Extremely Randomized Trees algorithm is one step further than the Random Forests algorithm in the way it chooses to split each node in the decision tree during the construction of the decision tree and how the parameters for the node is computed. [19]

Algorithmic trading (AT) is using the computational power at our disposal in the stock market. Computers programmed with a specific set of instructions, large amounts of data, and mathematical models that decide how to trade in a speed and frequency that humans are not capable of achieving, in order to generate more profit ruling out human errors and emotions. Multiple studies show the effect of algorithmic trading on the stock market. A study was done from 2001 to 2011 on the stock market and how AT affects it and it showed that it improved liquidity, efficiency, but also increased volatility [20]. However, a paper showed that results were not uniform across different stocks and there were different outcomes under different conditions [21].

Machine learning (ML) has been a hot topic between researchers for its use in a lot of fields. We are concerned with it being used in the stock market to assist investors in trading, by trying to predict the behavior of the stock market through computations of large amounts of historical stock market data. A considerable amount of effort was also put into using Neural Networks as a prediction technique. One of the first papers that attempted to apply that to the stock market was used to predict the index of the Tokyo Stock Market [22]. A much recent paper about Neural Networks used two kinds of neural networks, namely a feed forward Multilayer Perception (MLP) and an Elman recurrent network [23]. The paper concluded that MLP has more potential in predicting stock value changes than Elman recurrent network and linear regression, although a simple linear regression model was better than the other two when it comes to predicting the direction of stock price changes one day ahead [24]. The authors of [25] proposed a trading agent that is based on deep reinforcement learning, to autonomously make trading decisions and gain profits in the dynamic financial markets. [26] paper developed a machine learning framework for algorithmic trading with virtual bids in electricity markets. Also, a budget and risk constrained portfolio optimization problem was solved.

Another paper proposed a model that combined the Support Vector Machine algorithm with other classification methods, in a way such that the weakness of a method will be balanced out by the strength of another (i.e., early attempts at ensemble methods in stock market prediction) [27]. Papers that used technical indicators for their machine learning methods typically computed the Exponential Moving Average (EMA) ⁽⁷⁾ and compared it to the stock markets, specifically using the Google and Yahoo stocks (NYSE: GOOG and NASDAQ: YHOO); in one particular paper, the authors suggested using other indicators as they believe that might provide more accurate results instead of just using the EMA [28].

Results from papers in the field have been both positive and negative towards the idea of using ML in AT. An example of a paper that was negative towards the idea used ML to facilitate *automated stock portfolio optimization*; the authors used the Dow Jones Industrial Average Index as a benchmark; they concluded that none of the techniques they used outperform the index, mainly because the index resulted in more returns at a lower risk than their proposed method [29]. An example of another paper that was positive towards the idea used a method that consisted of linear regression, generalized linear model, with the aid of the Support Vector Machine algorithm, to predict future stock market prices; results were desirable and they generated a higher profit than the selected benchmark [30]. Another positive paper proposed a stock price prediction system also based on the Support Vector Machine algorithm and was tested on the Taiwan stock market; the method performed better than conventional stock market prediction systems (in terms of accuracy) [31].

More advanced papers have used hybrid combinatorial methods of clustering⁽⁸⁾ and classification. One of these papers first applies a clustering algorithm such as K-Nearest Neighbors and partitions the clustered values into number of parties, and then applies a horizontal partition based decision tree algorithm; the paper used the algorithm on data from the Shanghai Stock Exchange and their predicted results were very close to the actual values [32].

In this project, we will compare our efficient ensemble methods with the K-Nearest Neighbors and the Support Vector Machine algorithms as a way of comparing our methods to those used in previous literature. Our simulation results will show that our efficient ensemble methods outperform those used in previous literature in predictive accuracy.

The use of Quantopian in academic research is rare; one of the few papers to use it begins with an explanation of the Efficient Market Hypothesis² and Self-Defeating Strategies³, and uses these two ideas to reason why there aren't enough academic papers showing positive results predicting the market using machine learning; in the author's opinion, if a model succeeds and is distributed to the public, it will not be successful for too long. The author also used different methods of trading using Quantopian and machine learning, but showed that results were undesirable [33].

As we can see from the aforementioned literature, there have been many different techniques tried and tested in an attempt to predict the stock market and automate stock market trading. All methods used different algorithms, factors, and parameters that could be tuned to

deliver better results. In this paper, we will use what we consider to be the latest machine learning methods to try and produce positive results in prediction and simulation.

3 Trading strategy

3.1 Model creation

In this section, we will explain the trading strategy that we will simulate. It is coded entirely in the Python language and it runs on the Quantopian simulator. As mentioned earlier, we will use three machine learning methods for our daily predictions. The classifiers used are the Gradient Boosting, Extremely Randomized Trees, and Random Forest classifiers, and they are all part of the open source *scikit-learn* library [34].

Creating the model is the first step in the algorithm, and the model creation is scheduled to happen at the beginning of every month throughout the simulation period. It is created by training the classifier⁴ data based on the previous 1000 days (which we define as the history range) relative to the model creation date, and based on this data we generate features, namely the Average True Range (ATR) and the Bollinger Bands (BB). The ATR is a measure of the volatility (volatility is defined later in Section 4.2) for the stocks: it is calculated through a 14-day period by finding the moving average of the “true range”. Simply put, if stocks are experiencing high volatility, then they would have higher ATR, and they will have lower ATR at lower volatility, and the difference between the maximum and minimum moving average is deemed the true range. The BB is another popular method to measure volatility: the prices of the stock along with a ten-day period moving average are banded by an upper band and a lower band, and the bands keep changing according to the market conditions. A wider band from the moving average means that the stock price is becoming more volatile, whereas tighter bands mean that the volatility is decreasing. If stock price moves closer to the upper band, this means that the stock is being overbought, and the stock is being oversold if the prices are moving closer to the lower band.

The following list outlines the organization of the features and the predicted target, before being used to train the classifiers. There is a total of 89 features⁵, and the 90th column contains the target to be predicted by the classifier. The value of the prediction target is a function that is detailed in Equation 1. The feature organization in the dataset used to train the classifiers is as follows:

- Price Changes
- ATR Upside Signal

2 In financial economics, the efficient-market hypothesis states that current stock prices fully reflect all available information. It is therefore, according to the hypothesis, impossible to find a pattern in stock price movement.

3 A self-defeating strategy is a term used for a strategy that will eventually stop working (or reduce in effectiveness) after it is applied to the stock market.

4 In machine learning, creating a model by training a classifier means that we feed the classifier with historical data to ‘train’ on. The created model will decide which class to allocate the newly observed data based on previous data.

5 The selection of 89 features is arbitrary. The number 89 comes from six technical indicators, each of which has a 14-day period. The use of a 14-day period is also arbitrary.

<i>Output of One Classifier</i>	<i>Outputs of Two Classifiers</i>	<i>Action taken by the AT program</i>
Classifier predicts increasing price with strong probability.	Both classifiers agree on an increasing price prediction.	Begin longing the stock. If we are already shorting the stock (betting that it will decrease), stop trading the stock.
Classifier predicts decreasing price with strong probability.	Both classifiers agree on an increasing price prediction.	Begin shorting the stock. If we are already longing the stock (betting that it will increase), stop trading the stock.
Classifier either predicts no change with strong probability or predicts any outcome with weak probability.	The classifiers either agree on no change in stock price or disagree on a prediction.	Make no changes to our ongoing action with the stock.

Table 1: This table outlines the workflow for each of the two trading strategie.

- ATR Downside Signal
- Upper Bollinger Band
- Middle Bollinger Band
- Lower Bollinger Band

The following equation is used to determine the target for prediction.

$$PCT(p) = \begin{cases} +1 & \text{if } p > \text{acertainpercentageofthepricechangethedaybefore}, \wedge \text{ ispositive} \\ 0 & \text{if } p \text{ within a certain percentage of the price change the day before} \\ -1 & \text{if } p > \text{acertainpercentageofthepricechangethedaybefore}, \wedge \text{ isnegative} \end{cases} \quad (Eq. 1)$$

Where p is the price change for tomorrow

3.2 Automatic stock selection

The algorithm is also able to choose certain stocks automatically every month, and therefore fully automates the trading process and keeps our simulations free of survivorship bias. The selection is based on fundamental data ⁶, and it does so by filtering according to a stock’s Price-to-Earnings Ratio (PER) and Market Capitalization (MC). The PER of a stock is measured by dividing the current share price over its earnings per share, and this is used as an indication of the value of the company. MC is calculated by multiplying the current market price of one share with the company’s total number of shares, and this shows the total market value of the shares in a company.

3.3 Longing and shorting stocks

The final stage of the trading strategy of our AT program is the longing and shorting of the selected stocks and the program is scheduled to long and short stocks daily (i.e.: every trading day in the NYSE during the selected time period). Our AT programs will use two different techniques to base our trading on: the first technique is using one classifier and the other is using two classifiers working simultaneously. If one classifier is used, the algorithm will long or short based on how sure the

predictor is of its prediction, and we specify that it should be more than a certain value (defined as the minimum probability) for the AT program to take the appropriate action of longing or shorting. When two classifiers are used, the AT program takes action when both predictions are the same. The actions taken by either of the classification methods is outlined in detail in Table 1 below.

3.4 Slippage and commission

For all simulations in this project, we are using the default *slippage* and *commission* models that are being used on the Quantopian simulator. *Slippage* calculates and simulates the impact of our order on the market, and it is measured by assessing how large our order is in comparison with the current trading volume; this is used to check if an order is too big (given that a trader cannot trade more than the market’s volume at any given time); therefore, our algorithm will be limited to ordering up to 2.5% of the total available stocks, a percentage defined by the simulator to make the simulation results more realistic. The *commission* is set to \$0.03 dollars per share, as is the default on the simulator.

⁶ The fundamental data of a stock is in the broadest terms any data, besides the trading patterns of the stock itself,

which can be expected to impact the price or perceived value of a stock.

4 Methodology

4.1 Testing the chosen machine learning methods in predictive accuracy

Before beginning to trade with the model predictions, it is better to first test the accuracy of the algorithms in predicting the future stock price movement. This would give us a better understanding of each algorithm’s performance in prediction only, and lets us tune the algorithm’s parameters to get better accuracy. Quantopian provides a research environment to experiment and try out trading strategies without running them through a simulator. We will assess the accuracy of each algorithm by using a confusion matrix (a table that counts the predictions that were classified and misclassified) and repeat each simulation multiple times to gain confidence in the results. Each algorithm has its own set of adjustable parameters, which we will try to fine-tune to attain the best accuracy from each algorithm.

4.2 Performance indicators

After finding the best parameters for an accurate prediction of stock price movement, we can move our algorithms from the research environment to the simulation. The algorithms will be part of the larger workflow, which was discussed in detail in the previous section. We define a certain time-period (greater than two years) for the simulation to run through day-by-day, and a fixed starting capital of 1 million US dollars. At the end of each simulation, the algorithm’s performance is assessed automatically through eight performance indicators that

are usually used to assess and compare different trading strategies together; they are outlined in Table 2 below.

We will provide the mathematical equations that were used in determining six of the eight performance indicators below. The remaining two indicators (i.e., cumulative returns and maximum draw-down) are considered to be straight forward and will not be explained due to lack of space. We have chosen to consider *the market* to be reasonably approximated by the Standard and Poor 500 index (NYSE: SPY), and *the risk-free rate* to be reasonably approximated by the US Treasury Index (NYSE: BIL).

4.2.1 Alpha and Beta

The values for alpha and beta are found from an equation that is a part of the capital asset pricing model (CAPM), shown below.

$$\alpha = R_p - [R_f + (R_m - R_f) \cdot \beta] \tag{Eq. 2}$$

R_p is the realized return of portfolio (this is the portfolio that is being simulated). R_m is the market return (this can be approximated by a portfolio with only the SPY Standard & Poor 500 stock longed with initial capital). R_f is the risk-free rate (this can be approximated by a portfolio with only the BIL US Treasury Bill Index stock longed with initial capital). β is calculated as in Eq. 3.

We find beta first using the Eq. 3, then we substitute it in Eq. 2 to get alpha:

$$\beta = \frac{Cov(R_p, R_m)}{Var(R_p)}$$

<i>Performance indicator</i>	<i>Brief description of the indicator</i>
Algorithm Returns	Cumulative returns (as a percentage) of the algorithm relative to the starting capital at the beginning of the simulation
Alpha	The return on an investment that is not a result of general movement in the greater market.
Beta	The tendency of the algorithm’s price movement to respond to swings in the market. A beta value of 0 means the algorithm is uncorrelated to the market, and in some sense is risk-free.
Sharpe Ratio	A measure for calculating risk-adjusted return; it is defined as the average return earned in excess of the risk-free rate per unit of volatility or total risk.
Sortino Ratio	A modification of the Sharpe ratio that differentiates harmful volatility from general volatility by taking into account the standard deviation of negative asset returns (downside deviation). A large Sortino ratio indicates that there is a low probability of a large loss.
Information Ratio	A ratio of portfolio returns above the returns of a benchmark (usually an index) to the volatility of those returns. The information ratio (IR) measures a portfolio manager’s ability to generate excess returns relative to a benchmark, but also attempts to identify the consistency of the investor.
Volatility	An identification of price ranges and breakouts; the ratio uses a true price range to determine an algorithm’s true trading range and is able to identify situations where the price has moved out of this true range.
Maximum Draw-down	The maximum draw-down experienced by the cumulative returns of the algorithm during a certain period of time defined by the simulator.

Table 2: Description of the eight performance indicators the simulator produces.

	Stock actually decreased in price	Stock price actually stayed almost the same	Stock actually increased in price
Stock predicted to decrease in price	36%	7%	6%
Stock price predicted to stay almost the same	12%	17%	15%
Stock predicted to increase in price	1%	2%	4%

Table 3: The confusion matrix from the best classifier (Gradient Boosting) with fine-tuned parameters (obtained through a grid-search) after predicting 970 stock price movements.

(Eq. 3)

$Cov(X, Y)$ is the covariance between the two variables X and Y .

$Var(X)$ is the variance in the variable X .

4.2.2 Sharpe ratio

$$Sharpe = \frac{Mean(R) - |p - R_f|}{StdDev(R_p - R_f)} \quad (\text{Eq. 4})$$

$StdDev(x)$ is the standard deviation in x , and $Mean(x)$ is the average value of x .

4.2.3 Sortino ratio

$$Sortino = \frac{Mean(R) - |p - R_f|}{StdDev} \quad (\text{Eq. 5})$$

$F(x, y)$ is a set of values that only contain the value of x - y when y was greater than x .

4.2.4 Information ratio

$$Information = \frac{Mean(R) - |p - R_m|}{StdDev(R_p - R_m)} \quad (\text{Eq. 6})$$

4.2.5 Volatility ratio

$$Volatility = \frac{T}{EMA_n(T)} \quad (\text{Eq. 7})$$

$$\rightarrow T = Max(H_t - L_t, H_t - C_{t-1}, C_{t-1} - L_t) \quad (\text{Eq. 8})$$

T is coined the “true range” and is determined by Eq. 8.

$EMA_n(T)$ is the exponentially moving average of T over a time period of n days.

H is the highest price a stock reached during the day.

L is the lowest price a stock reached during the day.

C is the closing price of the day for a given stock.

The subscripts of the variables in the true range definition indicate which day the variables are taken from.

4.3 Other considerations

It is a difficult task to compare all trading strategies with only one performance indicator. During our experimentation we will find strategies that perform well through some indicators but poorly in others, so we will compare algorithms using all indicators and leave it to the investor to decide which algorithms are the most favorable. We will first show how changing the prediction algorithm affects the performance indicators, and then we will show how fine-tuning the different algorithm parameters affect the indicators too. In the conclusion of our simulations and analysis of all the different methods and trading strategies, we will try to find the strengths of each prediction algorithm when applied to a trading strategy and show their defining characteristics when that trading strategy is used in AT.

5 Results

5.1 Results from initial testing of predictive accuracy

We precede the simulations with initial testing in the research environment provided by Quantopian. Using a gradient boosting classifier, we trained the classifier on a normalized SPY index during a certain period of time (between 2006 and 2010), and then used that trained classifier to predict three randomly selected stocks, as shown in Table 3. The classifier outputs one of the three classes of prediction: the stock price one day from today will either increase by a certain percentage, decrease by that percentage, or stay within that percentage (which we considered to be negligible movement). Table 3 contains a confusion matrix⁷, and it counts the number of predictions and their outcomes, as a way of assessing the performance of the classifier. The result is the average of ten simulations. The matrix in Table 3 yields an accuracy of 57%, and the accuracy is calculated by summing the diagonal of the matrix.

Following that experiment, we can compare the accuracy of each classifier when used to predict each of the stocks separately. Because there are three classes to predict, we can consider a random guess to be a uniform

⁷ A confusion matrix is a commonly used tool in classification tasks to assess the accuracy of a classifier.

	Apple Inc. (NYSE: AAPL)	JPMorgan Chase (NYSE: JPM)	Microsoft Corp. (NYSE: MSFT)
K-Nearest Neighbors Classification	41.9%	42.6%	48.0%
Support Vector Machine	39.4%	41.2%	42.4%
Random Forest Classification	50.5%	56.6%	51.1%
Extremely Randomized Trees Classification	50.1%	56.0%	50.5%
Gradient Boosting Classification	52.2%	57.0%	53.1%

Table 4: This table provides a useful side by side comparison of the predictive performance each of the three classifiers with fine-tuned parameters, along with two classifiers from previous literature.

distribution between the three classes (i.e.: 33%). This is a good reference to be used when comparing the classifiers, because any classifier that has a predictive accuracy below random guessing is not considered useful. Observing the values in Table 4, all classifiers achieve significantly higher accuracy than both the reference and classifiers from previous work (refer to Section 2); this leads us to believe *the market is not random*⁸ and we can try to use these predictions in trading.

In Table 4, all classifiers are trained on the Standard and Poor 500 index (NYSE: SPY) and then used to predict the stock indicated in each column. In most cases, the Gradient Boosting classifier reaches the highest accuracy amongst the three classifiers, reaching almost two times the accuracy of the reference.

5.2 Simulations using pre-selected stocks

Following the results presented in Table 4, we have confidence in the prediction ability of our ensemble methods, and we can move these methods onto the trading simulation. The details of the inner workings of the trading algorithm are in the previous section (i.e.: Trading Strategy). As described before, there are two versions of the algorithm, each differing by either using one or two classifiers, and by either using preselected stocks or automatic stock selection. We will present the cumulative returns of each version of the algorithm and discuss them briefly. All methods are compared to the Standard and Poor 500 index (NYSE: SPY) as a benchmark for assessing performance. The time period throughout which the classifiers are simulated is selected based on simulation complexity, and we kept all periods to a minimum of two years, usually starting no earlier than 2010, and all algorithms traded daily. In Figure 2, we will show the cumulative returns of each of the three classifiers when using preselected stocks and the one classifier

method.⁹ The preselected stocks are a random selection of 36 stocks that were constituents of the Standard and Poor 500 index (NYSE: SPY) during the year of 2010, which is the starting year for all of the simulations in this project.

It can be deduced from Figure 2 that the Extremely Randomized Trees classifier and the Random Forest Classifier strongly outperform the benchmark, with the Random Forest Classifier having higher cumulative returns towards the end of the period. The Gradient Boosting Classifier under-performs compared to the other classifiers if we compare them using cumulative returns. Gradient Boosting outperforms the other two when it comes to stability and volatility of the simulation.

We will now discuss a problem that may arise when using only one classifier in a trading strategy for stock price prediction. The uncertainty a classifier has in its own prediction, as per the discussed trading strategy, can sometimes lead the AT program to not act upon the prediction. A more complex approach that solves this problem is to use two similar classifiers and only act upon their agreement; we will call this method hereafter the Two Classifier method, and the method that uses only one classifier and a probability threshold will be called hereafter the One Classifier method. The result of this alternative approach, namely the Two Classifier method, is shown in Figure 3 below with preselected stocks only.

The reader can see from Figure 3 that all classifiers easily outperform the benchmark, with the ETC and the RFC classifiers having the best overall cumulative returns. It is worth noting that, compared to the one classifier method the two classifier methods usually have less volatility but also less cumulative returns as well.

⁸ We refer to this because there is a popular hypothesis in financial literature named the “Efficient Market Hypothesis”, and it is an investment theory that states it is impossible to find a pattern in the stock market because stock market efficiency causes existing stock prices to always incorporate and reflect all relevant information.

⁹ The Gradient Boosting Classifier uses fewer estimators than its counterparts, and this is due to the greater

complexity of the algorithm compared to the other algorithms. In general, it takes significantly less processing time to train a Random Forest Classifier (RFC) or Extremely Randomized Trees (ETC) Classifier than a Gradient Boosting Classifier (GBC) with an equal number of estimators for all three classifiers.

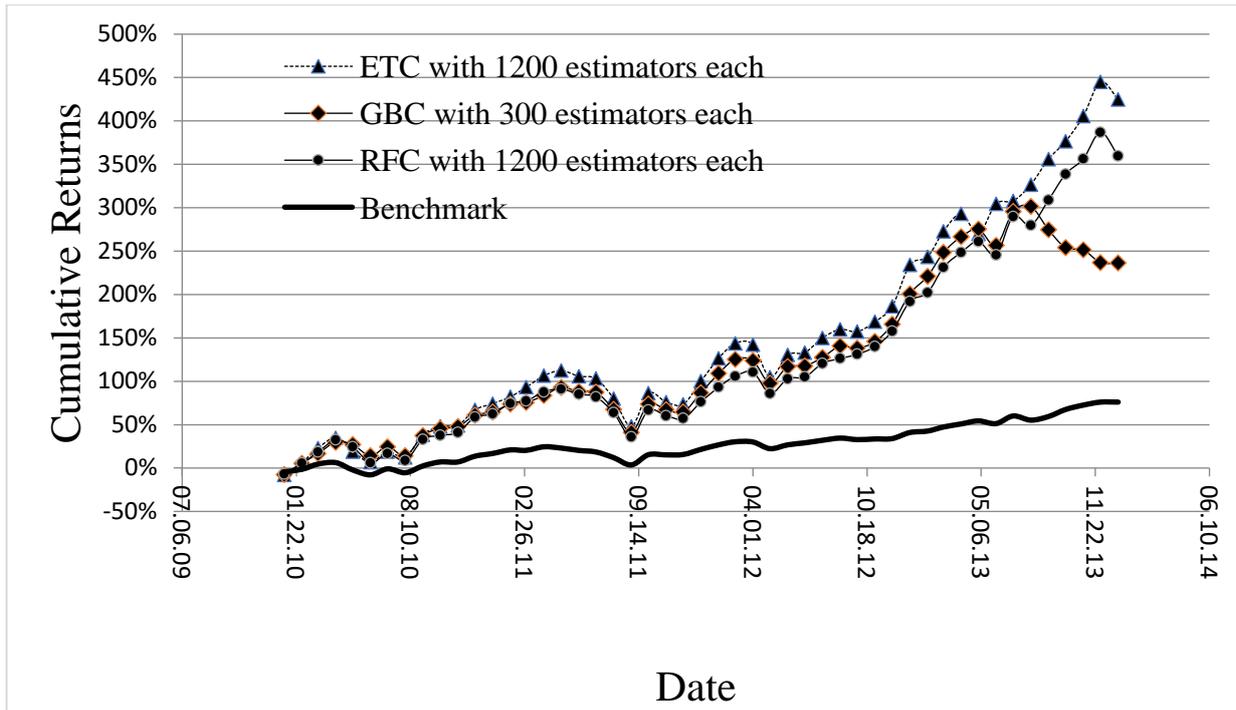


Figure 3: The graph shows the cumulative returns of each of the three algorithms when working with 36 preselected stocks and using the agreement of two classifiers to predict the trading stocks.

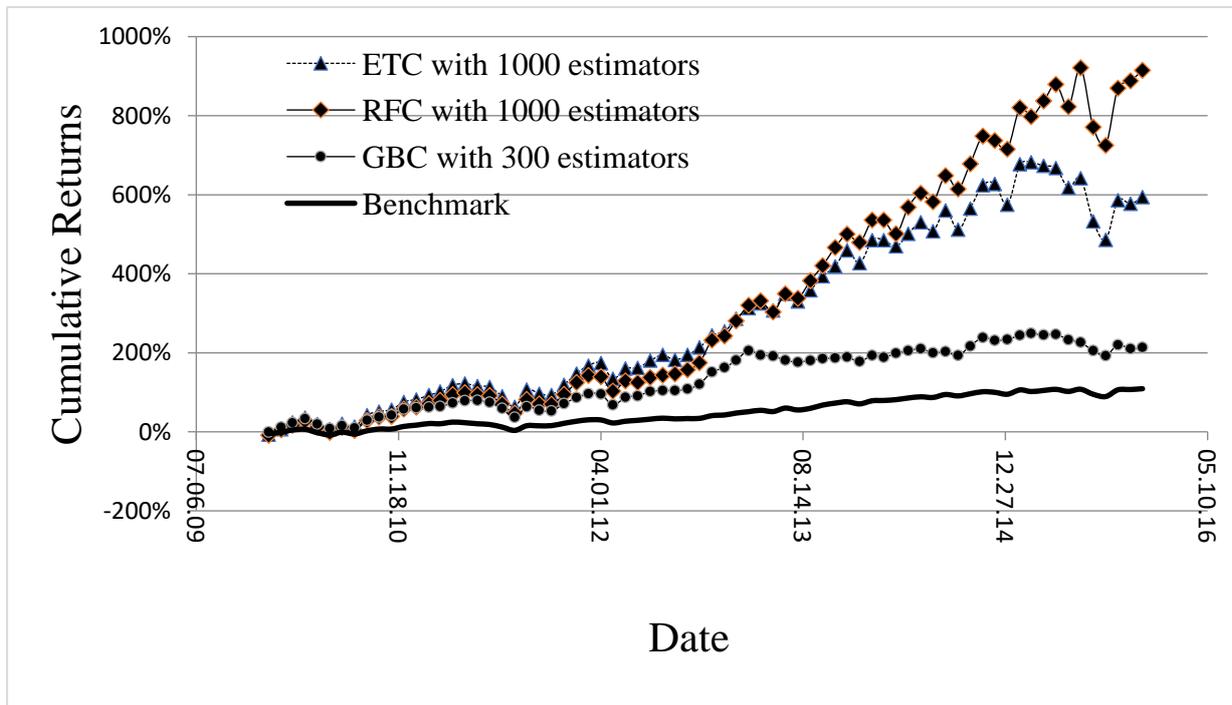


Figure 2: The graph shows the cumulative returns of each of the three algorithms when working with 36 preselected stocks and using one classifier to predict the trading stocks.

5.3 Simulations using automatically selected stocks

The main problem with using preselected stocks is that we are prone to survivorship bias; this means that our selection of stocks manually uses our information of the

future, relative to the time of the simulation. Our proposed solution to survivorship bias is to let our AT program automatically select stocks every month using basic fundamental analysis. The selection scheme for the 100 stocks we used was a simple one based on the stock’s Price-to-Earnings Ratio (PER) and Market Capitalization (MC), and all stocks are selected from the NYSE and

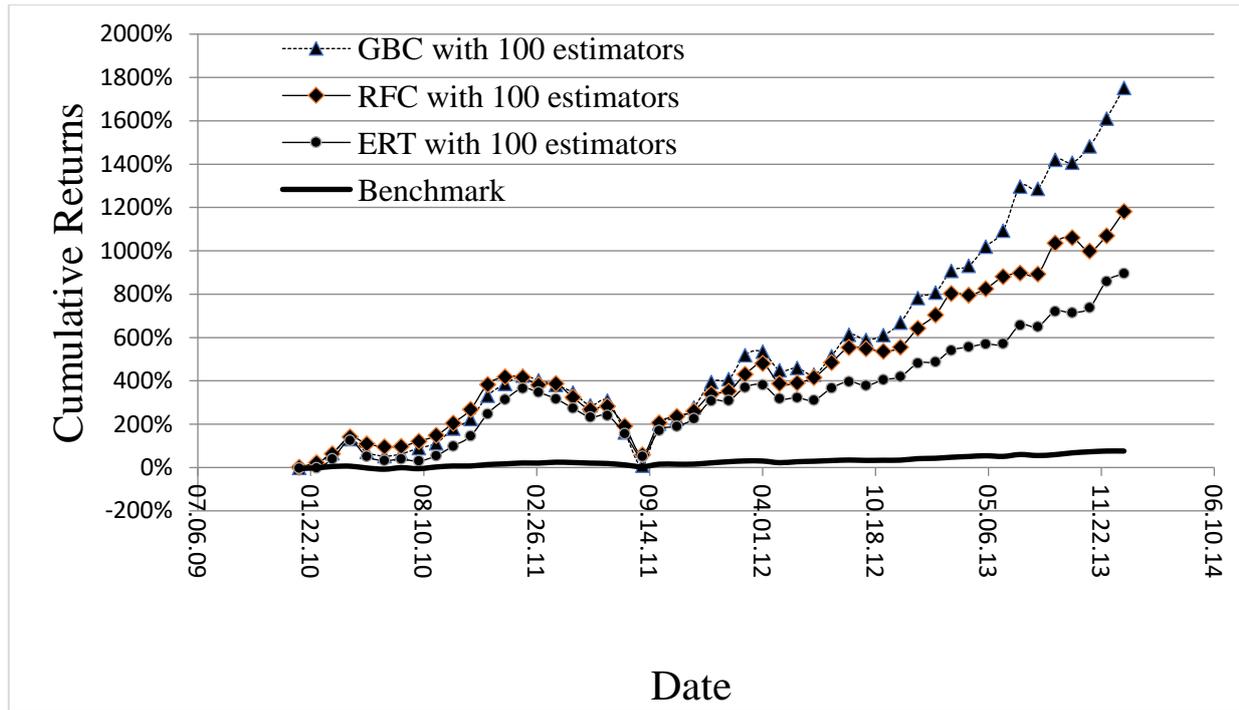


Figure 4: The graph shows the cumulative returns of each of the three algorithms when working with 100 automatically selected stocks (selected at the start of each month) and using one classifier to predict the trading stock.

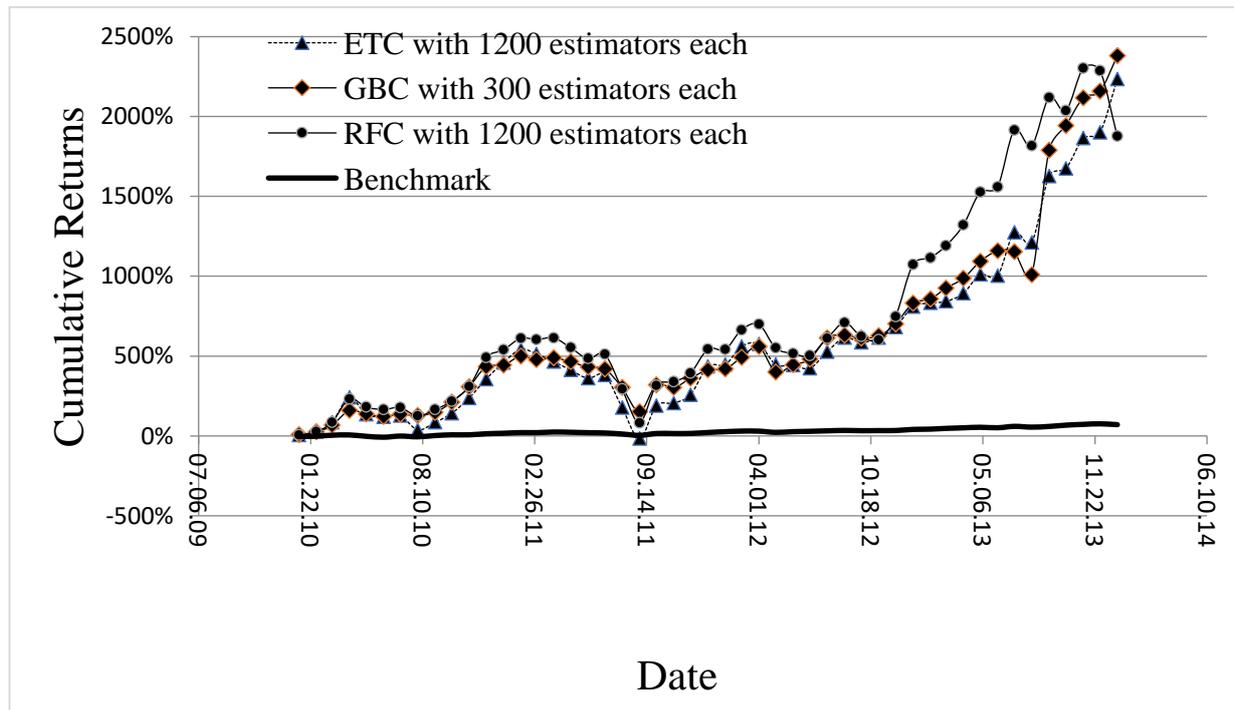


Figure 5: Cumulative returns of each of the three algorithms when working with 100 automatically selected stocks (selected at the start of each month) and using the agreement of two classifiers to predict the trading stock.

NASDAQ exchanges. Our selection used stocks the first 100 stocks that were above \$100 million in MC, had a PER less than 10, and were sorted by their MC value in descending order. Our choice of filter values for the automatic stock selection was based on trial and error through multiple simulations.

At the beginning of each month, the available stocks are filtered and reselected, ensuring that the algorithm has

the best selection of stocks to analyze and trade. The two graphs that follow, Figures 4 and 5, will show the cumulative returns over time when using the one classifier and two classifier methods respectively with automatic selection of trading stocks.

In Figure 4, we can see that all three algorithms seem to be prone to a heavy decline between 2011 and 2012, and we will see that this decline is less prominent in the

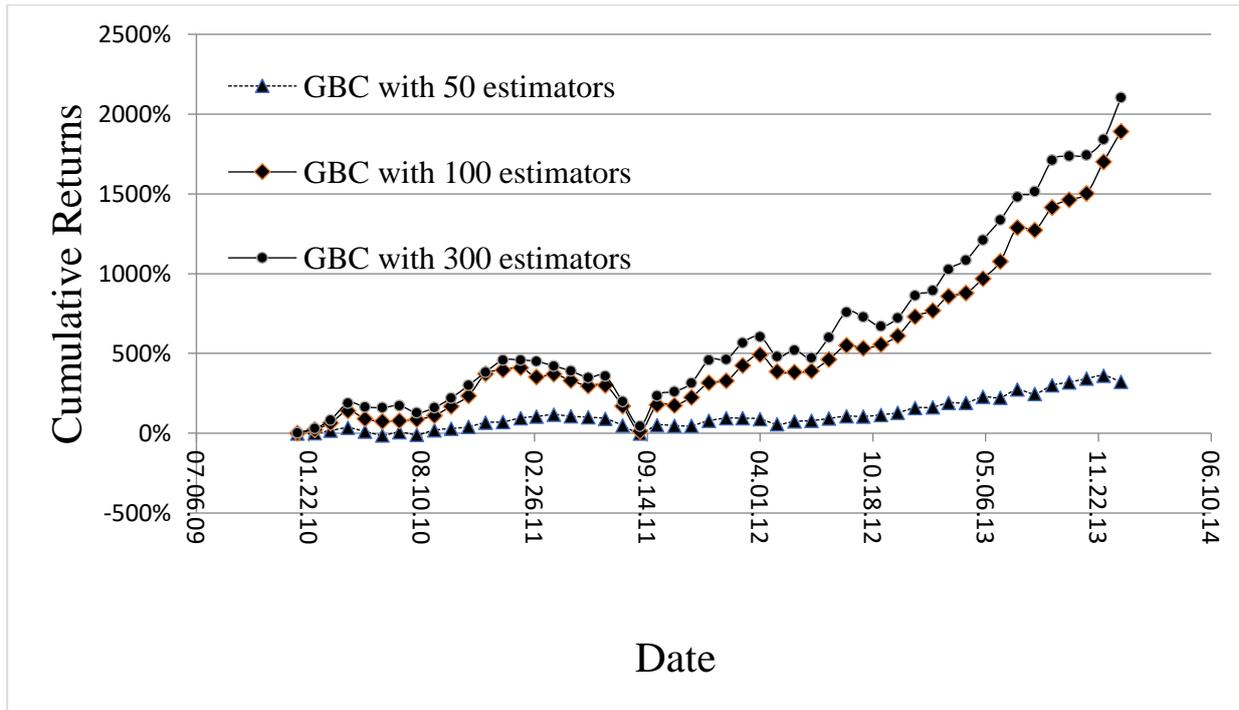


Figure 6: The change in cumulative returns as we change the number of estimators (trees) in the gradient boosting classifier when only one classifier is used and stocks are automatically selected.

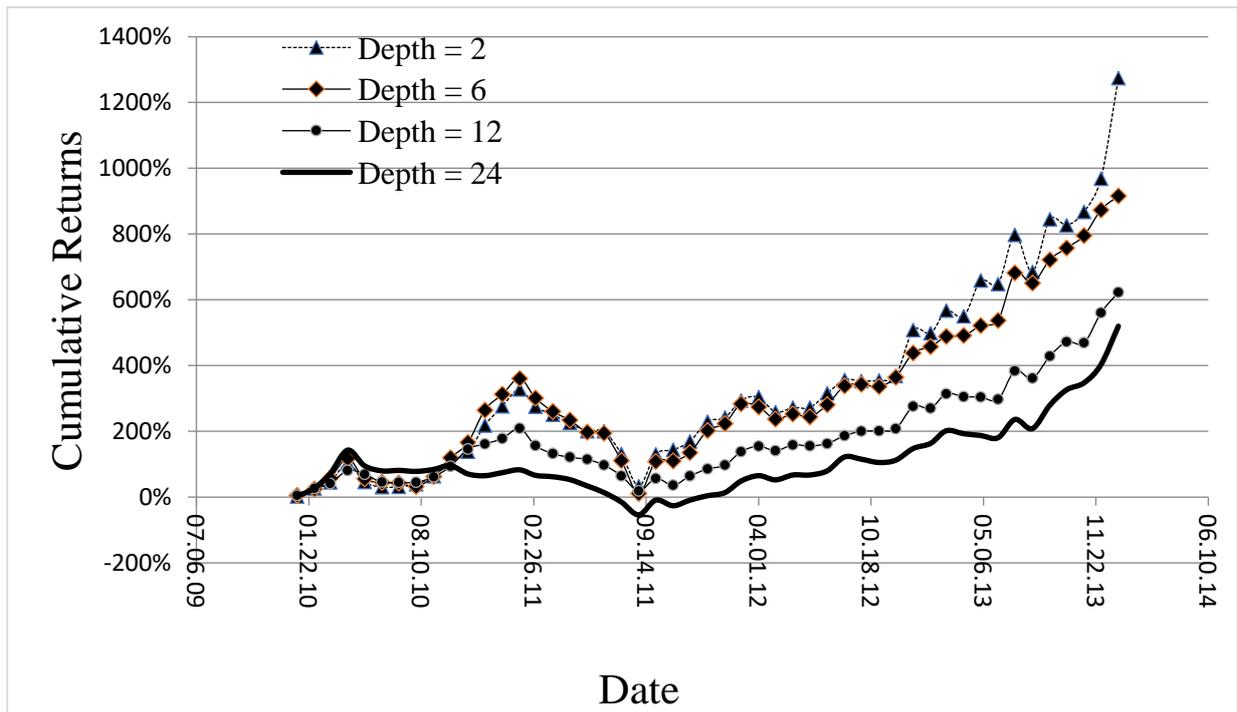


Figure 7: The graph shows the change in cumulative returns as we change the maximum depth of the estimators (trees) in the Gradient Boosting Classifier when only one classifier is used and stocks are automatically selected.

two-classifier method. The Gradient Boosting Classifier outperforms the other two classifiers considerably over the time period following the mass decline. All three classifiers outperform the benchmark significantly.

In Figure 5, it should be noted that the maximum possible number of estimators for each classifier was selected, which is why the more complex Gradient Boosting algorithm has less estimators than the simpler

Extremely Randomized Trees and Random Forest algorithm. Nonetheless, they all achieve similar performance, although the Gradient Boosting algorithm seems to outperform the others and maintains less volatility, especially towards the end of the time period.

	12-month Alpha		12-month Beta	
	One Classifier Method	Two Classifiers Method	One Classifier Method	Two Classifiers Method
Random Forest Classifier	0.40	1.29	1.89	2.79
Extremely Randomized Trees Classifier	0.40	1.05	1.25	2.77
Gradient Boosting Classifier	0.62	1.37	1.74	4.70

Table 5: The table compares the average values of the alpha and beta coefficients over 12-month periods for each of the three classification methods when used in simulation over the time-period 2010 to 2015.

5.4 Simulations with parameter adjustments

Following the results that were observed in the previous section, we will now move on to investigate how fine-tuning the classifier parameters affects cumulative returns of the program. We chose to change two parameters in the Gradient Boosting Classifier, namely the number of estimators in the classifier, and the maximum depth of each estimator in the classifier. It can be seen from Figure 6 that an increase in number of estimators yields better cumulative returns and more resistance to negative changes in the market.

From Figure 7, the reader can observe from the graph that a decrease in the maximum depth allowed yields better cumulative returns. This is most likely due to the classifier not being prone to over-fitting when the depth is limited. While the changes were only performed on the Gradient Boosting classifier, the construction of the other two classifiers is similar and the results are almost the same (we will not show those results due to lack of space). This is evident from when we fine-tuned the classifiers earlier in the research section of Quantopian and arrived at similar results. It can be shown that, in general, increasing the number of estimators and decreasing the maximum depth of the estimators will increase the predictive performance of ensemble methods.

5.5 Performance indicators resulting from simulations

Following the fine-tuning investigation, we start to compare performance indicators other than cumulative returns, using the average of their 12-month value over multiple simulations and taking them to a confidence level of 90%. We will start by comparing the averaged values of alpha and beta over a 12-month period for each of the three classifiers (taking the Standard and Poor 500 index, NYSE: SPY, as a benchmark) and for each of the classification scenarios (one classifier and two classifiers respectively). By observing the data in Table 5, it can be seen that in general the two-classifier method produces a higher value of alpha and beta than the one classifier method.

When looking for a good trading strategy, it is preferable to find one that has low correlation to the market (in this case, the market is considered to be the

Standard and Poor 500 index, NYSE: SPY). A low long-term beta value and a high long-term alpha value for a trading strategy are desirable characteristics of a trading strategy because they indicate low market correlation and high “risk-free” return respectively.

From observing the data in Table 5 we can deduce that, while most strategies have undesirably high values of beta, they also possess large values of alpha and the two classifiers method produces higher alpha than its one classifier counterpart on average. This leads us to presume there is value in these strategies from a financial perspective, although they would need further work to decrease market correlation. We will also compare the ratio indicators which give the reader some indicator of return versus risk for each of the classifiers in each of the classification scenarios, but in different ways. It should be noted that the ratios will seem unusually high compared to traditional methods, and this is expected with the high cumulative returns we saw earlier. The next table, Table 6, compares the Sharpe, Sortino, and Information ratios for all three ensemble methods for both the one classifier and two classifier methods.

We can observe from the values in Table 6 that in general, the Gradient Boosting Classifier method produces higher performance ratios than its counterparts and this may indicate that the Gradient Boosting Classifier method results in higher returns with lower risk relative to other trading strategies. The reader can also see that the two-classifier method produces higher performance ratios than the one classifier method in most of the cases, except for the Information Ratio in which both methods have similar performance. The final two performance indicators, outlined in Table 7 below, are the volatility and maximum draw-down, both of which describe the risk associated with the trading algorithm.

It can be observed from Table 7 that in general, the two-classifier method has higher volatility than the one classifier method. This may be because of the greater condition placed on the one classifier method (i.e.: to be sure of the result with a certain probability defined beforehand) than that of the two classifiers method. Reading the three previous tables, namely Tables 5, 6, and 7, we can see that almost all trading strategies have high volatility and large draw-downs, despite them having large amounts of cumulative return and good return-versus-risk ratios. It is worth noting that the two classifier

	Sharpe Ratio		Sortino Ratio		Information Ratio	
	One Classifier Method	Two Classifiers Method	One Classifier Method	Two Classifiers Method	One Classifier Method	Two Classifiers Method
Random Forest Classifier	2.26	3.42	4.06	5.28	0.11	0.10
Extremely Randomized Trees Classifier	2.68	3.24	4.07	3.25	0.10	0.10
Gradient Boosting Classifier	3.61	3.84	5.28	5.73	0.15	0.13

Table 6: The table compares the average values of the Sharpe, Sortino and Information ratios over 12-month periods for each of the three classification methods when used in simulation over the time-period 2010 to 2015.

	Volatility		Maximum Draw-down	
	One Classifier	Two Classifiers	One Classifier	Two Classifiers
Random Forest Classifier	0.24	0.35	11.55%	21.45%
Extremely Randomized Trees Classifier	0.23	0.49	11.69%	25.25%
Gradient Boosting Classifier	0.22	0.38	24.00%	24.02%

Table 7: The table compares the average values of the volatility and maximum draw-down indicators over 12-month periods for each of the three classification methods when used in simulation over the time-period 2010 to 2015.

methods, on average, have higher return and risk ratios than their one classifier counterparts.

6 Conclusions and future work

It has always been a difficult task to predict the stock market, and even after many attempts by researchers to restore confidence in stock market prediction, there are many other researchers and economists that believe the market is random and cannot be predicted. It is our aim that the simulation results presented in this project provide the reader with a new positive look towards machine learning and stock market prediction.

The simulation results and investigation show both advantages and disadvantages of using ensemble methods in algorithmic trading; all three classification methods were able to *outperform the benchmark* in cumulative returns *over a long trading period* and had *significant alpha coefficients*. Additionally, all three classification methods outperformed the methods from previous literature in predictive accuracy. However, when our methods were applied in algorithmic trading, they all showed volatile and risky trading behavior, and they all had an undesirably large correlation to the market. Through basic parameter adjustment (such as maximum depth and number of estimators) and changing of trading strategies (such as one and two classifier methods), we were able to greatly modify the performance of all classifiers and their simulated trading portfolios.

It is in our opinion that the deployment of the algorithmic trading strategies discussed in this paper is a good move by any prospective investor. The strategies

outlined in this paper provide a solid foundation for further improvement, and they can be deployed once they have been adjusted to have less volatility and risk. The use of these ensemble methods specifically, and machine learning in general, in real algorithmic trading in the future is realistic, and we expect further work and research on them.

The field of machine learning algorithms is a very vast one and we have only explored the field of ensemble methods in this project. There are many new machine learning methods being developed that have been shown to excel at time-series prediction. For example, deep neural networks have recently shown good performance in time-series prediction, and there have been some attempts at using them on stock market data [35]. It is also possible to create ensemble methods from deep neural networks and apply similar ideas presented in this project to those ensemble methods.

We have also only explored one specific trading strategy that encompassed our prediction algorithms. The field of algorithmic trading has many different trading strategies, and these can differ in the inputs used to feed the machine learning algorithm, or in the way the strategy handles the output of the machine learning algorithm. For the former, it would be wise to investigate the use of both fundamental data and technical indicators together as inputs to the predicting algorithm, and assess the results. For the latter, it would also be wise to investigate a larger number of output prediction classes than the three target classes we used in this project.

Glossary of Technical Terms

Asset Ratios	The ratio of company's total sales relative to the value of their assets.
Liquidity Ratio	Determines company's ability to pay off its short-term debt obligations.
Debt Ratio	Describes the financial health of the company. Determined by dividing total liabilities by total assets.
Algorithm	Computer program consisting of a set of instructions to achieve a well-defined task in a finite number of steps.
Classifier	Algorithm responsible of classifying new observed data into a set of categories.
Ensemble Methods	Methods that use multiple learning algorithms to obtain a better performance than any of the constituent algorithms.
Exponential Moving Average	The average of the stock prices over a defined number of time periods, giving more weight to more recent prices.
Clustering	A method, algorithm, that assigns a set of observations into subsets 'clusters', in which observations in the same cluster are similar in some sense. Different method than classifiers but used for the same purposes.

References

- [1] Mladjenovic, Paul. *Stock investing for dummies*. John Wiley & Sons (2005).
- [2] Levišauskait, Kristina. "Investment Analysis and Portfolio Management." Leonardo da Vinci programme project (2010).
- [3] Bauman, Mark P. "A review of fundamental analysis research in accounting." *Journal of Accounting Literature* 15 (1996): 1.
- [4] Edwards, Robert D., John Magee, and WH Charles Bassetti. *Technical analysis of stock trends*. CRC Press (2007).
- [5] Nuti, Giuseppe, et al. "Algorithmic trading." *Computer* 44.11 (2011): 61-69.
- [6] Glantz, Morton, and Robert Kissell. *Multi-Asset Risk Modeling: Techniques for a Global Economy in an Electronic and Algorithmic Trading Era*. Academic Press (2013).
- [7] Hendershott, Terrence, Charles M. Jones, and Albert J. Menkveld. "Does algorithmic trading improve liquidity?" *The Journal of Finance* 66.1 (2011): 1-33.
- [8] Chaboud, Alain P., et al. "Rise of the machines: Algorithmic trading in the foreign exchange market." *The Journal of Finance* 69.5 (2014): 2045-2084.
- [9] Tudball, Dan. "Quant Insights." *Wilmott* 2016.82 (2016): 27-30.
- [10] Brock, William, Josef Lakonishok, and Blake LeBaron. "Simple technical trading rules and the stochastic properties of stock returns." *The Journal of finance* 47.5 (1992): 1731-1764.
- [11] Atsalakis, George S., and Kimon P. Valavanis. "Surveying stock market forecasting techniques—Part II: Soft computing methods." *Expert Systems with Applications* 36.3 (2009): 5932-5941.
- [12] Fama, Eugene F. "Random walks in stock market prices." *Financial analysts journal* 51.1 (1995): 75-80.
- [13] Shah, Vatsal H. "Machine learning techniques for stock prediction." *Foundations of Machine Learning* | Spring (2007).
- [14] Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." *Annals of statistics* (2001): 1189-1232.
- [15] Breiman, Leo. "Random forests." *Machine learning* 45.1 (2001): 5-32.
- [16] Geurts, Pierre, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees." *Machine learning* 63.1 (2006): 3-42.
- [17] Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." *Annals of statistics* (2001): 1189-1232.2. Tin Kam (1995); *Random Decision Forest*.
- [18] Li, Ping, Qiang Wu, and Christopher J. Burges. "Mcrank: Learning to rank using multiple classification and gradient boosting." *Advances in neural information processing systems*. (2007).
- [19] Geurts, Pierre, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees." *Machine learning* 63.1 (2006): 3-42.
- [20] Hendershott, Terrence, Charles M. Jones, and Albert J. Menkveld. "Does algorithmic trading improve liquidity?" *The Journal of Finance* 66.1 (2011): 1-33.
- [21] Boehmer, Ekkehart, Kingsley YL Fong, and Juan Julie Wu. "International evidence on algorithmic trading." *AFA 2013 San Diego Meetings Paper*. (2014).
- [22] Kimoto, Tatsuya, et al. "Stock market prediction system with modular neural networks." *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*. IEEE (1990).
- [23] Elman, Jeffrey L. "Distributed representations, simple recurrent networks, and grammatical structure." *Machine learning* 7.2-3 (1991): 195-225.
- [24] Naeini, Mahdi Pakdaman, Hamidreza Taremiyan, and Homa Baradaran Hashemi. "Stock market value prediction using neural networks." *Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on*. IEEE (2010).
- [25] Li, Y., Zheng, W. and Zheng, Z., 2019. Deep robust reinforcement learning for practical algorithmic trading. *IEEE Access*, 7, pp.108014-108022.
- [26] Wang, W. and Yu, N., 2019, August. A machine learning framework for algorithmic trading with virtual bids in electricity markets. In *2019 IEEE Power & Energy Society General Meeting (PESGM)* (pp. 1-5). IEEE.

- [27] Huang, Wei, Yoshiteru Nakamori, and Shou-Yang Wang. "Forecasting stock market movement direction with support vector machine." *Computers & Operations Research* 32.10 (2005): 2513-2522.
- [28] Shah, Vatsal H. "Machine learning techniques for stock prediction." *Foundations of Machine Learning*, Springer (2007).
- [29] Andersen, André Christoffer. *A Novel Algorithmic Trading Framework Applying Evolution and Machine Learning for Portfolio Optimization*. Diss. Master's Thesis, Faculty of Social Science and Technology Management, Department of Industrial Economics and Technology Management (2012).
- [30] Shen, Shunrong, Haomiao Jiang, and Tongda Zhang. "Stock market forecasting using machine learning algorithms." (2012).
- [31] Lin, Yuling, Haixiang Guo, and Jinglu Hu. "An SVM-based approach for stock market trend prediction." *IJCNN, The 2013 International Joint Conference, IEEE*, (2013).
- [32] Gupta, Abhishek, and Samidha D. Sharma. "Clustering-Classification Based Prediction of Stock Market Future Prediction." *IJCSIT) International Journal of Computer Science and Information Technologies* 5.3 (2014): 2806-2809.
- [33] Mark Dunne. "Stock Market Prediction". University College Cork (2015).
- [34] Scikit-learn: Machine Learning in Python, Pedregosa et al., *JMLR* 12, pp. 2825-2830 (2011).
- [35] Jia, Hengjian. "Investigation Into The Effectiveness Of Long Short Term Memory Networks For Stock Price Prediction." *arXiv preprint arXiv:1603.07893* (2016).

Increasing the Engagement Level in Algorithms and Data Structures Course by Driving Algorithm Visualizations

Slavomír Šimoňák

Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics
Technical University of Košice, Slovak Republic
E-mail: slavomir.simonak@tuke.sk

Keywords: algorithms, data structures, algorithm visualization, study supporting system, active learning

Received: July 12, 2019

The paper presents the results of our research in the field of applying algorithm visualizations within Data structures and algorithms subject. We accomplished several experiments relating the ability of students to solve simple problems in a pure visual way in one case and by programming the solution using a particular programming language in another one. The experiments are described and the results are analyzed within the paper. In accordance with our previous informal experiences and the results of the analysis we found there can be some part of students, which will be able to apply an algorithm to concrete problem in a visual way, but will not be able to express it clearly enough and implement it in given programming language. As an attempt to cope with the situation, we propose a new teaching approach, together with a prototype of study supporting system, based on the idea that students would participate on creating simple visualizations, not just using them. The purpose of such approach is to help students to develop both types of skills - understanding the algorithms and implementing them as well, by increasing the engagement level and supporting the active learning.

Povzetek: Predstavljen je način predavanja o algoritmih in podatkovnih strukturah s pomočjo algoritmične vizualizacije.

1 Introduction

Data structures and algorithms, the subject of our interest within the paper, is one of fundamental subjects taught within the bachelor study program at our department. Since the subject is positioned in the second year of study, students are supposed to have the basic knowledge and some practical skills in programming [12]. The goal of the subject is to further enhance this skills and to provide the students with fundamental knowledge on data structures, methods for designing algorithms and to asses their efficiency. As the area of algorithms and data structures is more general and abstract, it is also more complicated to learn for many students [5]. One of widely adopted approaches to help with this situation is based on using algorithm visualizations within the subjects covering the area [8, 6]. Also within our subject, except the conventional ways of teaching (using pseudocodes of algorithms or diagrams), we also use algorithm visualizations for couple of years [17, 16]. Although the results achieved by using visualizations are promising [19, 20], we further try to improve the educational process in order to make it more attractive and efficient.

If we want to write a correct program for solving a given problem, we need to have pretty clear idea on the algorithm solving the problem. But we also need some practical skills to be able to implement the algorithm in a particular pro-

gramming language. So if one of this two basic elements is missing it would be very challenging task, so we can give up, or it will take too long to write the required program. While the first one seems to be absolutely necessary, as without knowing the particular algorithm we will not be able to create its implementation, it does not automatically imply, that a student knowing the algorithm will be able to create the corresponding program. So our hypothesis is, that there will be some part of students, which will be able to apply an algorithm to concrete problem in a visual way, but will not be able to create its corresponding program implementation within the reasonable time limit.

In experiments described further within the paper we want to contribute to this idea by comparing the results of solving simple problems by students just in visual way in one case and by programming the solution in another one. We think it is very useful to utilize algorithm visualizations in order to help students to understand the basic principles of algorithms operation. But according to our experiences, we also believe there is still some gap between understanding the basic principles of particular algorithm and the ability to express it clearly enough in a given programming language. By the experiments we wanted to obtain some empirical results in order to support or to invalidate our informal experiences in this respect. After evaluating the results of experiments we provide some of our ideas how the quality of the understanding of the given topic as well

as the ability to implement particular algorithms could be improved jointly.

The rest of the paper is organized as follows. In section 2 we provide a short description on selection of some interesting and influential works on the topic of algorithm visualization and its effectiveness. Section 3 gives an overview of the algorithm visualization systems we developed to support the teaching process in the field of algorithms and data structures. Experiments we conducted are described in section 4 and the results of experiments are evaluated and analyzed in section 5 of the paper. Section 6 proposes a new approach, aiming at increasing the engagement level, supported by the prototype of new study supporting system. Section 7 concludes the paper and provides some ideas for further development.

2 Related work

There are many tools for algorithm visualizations available presently, which would indicate that algorithm visualizations are widely used in a field of algorithm and data structures education. But, as the results from the recent research in the field indicate, the important part of effectiveness of algorithm visualization is how students are engaged in a learning activity. Authors of a meta-study [7] conclude, that studies in which students only viewed visualizations, usually did not indicate significant learning advantages over students using conventional learning materials. This can be perceived in a way, that the mere presence of visualizations does not guarantee that students will better understand algorithms. The results of this research also suggest that the most successful educational applications of algorithm visualizations are those in which the visualization is used as a vehicle for engaging students into the process of learning. So the form of the learning activity in which visualization technology is used is more important than the style of visualizations used.

One of modern approaches is based on learning through playing educational games [3, 4]. It is believed that using educational games can provide a wide range of benefits (like increased effectiveness, interest and motivation), but those are questionable or at least not rigorously established [11]. In the paper [5] authors present an educational game intended to help students in understanding the stack data structure on conceptual as well as practical level. The Stack Game was developed in three parts, corresponding to learning objectives (understanding the concept of stack, application of stacks, stack implementation), bound together by a meaningful storyline.

An interesting exercise support system, based on combining exercise tasks with automatic evaluation and integrated algorithm animation is described in [14]. The system is based on the established ANIMAL system, since it supports the ad-hoc generation of animation based on data provided by a user. So the creation of new exercise sheets based on existing sheets by the means of modifying the

input parameters is supported by the system. The correct answer does not have to be given directly, since it can be determined by the system automatically, based on provided evaluation scripts.

While the above mentioned approaches are interesting, the approach described in this work for increasing students' engagement and motivation is slightly different and it is based on the idea that they would participate on creating visualizations not just using them. The approach, together with the prototype of associated supporting tool, will be described in greater details later in this paper.

Very interesting in this respect is the engagement taxonomy [10], which defines five levels of interaction [8] between a student and an algorithm visualization:

- viewing,
- responding,
- changing,
- constructing,
- presenting.

Several hypotheses are proposed in [10], which can be interpreted in a way, that the higher level or the more forms of engagement are used, the more efficient the learning becomes. Within our currently available tools (like Algomaster or VizAlgo, described briefly in the following section) first three levels of interaction from the engagement taxonomy are easily accessible for students. Mastering the fourth level (constructing visualizations) however, is bit more complicated, since it requires some knowledge about the structure of the application and its plugin modules. There has been some attempts to simplify the process of creating plugins for Algomaster platform [2], but the solution created has still limited area of usability.

As the approach presented later in this work is based on the idea of involving students deeper into the process of creating particular visualizations, it can be considered to provide the fourth level of interaction within the engagement taxonomy. Hence we hope it could help to increase the learning efficiency by providing the higher level of interaction, as well as to develop practical programming skills by implementing the algorithm under consideration.

3 Algorithm visualization tools

Within our subject we use algorithm visualizations as an education supporting tools for several years. We started with visualizations available from different authors. But, while it was a quick and simple solution, we encountered the limitations of various kinds time to time, so we started developing our platforms for visualizing algorithms and data structures. Platforms were developed in order to fulfill our specific needs regarding the selection of algorithms, naming conventions or the ability to adapt the visualizations whenever we decided. The platforms are only briefly

described here, to provide a reader with basic context, since one of them (Algomaster) was also involved in experiments described in the next section. For more information, additional references are provided as well. The first of the platforms was named VizAlgo [16] and it was developed with emphasis on two main goals: extensibility and portability. The first of the goals is reflected within the structure of the application, which consists of two cooperating parts - the core module and the set of relatively independent plugin modules. The core module is intended to provide the support for displaying and controlling the algorithm execution, while plugin modules are responsible for visualizations of particular algorithms. Choosing the Java development platform was connected with the second of the goals mentioned above. The platform is still in development and over the years not only the set of available visualizations was changing, but also the core functionalities and user interface [1] evolved (Figure 1).

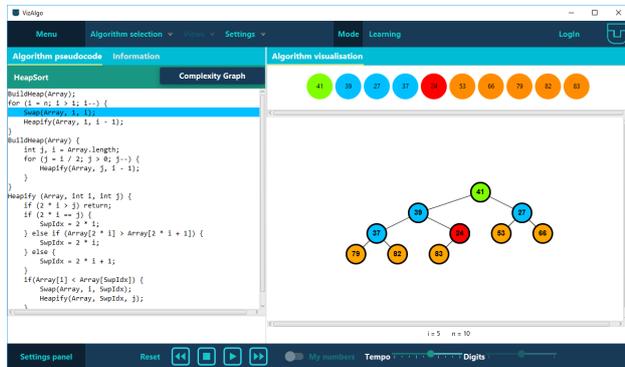


Figure 1: VizAlgo platform

According to experiences gained with the VizAlgo, the second of the platforms, Algomaster [18], also has the plugin-based architecture, but it was intended to provide some more advanced features. The features include functionality for algorithm stepping in both directions [13], call stack visualization for recursive algorithms and a special mode for practical student testing in a visual way. In contrast to the VizAlgo, the Algomaster is based on .NET framework development and execution platform [9]. Later on, the platform was extended significantly [2] in order to provide the support for visualization of complex algorithms with the ability of changing input data during the visualization. Examples of visualizations using the new features are operations on B-tree, 2-3 tree or AVL tree (Figure 2).

In addition to the ability to define input data dynamically, extensions were also oriented towards a real-time student testing and support for simplified development of plugins for the platform. In order to simplify the creation of Algomaster plugins, a separate application named AlgoCreator (Figure 3) was developed.

The application uses a pattern for generating plugins of particular algorithm class, e.g. pattern for comparison-based sorting algorithms. A pattern consists of text tem-

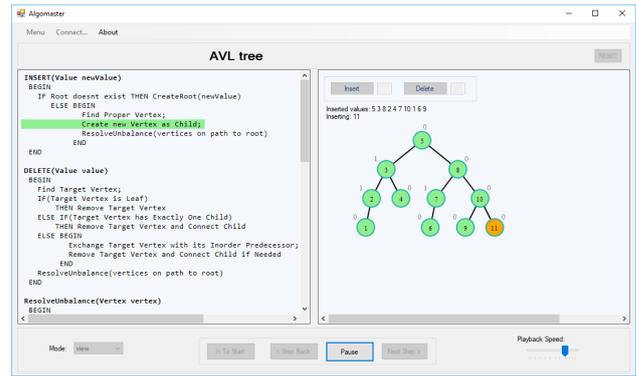


Figure 2: Algomaster platform

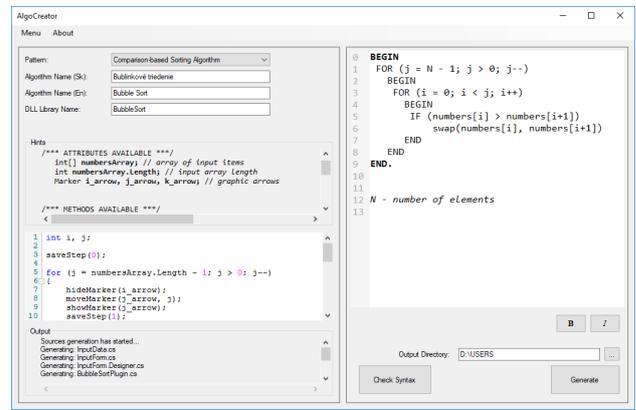


Figure 3: An environment of AlgoCreator application

plates for source code generation and an interpreter for interpreting user defined model. In short, a process of creating a plugin module can be described as follows: a user can select one of available patterns, provide basic algorithm-related information and the algorithm pseudocode, define the behavior of the algorithm and initiate library generation. The process is described in deeper detail in [2].

4 Experiments

As it was yet mentioned within introductory part of the paper, the main motivation behind the accomplished experiments was the comparison of the ability of students to solve algorithmic problems in two distinct ways. One of them was based on visual “simulation” of given algorithm operation, using one of our visualization tools, described in section 3. The another one consisted of programming the particular algorithm in given programming language. Experiments considered in this work were conducted with students of four study groups (G_1 - G_4) and they were focused on two basic areas. The first area was oriented on traversing trees using different strategies (T) and the second one on simple comparison-based sorting algorithms (S).

Thus assignments of the particular area consisted of two

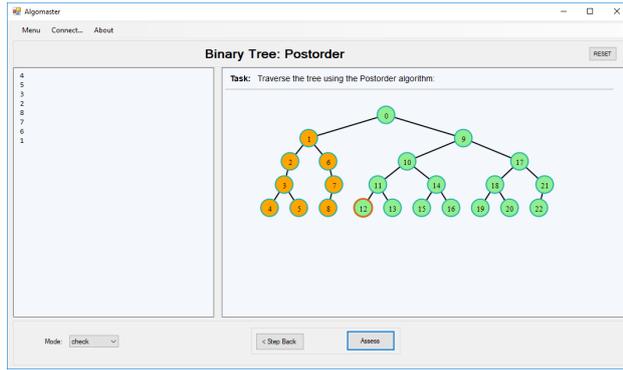


Figure 4: Algomaster in check mode - traversing binary tree

parts: solving the problem in pure visual way using the Algomaster platform (*V*) in one case (Figure 4) and programming the particular algorithm in C programming language (*P*) in the another one. This way we got the four combinations (two areas and two ways of solving a problem from the given area - *TP*, *TV*, *SP* and *SV*) for each of four study groups (G_1 - G_4).

In the area of tree traversing, three basic traversing strategies were used (in-order, pre-order and post-order). In the area of sorting, simple sorting algorithms (like Insert sort and Bubble sort) were used. Within the following four tables (Table 1, Table 2, Table 3, and Table 4), individual scores are presented, achieved by students of particular study groups (G_1 - G_4) in all experiments (*TP*, *TV*, *SP*, *SV*).

Experiment	Results
<i>TP</i> - G_1	0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0
<i>TV</i> - G_1	0.06 1 1 1 1 1 1 1 0.9 0 0 1 1 1 0.2 0.07 1 1 1 0 1 0
<i>SP</i> - G_1	0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 - 0 0 1 0
<i>SV</i> - G_1	1 1 1 1 1 0.06 0.15 1 1 0.77 1 1 1 0.57 0.05 1 - 1 0 1 0.18

Table 1: Results achieved by the students of the study group G_1

Experiments described within this section were conducted in Fall 2017. 82 students were considered on experiments in total, of which 72 were males and 10 were females. Since all the activities were not necessarily conducted on a single class, not all students were necessarily present on all activities. Such situation can be distinguished in particular table by the presence of “-” character within the Results column. This fact can be perceived as a slight disadvantage, but it is generally hard to influence the presence of students on classes. And since it was registered only in few individual cases from all considered students, we believe it was not affecting the results significantly.

Experiment	Results
<i>TP</i> - G_2	0 0 0 0 1 0 1 - 0 1 0 0 0 0 0 1 1 1 0 1 1
<i>TV</i> - G_2	1 0.67 0.09 1 1 0.21 0 1 0.38 0.4 1 0.14 1 1 1 0.6 1 1 1 0.79 0.67
<i>SP</i> - G_2	1 0 1 1 1 0 1 1 0 1 1 0 0 1 0 1 1 1 0 1 1
<i>SV</i> - G_2	1 1 0.33 1 0 1 1 1 1 1 0.81 0 0.05 0.62 1 1 1 1 1 1 1

Table 2: Results achieved by the students of the study group G_2

Experiment	Results
<i>TP</i> - G_3	0 0 0 0 0 1 1 1 1 1 - 1 1 1 1 1 - 0 1 0
<i>TV</i> - G_3	0.46 0 1 0.75 1 0.14 1 1 0.13 1 0.2 1 0.1 1 1 1 1 1 1 1
<i>SP</i> - G_3	0 1 1 0 0 1 0 1 1 0 0 0 1 1 1 1 1 0 1 0
<i>SV</i> - G_3	1 1 1 1 1 1 1 0.93 1 1 1 1 1 1 1 1 0.04 0.29 1 1

Table 3: Results achieved by the students of the study group G_3

5 Analysis of the results of experiments

Within this section we provide a sketch of approach for calculating some of the resulting values, summarize the obtained results and formulate some comments on them. For calculating the average scores (mean) of the first group (G_1) of students in particular experiments (*TP*, *TV*, *SP*, *SV*), the following formulas (1 - 4) were used. The *average score* (G_1Av_{TP}) achieved by the study group (G_1) in the experiment *TP* is given by the formula (1). Within the formula, G_1TS_{TP} represents the *total score* achieved by the group G_1 in the *TP* experiment and G_1NS_{TP} the *number of students* participating in the experiment. The mean values (G_1Av_{TV} , G_1Av_{SP} , G_1Av_{SV}) for remaining experiments (*TV*, *SP*, *SV*) of the study group G_1 were calculated analogically.

Experiment	Results
<i>TP</i> - G_4	1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 1 0 1 0 0
<i>TV</i> - G_4	1 0.63 1 0 0.4 1 0 1 0.33 1 1 0 1 1 1 0.86 1 1 1 1
<i>SP</i> - G_4	1 0 0 0 0 0 1 1 - 1 0 1 1 1 0 1 1 0 1 0
<i>SV</i> - G_4	1 0.7 0.16 1 1 1 1 1 - 1 1 1 1 1 1 1 1 1 1 1

Table 4: Results achieved by the students of the study group G_4

$$G_1Av_{TP} = \frac{G_1TS_{TP}}{G_1NS_{TP}} = \frac{6}{21} = 0.286, \quad (1)$$

$$G_1Av_{TV} = \frac{G_1TS_{TV}}{G_1NS_{TV}} = \frac{14.23}{21} = 0.678, \quad (2)$$

$$G_1Av_{SP} = \frac{G_1TS_{SP}}{G_1NS_{SP}} = \frac{7}{20} = 0.35, \quad (3)$$

$$G_1Av_{SV} = \frac{G_1TS_{SV}}{G_1NS_{SV}} = \frac{14.78}{20} = 0.739. \quad (4)$$

Similarly, the mean values were calculated for remaining groups ($G_2 - G_4$), based on the data presented in tables Table 2, Table 3, and Table 4. Variance and standard deviation values for all experiments were calculated as well and the overall results are available in the table Table 5.

Exper.	Group	Mean	Variance	Std. deviat.
TP	G1	0.286	0.204	0.452
TV		0.678	0.201	0.448
SP		0.35	0.228	0.477
SV		0.739	0.153	0.391
TP	G2	0.4	0.24	0.490
TV		0.712	0.125	0.353
SP		0.667	0.222	0.471
SV		0.8	0.128	0.358
TP	G3	0.611	0.238	0.487
TV		0.739	0.147	0.383
SP		0.55	0.248	0.497
SV		0.913	0.064	0.253
TP	G4	0.35	0.228	0.477
TV		0.761	0.141	0.376
SP		0.526	0.249	0.499
SV		0.94	0.038	0.196

Table 5: Statistical results of experiments

As we can observe from the graph of average scores (Figure 5) achieved by students in particular activities, the scores achieved in visual tasks are usually significantly higher than the scores achieved in corresponding programming tasks. The only differences are TP/TV relation for the group G_3 and SP/SV relation for the group G_2 . Also in these cases the scores achieved by solving problems in visual way are higher, but maybe not so significantly.

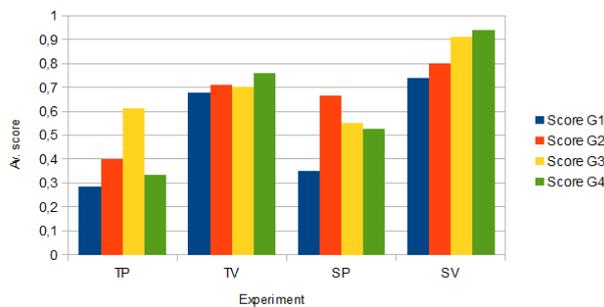


Figure 5: Graph of average scores (study groups)

When we further average the results obtained in particular experiments, better results in visual tasks become clearly visible (Figure 6). These results practically support our informal experiences and the hypothesis expressed within the introductory part of the paper.

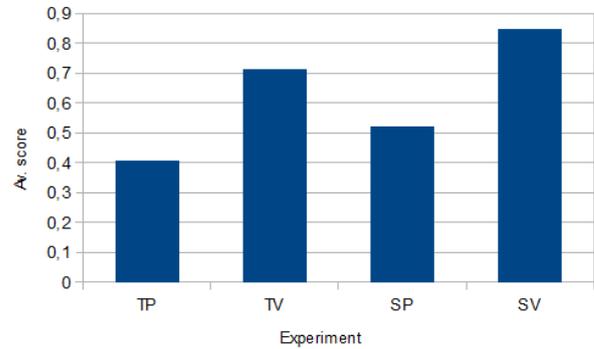


Figure 6: Graph of average scores (experiments)

The results can be also interpreted in a way, that algorithm visualizations provide the solid potential we would like to build upon and examine the new ways of utilizing them in the filed of algorithms and data structures education.

6 Proposal of the new approach and study supporting system

In order to cope with the situation and stimulate further students’ algorithmic and programming skills, while taking advantage of algorithm visualizations, we propose a new teaching approach supported by the prototype of new study supporting system. As it was mentioned before, the teaching approach is based on the idea that students would participate on creating simple visualizations, and this way interact with algorithm visualization on a higher level of the engagement taxonomy. The role of the proposed system is to provide the environment, that allows students to control the pre-arranged visualizations from their code by using simple programming constructs. The approach, together with the system are intended to be used in conjunction with other teaching methods, not to replace them. The prototype of the system with a working name DSAV (Figure 7) combines algorithm visualizations with programming tasks and so increases the engagement level and supports active learning.

As a proof of concept, we implemented the support for several (Bubble sort, Selection sort, Insertion sort, Quick sort, Heap sort, Merge sort) sorting algorithms [15] and algorithms for traversing binary trees (Figure 8) using various strategies (Inorder, Preorder, Postorder and Level-order). We would like to enhance the system in the future, and perspective areas for such enhancements would be visualizations of operations on lists, trees, or graphs.

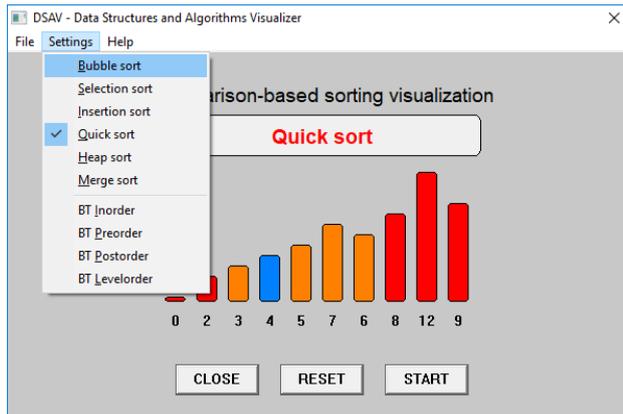


Figure 7: The working prototype of DSAV system

Technically, the system essentially consists of two parts: the main part, managing the user interface and visualization, and a separate thread implementing the algorithms to be visualized. There is a simple API consisting of several supporting operations which can be used appropriately by a programmer implementing a particular algorithm. The basic operation available is (`RedrawAndWait(int millis)`) telling the system to update the visualization according to current values in a data structure shared by both parts and wait for a specified amount of time.

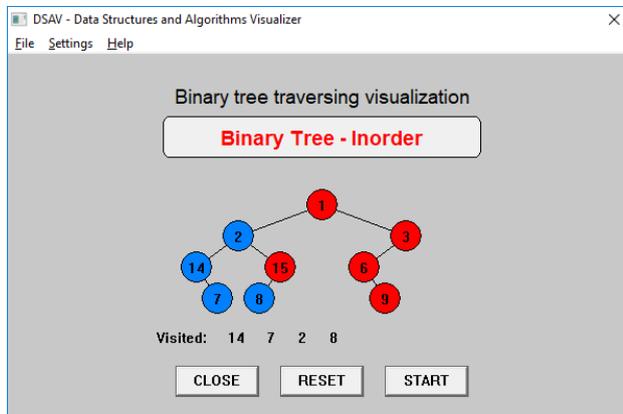


Figure 8: Visualization of traversing binary tree

In case of sorting algorithms we provide several simple API calls for rendering some elements of sorted array in different color. They can be useful in cases we want to put special emphasis on particular element (elements) of the sorted sequence (Figure 7). Some of them are given in the following list:

- `CSortClearColorArr()`
- `CSortSetColor(int index)`
- `CSortClrSetColor(int index)`

- `CSortSetColorInt(int begidx, int endidx)`
- `CSortClrSetColorRW(int index, int millis)`

As some of operations tend to be used often together, we also provide special calls for performing combined operations (e.g. `CSortClrSetColorRW(index, millis)` combines `CSortClrSetColor(index)` for rendering specified element in different color and `RedrawAndWait(millis)`). The reason for introducing such combined calls is to leave the code of particular algorithm closer to its original form. Figure 9 provides an example of using one of the operations within a simple sorting algorithm.

```
void selectionsort(int a[], int size)
{
    int i, j, k;
    for(i=0; i<size; i++)
    {
        int min = i;
        for(j=i+1; j<=9; j++)
            if(a[j]<a[min])
                min = j;
        k = a[i];
        CSortClrSetColorRW(min, 500);
        a[i] = a[min];
        a[min] = k;
        CSortClrSetColorRW(i, 500);
    }
}
```

Figure 9: A simple sorting algorithm implementation within the DSAV system

Analogically, there is a set of simple supporting API calls for visualization of tree traversing algorithms. Some of them are provided in the following list:

- `Btree3AGetDepth(int root)`
- `Btree3AGetLevel(int root, int d)`
- `BT3GetListVisited()`
- `Btree3ASetVisited(int root)`

An example of implementation of simple traversing algorithm is given in Figure 10.

Debugging outputs for particular algorithm can be printed using console output, if needed. The DSAV is a Win32 application, written in C/C++ programming language, since students mainly use this language in exercises within our subject presently.

7 Conclusion

Within the paper we described our experiments based on solving problems from given areas by students in two different ways. The first way was purely visual, accomplished

```

void inorder(int root){
    if(left[root]!=0)inorder(left[root]);
    Btree3ASetVisited(root);
    RedrawAndWait(300);
    if(right[root]!=0)inorder(right[root]);
}

```

Figure 10: A simple traversing algorithm implementation within the DSAV system

by using the Algomaster platform and the second way was based on programming a particular algorithm using C programming language.

The results acquired are presented and analyzed. We found that the scores achieved in visual tasks are usually significantly higher than the scores achieved in corresponding programming tasks. This correlates with our previous informal experiences and supports the validity of the hypothesis expressed within the introductory part of the paper.

The solution is proposed based on the idea of involving students into the process of creating algorithm visualizations. By the proposed solution we would like to help students not only to understand the basic principles of the particular algorithm in a convenient visual way, but also to stimulate their ability to implement it in particular programming language. Based on our experiences, confirmed by the results of accomplished experiments we believe, we should develop both of the skills in order to better prepare our students for their future professional career.

It would be interesting to further develop the proposed approach and the supporting system and study the contributions of the approach. Except the additional sorting algorithms, perspective areas for further extension would include visualizations of lists, trees, graphs or hash tables. We believe, that if system is enhanced properly and utilized in a right way, it would contribute to the quality of education in the subject. However, the further research is required in order to evaluate the benefits and efficiency of the proposed solution.

References

- [1] Bačková M., Porubán J.: Ergonomic vs. Domain Usability of User Interfaces, HSI 2013: 6th International Conference on Human System Interaction, June 6. - 8. 2013, Sopot, Poland, Piscataway, IEEE, 2013, pp. 1-8. <https://doi.org/10.1109/hsi.2013.6577817>
- [2] Benej M., Šimoňák S.: Algomaster platform extension for improved usability, Journal of Electrical and Electronics Engineering, vol. 10, no. 1, 2017, pp. 27-30.
- [3] Boyle E.A., Connolly T.M., Hailey T.: The role of psychology in understanding the impact of computer games, Entertainment Computing, vol. 2, no. 2, 2011, pp. 69-74. <https://doi.org/10.1016/j.entcom.2010.12.002>
- [4] Boyle E.A., Hailey T., Connolly T.M., Gray G., Earp J., Ott M., et al. An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games, Computers & Education, 94, 2016, pp. 178-192. <https://doi.org/10.1016/j.compedu.2015.11.003>
- [5] Dicheva D., Hodge A.: Active Learning through Game Play in a Data Structures Course, Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18), ACM, New York, NY, USA, 2018, pp. 834-839. <https://doi.org/10.1145/3159450.3159605>
- [6] Grissom S., McNally M.F., Naps T.: Algorithm visualization in CS education: comparing levels of student engagement, Proceedings of the 2003 ACM symposium on Software visualization (SoftVis '03), ACM, New York, USA, 87-94. <https://doi.org/10.1145/774833.774846>
- [7] Hundhausen C. D., Douglas S. A. and Stasko J. T.: A meta-study of algorithm visualization effectiveness, Journal of Visual Languages and Computing, 13, 2002, pp. 259-290. <https://doi.org/10.1006/jv1c.2002.0237>
- [8] Karavirta V., Shaffer C. A.: Creating Engaging Online Learning Material with the JSAV JavaScript Algorithm Visualization Library, IEEE Transactions on Learning Technologies, vol. 9, no. 2, pp. 171-183, April-June 2016. <https://doi.org/10.1109/tlt.2015.2490673>
- [9] Microsoft .NET, <https://dotnet.microsoft.com/>
- [10] Naps T. L., Röbling G, et al.: Exploring the role of visualization and engagement in computer science education, Working group reports from ITiCSE on Innovation and technology in computer science education (ITiCSE-WGR '02), ACM, New York, NY, USA, 131-152. <https://doi.org/10.1145/960568.782998>
- [11] Petri G., von Wangenheim C. G.: How games for computing education are evaluated? A systematic literature review, Computers & Education, vol. 107, April 2017, pp. 68-90. <https://doi.org/10.1016/j.compedu.2017.01.004>
- [12] Pietriková E., Chodarev S.: Towards Programmer Knowledge Profile Generation, Acta Electrotechnica et Informatica, vol. 16, no. 1, 2016,

- pp. 15-19. <https://doi.org/10.15546/aeei-2016-0003>
- [13] Rößling G.: A First Set of Design Patterns for Algorithm Animation, *Electronic Notes in Theoretical Computer Science*, Volume 224, 2009, pp. 67-76. <https://doi.org/10.1016/j.entcs.2008.12.050>
- [14] Rößling G., Mihaylov M., Saltmarsh J.: AnimalSense: Combining Automated Exercise Evaluations with Algorithm Animations, *Proceedings of the 16th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE 2011, Darmstadt, Germany, June 27-29, 2011*, pp. 298-302. <https://doi.org/10.1145/1999747.1999831>
- [15] Silváši F., Tomášek M.: Lean Formalization of Insertion Sort Stability and Correctness, *Acta Electrotechnica et Informatica*, vol. 18, no. 2, 2018, pp. 42-49. <https://doi.org/10.15546/aeei-2018-0015>
- [16] Šimoňák S.: Algorithm Visualization Using the VizAlgo Platform, *Acta Electrotechnica et Informatica*, vol. 13, no. 2, 2013, pp. 54-64. http://aei.tuke.sk/papers/2013/2/08_%C5%A0imo%C5%88%C3%A1k.pdf
- [17] Šimoňák S.: Using algorithm visualizations in computer science education, *Central European Journal of Computer Science*, vol. 4, no. 3, 2014, pp. 183-190. <https://doi.org/10.2478/s13537-014-0215-4>
- [18] Šimoňák S., Benej M.: Visualizing Algorithms and Data Structures Using the Algomaster Platform, *Journal of Information, Control and Management Systems*, vol. 12, no. 2, 2014, pp. 189-201.
- [19] Šimoňák S.: Algorithm visualizations as a way of increasing the quality in computer science education, *SAMI 2016, Danvers, IEEE, 2016*, pp. 153-157. <https://doi.org/10.1109/sami.2016.7422999>
- [20] Urquiza-Fuentes J., Velázquez-Iturbide J. Á.: Pedagogical Effectiveness of Engagement Levels - A Survey of Successful Experiences, *Electronic Notes in Theoretical Computer Science*, Volume 224, 2009, pp. 169-178. <https://doi.org/10.1016/j.entcs.2008.12.061>

Similarity Measure of Multiple Sets and its Application to Pattern Recognition

Shijina V, Adithya Unni and Sunil Jacob John

National Institute of Technology Calicut, Kerala, India- 673601

E-mail: shiji.chan@gmail.com, adithyaunni1998@gmail.com and sunil@nitc.ac.in

<http://nitc.ac.in/index.php?url=users/view/178/11/3>

Keywords: fuzzy sets, similarity measure, multiple sets, pattern recognition

Received: July 17, 2019

Multiple set is a newborn member of the family of generalized sets, which can model uncertainty together with multiplicity. It has the power to handle numerous uncertain features of objects in a multiple way. Multiple set theory has the edge over the well established fuzzy set theory by its capability to handle uncertainty and multiplicity simultaneously. Similarity measure of fuzzy sets is well addressed in literature and has found prominent applications in various domains. As multiple set is an efficient generalization of fuzzy set, the concept and theory of similarity measure can be extended to multiple set theory and can be developed probable applications in various real-life problems. This paper introduces the concept of similarity measure of multiple sets and proposes two different similarity measures of multiple sets and investigates their properties. Finally, this work substantiates application of the concept of similarity measure of multiple sets to pattern recognition. A numerical illustration demonstrates the effectiveness of the proposed technique to this application.

Povzetek: V članku je predstavljena teorija podobnosti multipnih množic z namenom uporabe prepoznavanja vzorcev.

1 Introduction

Various mathematical models are available in the literature to represent the concepts like uncertainty, vagueness and inexactness. Such models includes fuzzy sets, L-fuzzy sets [1], multisets [2], rough sets[3], intuitionistic fuzzy sets[4], fuzzy multisets[5], vague sets[6], multi fuzzy sets[7], etc. Each of these models has advanced into an elaborated theory and has numerous practical applications.[3]

A fuzzy set is characterized by a membership function which assigns a grade of membership to each object in the universal set. Even though, the concept of fuzzy set is strong enough to handle uncertain data successfully, it can manage only one uncertain feature of the object at a time. Also, fuzzy set fails to handle the multiplicity of objects. Later, The notion of fuzzy multiset was defined as an extension of a fuzzy set. Fuzzy multiset gives fuzzy membership values for identical copies of each object. The main advantage of fuzzy multiset over fuzzy set is that it can handle the multiplicity of objects. However, it can handle only one feature of the object at a time. On the other hand, multi fuzzy set is also an extension of fuzzy set, and gives fuzzy membership values for different features of objects. The main advantage of multi fuzzy set over fuzzy set is that it can simultaneously manage numerous uncertain characteristics of objects, but fails to handle the multiplicity of objects. Recently, multiple set is introduced to model uncertainty together with multiplicity. The advantage of multiple set lies in the fact that it simultaneously accumulates numerous uncertain features of objects together

with its multiplicity, in a better way. It was put forward by Shijina et al.[8, 9] as a generalization of fuzzy set, multiset, fuzzy multiset and multi fuzzy set. Later, Shijina et al.[10, 11] defined more operations, viz. aggregation operators and matrix norms on multiple sets. Then, the concept of relation on multiple sets is introduced and applied this concept in medical diagnosis problem[12]. As a continuation, this work is aspired as an attempt to extend the concept of similarity measure to multiple sets.

Measuring the similarity between objects plays a crucial role in many real life problems involving image processing, image retrieval, image compression, pattern recognition, clustering, information retrieval problems, etc. Many measures of similarity have been proposed and researched in literature and it has been shown that similarity measure is proficient in coping with uncertain information. For example, the theory of fuzzy sets, introduced by Zadeh[13], is a successful approach in confronting uncertainty. Fuzzy set has enormous power to describe the objective world that we live in and the strength of fuzzy set has transpired in several real life applications. Zadeh himself initiated the idea of similarity measure of fuzzy sets[14]. Later, similarity measure of fuzzy sets has been explored widely by many researchers[15, 16, 17, 18, 19, 20, 21, 22, 23] and have applied them to real life problems involving pattern recognition[24], image processing[25, 26, 27, 28, 29, 30], etc. As an extension of fuzzy set theory, intuitionistic fuzzy set theory has found to be highly useful in dealing with imprecision and uncertainty. Many different similarity measures between intuitionistic fuzzy

sets have been proposed and are extensively applied to many areas such as decision making[31, 32], pattern recognition[33, 34, 35, 36, 37, 38, 39], etc. As a combined concept of intuitionistic fuzzy set and interval valued fuzzy set, Atanassov[40] introduced interval valued intuitionistic fuzzy sets. It greatly furnishes the additional capability to deal with vague information and model non-statistical uncertainty by providing both membership interval and non-membership intervals. Similarity measure of interval valued intuitionistic fuzzy sets was also proposed and it has found applications in pattern recognition and multi-criteria decision making[41]. Type-2 fuzzy sets, which is an extension of fuzzy sets was also proposed by Zadeh[42]. Their membership values are fuzzy sets on the interval $[0, 1]$. Type-2 fuzzy sets can improve certain kinds of inference better than fuzzy sets with increasing imprecision, uncertainty and fuzziness in information. Hung and Yang[43] presented a similarity measure of type-2 fuzzy sets based on the fuzzy Hausdor distance. There were further studies of similarity measures on Type-2 fuzzy sets[44, 45, 46] and have found applications in clustering[47, 48, 49], pattern recognition[50], students' evaluation[51], etc. Hesitant fuzzy set was first introduced by Torra[52] and Torra and Narukawa[53]. It permits the membership degree of an element to a set comprising of several possible values between 0 and 1. Hesitant fuzzy sets are very useful in dealing with situations where people are hesitant in providing their preference over objects in a decision making process. Therefore hesitant fuzzy set has played a significant role in the uncertain system and received much attention from researchers. Similarity measures of hesitant fuzzy sets[54] have been proposed, but it has not yet gained wide acceptance.

Analogously, several similarity measures between sets have been proposed and have found many real life applications. But, here we will restrict our attention to the theory of similarity measures of fuzzy sets and its various applications, so that it can be explored to define the similarity measure of multiple sets. Before presenting the theory of similarity measure of fuzzy sets, it is desirable to have a short discussion on its application in day-to-day life. So, in the following, the potential of similarity measure of fuzzy sets in real life applications is reviewed.

Weken et al.[25] gave an overview of similarity measures of fuzzy sets which can be applied to images. These similarity measures are all pixel-based and fail to produce satisfactory results consistently. To overcome this drawback, Weken et al.[26] extended their work to propose similarity measures based on neighbourhoods so that the relevant structures of the images are observed better. In his survey paper on similarity measures of fuzzy sets, Weken et al.[27] established measures for image comparison. The same authors presented an overview of the possible application of similarity measures of fuzzy sets to colour images in[28]. Nachtegael et al.[30] presented a color image retrieval system using a specific similarity measure of fuzzy sets. Li et al.[55] presented a faster algorithm on similarity measure using cen-

ter of gravity of fuzzy sets in content-based image retrieval. The discussion in [55] nearly covers all the similarity measures of fuzzy sets, which may be greatly helpful to both the development and application of fuzzy set theory for content based image retrieval. Chen et al.[29] proposed a novel algorithm viz., normalized fuzzy similarity measure to deal with the nonlinear distortion in finger print images. Chaira and Ray[24] presented a region extraction algorithm to identify a color region similar to the query image from an image database containing images with different types of colors. Here, the matching process is based on similarity measure of fuzzy sets between the query image and the images in the database. Capitaine[56] proposed a general framework of designing similarity measures based on residual implication functions. They presented some new families of parametric similarity measures using parametric residual implications and modeled an algorithm to learn the parameter of each similarity measure based on relevance degrees. El-Sayed and Aboelwafa[57] introduced a new approach for face recognition based on similarity measure of fuzzy sets. Xu et al.[58] proposed a new similarity measure of fuzzy sets based on the extension of the Dice and cosine similarity measures and then applied the variation coefficient similarity to the emergency group decision-making problems. Also, they gave a practical example to evaluate the emergency management capability of major snow disaster in Hunan province of China. Baccour[59] applied similarity measures of fuzzy sets reported in existing literature to classification of shapes, mosaic recognition and Arabic sentence recognition.

As discussed above, similarity measure of fuzzy sets have found widespread application in various fields such as image processing, pattern recognition, decision making, etc. Multiple set, which is an extension of fuzzy set, is capable of handling uncertainty and multiplicity simultaneously. Motivated by the benefits of similarity measure of fuzzy sets, this work intends to extend similarity measure to multiple sets. This paper proposes two different types of similarity measures- one is based on similarity measure of fuzzy sets; other one is based on the similarity measure of fuzzy sets and fuzzy aggregation operators. We strongly believe that similarity measure of multiple set can handle uncertain information in a better way. It must, therefore, have a better scope of real life applications. To substantiate our claim, we have applied the concept of similarity measure of multiple sets to pattern recognition, which is the first of its kind. The rest of the paper is organized as follows. In section 2, we briefly review some standard facts on multiple sets and the similarity measures of fuzzy sets. In section 3, we derive two interesting formulas for similarity measure on multiple sets and establish some of their properties. In section 4, we indicate how these techniques may be used to pattern recognition problems. In section 5, we end the paper by encapsulating the main conclusions.

2 Preliminaries

In this section, we first give some basic concepts related to multiple sets. Then, we proceed with a brief exposition of similarity measures of fuzzy sets.

Throughout this paper, the following notations are used. $R^+ = [0, \infty)$; X is the universe of discourse; $|X|$ is the cardinality of X ; capital letters A, B, C , etc. are fuzzy sets on X and also represents corresponding membership functions; $A(x)$ is the fuzzy membership value of the element x in X ; ϕ is the fuzzy set with all membership values equal to 0; I is the fuzzy set with all membership values equal to 1; M is the fuzzy set with all membership values equal to 0.5; \bar{A} is the complement of fuzzy set A ; $FS(X)$ is the class of all fuzzy sets of X ; $P(X)$ is the class of all crisp subsets of X .

Let $\mathbb{M} = \mathbb{M}_{n \times k}([0, 1])$ denotes the set of all matrices of order $n \times k$ with entries from $[0, 1]$ and for $\epsilon \in [0, 1]$, $[\epsilon]_{n \times k}$ denotes the matrix in \mathbb{M} with all its entries equal to ϵ .

Definition 2.1. Let $M = [M_{ij}], N = [N_{ij}] \in \mathbb{M}$. Then,

1. $M \leq N \iff M_{ij} \leq N_{ij}$ for every $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, k$.
2. $M \geq N \iff M_{ij} \geq N_{ij}$ for every $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, k$.
3. $M = N \iff M_{ij} = N_{ij}$ for every $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, k$.
4. Join of M and N , denoted by $M \vee N$, is a matrix in \mathbb{M} defined by $(M \vee N)_{ij} = M_{ij} \vee N_{ij}$ for every $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, k$.
5. Meet of M and N , denoted by $M \wedge N$, is a matrix in \mathbb{M} defined by $(M \wedge N)_{ij} = M_{ij} \wedge N_{ij}$ for every $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, k$.

From this definition it can be noted that, $\langle M, \leq, [0]_{n \times k}, [1]_{n \times k} \rangle$ is a bounded lattice.

2.1 Multiple sets

Multiple set is a unified structure to represent numerous uncertain features of objects simultaneously, in a multiple way. Multiple set utilizes distinct fuzzy membership functions to delineate each uncertain features of the object and assigns various values to each membership function according to the multiplicity. This is symbolized by assigning a matrix to each object, where each row in the matrix indicates distinct fuzzy membership function corresponding to each feature of the object. Further, entries in a row points out different values of the corresponding membership function according to its multiplicity. Multiple set can be defined as follows:

Definition 2.2. Let X be a non-empty crisp set called the universal set and A_1, A_2, \dots, A_n be n distinct fuzzy sets of X . For each $i = 1, 2, \dots, n$, $A_i^1(x), A_i^2(x), \dots, A_i^k(x)$

are membership values of the fuzzy set A_i for k identical copies of the element $x \in X$, in descending order. Then, multiple set \mathcal{A} of order (n, k) over X is an object of the form

$$\mathcal{A} = \{(x, \mathcal{A}(x)) : x \in X\}$$

where for each $x \in X$ its membership value is an $n \times k$ matrix in \mathbb{M} given by

$$\mathcal{A}(x) = \begin{bmatrix} A_1^1(x) & A_1^2(x) & \dots & A_1^k(x) \\ A_2^1(x) & A_2^2(x) & \dots & A_2^k(x) \\ \vdots & \vdots & \dots & \vdots \\ A_n^1(x) & A_n^2(x) & \dots & A_n^k(x) \end{bmatrix}$$

The matrix $\mathcal{A}(x)$ is called the membership matrix of the element x .

Note that, fuzzy sets A_1, A_2, \dots, A_n evaluates n distinct properties of objects and are called underlying fuzzy sets of the multiple set \mathcal{A} . Further, each underlying fuzzy set A_i corresponds to k fuzzy sets $A_i^j = \{(x, A_i^j(x)) : x \in X\}$, for $j = 1, 2, \dots, k$. Clearly, for every $i = 1, 2, \dots, n$, $A_i^1 \supset A_i^2 \supset \dots \supset A_i^k$.

The universal multiple set \mathcal{X} is a multiple set of order (n, k) over X for which the membership matrix for each $x \in X$ is $[1]_{n \times k}$. The empty multiple set Φ is a multiple set of order (n, k) over X for which the membership matrix for each $x \in X$ is $[0]_{n \times k}$.

The set of all multiple sets of order (n, k) over X is denoted by $MS_{(n,k)}(X)$. It is perceived that a multiple set \mathcal{A} of order (n, k) over X can be viewed as a function $\mathcal{A} : X \rightarrow \mathbb{M}$, which maps each $x \in X$ to its $n \times k$ membership matrix $\mathcal{A}(x)$ in \mathbb{M} .

As an example, multiple set can be used to represent the evaluation of a set of students under the characteristics of intelligence, extra curricular activities, communication skill and personality by three experts.

Example 2.3. Suppose $X = \{x_1, x_2, x_3\}$ is the universal set of students under consideration and there is a panel consisting of three experts evaluating the students under the criteria of intelligence, extra curricular activities, communication skill and personality. Then the performance of the students can be represented by a multiple set of order $(4, 3)$ as follows:

$$\mathcal{A} = \{(x_1, \mathcal{A}(x_1)), (x_2, \mathcal{A}(x_2)), (x_3, \mathcal{A}(x_3))\}$$

where $\mathcal{A}(x_i)$ for $i = 1, 2, 3$ are 4×3 matrices given as follows;

$$\mathcal{A}(x_1) = \begin{bmatrix} 0.7 & 0.6 & 0.5 \\ 0.6 & 0.5 & 0.4 \\ 0.7 & 0.5 & 0.3 \\ 0.9 & 0.9 & 0.8 \end{bmatrix} \quad \mathcal{A}(x_2) = \begin{bmatrix} 0.8 & 0.6 & 0.6 \\ 0.6 & 0.5 & 0.4 \\ 0.7 & 0.5 & 0.4 \\ 0.9 & 0.8 & 0.7 \end{bmatrix}$$

$$\mathcal{A}(x_3) = \begin{bmatrix} 0.8 & 0.7 & 0.5 \\ 0.7 & 0.6 & 0.4 \\ 0.7 & 0.4 & 0.4 \\ 0.8 & 0.8 & 0.7 \end{bmatrix}$$

Here, first, second, third and fourth row of the membership matrix indicates the fuzzy membership function corresponding to the features intelligence, extra curricular activities, communication skill and personality, respectively. Corresponding to each feature, three entries in the row are the values given by the three experts, written in descending order. For example, for the student x_1 the membership values corresponding to intelligence are 0.7, 0.6 and 0.5, corresponding to extra curricular activities are 0.6, 0.5 and 0.4 and so on.

Next, we discuss the standard operations on multiple sets. Let \mathcal{A} and \mathcal{B} be two multiple sets in $MS_{(n,k)}(X)$.

Definition 2.4. \mathcal{A} is a subset of \mathcal{B} , denoted as $\mathcal{A} \subseteq \mathcal{B}$, if and only if $\mathcal{A}(x) \leq \mathcal{B}(x)$ for every $x \in X$.

Definition 2.5. \mathcal{A} is equal to \mathcal{B} , denoted as $\mathcal{A} = \mathcal{B}$, if and only if $\mathcal{A} \subseteq \mathcal{B}$ and $\mathcal{B} \subseteq \mathcal{A}$ that is, if and only if $\mathcal{A}(x) = \mathcal{B}(x)$ for every $x \in X$.

Definition 2.6. The union of \mathcal{A} and \mathcal{B} is a multiple set in $MS_{(n,k)}(X)$, denoted as $\mathcal{A} \cup \mathcal{B}$, whose membership matrix is $(\mathcal{A} \cup \mathcal{B})(x) = \mathcal{A}(x) \vee \mathcal{B}(x)$ for every $x \in X$.

Definition 2.7. The intersection of \mathcal{A} and \mathcal{B} is a multiple set in $MS_{(n,k)}(X)$, denoted as $\mathcal{A} \cap \mathcal{B}$, whose membership matrix is $(\mathcal{A} \cap \mathcal{B})(x) = \mathcal{A}(x) \wedge \mathcal{B}(x)$ for every $x \in X$.

Definition 2.8. The complement of \mathcal{A} is a multiple set in $MS_{(n,k)}(X)$, denoted as $\bar{\mathcal{A}}$, whose membership matrix for each $x \in X$ is an $n \times k$ matrix, $\bar{\mathcal{A}}(x) = [\bar{A}_i^j(x)]$ where $\bar{A}_i^j(x) = 1 - A_i^{(k-j+1)}(x)$ for every $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, k$.

2.2 Similarity measure of fuzzy sets

Being an important topic in the theory of fuzzy sets, similarity measure of fuzzy sets has been investigated extensively by many researchers from different point of view. But, there does not exist a unique definition of similarity measure of fuzzy sets. There do exist many special purpose definitions which have been employed with success in cluster analysis, pattern recognition, image processing, classification, diagnostics and many other fields. Recently, several similarity measures are proposed and used for various purposes. For example, Zwick et al.[15] reviewed 19 measures of similarity and compared their performance in a behavioral experiment. Xuecheng[16] systematically gave an axiom definition of similarity measure of fuzzy sets as:

Definition 2.9. A real function $S : FS(X) \times FS(X) \rightarrow \mathbb{R}^+$ is called a similarity measure, if S has the following properties:

1. $S(A, B) = S(B, A)$ for all $A, B \in FS(X)$.
2. $S(D, \bar{D}) = 0$ for all $D \in P(X)$.
3. $S(C, C) = \max_{A, B \in FS(X)} S(A, B)$ for all $C \in FS(X)$.

4. For all $A, B, C \in FS(X)$, if $A \subseteq B \subseteq C$, then $S(A, B) \geq S(A, C)$ and $S(B, C) \geq S(A, C)$.

On account of this definition, Xuecheng proposed a similarity measure on the basis of a measurable function with respect to borel field \mathcal{B}^1 : Let $X = [0, 1]$ and

$$\mathcal{F} = \{A \in FS(X); A(x) \text{ is a measurable function with respect to borel field } \mathcal{B}^1\}$$

Then, for $p \geq 1$

$$S_p(A, B) = 1 - \left(\int_0^1 |A(x) - B(x)|^p dx \right)^{1/p} \tag{2.1}$$

for all $A, B \in \mathcal{F}$, is a similarity measure on \mathcal{F} .

Pappis and Karacapilidis[17] presented three similarity measures as follows:

- (1) Measure based on the operations of union and intersection:

$$S(A, B) = \frac{\sum_{x \in X} \min\{A(x), B(x)\}}{\sum_{x \in X} \max\{A(x), B(x)\}} \tag{2.2}$$

- (2) Measure based on the maximum difference:

$$S(A, B) = 1 - \max_{x \in X} \{|A(x) - B(x)|\} \tag{2.3}$$

- (3) Measure based on the difference and the sum of grades of membership:

$$S(A, B) = 1 - \frac{\sum_{x \in X} |A(x) - B(x)|}{\sum_{x \in X} (A(x) + B(x))} \tag{2.4}$$

The authors summarized that similarity measures (2.2) and (2.4) satisfies the following properties: xcvbnm,.

- (p1) $S(A, B) = S(B, A)$.
- (p2) $A = B \Leftrightarrow S(A, B) = 1$.
- (p3) $A \cap B = \phi \Leftrightarrow S(A, B) = 0$.
- (p4) $S(A, \bar{A}) = 1 \Leftrightarrow A = M$.
- (p5) $S(A, \bar{A}) = 0 \Leftrightarrow A = I$ or $A = \phi$.

The similarity measure (2.3) satisfies properties (p1), (p2) and (p4) and

$$(p3^*) A \cap B = \phi \Leftrightarrow S(A, B) = 1 - \max_{x \in X} \{A(x), B(x)\}.$$

$$(p5^*) S(A, \bar{A}) = 0 \Leftrightarrow A \text{ and } \bar{A} \text{ are normal fuzzy sets.}$$

Hyung et al.[18] proposed a similarity measure of fuzzy sets using maximum and minimum operators:

$$S(A, B) = \max_{x \in X} \min\{A(x), B(x)\} \tag{2.5}$$

and showed that it satisfies the properties (p1) and

(p6) The similarity degree is bounded:
 $0 \leq S(A, B) \leq 1$.

(p7) If A and B are normalized and $A = B$ then
 $S(A, B) = 1$.

(p8) $A \cap B = \phi \Rightarrow S(A, B) = 0$.

(p9) If A and B are crisp sets, then $S(A, B) = 0$ if $A \cap B = \phi$ and $S(A, B) = 1$ if $A \cap B \neq \phi$.

Chen et al.[20] extended the work of Pappis to further investigate measures of similarity of fuzzy values. They proposed 3 similarity measures:

(1) Measure based on geometric distance model:

$$S(A, B) = 1 - \frac{\sum_{x \in X} |A(x) - B(x)|}{|X|} \tag{2.6}$$

(2) Measure based on the set theoretic approach:

$$S(A, B) = \sup_{x \in X} (A \cap B)(x) \tag{2.7}$$

(3) Measure based on the matching function[60]:

$$S(A, B) = \frac{\sum_{x \in X} A(x)B(x)}{\max \left\{ \sum_{x \in X} A(x)^2, \sum_{x \in X} B(x)^2 \right\}} \tag{2.8}$$

They summarized that similarity measure (2.6) satisfies the properties (p1), (p2), (p4) and (p5) and fails to satisfy (p3), similarity measure (2.7) satisfies the properties (p1) and (p3) and fails to satisfy (p2), (p4) and (p5) and similarity measure (2.8) satisfies the properties (p1) to (p5). Later, Wang et al.[19] made a comparative study of similarity measures. They commended on the study of similarity measures introduced by Pappis[17]. Also, they introduced a new class of similarity measures extracted from the work of Bandler and Kohout on fuzzy power sets[61], as:

$$S(A, B) = \min \left\{ \inf_{x \in X} I(A(x), B(x)), \inf_{x \in X} I(B(x), A(x)) \right\} \tag{2.9}$$

where I is any fuzzy implication operator. Wang[21] proposed two new similarity measures of fuzzy sets:

$$S(A, B) = \frac{\sum_{x \in X} \left(\frac{\min\{A(x), B(x)\}}{\max\{A(x), B(x)\}} \right)}{|X|} \tag{2.10}$$

$$S(A, B) = \frac{\sum_{x \in X} (1 - |A(x) - B(x)|)}{|X|} \tag{2.11}$$

They examined that similarity measures (2.10) and (2.11) satisfies the Definition 2.9. They also made a comparison between similarity measures put forward by them with that

of [17] and [18]. Razaeei et al.[22] developed a new similarity measure of fuzzy sets based on their relative sigma count:

$$S(A, B) = \frac{\sum_{x \in X} \min \{A(x), B(x)\}}{\max \left\{ \sum_{x \in X} A(x), \sum_{x \in X} B(x) \right\}} \tag{2.12}$$

where $A \neq \phi$ or $B \neq \phi$ and also define $S(\phi, \phi) = 1$. They probed that this similarity measure satisfies the Definition 2.9 and also satisfies the properties (p1) to (p5).

3 Similarity measure of multiple sets

In this section, we first introduce the axiom definition of similarity measure of multiple sets. Let $\xi_{(n,k)}(X)$ be the subset of $MS_{(n,k)}(X)$, which is the collection of all multiple sets over X whose membership matrices are either $[0]_{n \times k}$ or $[1]_{n \times k}$.

Definition 3.1. A real function $S : MS_{(n,k)}(X) \times MS_{(n,k)}(X) \rightarrow \mathbb{R}^+$ is called a similarity measure of multiple sets, if S satisfies the following axioms;

1. $S(A, B) = S(B, A)$ for all $A, B \in MS_{(n,k)}(X)$.
2. $S(D, \bar{D}) = 0$ for all $D \in \xi_{(n,k)}(X)$.
3. $S(C, C) = \max_{A, B \in MS_{(n,k)}(X)} S(A, B)$ for all $C \in MS_{(n,k)}(X)$.
4. For all $A, B, C \in MS_{(n,k)}(X)$, if $A \subseteq B \subseteq C$, then $S(A, B) \geq S(A, C)$ and $S(B, C) \geq S(A, C)$

In the following, we propose two similarity measures between multiple sets, one is based on the similarity measure of fuzzy sets; other is based on similarity measure of fuzzy sets and a fuzzy aggregation operator.

Let S be any similarity measure of fuzzy sets satisfying the Definition 2.9. For multiple sets A and B in $MS_{(n,k)}(X)$, denote

$$S(A, B) = \sum_{i=1}^n \max_{j=1,2,\dots,k} S(A_i^j, B_i^j) \tag{3.1}$$

Theorem 3.2. $S(A, B)$ is a similarity measure between the multiple sets A and B in X .

Proof. Axioms (1) and (2) are obvious, respectively, from axioms (1) and (2) of Definition 2.9 for fuzzy similarity measure S .

Axiom(3): Let C be any multiple set in $MS_{(n,k)}(X)$. Clearly, we have

$$S(C, C) \leq \max_{A, B \in MS_{(n,k)}(X)} S(A, B) \tag{3.2}$$

Now, for any multiple sets $A, B \in MS_{(n,k)}(X)$, from axiom (3) of Definition 2.9 for fuzzy similarity measure S ,

we have $S(C_i^j, C_i^j) \geq S(A_i^j, B_i^j)$ for every $j = 1, 2, \dots, k$ and $i = 1, 2, \dots, n$. Therefore,

$$\max_{j=1,2,\dots,k} S(C_i^j, C_i^j) \geq \max_{j=1,2,\dots,k} S(A_i^j, B_i^j)$$

for every $i = 1, 2, \dots, n$, which implies

$$\sum_{i=1}^n \max_{j=1,2,\dots,k} S(C_i^j, C_i^j) \geq \sum_{i=1}^n \max_{j=1,2,\dots,k} S(A_i^j, B_i^j)$$

So, we have $\mathcal{S}(\mathcal{C}, \mathcal{C}) \geq \mathcal{S}(\mathcal{A}, \mathcal{B})$ for all $\mathcal{A}, \mathcal{B} \in MS_{(n,k)}(X)$. Therefore,

$$\mathcal{S}(\mathcal{C}, \mathcal{C}) \geq \max_{\mathcal{A}, \mathcal{B} \in MS_{(n,k)}(X)} \mathcal{S}(\mathcal{A}, \mathcal{B}) \tag{3.3}$$

Combining inequalities (3.2) and (3.3), it follows that

$$\mathcal{S}(\mathcal{C}, \mathcal{C}) = \max_{\mathcal{A}, \mathcal{B} \in MS_{(n,k)}(X)} \mathcal{S}(\mathcal{A}, \mathcal{B})$$

Axiom(4): Suppose \mathcal{A}, \mathcal{B} and \mathcal{C} are multiple sets in $MS_{(n,k)}(X)$ such that $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{C}$. Then $A_i^j \subseteq B_i^j \subseteq C_i^j$ for every $j = 1, 2, \dots, k$ and $i = 1, 2, \dots, n$. Then, from axiom (4) of Definition 2.9 for fuzzy similarity measure S , we have $S(A_i^j, B_i^j) \geq S(A_i^j, C_i^j)$ for every $j = 1, 2, \dots, k$ and $i = 1, 2, \dots, n$. Therefore,

$$\sum_{i=1}^n \max_{j=1,2,\dots,k} S(A_i^j, B_i^j) \geq \sum_{i=1}^n \max_{j=1,2,\dots,k} S(A_i^j, C_i^j)$$

and hence $\mathcal{S}(\mathcal{A}, \mathcal{B}) \geq \mathcal{S}(\mathcal{A}, \mathcal{C})$. In a similar way, we can prove that $\mathcal{S}(\mathcal{B}, \mathcal{C}) \geq \mathcal{S}(\mathcal{A}, \mathcal{C})$. That is, $\mathcal{S}(\mathcal{A}, \mathcal{B})$ satisfies all the axioms of Definition 3.1. Thus $\mathcal{S}(\mathcal{A}, \mathcal{B})$ is a similarity measure between the multiple sets \mathcal{A} and \mathcal{B} in X . \square

Example 3.3. Let $X = \{x_1, x_2, x_3\}$ be the universal set and \mathcal{A} and \mathcal{B} be multiple sets in $MS_{(3,3)}(X)$ given by the following membership matrices;

$$\begin{aligned} \mathcal{A}(x_1) &= \begin{bmatrix} 0.9 & 0.9 & 0.8 \\ 0.5 & 0.5 & 0.5 \\ 0.3 & 0.2 & 0.1 \end{bmatrix} & \mathcal{A}(x_2) &= \begin{bmatrix} 0.9 & 0.5 & 0.5 \\ 0.6 & 0.5 & 0.3 \\ 0.5 & 0.4 & 0.3 \end{bmatrix} \\ \mathcal{A}(x_3) &= \begin{bmatrix} 0.8 & 0.7 & 0.7 \\ 0.6 & 0.5 & 0.3 \\ 0.8 & 0.7 & 0.6 \end{bmatrix} & \mathcal{B}(x_1) &= \begin{bmatrix} 0.9 & 0.5 & 0.5 \\ 0.8 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.2 \end{bmatrix} \\ \mathcal{B}(x_2) &= \begin{bmatrix} 0.8 & 0.5 & 0.5 \\ 0.6 & 0.6 & 0.4 \\ 0.6 & 0.5 & 0.2 \end{bmatrix} & \mathcal{B}(x_3) &= \begin{bmatrix} 0.9 & 0.7 & 0.5 \\ 0.8 & 0.6 & 0.5 \\ 0.7 & 0.5 & 0.3 \end{bmatrix} \end{aligned}$$

Consider 3 similarity measures S_1, S_2 and S_3 of fuzzy sets, given by the equations 2.10, 2.11 and 2.12, respectively. Then, from simple mathematical calculations, we have the similarity measures $\mathcal{S}(\mathcal{A}, \mathcal{B})$ between multiple sets \mathcal{A} and \mathcal{B} based on similarity measure S_1 is 2.584, based on S_2 is 2.733 and based on S_3 is 2.677.

Using the properties of fuzzy similarity measure and definition of similarity measure of multiple set the following properties can be proved easily:

Theorem 3.4. Let \mathcal{A} and \mathcal{B} be multiple sets in $MS_{(n,k)}(X)$ and \mathcal{M} be the multiple set in $MS_{(n,k)}(X)$ for which membership matrices for each $x \in X$ is $[0.5]_{n \times k}$. Then, similarity measure $\mathcal{S}(\mathcal{A}, \mathcal{B})$ defined in equation (3.1) satisfies the following properties:

1. Suppose fuzzy similarity measure S satisfies the property $A = B \Leftrightarrow S(A, B) = 1$. Then $\mathcal{A} = \mathcal{B} \Rightarrow \mathcal{S}(\mathcal{A}, \mathcal{B}) = n$.
2. Suppose fuzzy similarity measure S satisfies the property $A \cap B = \phi \Leftrightarrow S(A, B) = 0$. Then $\mathcal{A} \cap \mathcal{B} = \Phi \Leftrightarrow \mathcal{S}(\mathcal{A}, \mathcal{B}) = 0$.
3. Suppose fuzzy similarity measure S satisfies the properties $A = M \Leftrightarrow S(A, \bar{A}) = 1$ and $0 \leq S(A, B) \leq 1$. Then $\mathcal{A} = \mathcal{M} \Rightarrow \mathcal{S}(\mathcal{A}, \bar{\mathcal{A}}) = n$.
4. Suppose fuzzy similarity measure S satisfies the property $S(A, \bar{A}) = 0 \Leftrightarrow A = I$ or $A = 0$. Then $\mathcal{S}(\mathcal{A}, \bar{\mathcal{A}}) = 0 \Leftrightarrow \mathcal{A} \in \xi_{(n,k)}(X)$.

Remark 3.5. 1. Converse of (1) in Theorem 3.4 need not be true. For example, let $X = \{x_1, x_2, x_3\}$ be the universal set and \mathcal{A} and \mathcal{B} be multiple sets in $MS_{(3,3)}(X)$ given by the following membership matrices;

$$\begin{aligned} \mathcal{A}(x_1) &= \begin{bmatrix} 0.8 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.1 \end{bmatrix} \\ \mathcal{A}(x_2) &= \begin{bmatrix} 0.9 & 0.5 & 0.5 \\ 0.6 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.3 \end{bmatrix} \\ \mathcal{A}(x_3) &= \begin{bmatrix} 0.8 & 0.5 & 0.4 \\ 0.5 & 0.5 & 0.2 \\ 0.6 & 0.5 & 0.2 \end{bmatrix} \\ \mathcal{B}(x_1) &= \begin{bmatrix} 0.9 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.4 \\ 0.5 & 0.5 & 0.2 \end{bmatrix} \\ \mathcal{B}(x_2) &= \begin{bmatrix} 0.8 & 0.5 & 0.5 \\ 0.6 & 0.5 & 0.5 \\ 0.6 & 0.5 & 0.2 \end{bmatrix} \\ \mathcal{B}(x_3) &= \begin{bmatrix} 0.8 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.1 \end{bmatrix} \end{aligned}$$

Choose the fuzzy similarity measure S_1 given by the equation (2.10). From simple calculations, we obtain $\mathcal{S}(\mathcal{A}, \mathcal{B}) = 3$. But, here $\mathcal{A} \neq \mathcal{B}$.

2. Converse of (3) in Theorem 3.4 need not be true. Consider the multiple set \mathcal{A} given in above example. Then complement of \mathcal{A} is given by the following membership matrices;

$$\bar{\mathcal{A}}(x_1) = \begin{bmatrix} 0.5 & 0.5 & 0.2 \\ 0.5 & 0.5 & 0.5 \\ 0.9 & 0.5 & 0.5 \end{bmatrix}$$

$$\bar{\mathcal{A}}(x_2) = \begin{bmatrix} 0.5 & 0.5 & 0.1 \\ 0.6 & 0.5 & 0.5 \\ 0.7 & 0.5 & 0.5 \end{bmatrix}$$

$$\bar{\mathcal{A}}(x_3) = \begin{bmatrix} 0.6 & 0.5 & 0.2 \\ 0.8 & 0.5 & 0.5 \\ 0.8 & 0.5 & 0.4 \end{bmatrix}$$

Choose the fuzzy similarity measure S_1 given by the equation (2.10). From simple calculations, we obtain $S(\mathcal{A}, \bar{\mathcal{A}}) = 3$. But, here $\mathcal{A} \neq \mathcal{M}$.

Based on the similarity measure of fuzzy sets and fuzzy aggregation operator, we give a similarity measure formula for multiple sets as follows:

Let S be any similarity measure of fuzzy sets satisfying Definition 2.9 and H be any fuzzy aggregation operator [62]. For multiple sets \mathcal{A} and \mathcal{B} in $MS_{(n,k)}(X)$, denote

$$S_H(\mathcal{A}, \mathcal{B}) = \sum_{i=1}^n S(H(A_i^1, A_i^2, \dots, A_i^k), H(B_i^1, B_i^2, \dots, B_i^k)) \tag{3.4}$$

Theorem 3.6. $S_H(\mathcal{A}, \mathcal{B})$ is a similarity measure between the multiple sets \mathcal{A} and \mathcal{B} in X .

Proof. Axioms (1) and (2) are obvious, respectively, from axioms (1) and (2) of Definition 2.9 for fuzzy similarity measure S .

Axiom(3): Clearly, we have

$$S_H(\mathcal{C}, \mathcal{C}) \leq \max_{\mathcal{A}, \mathcal{B} \in MS_{(n,k)}(X)} S_H(\mathcal{A}, \mathcal{B}) \tag{3.5}$$

Now, for any $\mathcal{A}, \mathcal{B} \in MS_{(n,k)}(X)$, we have

$$\frac{S(H(C_i^1, C_i^2, \dots, C_i^k), H(C_i^1, C_i^2, \dots, C_i^k))}{S(H(A_i^1, A_i^2, \dots, A_i^k), H(B_i^1, B_i^2, \dots, B_i^k))} \geq 1$$

for every $i = 1, 2, \dots, n$. Therefore

$$\sum_{i=1}^n S(H(C_i^1, C_i^2, \dots, C_i^k), H(C_i^1, C_i^2, \dots, C_i^k)) \geq \sum_{i=1}^n S(H(A_i^1, A_i^2, \dots, A_i^k), H(B_i^1, B_i^2, \dots, B_i^k))$$

So we have $S_H(\mathcal{C}, \mathcal{C}) \geq S_H(\mathcal{A}, \mathcal{B})$ for all $\mathcal{A}, \mathcal{B} \in MS_{(n,k)}(X)$. Therefore,

$$S_H(\mathcal{C}, \mathcal{C}) \geq \max_{\mathcal{A}, \mathcal{B} \in MS_{(n,k)}(X)} S_H(\mathcal{A}, \mathcal{B}) \tag{3.6}$$

Combining equations (3.5) and (3.6), it follows that

$$S_H(\mathcal{C}, \mathcal{C}) = \max_{\mathcal{A}, \mathcal{B} \in MS_{(n,k)}(X)} S_H(\mathcal{A}, \mathcal{B})$$

Axiom(4): Suppose $\mathcal{A}, \mathcal{B}, \mathcal{C} \in MS_{(n,k)}(X)$, such that $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{C}$. Then $A_i^j \subseteq B_i^j \subseteq C_i^j$ for every $j = 1, 2, \dots, k$ and $i = 1, 2, \dots, n$. Then, from axiom (4) of Definition 2.9 for fuzzy similarity measure S , we have

$$\frac{S(H(A_i^1, A_i^2, \dots, A_i^k), H(B_i^1, B_i^2, \dots, B_i^k))}{S(H(A_i^1, A_i^2, \dots, A_i^k), H(C_i^1, C_i^2, \dots, C_i^k))} \geq 1$$

for every $i = 1, 2, \dots, n$. Therefore,

$$\sum_{i=1}^n S(H(A_i^1, A_i^2, \dots, A_i^k), H(B_i^1, B_i^2, \dots, B_i^k)) \geq \sum_{i=1}^n S(H(A_i^1, A_i^2, \dots, A_i^k), H(C_i^1, C_i^2, \dots, C_i^k))$$

and hence $S_H(\mathcal{A}, \mathcal{B}) \geq S_H(\mathcal{A}, \mathcal{C})$. In a similar way, we can prove that $S_H(\mathcal{B}, \mathcal{C}) \geq S_H(\mathcal{A}, \mathcal{C})$. That is, $S(\mathcal{A}, \mathcal{B})$ satisfies all the axioms of Definition 3.1. Thus $S_H(\mathcal{A}, \mathcal{B})$ is a similarity measure between the multiple sets \mathcal{A} and \mathcal{B} in X . \square

Example 3.7. Let \mathcal{A} and \mathcal{B} be multiple sets given in example (3.3). Consider 3 similarity measures S_1, S_2 and S_3 of fuzzy sets, given by the equations 2.10, 2.11 and 2.12, respectively. Here we consider three fuzzy aggregation operators $H = avg, max$ or min . Then, the similarity measures $S(\mathcal{A}, \mathcal{B})$ between multiple sets \mathcal{A} and \mathcal{B} based on similarity measures S_1, S_2 or S_3 of fuzzy sets and fuzzy aggregation operators $H = avg, max$ or min are given in Table 1.

-	avg	max	min
S_1	2.405	2.487	2.117
S_2	2.645	2.633	2.566
S_3	2.503	2.568	2.136

Table 1: similarity measures $S(\mathcal{A}, \mathcal{B})$ between multiple sets \mathcal{A} and \mathcal{B} given by the Definition (3.4).

Using the properties of fuzzy similarity measure and definition of similarity measure of multiple set the following properties can be proved easily:

Theorem 3.8. Let \mathcal{A} and \mathcal{B} be multiple sets in $MS_{(n,k)}(X)$ and \mathcal{M} be the multiple set in $MS_{(n,k)}(X)$ for which the membership matrix for each $x \in X$ is $[0.5]_{n \times k}$. Let H denotes the fuzzy aggregation operators average, maximum or minimum. The similarity measure $S_H(\mathcal{A}, \mathcal{B})$ defined in equation (3.4) satisfies the following properties:

1. Suppose fuzzy similarity measure S satisfies the property $A = B \Leftrightarrow S(A, B) = 1$. Then $\mathcal{A} = \mathcal{B} \Rightarrow S_H(\mathcal{A}, \mathcal{B}) = n$.
2. Suppose fuzzy similarity measure S satisfies the property $A \cap B = \phi \Leftrightarrow S(A, B) = 0$ and $H = max$ or avg . Then $\mathcal{A} \cap \mathcal{B} = \Phi \Leftrightarrow S_H(\mathcal{A}, \mathcal{B}) = 0$. Moreover, if $H = min$, then $\mathcal{A} \cap \mathcal{B} = \Phi \Rightarrow S_H(\mathcal{A}, \mathcal{B}) = 0$.
3. Suppose fuzzy similarity measure S satisfies the properties $A = M \Leftrightarrow S(A, \bar{A}) = 1, 0 \leq S(A, B) \leq 1$ and $H = max$ or min . Then $\mathcal{A} = \mathcal{M} \Leftrightarrow S_H(\mathcal{A}, \bar{\mathcal{A}}) = n$. Moreover, if $H = avg$, then $\mathcal{A} = \mathcal{M} \Rightarrow S_H(\mathcal{A}, \bar{\mathcal{A}}) = n$.
4. Suppose fuzzy similarity measure S satisfies the property $S(A, \bar{A}) = 0 \Leftrightarrow A = I$ or $A = 0$ and $H =$

max or avg. Then $S_H(\mathcal{A}, \bar{\mathcal{A}}) = 0 \Leftrightarrow \mathcal{A} \in \xi_{(n,k)}(X)$.
 Moreover, if $H = \min$, then $\mathcal{A} \in \xi_{(n,k)}(X) \Rightarrow S(\mathcal{A}, \bar{\mathcal{A}}) = 0$.

Remark 3.9. 1. Converse of (1) in Theorem 3.8 need not be true. For example, let $X = \{x_1, x_2, x_3\}$ be the universal set and \mathcal{A} and \mathcal{B} be multiple sets in $MS_{(3,3)}(X)$ given by the following membership matrices;

$$\mathcal{A}(x_1) = \begin{bmatrix} 0.8 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.6 & 0.5 & 0.1 \end{bmatrix}$$

$$\mathcal{A}(x_2) = \begin{bmatrix} 0.9 & 0.5 & 0.4 \\ 0.6 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.3 \end{bmatrix}$$

$$\mathcal{A}(x_3) = \begin{bmatrix} 0.8 & 0.5 & 0.4 \\ 0.8 & 0.5 & 0.2 \\ 0.6 & 0.5 & 0.2 \end{bmatrix}$$

$$\mathcal{B}(x_1) = \begin{bmatrix} 0.8 & 0.6 & 0.4 \\ 0.5 & 0.5 & 0.5 \\ 0.6 & 0.4 & 0.2 \end{bmatrix}$$

$$\mathcal{B}(x_2) = \begin{bmatrix} 0.9 & 0.5 & 0.4 \\ 0.6 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.3 \end{bmatrix}$$

$$\mathcal{B}(x_3) = \begin{bmatrix} 0.8 & 0.5 & 0.4 \\ 0.8 & 0.5 & 0.2 \\ 0.6 & 0.4 & 0.3 \end{bmatrix}$$

Choose the fuzzy similarity measure S_1 given by the equation (2.10). From simple calculations we obtain; $S_{max}(\mathcal{A}, \mathcal{B}) = 3$ and $S_{avg}(\mathcal{A}, \mathcal{B}) = 3$. But, here $\mathcal{A} \neq \mathcal{B}$. Now, let \mathcal{C} be a multiple set in $MS_{(3,3)}(X)$ given by the following membership matrices;

$$\mathcal{C}(x_1) = \begin{bmatrix} 0.8 & 0.6 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.6 & 0.4 & 0.1 \end{bmatrix}$$

$$\mathcal{C}(x_2) = \begin{bmatrix} 0.9 & 0.5 & 0.4 \\ 0.6 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.3 \end{bmatrix}$$

$$\mathcal{C}(x_3) = \begin{bmatrix} 0.8 & 0.5 & 0.4 \\ 0.8 & 0.5 & 0.2 \\ 0.6 & 0.4 & 0.2 \end{bmatrix}$$

Choose the fuzzy similarity measure S_1 given by the equation (2.10). From simple calculations we obtain, $S_{min}(\mathcal{A}, \mathcal{C}) = 3$. But, here $\mathcal{A} \neq \mathcal{C}$.

2. $S_{min}(\mathcal{A}, \mathcal{B}) = 0$ need not imply $\mathcal{A} \cap \mathcal{B} = \Phi$. For example, let $X = \{x_1, x_2, x_3\}$ be the universal set and \mathcal{A} and \mathcal{B} be multiple sets in $MS_{(3,3)}(X)$ given by the following membership matrices;

$$\mathcal{A}(x_1) = \begin{bmatrix} 0.8 & 0.5 & 0.5 \\ 0.5 & 0.3 & 0.0 \\ 0.6 & 0.5 & 0.0 \end{bmatrix}$$

$$\mathcal{A}(x_2) = \begin{bmatrix} 0.2 & 0.2 & 0.0 \\ 0.1 & 0.1 & 0.0 \\ 0.5 & 0.5 & 0.3 \end{bmatrix}$$

$$\mathcal{A}(x_3) = \begin{bmatrix} 0.8 & 0.5 & 0.5 \\ 0.8 & 0.5 & 0.4 \\ 0.3 & 0.2 & 0.0 \end{bmatrix}$$

$$\mathcal{B}(x_1) = \begin{bmatrix} 0.4 & 0.2 & 0.0 \\ 0.5 & 0.5 & 0.5 \\ 0.6 & 0.4 & 0.2 \end{bmatrix}$$

$$\mathcal{B}(x_2) = \begin{bmatrix} 0.9 & 0.5 & 0.5 \\ 0.6 & 0.5 & 0.5 \\ 0.5 & 0.2 & 0.0 \end{bmatrix}$$

$$\mathcal{B}(x_3) = \begin{bmatrix} 0.1 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.6 & 0.4 & 0.3 \end{bmatrix}$$

Choose the fuzzy similarity measure S_1 given by the equation (2.10). From simple calculations we obtain, $S_{min}(\mathcal{A}, \mathcal{B}) = 0$. But, here $\mathcal{A} \cap \mathcal{B} \neq \Phi$.

3. $S_{avg}(\mathcal{A}, \bar{\mathcal{A}}) = n$ need not imply $\mathcal{A} = \mathcal{M}$. For example, let $X = \{x_1, x_2, x_3\}$ be the universal set and \mathcal{A} be multiple set in $MS_{(3,3)}(X)$ given by the following membership matrices;

$$\mathcal{A}(x_1) = \begin{bmatrix} 0.8 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.1 \end{bmatrix}$$

$$\mathcal{A}(x_2) = \begin{bmatrix} 0.9 & 0.5 & 0.5 \\ 0.6 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.3 \end{bmatrix}$$

$$\mathcal{A}(x_3) = \begin{bmatrix} 0.8 & 0.5 & 0.4 \\ 0.5 & 0.5 & 0.2 \\ 0.6 & 0.5 & 0.2 \end{bmatrix}$$

Then, the complement of \mathcal{A} is obtained as follows;

$$\bar{\mathcal{A}}(x_1) = \begin{bmatrix} 0.5 & 0.5 & 0.2 \\ 0.5 & 0.5 & 0.5 \\ 0.9 & 0.5 & 0.5 \end{bmatrix}$$

$$\bar{\mathcal{A}}(x_2) = \begin{bmatrix} 0.5 & 0.5 & 0.1 \\ 0.6 & 0.5 & 0.5 \\ 0.7 & 0.5 & 0.5 \end{bmatrix}$$

$$\bar{\mathcal{A}}(x_3) = \begin{bmatrix} 0.6 & 0.5 & 0.2 \\ 0.8 & 0.5 & 0.5 \\ 0.8 & 0.5 & 0.4 \end{bmatrix}$$

Choose the fuzzy similarity measure S_1 given by the equation (2.10). From simple calculations we obtain, $S_{avg}(\mathcal{A}, \bar{\mathcal{A}}) = 3$. But, here $\mathcal{A} \neq \mathcal{M}$.

4. $S_{min}(\mathcal{A}, \bar{\mathcal{A}}) = 0$ need not imply $\mathcal{A} \in \xi_{(n,k)}(X)$. For example, let $X = \{x_1, x_2, x_3\}$ be the universal set and \mathcal{A} be multiple set in $MS_{(3,3)}(X)$ given by the following membership matrices;

$$\begin{aligned} \mathcal{A}(x_1) &= \begin{bmatrix} 0.3 & 0.2 & 0.0 \\ 0.1 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \\ \mathcal{A}(x_2) &= \begin{bmatrix} 0.2 & 0.2 & 0.0 \\ 0.1 & 0.1 & 0.0 \\ 0.5 & 0.3 & 0.0 \end{bmatrix} \\ \mathcal{A}(x_3) &= \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.4 & 0.3 & 0.0 \\ 0.3 & 0.2 & 0.0 \end{bmatrix} \end{aligned}$$

Then, the complement of \mathcal{A} is obtained as follows;

$$\begin{aligned} \bar{\mathcal{A}}(x_1) &= \begin{bmatrix} 1.0 & 0.8 & 0.7 \\ 1.0 & 0.9 & 0.9 \\ 1.0 & 1.0 & 1.0 \end{bmatrix} \\ \bar{\mathcal{A}}(x_2) &= \begin{bmatrix} 1.0 & 0.8 & 0.8 \\ 1.0 & 0.9 & 0.9 \\ 1.0 & 0.7 & 0.5 \end{bmatrix} \\ \bar{\mathcal{A}}(x_3) &= \begin{bmatrix} 1.0 & 1.0 & 1.0 \\ 1.0 & 0.7 & 0.6 \\ 1.0 & 0.8 & 0.7 \end{bmatrix} \end{aligned}$$

Choose the fuzzy similarity measure S_1 given by the equation (2.10). From simple calculations we obtain, $S_{min}(\mathcal{A}, \bar{\mathcal{A}}) = 0$. But, here $\mathcal{A} \notin \xi_{(3,3)}(X)$.

4 Applications of similarity measures to pattern recognition

The capability of recognizing and classifying patterns is one of the most fundamental characteristics of human intelligence. Pattern recognition may be defined as a process by which we search for structures in data and classify these structures into categories such that the degree of association is high among structures of the same category and low between structures of different categories.

There are three fundamental problems in pattern recognition. The first one is sensing problem which is concerned with the representation of input data obtained by measurements on objects that are to be recognized. In general, each object is represented by a vector, known as pattern vector, in which each component represents a particular characteristic of the object. The second problem is feature extraction problem, which concerns the extraction of characteristic features from the input data in terms of which the dimensionality of pattern vectors can be reduced. The features should be characterizing attributes by which the given pattern classes are well discriminated. The third problem is classification of given patterns. This is usually done by defining an appropriate discrimination function for each class, which assigns a real number to each pattern vector. Individual pattern vectors are evaluated by these discrimination functions, and their classification is decided by the resulting values. Each pattern vector is classified to that class whose discrimination function yields the largest value. Pattern recognition systems have found vast applications in many areas such as handwritten character and

word recognition; automatic screening and classification of X-ray images; electrocardiograms, electroencephalograms, and other medical diagnostic tools; speech recognition and speaker identification; fingerprint recognition; classification of remotely sensed data; analysis and classification of chromosomes; image understanding; classification of seismic waves; target identification and human face recognition.

The utility of fuzzy set theory in pattern recognition was already recognized and the literature dealing with fuzzy pattern recognition is now quite extensive. In their position paper[63], Mitra et al. gave an outline to the contribution of fuzzy sets to pattern recognition. They mentioned that the concept of fuzzy sets can be used at the feature level in representing input data as an array of membership values denoting the degree of possession of certain properties; in representing linguistically phrased input features for their processing; in weakening the strong commitments for extracting ill-defined image regions, properties, primitives, and relations among them. Also, fuzzy sets can be used at the classification level, for representing class membership of objects, and for providing an estimate (or representation) of missing information in terms of membership values.

As mentioned above, fuzzy sets are very effective in representing different patterns in pattern recognition. Since multiple set is a generalization of fuzzy sets and it has the capability to represent numerous features simultaneously, they are well suited to model patterns. In this section, we establish a new procedure for pattern recognition with the aid of similarity measure on multiple sets.

Assume that there exist m patterns which are represented by multiple sets \mathcal{A}_r for $r = 1, 2, \dots, m$. Suppose that there be a sample to be recognized which is represented by a multiple set \mathcal{B} . According to the principle of the maximum degree of similarity between multiple sets, we can decide that the sample belongs to the pattern \mathcal{A}_r with maximum $S(\mathcal{A}_r, \mathcal{B})$. In the following, a fictitious numerical example is given to show application of the similarity measures to pattern recognition problems. Let three patterns be represented by multiple sets $\mathcal{A}_1, \mathcal{A}_2$ and \mathcal{A}_3 on $X = \{x_1, x_2, x_3\}$, given by the following membership matrices;

$$\mathcal{A}_1(x_1) = \begin{bmatrix} 0.9 & 0.8 & 0.8 \\ 0.4 & 0.4 & 0.4 \\ 0.2 & 0.2 & 0.1 \\ 0.8 & 0.8 & 0.7 \end{bmatrix}$$

$$\mathcal{A}_1(x_2) = \begin{bmatrix} 0.8 & 0.7 & 0.6 \\ 0.5 & 0.4 & 0.2 \\ 0.1 & 0.1 & 0.0 \\ 0.7 & 0.6 & 0.6 \end{bmatrix}$$

$$\mathcal{A}_1(x_3) = \begin{bmatrix} 0.7 & 0.6 & 0.5 \\ 0.4 & 0.3 & 0.3 \\ 0.2 & 0.1 & 0.0 \\ 0.7 & 0.7 & 0.7 \end{bmatrix}$$

$$\mathcal{A}_2(x_1) = \begin{bmatrix} 0.7 & 0.7 & 0.6 \\ 0.5 & 0.5 & 0.3 \\ 0.2 & 0.2 & 0.2 \\ 0.9 & 0.9 & 0.8 \end{bmatrix}$$

$$\mathcal{A}_2(x_2) = \begin{bmatrix} 0.6 & 0.5 & 0.4 \\ 0.7 & 0.6 & 0.5 \\ 0.3 & 0.3 & 0.1 \\ 0.9 & 0.8 & 0.8 \end{bmatrix}$$

$$\mathcal{A}_2(x_3) = \begin{bmatrix} 0.8 & 0.8 & 0.7 \\ 0.8 & 0.7 & 0.6 \\ 0.3 & 0.0 & 0.0 \\ 0.9 & 0.7 & 0.7 \end{bmatrix}$$

$$\mathcal{A}_3(x_1) = \begin{bmatrix} 0.2 & 0.2 & 0.0 \\ 0.5 & 0.5 & 0.4 \\ 0.8 & 0.7 & 0.5 \\ 0.5 & 0.4 & 0.2 \end{bmatrix}$$

$$\mathcal{A}_3(x_2) = \begin{bmatrix} 0.6 & 0.2 & 0.1 \\ 0.5 & 0.5 & 0.3 \\ 0.9 & 0.8 & 0.8 \\ 0.4 & 0.4 & 0.4 \end{bmatrix}$$

$$\mathcal{A}_3(x_3) = \begin{bmatrix} 0.5 & 0.3 & 0.2 \\ 0.7 & 0.6 & 0.6 \\ 0.9 & 0.9 & 0.9 \\ 0.4 & 0.2 & 0.2 \end{bmatrix}$$

Consider a sample \mathcal{B} in $MS_{(4,3)}(X)$ which will be recognized, where \mathcal{B} is given by the following membership matrices;

$$\mathcal{B}(x_1) = \begin{bmatrix} 1.0 & 0.9 & 0.9 \\ 0.3 & 0.3 & 0.3 \\ 0.2 & 0.2 & 0.1 \\ 0.7 & 0.7 & 0.6 \end{bmatrix}$$

$$\mathcal{B}(x_2) = \begin{bmatrix} 0.7 & 0.6 & 0.5 \\ 0.5 & 0.4 & 0.2 \\ 0.2 & 0.2 & 0.1 \\ 0.7 & 0.7 & 0.6 \end{bmatrix}$$

$$\mathcal{B}(x_3) = \begin{bmatrix} 0.7 & 0.6 & 0.5 \\ 0.5 & 0.3 & 0.3 \\ 0.3 & 0.2 & 0.1 \\ 0.7 & 0.7 & 0.6 \end{bmatrix}$$

Consider 3 similarity measures S_1, S_2 and S_3 of fuzzy sets, given by the equations 2.10, 2.11 and 2.12, respectively. Then the similarity measures $\mathcal{S}(\mathcal{A}_r, \mathcal{B})$ for $r = 1, 2, 3$, given by the Definition (3.1) based on similarity measures S_1, S_2 and S_3 of fuzzy sets are obtained in Table 2; Now, the similarity measures $\mathcal{S}(\mathcal{A}_r, \mathcal{B})$ for $r = 1, 2, 3$, given by the Definition (3.4), based on similarity measures S_1, S_2 and S_3 of fuzzy sets and fuzzy aggregation operators $H = min, max$ or avg are given in tables 3, 4 and 5. From the tables 2, 3, 4 and 5, we can see that $\mathcal{S}(\mathcal{A}_1, \mathcal{B})$ has the maximum value. The important point to note here is

-	S_1	S_2	S_3
$\mathcal{S}(\mathcal{A}_1, \mathcal{B})$	3.522	3.8	3.55
$\mathcal{S}(\mathcal{A}_2, \mathcal{B})$	3.23	3.5	3.257
$\mathcal{S}(\mathcal{A}_3, \mathcal{B})$	2.098	2.533	2.068

Table 2: similarity measures $\mathcal{S}(\mathcal{A}_r, \mathcal{B})$ for $r = 1, 2, 3$, given by the Definition (3.1).

-	S_1	S_2	S_3
$\mathcal{S}_{min}(\mathcal{A}_1, \mathcal{B})$	3.062	3.766	3.069
$\mathcal{S}_{min}(\mathcal{A}_2, \mathcal{B})$	2.464	3.366	2.81
$\mathcal{S}_{min}(\mathcal{A}_3, \mathcal{B})$	1.428	2.334	1.353

Table 3: similarity measures $\mathcal{S}(\mathcal{A}_r, \mathcal{B})$ for $r = 1, 2, 3$, given by the Definition (3.4) based on $H = min$.

-	S_1	S_2	S_3
$\mathcal{S}_{max}(\mathcal{A}_1, \mathcal{B})$	3.455	3.766	3.55
$\mathcal{S}_{max}(\mathcal{A}_2, \mathcal{B})$	3.124	3.367	3.136
$\mathcal{S}_{max}(\mathcal{A}_3, \mathcal{B})$	2.093	2.5	2.068

Table 4: similarity measures $\mathcal{S}(\mathcal{A}_r, \mathcal{B})$ for $r = 1, 2, 3$, given by the Definition (3.4) based on $H = max$.

-	S_1	S_2	S_3
$\mathcal{S}_{avg}(\mathcal{A}_1, \mathcal{B})$	3.361	3.765	3.424
$\mathcal{S}_{avg}(\mathcal{A}_2, \mathcal{B})$	2.884	3.368	3.049
$\mathcal{S}_{avg}(\mathcal{A}_3, \mathcal{B})$	1.772	2.4	1.728

Table 5: similarity measures $\mathcal{S}(\mathcal{A}_r, \mathcal{B})$ for $r = 1, 2, 3$, given by the Definition (3.4) based on $H = avg$.

that all formulae of multiple similarity measure mentioned here, results the same conclusion. Obviously, the sample \mathcal{B} belongs to the pattern represented by the multiple set \mathcal{A}_1 .

5 Conclusion

Similarity measure of fuzzy sets is a mature research field and has found applications in diverse areas such as pattern recognition, image processing, decision making, etc. Comparatively, similarity measure of multiple sets is a new topic. This paper deals with the similarity measure of multiple sets. Two formulas for similarity measure of multiple sets are proposed and their properties are investigated. This new concept is applied to pattern recognition problem and the suitability of proposed method is demonstrated using a numerical example. We believe that the concept can be extended to other applications such as image processing, decision making, etc. Investigation along these lines will be considered as a part of future work.

Acknowledgement

The second author acknowledges the financial assistance given by Kerala State Council for Science Technology and Environment (KSCSTE), INDIA through Student Project Scheme (541/SPS63/2018/KSCSTE) for a part of the work carried out and included in this this paper.

References

- [1] Goguen, Joseph A, L-fuzzy sets, *Journal of mathematical analysis and applications* 18(1) (1967): 145-174. [https://doi.org/10.1016/0022-247X\(67\)90189-8](https://doi.org/10.1016/0022-247X(67)90189-8)
- [2] Cerf V , Fernandez E , Gostelow K , Volansky S, Formal Control-Flow Properties of a Model of Computation, Tech. rep. California Univ., Los Angeles. Dept. of Computer Science(1971)
- [3] Z. Pawlak, Rough sets, *International Journal of Computer and Information Sciences* 11(5) (1982): 341-356. <https://doi.org/10.1007/BF01001956>
- [4] Atanassov KT, Intuitionistic fuzzy sets, VII ITKR's Session, Sofia deposited in Central Sci, Technical Library of Bulg. Acad. of Sci 1697(84) (1983): 84.
- [5] Yager, Ronald R, On the theory of bags, *International Journal of General System* 13(1) (1986): 23-37. <https://doi.org/10.1080/03081078608934952>
- [6] Goguen, Joseph A, Vague sets, *IEEE transactions on systems, man, and cybernetics* 23(2) (1993): 610-614. <https://doi.org/10.1109/21.229476>
- [7] Sebastian, Sabu and Ramakrishnan, TV, Multi-fuzzy sets, *International Mathematical Forum* 5(50) (2010): 2471-2476.
- [8] Shijina, V., Sunil, J.J & Anitha, S. Multiple sets. *Journal Of New Results In Science* (9)(2015) 18-27.
- [9] Shijina, V., Sunil, J.J & Anitha, S. Multiple sets: A unified approach towards modelling vagueness and multiplicity. *Journal Of New Theory*(11)(2016) 29-53.
- [10] Shijina, V. & Sunil, J.J Aggregation operations on multiple sets. *International Journal Of Scientific & Engineering Research*. 5(9),(2014) 39-42
- [11] Shijina, V. & Sunil, J.J Matrix-norm aggregation operators. *Iosr Journal Of Mathematics*. 1(2),(2017) 5-11
- [12] Shijina, V. & Sunil, J.J Multiple Relations and its Application in Medical Diagnosis. *International Journal Of Fuzzy System Applications*, 6(4) (2017) 16 Pages. <https://doi.org/10.4018/IJFSA.2017100104>
- [13] Zadeh, LA. Fuzzy sets. *Information And Control*. 8(3),(1965) 338–353. [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- [14] Zadeh, LA. Similarity relations and fuzzy orderings. *Information Sciences*. 3(2) (1971) 177–200. [https://doi.org/10.1016/S0020-0255\(71\)80005-1](https://doi.org/10.1016/S0020-0255(71)80005-1)
- [15] Zwick, R., Carlstein, E. & Budescu, D. Measures of similarity among fuzzy concepts: A comparative analysis. *International Journal Of Approximate Reasoning*. 1(2) (1987) 221–242. [https://doi.org/10.1016/0888-613x\(87\)90015-6](https://doi.org/10.1016/0888-613x(87)90015-6)
- [16] L.Xuecheng, Entropy, distance measure and similarity measure of fuzzy sets and their relations. *Fuzzy Sets And Systems*. 52(3),(1992) 305–318. [https://doi.org/10.1016/0165-0114\(92\)90239-z](https://doi.org/10.1016/0165-0114(92)90239-z)
- [17] C P. Pappis, & N.I. Karacapilidis, , A comparative assessment of measures of similarity of fuzzy values. *Fuzzy Sets And Systems*. 56(2),(1993) 171–174. [https://doi.org/10.1016/0165-0114\(93\)90141-4](https://doi.org/10.1016/0165-0114(93)90141-4)
- [18] Lee-kwang, H.,Y.S Song, & K.M Lee, Similarity measure between fuzzy sets and between elements. *Fuzzy Sets And Systems*. 62 (1994) 291–293. [https://doi.org/10.1016/0165-0114\(94\)90113-9](https://doi.org/10.1016/0165-0114(94)90113-9)
- [19] X.Wang, ,B. De Baets, & E. Kerre, A comparative study of similarity measures. *Fuzzy Sets And Systems*. 73(2), 259-268 (1995). [https://doi.org/10.1016/0165-0114\(94\)00308-t](https://doi.org/10.1016/0165-0114(94)00308-t)
- [20] S.M Chen,,M.S Yeh, & P.Y Hsiao, A comparison of similarity measures of fuzzy values. *Fuzzy Sets And Systems* 72(1) (1995) 79-89. [https://doi.org/10.1016/0165-0114\(94\)00284-e](https://doi.org/10.1016/0165-0114(94)00284-e)
- [21] W.J Wang, New similarity measures on fuzzy sets and on elements, *Fuzzy Sets And Systems*. 85(3)(1997) 305-309. [https://doi.org/10.1016/0165-0114\(95\)00365-7](https://doi.org/10.1016/0165-0114(95)00365-7)
- [22] H.Rezaei, , M. Emoto, & M.Mukaidono, New Similarity Measure Between Two Fuzzy Sets. *JACIII*. 10(6)(2006), 946-953. <https://doi.org/10.20965/jaciii.2006.p0946>
- [23] S.Omran, & M. Hassaballah, A new class of similarity measures for fuzzy sets. *International Journal Of Fuzzy Logic And Intelligent Systems*. 6(2),(2006) 100-104 . <https://doi.org/10.5391/ijfis.2006.6.2.100>
- [24] T.Chaira, & A.Ray, Fuzzy approach for color region extraction. *Pattern Recognition Letters*. 24(12)(2003) 1943–1950. [https://doi.org/10.1016/S0167-8655\(03\)00033-3](https://doi.org/10.1016/S0167-8655(03)00033-3)

- [25] D. Van der Weken, M. Nachtegael, & E. E. Kerre, An overview of similarity measures for images, in: Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on, Vol. 4, IEEE, 2002, pp. IV-3317. <https://doi.org/10.1109/icassp.2002.1004621>
- [26] D. Van der weken, , M. Nachtegael, & E. E. Kerre, Using similarity measures and homogeneity for the comparison of images. *Image And Vision Computing*. 22(9)(2004) 695-702. <https://doi.org/10.1016/j.imavis.2004.03.002>
- [27] D. Van der Weken, M. Nachtegael, V. De Witte, S. Schulte, & E. Kerre, A survey on the use and the construction of fuzzy similarity measures in image processing. in: *Computational Intelligence for Measurement Systems and Applications*, 2005. CIMS A. 2005 IEEE International Conference on, IEEE, 2005, pp. 187-192. <https://doi.org/10.1109/icip.2007.4379511>
- [28] D. Van Der Weken, V. De Witte, M. Nachtegael, S. Schulte, & E. Kerre, Fuzzy similarity measures for colour images, in: *Cybernetics and Intelligent Systems*, 2006 IEEE Conference on, IEEE, 2006, pp. 1-6. <https://doi.org/10.1109/iccis.2006.252363>
- [29] X. Chen, J. Tian & X. Yang, A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure, *IEEE Transactions on Image Processing* 15 (3) (2006) 767-776. <https://doi.org/10.1109/tip.2005.860597>
- [30] M. Nachtegael, D. Van der Weken, V. De Witte, S. Schulte, T. Mélangé, E. E. Kerre, Color image retrieval using fuzzy similarity measures and fuzzy partitions, in: *Image Processing, 2007. ICIP 2007. IEEE International Conference on, Vol. 6, IEEE, 2007, pp. VI-13.* <https://doi.org/10.1109/icip.2007.4379511>
- [31] E. Szmidt & J. Kacprzyk, A concept of similarity for intuitionistic fuzzy sets and its use in group decision making, in: *Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on, Vol. 2, IEEE, 2004, pp. 1129-1134.* <https://doi.org/10.1109/fuzzy.2004.1375570>
- [32] E. Szmidt & J. Kacprzyk, A new concept of a similarity measure for intuitionistic fuzzy sets and its use in group decision making, in: *International Conference on Modeling Decisions for Artificial Intelligence*, Springer, 2005, pp. 272-282. https://doi.org/10.1007/11526018_27
- [33] L. Dengfeng, C. Chuntian, New similarity measures of intuitionistic fuzzy sets and application to pattern recognitions, *Pattern recognition letters* 23 (1) (2002) 221-225. [https://doi.org/10.1016/s0167-8655\(01\)00110-6](https://doi.org/10.1016/s0167-8655(01)00110-6)
- [34] Z. Liang, P. Shi, Similarity measures on intuitionistic fuzzy sets, *Pattern Recognition Letters* 24 (15) (2003) 2687-2693. [https://doi.org/10.1016/s0167-8655\(03\)00111-9](https://doi.org/10.1016/s0167-8655(03)00111-9)
- [35] W.L. Hung, & M.S. Yang, Similarity measures of intuitionistic fuzzy sets based on Hausdorff distance. *Pattern Recognition Letters*. 25(14)(2004) 1603-1611. <https://doi.org/10.1016/j.patrec.2004.06.006>
- [36] H. B. Mitchell, On the dengfeng-chuntian similarity measure and its application to pattern recognition, *Pattern Recognition Letters* 24 (16) (2003) 3101-3104. [https://doi.org/10.1016/s0167-8655\(03\)00169-7](https://doi.org/10.1016/s0167-8655(03)00169-7)
- [37] W.L. Hung, M.S. Yang, Similarity measures of intuitionistic fuzzy sets based on L^p metric, *International Journal of Approximate Reasoning*, 46 (14) (2007) 120-136. <https://doi.org/10.1016/j.ijar.2006.10.002>
- [38] P. Julian, K.-C. Hung, S.-J. Lin, On the Mitchell similarity measure and its application to pattern recognition, *Pattern Recognition Letters* 33 (9) (2012) 1219-1223. <https://doi.org/10.1016/j.patrec.2012.01.008>
- [39] G. A. Papakostas, A. G. Hatzimichailidis & V. G. Kaburlasos, Distance and similarity measures between intuitionistic fuzzy sets: A comparative analysis from a pattern recognition point of view, *Pattern Recognition Letters* 34 (14) (2013) 1609-1622. <https://doi.org/10.1016/j.patrec.2013.05.015>
- [40] K. Atanassov & G. Gargov, Interval valued intuitionistic fuzzy sets, *Fuzzy sets and systems* 31 (3)(1989) 343-349. [https://doi.org/10.1016/0165-0114\(89\)90205-4](https://doi.org/10.1016/0165-0114(89)90205-4)
- [41] C.P. Wei, P. Wang & Y.Z. Zhang, Entropy, similarity measure of interval-valued intuitionistic fuzzy sets and their applications, *Information Sciences* 181 (19) (2011) 4273-4286. <https://doi.org/10.1016/j.ins.2011.06.001>
- [42] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-I, *Information sciences* 8 (3) (1975) 199-249. [https://doi.org/10.1016/0020-0255\(75\)90036-5](https://doi.org/10.1016/0020-0255(75)90036-5)
- [43] W.-L. Hung & M.-S. Yang, Similarity measures between type-2 fuzzy sets, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 12 (06) (2004) 827-841. <https://doi.org/10.1142/s0218488504003235>

- [44] G. Zheng, J. Wang, W. Zhou & Y. Zhang, A similarity measure between interval type-2 fuzzy sets, in: *Mechatronics and Automation (ICMA), 2010 International Conference on*, IEEE, 2010, pp. 191-195. <https://doi.org/10.1109/icma.2010.5589072>
- [45] C.M. Hwang, M.S. Yang, W.L. Hung & E. S. Lee, Similarity, inclusion and entropy measures between type-2 fuzzy sets based on the sugeno integral, *Mathematical and Computer Modelling* 53 (9) (2011) 1788-1797. <https://doi.org/10.1016/j.mcm.2010.12.057>
- [46] C.M. Hwang, M.S. Yang & W.L. Hung, New similarity and inclusion measures between type-2 fuzzy sets, in: *Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), 2011 IEEE Symposium on*, IEEE, 2011, pp. 82-87. <https://doi.org/10.1109/t2fuzz.2011.5949547>
- [47] M.-S. Yang & D.C. Lin, On similarity and inclusion measures between type-2 fuzzy sets with an application to clustering, *Computers & Mathematics with Applications* 57 (6) (2009) 896-907. <https://doi.org/10.1016/j.camwa.2008.10.028>
- [48] S. S. Mohamed & A. S. AbdAla, Applying a new similarity measure between general type-2 fuzzy sets to clustering, in: *Computational Intelligence and Informatics (CINTI), 2011 IEEE 12th International Symposium on*, IEEE, 2011, pp. 283-286. <https://doi.org/10.1109/cinti.2011.6108514>
- [49] D.-C. Lin & M.S. Yang, A similarity measure between type-2 fuzzy sets with its application to clustering, in: *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, Vol. 1, IEEE, 2007, pp. 726-731. <https://doi.org/10.1109/fskd.2007.123>
- [50] H. B. Mitchell, Pattern recognition using type-II fuzzy sets, *Information Sciences* 170 (2) (2005) 409-418. <https://doi.org/10.1016/j.ins.2004.02.027>
- [51] P. Singh, Similarity measure for type-2 fuzzy sets with an application to students' evaluation, *Computer Applications in Engineering Education* 23 (5) (2015) 694-702. <https://doi.org/10.1002/cae.21642>
- [52] V. Torra, Hesitant fuzzy sets, *International Journal of Intelligent Systems* 25 (6) (2010) 529-539. <https://doi.org/10.1002/int.20418>
- [53] V. Torra, Y. Narukawa, On hesitant fuzzy sets and decision, in: *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*, IEEE, 2009, pp. 1378-1382. <https://doi.org/10.1109/fuzzy.2009.5276884>
- [54] Z. Xu, M. Xia, Distance and similarity measures for hesitant fuzzy sets, *Information Sciences* 181 (11) (2011) 2128-2138. <https://doi.org/10.1016/j.ins.2011.01.028>
- [55] Y. Li, J.M. Liu, J. Li, W. D. C.X. Ye & Z.F. Wu, The fuzzy similarity measures for content-based image retrieval, in: *Machine Learning and Cybernetics, 2003 International Conference on*, Vol. 5, IEEE, 2003, pp. 3224-3228. <https://doi.org/10.1109/icmlc.2003.1260136>
- [56] H. Le Capitaine, A relevance-based learning model of fuzzy similarity measures, *IEEE Transactions on Fuzzy Systems* 20 (1) (2012) 57-68. <https://doi.org/10.1109/tfuzz.2011.2166079>
- [57] M. A. El-Sayed & N. Aboelwafa, Study of face recognition approach based on similarity measures, *International Journal of Computer Science Issues (IJCSI)* 9 (2) (2012) 133-139.
- [58] X. Xu, L. Zhang & Q. Wan, A variation coefficient similarity measure and its application in emergency group decision-making, *Systems Engineering Procedia* 5 (2012) 119-124. <https://doi.org/10.1016/j.sepro.2012.04.019>
- [59] L. Baccour, A. M. Alimi & R. I. John, Some notes on fuzzy similarity measures and application to classification of shapes recognition of arabic sentences and mosaic, *IAENG International Journal of Computer Science* 41 (2) (2014) 81-90. <https://doi.org/10.1109/fuzzy.2009.5276877>
- [60] S.M. Chen, A new approach to handling fuzzy decision-making problems, *IEEE Transactions on Systems, Man, and Cybernetics* 18 (6) (1988) 1012-1016. <https://doi.org/10.1109/21.23100>
- [61] W. Bandler, L. Kohout, Fuzzy power sets and fuzzy implication operators, *Fuzzy Sets and Systems* 4 (1) (1980) 13-30. [https://doi.org/10.1016/0165-0114\(80\)90060-3](https://doi.org/10.1016/0165-0114(80)90060-3)
- [62] K. George J & Y. Bo, *Fuzzy sets and fuzzy logic, theory and applications*, Prentice Hall PTR, (1995).
- [63] S. Mitra & S. K. Pal, Fuzzy sets in pattern recognition and machine intelligence, *Fuzzy Sets and systems* 156 (3) (2005) 381-386. <https://doi.org/10.1016/j.fss.2005.05.035>

Performance Assessment of a Set of Multi-Objective Optimization Algorithms for Solution of Economic Emission Dispatch Problem

Sarat Kumar Mishra

Department of Electrical and Electronics Engineering, Padmanava College of Engineering, Rourkela, India

E-mail: mishra.sarat@gmail.com

Sudhansu Kumar Mishra

Department of Electrical and Electronics Engineering, Birla Institute of Technology, Mesra, Ranchi, India

E-mail: sudhansu.nit@gmail.com

Keywords: differential evolution, economic emission dispatch, multi-objective optimization, non-dominated sorting, particle swarm optimization

Received: October 30, 2018

This paper addresses the realistic economic emission dispatch (EED) problem of power system by considering the operating fuel cost and environmental emission as two conflicting objectives, and power balance and generator limits as two constraints. A novel dynamic multi-objective optimization algorithm, namely the multi-objective differential evolution with recursive distributed constraint handling (MODE-RDC) has been proposed and successfully employed to address this challenging EED problem. It has been thoroughly investigated in two different test cases at three different load demands. The efficiency of the MODE-RDC is also compared with two other multi-objective evolutionary algorithms (MOEAs), namely, the non-dominated sorting genetic algorithm (NSGA-II) and multi-objective particle swarm optimization (MOPSO). Performance evaluation is carried out by comparing the Pareto fronts, computational time and three non-parametric performance metrics. The statistical analysis is also performed, to demonstrate the ascendancy of the proposed MODE-RDC algorithm. Investigation of the performance metrics revealed that the proposed MODE-RDC approach was capable of providing good Pareto solutions while retaining sufficient diversity. It renders a wide opportunity to make a trade-off between operating cost and emission under different challenging constraints.

Povzetek: Opisan je izvorni multi-kriterijski optimirni algoritem za energetske sisteme, ki kombinira kriterij onesnaževanja in kriterij energetske potrošnje.

1 Introduction

The Economic Load Dispatch (ELD) problem deals with the estimation of the scheduled real power generation from the committed units for best economic operation. Over the years the problem has become more complex due to the increasing effects of emissions from fossil fuel based power plants on the environment. The emission and fuel cost of each unit depend on the quantity of power to be generated. Both of them are nonlinear functions of power output. Minimum operating cost does not ensure minimum emission. Each operating condition must satisfy the power balance criterion and should obey the generating limits of the committed units. These can be considered as constraints. Generally, better quality fuel ensures less emission but it can be further reduced by proper scheduling of generation from different units. The cost coefficients and emission coefficients of these generating units do not match. Hence, achieving these two objectives, i.e. less cost and less emission is contradictory in nature. Thus, the EED problem has evolved as a modification of the ELD problem. Therefore, the EED problem is a multi-objective optimization problem with nonlinear constraints.

In [1-2], the power engineers solved the ELD problem by scheduling of the generation of multi-unit

systems using the derivative based Gauss-Siedel and Newton-Raphson algorithms along with the Lagrangian multiplier. These conventional methods suffer from the problem of getting trapped in local minima and also fail for system discontinuities due to prohibited zones. These techniques are inadequate to solve multi-objective problems with nonlinear constraints. Chang et al. [3] rehabilitated the inherently multi-objective EED problem to a single objective one by assigning weights to the operating cost and emission. This weighted sum approach requires many runs of the same algorithm to find the Pareto optimal front. The solutions arrived at by this method do not ensure a uniform Pareto front. The trade-off information is lost when the function is concave. To avoid this bottom-hole different evolutionary based heuristic approaches have been introduced by many researchers [4-5]. These evolutionary algorithms have considered the two objectives simultaneously and are shown to perform better as compared to the conventional ones. Chiang et al. [6] made a further refinement and proposed an improved genetic algorithm to speed up the search process. He used the ϵ -constraint technique for efficient

constraint handling and proposed a multiplier updating mechanism for better exploration of the search space.

Deb et al. [7] proposed the non-dominated sorting genetic algorithm which utilized rank and crowding distance as parameters to arrive at a compromise between the two conflicting objectives. This was applied to the multi-objective environmental economic load dispatch problem in [8]. The Pareto optimal front could be obtained by a single run of the algorithm. But, this population based genetic algorithm depends upon biologically inspired factors like mutation and crossover parameters. It needs further improvement in terms of exploring a wider area in the search space. Brar et al. [9] made improvements in the search space by adding the fuzzy inference system. Muthuswamy et al. [10] modified the non-dominated sorting technique by incorporating a dynamic crowding distance to improve the diversity of solutions in the search space. These algorithms fail when there are discontinuities in the cost function.

Nayak et al. [11] implemented another evolutionary algorithm, the artificial bee colony (ABC) optimization, and improved the convergence rate and reliability under the presence of the prohibited zones and ramp rate limits. Liang et al. [12] modified the ABC algorithm to form an improved artificial bee colony (IABC) by addition of a new skill called chaos ques in the search process. Mori et al. [13] made an excellent improvement in the exploration of search space through the implementation of the particle swarm optimization (PSO) for this multimodal problem. They also used adaptive parameter adjustment to improve the results. A significant improvement in search space exploration was made by Hadji et al. [14]. They incorporated a time varying acceleration of the particles to improve the robustness of the algorithm. Recently, a differential evolution (DE) algorithm came up which generates the next set of population of new particles by the addition of a differential vector obtained from the difference of the position vectors of two different particles other than the particle undergoing evolution [15]. This algorithm is still dependent on the bio-inspired parameters but is able to avoid premature convergence. Meza et al. [16] improved the algorithm by incorporating spherical pruning for better exploitation of the search space. Di et al. [17] introduced a marginal analysis correction operator to improve the constraint handling.

In [18], the particle swarm optimization algorithm has been developed which is based on the intelligence of flock of birds. The same has been improved and tested for multi-objective problems in [19-21]. The EED problem has been solved to decide the unit commitment of the power system by considering operational power flow and environmental constraints in [22]. But, it again utilized the method of conversion of the multi-objective problem to a single objective one. A new approach to optimization is proposed in [23] which hybridized adaptive PSO and DE for improvement of the search space. An improvement over ABC called as multi-objective global best artificial bee colony (MOGABC) optimization is suggested in [24] for better constraint

handling in EED problem. The EED problem has been further modified and applied to the micro-grid containing renewable sources along with the conventional thermal power stations in [25]. It also converts the problem to a single objective one by incorporating a h-index.

In this paper, a new constraint handling mechanism has been implemented, and a new multi-objective optimization (MOP) algorithm, namely the multi-objective differential evolution with recursive distributed constraint handling (MODE-RDC) has been proposed. The constraint handling mechanism is suitably incorporated in three multi-objective optimization (MOP) algorithms, and the effectiveness of the algorithms has been tested under various load conditions.

2 Multi-objective optimization: a review

The main aim of the multi-objective optimization technique is to optimize two or more conflicting objectives simultaneously. The MOP is denoted by a decision variable vector, each element of which represents the objective functions [21]. The solution to the MOP is the optimum value of the vector function by considering all the constraints. A multi-objective minimization problem can be generalized as follows:

$$\text{Minimize } f(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_M(\vec{x})) \quad (1)$$

Subject to constraints:

$$g_j(\vec{x}) \leq 0; j=1, 2, \dots, J \quad (2)$$

$$h_k(\vec{x}) = 0; k=1, 2, \dots, K \quad (3)$$

where, \vec{x} is a vector with N decision variables

$$\vec{x} = [x_1, x_2, \dots, x_N]^T$$

The search space may be limited by lower and upper bounds

$$lb_i \leq x_i \leq ub_i; i=1, 2, \dots, N \quad (4)$$

A solution vector $\vec{u} = [u_1, u_2, \dots, u_N]^T$ dominates over another solution $\vec{v} = [v_1, v_2, \dots, v_N]^T$ if and only if

$$\left. \begin{aligned} f_i(\vec{u}) &\leq f_i(\vec{v}) \forall i \in [1, 2, \dots, M] \\ f_i(\vec{u}) &< f_i(\vec{v}) \text{ for at least one } i \in [1, 2, \dots, M] \end{aligned} \right\} \quad (5)$$

Solutions that are not dominated by other solutions within the given solution space are said to be non-dominated solutions. The front obtained by mapping such points onto the objective space is said to be the Pareto optimal front (POF)

$$\text{POF} = f(\vec{x}) = \{ \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})\}; |\vec{x} \in p \} \quad (6)$$

where, p is the set of non-dominated particles.

3 Economic emission dispatch problem

The generation schedule for minimum operating cost is called ELD. This schedule is obtained when the committed units of the power system are able to supply the load demanded and the associated transmission losses by satisfying the generator limits. The thermal generating units are associated with emissions which are highly polluting in nature. Therefore, it is essential to minimize

the emissions along with the fuel cost. The problem has been transformed to EED problem. It is considered as a multi-objective optimization problem as minimum operating cost does not ensure minimum emission [5]. The operating cost of thermal power systems depends mostly on the cost of fuel used. The quantity of fuel used by each unit depends on the generated power, efficiency of turbine etc.

The fuel cost characteristics of the generating units are normally of the second order polynomial of the generated power. Thus, the operating fuel cost of the i th generator supplying a real power P_{Gi} is given by

$$F_i = a_i P_{Gi}^2 + b_i P_{Gi} + c_i \tag{7}$$

where, a_i , b_i and c_i are the coefficients of cost function.

The emission from the generator i can be approximated as

$$E_i = \alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2 \tag{8}$$

where, α_i , β_i , γ_i are the coefficients of emission function.

The aim of the ELD problem is to determine generation schedule for the minimum total fuel cost given by

$$F_T = \sum_{i=1}^N F_i \tag{9}$$

subject to the constraints.

The total real power generation must be equal to the demand plus transmission losses

$$\sum_{i=1}^N P_{Gi} = P_D + P_L \tag{10}$$

where, P_D is the load demand on the system and P_L is the transmission loss. It is given by Kron’s formula

$$P_L = \sum_{i=1}^N \sum_{j=1}^N P_{Gi} B_{ij} P_{Gj} + \sum_{i=1}^N B_{i0} P_{Gi} + B_{00}. \tag{11}$$

The constants B_{ij} , B_{i0} and B_{00} are dependent on the line parameters. The generated real power of each unit must be within the feasible lower and upper bounds.

$$P_{Gi(\min)} \leq P_{Gi} \leq P_{Gi(\max)}; i = 1, 2, \dots, N. \tag{12}$$

Thus, the sole objective of the EED problem is to optimize both the fuel cost and emission simultaneously. Hence, it is inherently a multi-objective optimization problem where these two objectives which are conflicting in nature need to be optimized. The set of all the potential compromised solutions is represented by the Pareto optimal front.

The problem can be stated as:

$$\text{minimize}_{P_G} [F_T(P_G), E_T(P_G)] \tag{13}$$

$$\text{Subject to: } g(P_G) = 0; h(P_G) \leq 0$$

where, the equality constraint is represented by equation (14) and inequality constraint by equation (15).

$$\sum_{i=1}^N P_{Gi} - P_D - P_L = 0 \tag{14}$$

$$P_{Gi} - P_{Gi}^{\max} \leq 0; P_{Gi}^{\min} - P_{Gi} \leq 0 \tag{15}$$

4 Multi-objective optimization algorithms

The multi-objective evolutionary algorithms can be categorized mostly into four types in accordance with the algorithmic framework, such as indicator based, convolution based, memetic based and non-dominated

sorting based. In this paper, we have solved the economic emission dispatch problem using three different multi-objective optimization algorithms that are based on non-dominated sorting.

4.1 Non-dominated Sorting Genetic Algorithm-II (NSGA-II)

This algorithm was formulated by Deb et al. [7], [8] for solving the multi-objective optimization problems. It is initialized with a random population, and used some operators for covering the objective space uniformly on the Pareto set. For multi-criteria optimization it uses three strategies: non-domination sorting, ranking based on density and crowding comparison. The individuals are classified into several layers based on their rank and crowding distance. The diversity in the solutions is maintained by rejecting the solutions with lower crowding distance. The quality of solutions is ensured by selecting the individuals with lower ranks. The advantage of this algorithm is that the complexity of computation is lowered and elitism is maintained.

4.2 Non-dominated Sorting Multi-Objective Particle Swarm Optimization (NS-MOPSO)

Kennedy and Eberhart proposed that any optimization problem can be solved by mimicking the movement of a flock of birds and school of fish [18]. The social behavior of the swarm is to change their position and velocity to maximize their chance of getting food and follow the best successful neighbor. This lead to the formulation of particle swarm optimization (PSO). In this method of optimization, a local best and a global best solution are identified. The i^{th} particle in the population having the best position (pbest) may be represented by p_i , that gives the best fitness value represented as

$$p_i = (p_{i1}, p_{i2}, \dots, p_{iN}) \tag{16}$$

The old and new velocity of the particles will be shown in equation (17) and (18) respectively.

$$V_i = (v_{i1}, v_{i2}, \dots, v_{iN}) \tag{17}$$

$$v_{id}(t) = wv_{id}(t-1) + c_1 r_1 (p_{id} - x_{id})(t-1) + c_2 r_2 (p_{id} - x_{id})(t-1) \tag{18}$$

and the new position of the particle will be

$$x_{id}(t) = x_{id}(t-1) + \chi v_{id}(t) \tag{19}$$

where, $d = 1, 2, \dots, D$ is the dimension of the decision variables and $i = 1, 2, \dots, N$, and χ is the constriction factor which constricts and controls the velocity magnitude. w , c_1 and c_2 are weight parameters and r_1 , r_2 are random numbers known as acceleration constants in the range $[0, 1]$. This method of improvement of position and velocity is applied to the non-dominating vectors to solve the multi-objective problem [14], [19] & [20].

4.3 Multi-Objective Differential Evolution with Recursive Distributed Constraint handling (MODE-RDC)

The differential evolution (DE) algorithm as developed by Stern and Price [15] is less dependent on bio-inspired mechanisms, and serves better for multivariable problems. The multi-objective optimization using differential evolution (MODE) proposed by Meza et al. [16] is an improvement of DE to solve multiple number of conflicting objectives simultaneously. The MODE is an evolutionary multi-objective optimization algorithm that retains the diversity of solutions on the Pareto front. This real coded stochastic algorithm uses an initial population to explore the search space by avoiding convergence to local optimal points. It uses two main operators: mutation and cross over. Each initial particle of the population is improved using these two operators. The mutation operator uses a differential vector selected from the particles other than the target particle. Three vectors $x_{r0,g}$, $x_{r1,g}$ and $x_{r2,g}$ are selected randomly from the population and the first one is updated with the difference of the other two. It is done as follows:

$$v_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g}) \tag{20}$$

where, $v_{i,g}$ is the mutant vector created from the target vector $x_{r0,g}$ and $x_{r1,g}$, $x_{r2,g}$ are two other vectors; $F \in (0,1+)$ is the scale factor that controls the rate of evolution.

A new trial vector (child) is created from the mutant vector and the target vector after cross over.

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } (\text{rand}_j(0,1) \leq C_r \text{ or } j = j_{\text{rand}}) \\ x_{j,i,g} & \text{otherwise} \end{cases} \tag{21}$$

where, the cross over probability $C_r \in [0,1]$. The child and the parent are tested for their fitness. The one with the best fitness is selected for participation in the next generation.

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{if } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{otherwise} \end{cases} \tag{22}$$

This is done for all $i=1, 2, \dots, n$ where $n =$ population size. The steps of the proposed MODE-RDC algorithm are summarized as follows:

- I. Initialize the number of individuals N and the population $P(0)$ by random selection within the limits of the search space. Set the fitness function, constraints, maximum number of generations, mutation factor, cross over rate and tolerance.
- II. Evaluate $P(0)$ using the fitness function and constraint
- III. If constraint violation is out of bounds call the recursive distributed constraint handler
- IV. Obtain the non-dominated solutions in $P(0)$ and store in $D(0)$
- V. Update the populations till the maximum number of generations or the convergence criterion is reached
 1. Randomly select a subpopulation of $N_s(k)$ with the proposed solution on $P(k)$
 2. Apply the DE operators on $N_s(k)$ to get the offspring $O(k)$:

- a) Perform the mutation operation.
 - b) Apply the fixing rule for boundary constraint violations.
 - c) Perform cross over operation.
3. Evaluate offspring $O(k)$ and determine constraint violations
 4. Call the recursive distributed constraint handler until constraint violation is within tolerable limits
 5. Compare the parent and offspring and select the best store in $D(k)$
 6. Apply dominance,
- VI. Modify $D(k)$, perform non dominated sorting on $D(k)$ and plot the Pareto front. Terminate the algorithm. Select the proper solution using the high level decision making rules. Here, we have used the allowable emission norms as the accepted solution.

4.3.1 Constraint handling

The main problem in finding the solution to the EED problem is that every new population evolved must satisfy the upper and lower bounds along with the nonlinear power balance constraint. The power balance constraint, being a polynomial of the individual solutions, makes it a complicated task to ensure convergence. Therefore, a continuous effort has to be made to restrict the solutions in the feasible area. Here, we propose a recursive distributed constraint handling method. The constraint c is the mismatch of power defined as

$$c = \sum_{i=1}^N P_{Gi} - P_D - P_L \tag{23}$$

if $c \leq \epsilon$, then $P_{Gi} = P_{Gi}$

$$\text{else } P_{Gi} = P_{Gi} - \frac{|c|}{N}$$

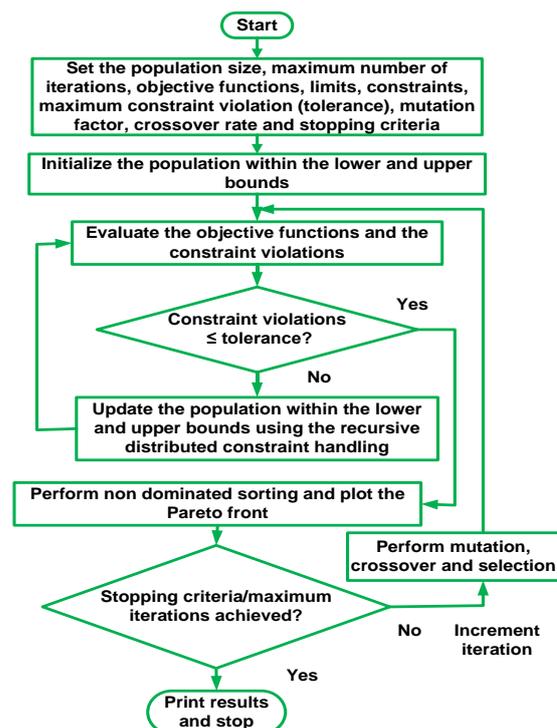


Figure 1: Flowchart of MODE-RDC algorithm.

$$\text{subject to } P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}$$

where, ϵ is the maximum allowable constraint violation. The constraint violation is evaluated and distributed over the decision variables. The values of the N decision variables will change until the constraint violation is restricted within the permissible limits. Thus the solutions are confined within the feasible work space.

The proposed algorithm is implemented using sequences shown in the flowchart (fig. 1). In the first block, the parameters including the limits are set. The second block initializes the population randomly between the upper and lower limits of the generating units. The population created is evaluated for the constraint violation; if the violation is high, then the population is updated using the constraint handler. The non-dominated sorting is performed, and Pareto optimal front is plotted from the updated population. This population represents the first iteration; subsequent iterations are performed to get further modified populations by applying equations (20), (21) and (22). This process is continued till convergence or the maximum number of iterations performed.

5 Performance measures

The following three performance metrics [21] have been applied to investigate the performance quality of the non-dominated solutions obtained in the form of Pareto fronts using different algorithms.

5.1 Generation Distance (GD)

It is the estimated distance between the non-dominated solution vectors from the standard efficient front. It is mathematically expressed as

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \tag{24}$$

where, n is the number of solution vectors and d_i is the minimum Euclidean distance between each of them. $GD=0$ indicates that all the solution vectors are present in the standard Pareto front. A smaller positive value of GD means the Pareto front obtained from the proposed algorithm is closer to the standard Pareto front.

5.2 Spacing (S)

Spacing is the measure of the spread of the solution vectors. It is expressed as

$$S \triangleq \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \tag{25}$$

where, $d_i = \min_j (|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|)$ for all $i, j=1, 2, \dots, n$ and \bar{d} =mean of all d_i and n is the number of non-dominated solution vectors found so far. The lower the value of S the better is the Pareto solution.

5.3 Diversity metric (Δ)

It measures how evenly the solution vectors are distributed in the search space, i.e. extent of the spread on the Pareto front. It is found from the Euclidean distances as follows

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (n-1)\bar{d}} \tag{26}$$

where, d_i is the distance between the consecutive solution vectors in the non-dominated solution set. The average of these distances is \bar{d} . Here, d_f and d_l are the first and last Euclidean distances. A low value of Δ indicates better diversity, and $\Delta=0$ means the non-dominated solution vectors are uniformly spread on the Pareto front.

6 Simulation study and results

The EED problem was simulated for two different standard test cases i.e., IEEE 14 bus and IEEE 30 bus. The system data of these two test cases were obtained from the website www.ee.washington.edu/research/pstca. The cost and emission coefficients were also recorded from standard sources [22] and are presented in the Appendix along with the B coefficients. Each test case is solved for three different load demands. The solutions are obtained by applying three different algorithms e.g., NSGA-II, MOPSO and proposed MODE-RDC. The algorithms are run in a MATLAB environment with a PC running on Microsoft windows 8 platform having a core i3 processor with a clock speed of 1.3 GHz and RAM of 4 GB. A maximum generation of 300 is taken with a population size of 100. The crossover rate is chosen as intermediate with the ratio set as 1.2 and mutation chosen as Gaussian with a scale of 0.1 and a shrink of 0.5 for NSGA-II. The velocity weight of MOPSO is selected as 0.4 and position weights as 1 with a population size of 100 and a maximum number of iterations of 100. The scaling factor of differential evolution is set as 0.5 and crossover rate as 0.5 with a population size of 100.

6.1 Test case I: IEEE-14 bus system

The algorithms were applied to this test case for three different load demands; i.e., for 200 MW, 259 MW and 300 MW. The performance was compared with respect to Pareto optimal front, computational time, fuel cost, transmission loss involved and statistical performance metrics.

6.1.1 IEEE-14 bus system: load demand 200 MW

The Pareto optimal fronts obtained by applying the three algorithms are compared and shown in Fig.2 for a load demand of 200MW.

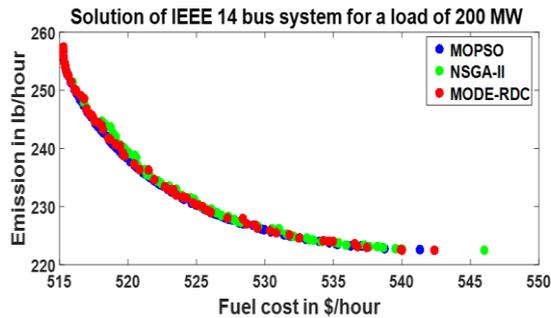


Figure 2: Solution of IEEE 14 bus system for a load demand of 200 MW.

Algorithm	NSGA-II	MOPSO	MODE-RDC
PG1 (MW)	121.894	117.4404	121.1744
PG2 (MW)	37.4252	41.0169	41.8528
PG3 (MW)	19.3125	19.9156	20.4068
PG4 (MW)	10.0000	13.9457	10.9482
PG5 (MW)	15.6575	11.8954	10.0000
Time (sec)	85.4620	0.963873	9.0003
PL (MW)	4.2892	4.2230	4.3822
Fuel Cost (\$/hour)	518.569	518.6977	518.3990
Emission(lb/hour)	244.963	241.1887	242.3576

Table 1: Results of EED of IEEE 14 bus system for a load demand of 200 MW.

Algorithm		NSGA-II	MOPSO	MODE-RDC
Generation Distance (GD)	Minimum	0.0452	0.0714	0.0273
	Maximum	0.0819	0.0714	0.0772
	Average	0.051075	0.0714	0.050145
	Standard Deviation	0.0144013	0.0000	0.0059599
Spacing (S)	Minimum	0.2266	0.5864	0.2659
	Maximum	0.7777	0.5864	0.5576
	Average	0.469715	0.5864	0.37959
	Standard Deviation	0.135470	0.0000	0.0643687
Diversity (A)	Minimum	1.2497	0.7446	0.4843
	Maximum	1.6853	0.7446	0.7300
	Average	1.47758	0.7446	0.613785
	Standard Deviation	0.134289	0.0000	0.0710588

Table 2: Performance of algorithms for IEEE 14 bus system at load demand of 200 MW.

The comparative generation schedule, fuel cost, emission and transmission loss are presented in Table 1. The Pareto front obtained by applying the proposed MODE-RDC algorithm and other two algorithms for different load conditions are depicted from Fig.2 to Fig. 7. The fuel cost value (518.399\$/hour) obtained is lowest among all the three competitive algorithms. The emission obtained from the proposed algorithm is also comparable with other two. Similarly, the algorithms were run for 25 times, and the performance metrics were calculated which are presented in Table 2. The average spacing obtained is 0.37959 which is the lowest among all the three algorithms. The average values of other two

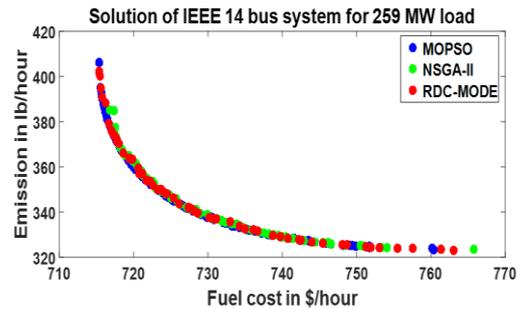


Figure 3: Solution of IEEE 14 bus system for a load demand of 259 MW.

Algorithm	NSGA-II	MOPSO	MODE-RDC
PG1 (MW)	150.416	150.6611	130.7742
PG2 (MW)	51.3048	50.2912	53.0186
PG3 (MW)	23.5338	23.7902	26.3320
PG4 (MW)	23.5837	22.8592	30.7226
PG5 (MW)	17.29	18.3925	24.2551
Time (sec)	91.4121	2.380883	8.0020
PL (MW)	7.1283	6.9942	6.1025
Fuel Cost (\$/hour)	720.3	720.1619	720.1591
Emission (lb/hour)	360	359.2373	359.1248

Table 3: Results of EED for IEEE 14 bus system at load demand of 259 MW.

Algorithm		NSGA-II	MOPSO	MODE-RDC
Generation Distance (GD)	Minimum	0.061130	0.1056	0.056818
	Maximum	0.247369	0.1056	0.215051
	Average	0.115768	0.1056	0.094455
	Standard Deviation	0.052146	0.0000	0.037827
Spacing (S)	Minimum	0.531774	0.6438	0.527058
	Maximum	2.442706	0.6438	1.926600
	Average	1.099879	0.6438	0.641592
	Standard Deviation	0.528886	0.0000	0.428131
Diversity (A)	Minimum	1.200668	0.5898	0.526846
	Maximum	1.702898	0.5898	0.825663
	Average	1.456543	0.5898	0.569785
	Standard Deviation	0.124752	0.0000	0.077072

Table 4: Performance of algorithms for IEEE 14 bus system at load demand of 259 MW.

performance matrices show improved performance of the proposed algorithm. This indicates that the Pareto solutions obtained by the proposed algorithm are superior to the competing algorithms.

6.1.2 IEEE-14 bus system: load demand 259 MW

Figure 3 shows the Pareto optimal fronts obtained for the test case with the load demand of 259 MW.

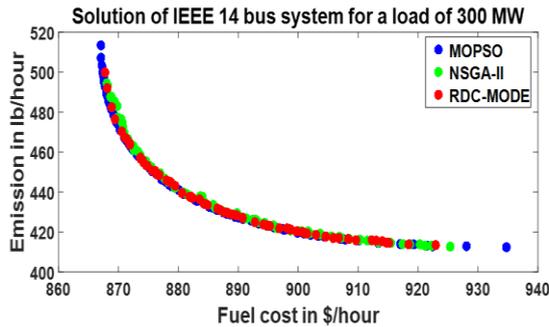


Figure 4: Solution of IEEE 14 bus system for a load demand of 300 MW.

Algorithm	NSGA-II	MOPSO	MODE-RDC
PG1 (MW)	158.385	159.5448	144.5312
PG2 (MW)	59.5374	57.2405	61.6878
PG3 (MW)	28.554	29.1715	30.9264
PG4 (MW)	34.3464	36.9012	41.7176
PG5 (MW)	27.8335	26.9604	29.1517
Time (sec)	94.0921	2.303318	8.0025
PL (MW)	8.6563	9.8184	8.0147
Fuel Cost (\$/hour)	880.909	879.916	880.4091
Emission (lb/hour)	440.116	440.862	440.234

Table 5: Results of EED of IEEE 14 bus system for load demand of 300 MW.

Algorithm		NSGA-II	MOPSO	MODE-RDC
Generation Distance (GD)	Minimum	0.056728	0.1245	0.055189
	Maximum	0.216685	0.1245	0.345505
	Average	0.123737	0.1245	0.120521
	Standard Deviation	0.044166	0.0000	0.058929
Spacing (S)	Minimum	0.510125	0.7691	0.568051
	Maximum	2.065131	0.7691	1.644507
	Average	1.154852	0.7691	0.764450
	Standard Deviation	0.428284	0.0000	0.666390
Diversity (Δ)	Minimum	1.269374	0.6529	0.491511
	Maximum	1.580342	0.6529	0.964161
	Average	1.426765	0.6529	0.648467
	Standard Deviation	0.087880	0.0000	0.102660

Table 6: Performance of algorithms for IEEE 14 bus system at load demand of 300 MW.

The generation schedule for this load is presented in Table 3, and the statistical performance metric values are shown in Table 4. The values of fuel cost and emission obtained by the proposed algorithm are better as compared to the other two algorithms. The generation schedule obtained by the former leads to less transmission loss. The proposed MODE-RDC algorithm shows lower average values of GD, S and Δ as compared to the other two competitive algorithms.

6.1.3 IEEE-14 bus system: load demand 300 MW

With a higher load of 300 MW the Pareto optimal fronts obtained using the algorithms are shown in Fig. 4.

Load (MW)	Algorithm	MODE-RDC Wins	MODE-RDC Losses	Detected differences
200	NSGA-II	17	8	$\alpha = 0.1$
	MOPSO	13	12	-
259	NSGA-II	18	7	$\alpha = 0.5$
	MOPSO	13	12	-
300	NSGA-II	18	7	$\alpha = 0.5$
	MOPSO	13	12	-

Table 7: Result of sign test on MODE-RDC algorithm for IEEE 14 bus system with GD metric as winning parameter.

Load (MW)	Algorithm	MODE-RDC Wins	MODE-RDC Losses	Detected differences
200	NSGA-II	14	11	-
	MOPSO	13	12	-
259	NSGA-II	16	9	-
	MOPSO	13	12	-
300	NSGA-II	14	11	-
	MOPSO	13	12	-

Table 8: Result of sign test on MODE-RDC algorithm for IEEE 14 bus system with S metric as winning parameter.

The generation schedule and performance metrics of the solution points are presented in Tables 5 and 6 respectively. The fuel cost and emission obtained using proposed MODE-RDC algorithm is better in terms of emission and PL in MW. The transmission losses involved due to the schedule obtained by the proposed algorithm is also lower. The average value of GD, S and Δ metrics of the solutions arrived from the proposed algorithm are less than that of the other two. Thus the quality of solutions is maintained.

6.1.4 IEEE-14 bus system: summary of results

The summary of generation schedules obtained for the three load conditions using the three algorithms as presented in Tables 1, 3 and 5 show that the fuel cost for the proposed algorithm provides improved performance for the load conditions. For the IEEE 14 bus test case, the two-tailed Sign tests [21] is conducted for the pair-wise comparison of the algorithms by considering three different performance metrics, and are presented in Tables 7, 8 and 9. The tests were carried out for all the three load conditions and by repeated run of the three algorithms for 25 times. It is observed from Tables 7, 8 and 9 that the proposed algorithm wins over the other two for all loads in terms of all the three performance metrics i.e. the generation distance, spacing and diversity metric as winning parameters. It can be pointed in Table-10 that in Sign test, if more than 17 wins are recorded, then the algorithm is better with a detected difference $\alpha=0.1$; and if more than 18 wins are recorded then, $\alpha=0.05$.

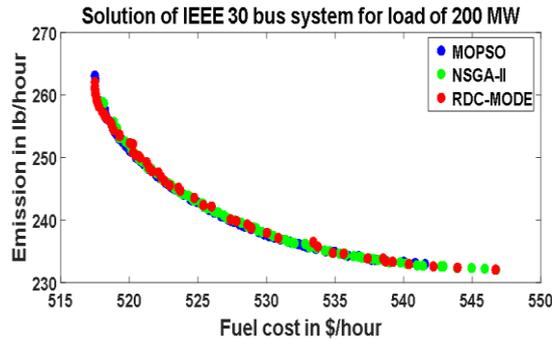


Figure 5: Solution of IEEE 30 bus system for load demand of 200 MW.

Load (MW)	Algorithm	MODE-RDC Wins	MODE-RDC Losses	Detected differences
200	NSGA-II	25	0	0.05
	MOPSO	13	12	-
259	NSGA-II	25	0	0.05
	MOPSO	13	12	-
300	NSGA-II	25	0	0.05
	MOPSO	13	12	-

Table 9: Result of sign test on MODE-RDC algorithm for IEEE 14 bus system with Δ metric as winning parameter.

No. of algorithm runs	$\alpha = 0.05$	$\alpha = 0.1$
5	5	5
6	6	6
7	7	6
8	7	7
9	8	7
10	9	8
11	9	9
12	10	9
13	10	10
14	11	10
15	12	11
16	12	12
17	13	12
18	13	13
19	14	13
20	15	14
21	15	14
22	16	15
23	17	16
24	18	16
25	18	17

Table 10: Significant values for decision on two-tailed sign test.

It is evident from the Table 7 and 8 that the proposed MODE-RDC algorithm shows better performance as compared to other two competitive algorithms. The algorithm is better with respect to NSGA-II a detected difference $\alpha=0.1$ for load of 200 MW, and a detected difference of $\alpha=0.05$ for load of 259MW and 300 MW. However, the proposed algorithm does not show much improvement as compared to MOPSO algorithm.

6.2 Test case II: IEEE-30 bus system

The algorithms were applied to this test case for three different load demands; i.e., for 200 MW, 283.4 MW and

Algorithm	NSGA-II	MOPSO	MODE-RDC
PG1 (MW)	103.927	104.4705	100.8991
PG2 (MW)	37.512	37.7012	41.8153
PG3 (MW)	18.996	19.3131	19.4211
PG4 (MW)	18.718	16.8090	14.9330
PG5 (MW)	13.021	12.7445	14.9143
PG6 (MW)	12.000	12.9950	12.0000
Time (sec)	90.8698	1.5908	8.0285
PL (MW)	4.1740	4.0333	3.9828
Fuel Cost (\$/hour)	524.966	523.9469	523.7250
Emission (lb/hour)	244.007	244.0324	244.9227

Table 11: Results of EED of IEEE 30 bus system for a load demand of 200 MW.

Algorithm		NSGA-II	MOPSO	MODE-RDC
Generation Distance (GD)	Minimum	0.018414	0.0457	0.013645
	Maximum	0.088774	0.0457	0.094384
	Average	0.034309	0.0457	0.032455
	Standard Deviation	0.015609	0.0000	0.009311
Spacing (S)	Minimum	0.159507	0.2623	0.156047
	Maximum	0.870246	0.2623	0.807493
	Average	0.304470	0.2623	0.255573
	Standard Deviation	0.157360	0.0000	0.099082
Diversity (Δ)	Minimum	0.975386	0.5672	0.547595
	Maximum	1.575610	0.5672	0.853479
	Average	1.223858	0.5672	0.558198
	Standard Deviation	0.173773	0.0000	0.052061

Table 12: Performance of algorithms for IEEE 30 bus system at load demand of 200 MW.

Algorithm	NSGA-II	MOPSO	MODE-RDC
PG1 (MW)	132.672	134.7225	111.077
PG2 (MW)	53.443	50.2415	51.679
PG3 (MW)	27.719	27.0213	31.856
PG4 (MW)	29.870	31.4431	33.083
PG5 (MW)	25.102	23.0641	30.000
PG6 (MW)	21.916	24.2446	31.969
Time (sec)	94.5415	2.0302	8.003
PL (MW)	7.32	7.3371	6.264
Fuel Cost (\$/hour)	821.269	820.1034	822.0048
Emission (lb/hour)	380.213	380.7899	379.5462

Table 13: Results of EED of IEEE 30 bus system for a load demand of 283.4 MW.

350 MW. The performance is compared with respect to Pareto optimal front, computational time, fuel cost, transmission loss involved and statistical performance metrics.

6.2.1 IEEE-30 bus system: load demand 200 MW

The Pareto optimal fronts obtained by applying the three algorithms for the load demand of 200 MW are presented in Figure 5.

From the simulation output it reveals that the Pareto curve obtained by applying the proposed algorithm covers a wider area of the search space. The generation

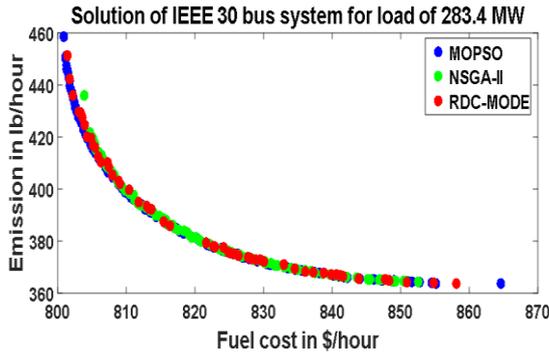


Figure 6: Solution of IEEE 30 bus system for load demand of 283.4 MW.

schedule for an acceptable emission level is achieved, and is presented in Table 11.

The quality of solutions is assessed by evaluating all the three performance metrics, and is presented in Table 12. The cost of fuel obtained by employing the proposed algorithm is lower, and the corresponding schedule causes lower transmission loss. It surpasses the performance of NSGA-II in terms of computational time. The lower average values of all the three performance matrices imply superior performance of the proposed algorithm over other two.

6.2.2 IEEE-30 bus system: load demand 283.4 MW

The test case is subjected to the rated load of 283.4 MW, and performance of the algorithms is assessed. The Pareto fronts obtained for this demand are presented in Figure 6.

The Pareto curve corresponding to the proposed algorithm covers a wide area of the search space. The generation schedules obtained by applying the algorithms are shown in Table 13. The proposed algorithm yields lower values of fuel cost and emission. The transmission losses involved with this schedule is less than the other two algorithms. The quality of solutions is assessed by running the algorithms for 25 times; statistical values of the performance metrics thus obtained are presented in Table 14. The average diversity of the solutions using the proposed algorithm is less than that of the other two algorithms. It also requires less computational time as compared to the NSGA-II.

6.2.3 IEEE-30 bus system: load demand 350 MW

The test case is further subjected to a higher load of 350 MW, and performance of the algorithms is assessed. The Pareto optimal fronts are shown in Figure 7. The generating schedule for the system obtained for this demand is presented in Table 15.

The statistical behavior of the performance metrics obtained for the solutions are shown in Table 16. It is observed that the proposed algorithm performs better than NSGA-II in terms of diversity of solutions, computational time, fuel cost and emission values.

Algorithm		NSGA-II	MOPSO	MODE-RDC
Generation Distance (GD)	Minimum	0.054373	0.1407	0.021384
	Maximum	0.116501	0.1407	0.233931
	Average	0.077291	0.1407	0.065103
	Standard Deviation	0.017481	0.0000	0.029863
Spacing (S)	Minimum	0.383454	0.9516	0.339348
	Maximum	1.026579	0.9516	2.075391
	Average	0.621816	0.9516	0.621204
	Standard Deviation	0.189580	0.0000	0.360044
Diversity (Δ)	Minimum	0.603026	0.6719	0.563586
	Maximum	1.229024	0.6719	0.767057
	Average	0.968746	0.6719	0.670803
	Standard Deviation	0.163341	0.0000	0.066364

Table 14: Performance of algorithms for IEEE 30 bus system at load demand of 283.4 MW.

Algorithm	NSGA-II	MOPSO	MODE-RDC
PG1 (MW)	159.003	163.500	162.974
PG2 (MW)	68.918	66.800	65.864
PG3 (MW)	46.060	32.600	36.256
PG4 (MW)	25.070	35.000	33.083
PG5 (MW)	30.000	30.000	27.768
PG6 (MW)	31.711	33.300	33.351
Time (sec)	89.7484	3.3319	8.03545
PL (MW)	10.762	11.200	10.970
Fuel Cost (\$/hour)	1111.24	1082.10	1081.3329
Emission (lb/hour)	540.295	539.90	539.1846

Table 15: Results of EED of IEEE 30 bus system for load demand of 350 MW.

Algorithm		NSGA-II	MOPSO	MODE-RDC
Generation Distance (GD)	Minimum	0.064384	0.1329	0.056721
	Maximum	0.162268	0.1329	0.391057
	Average	0.094781	0.1329	0.093724
	Standard deviation	0.025093	0.0000	0.056003
Spacing (S)	Minimum	0.474948	0.7443	0.403432
	Maximum	1.483643	0.7443	3.664206
	Average	0.789943	0.7443	0.741615
	Standard Deviation	0.255086	0.0000	0.609410
Diversity (Δ)	Minimum	0.807397	0.5516	0.509509
	Maximum	1.353364	0.5516	0.958338
	Average	1.095059	0.5516	0.531211
	Standard deviation	0.136457	0.0000	0.069968

Table 16: Performance of algorithms for IEEE 30 bus system at load demand of 350 MW.

Load (MW)	Algorithm	MODE-RDC Wins	MODE-RDC Losses	Detected differences
200	NSGA-II	14	11	-
	MOPSO	13	12	-
283.4	NSGA-II	13	12	-
	MOPSO	15	10	-
350	NSGA-II	13	12	-
	MOPSO	13	12	-

Table 17: Result of sign test on MODE-RDC algorithm on IEEE 30 bus system with GD metric as winning parameter.

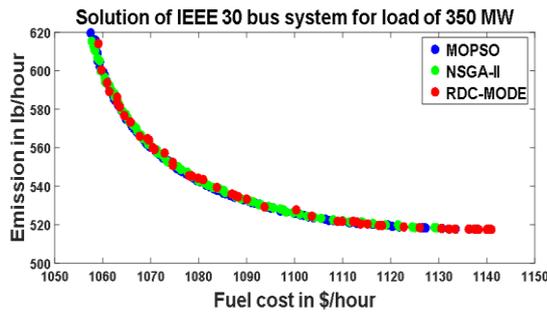


Figure 1: Solution of IEEE 30 bus system for load demand of 350 MW.

Load (MW)	Algorithm	MODE-RDC Wins	MODE-RDC Losses	Detected differences
200	NSGA-II	16	9	-
	MOPSO	13	12	-
283.4	NSGA-II	14	11	-
	MOPSO	16	9	-
350	NSGA-II	13	12	-
	MOPSO	13	12	-

Table 18: Result of sign test on MODE-RDC algorithm for IEEE 30 bus system with S metric as winning parameter.

6.2.4 IEEE-30 bus system: summary of results

The summary of results for this test case with the three load demands by applying all the three algorithms is presented in Tables 11, 13 and 15. It shows that the fuel cost obtained using the proposed algorithm provides improved performance as compared to other two algorithms. The transmission loss involved with the generation schedule thus arrived is also lower. The quality of solutions arrived using the algorithms is further estimated from pair wise sign test [21] on 25 runs of the algorithms. The results are presented in Tables 17, 18 and 19. Based on the number of wins and losses it is observed from Tables 17, 18 and 19 that the solutions obtained by applying the proposed algorithm yield lower average values in terms of all the three performance metrics. So, these solutions can be considered to be better than those from the other two algorithms. Also, the proposed MODE-RDC algorithm does not perform very well in terms of GD and S metric as in case of IEEE 30 bus system for load demand of 350 MW as depicted in Tables 17 and 18. This is clear in these tables as it does not have a significant detected difference that needs at least 17 wins out of 25 runs. This is due to the reduction in the number of non-dominated solutions in the successive iterations in the proposed algorithm.

Here, it can be pointed out that in most cases the average value of S metric has increased with rise in load demand for all the three algorithms this can be verified from Tables 2, 4 and 6 for test case I and Tables 12, 14 and 16 for test case II. This is due to the fact that when the load demand on the system rises, the size of the feasible space decreases due to the generator limits. The situation is further limited by the power balance constraint as the transmission losses increase with

Load (MW)	Algorithm	MODE-RDC Wins	MODE-RDC Losses	Detected differences
200	NSGA-II	15	10	0.05
	MOPSO	13	12	-
283.4	NSGA-II	14	11	0.05
	MOPSO	13	12	-
350	NSGA-II	23	2	0.05
	MOPSO	12	13	-

Table 19: Result of sign test on MODE-RDC algorithm for IEEE 30 bus system with Δ metric as winning parameter.

Gen no	1	2	3	4	5
Max MW	250	140	100	120	45
Min MW	10	20	15	10	10
γ	0.0126	0.02	0.027	0.0291	0.029
β	-0.9	-0.1	-0.01	-0.005	-0.004
α	22.983	25.313	25.505	24.9	24.7
a	0.00375	0.0175	0.0625	0.00834	0.025
b	2.0	1.75	1.0	3.25	3.0
c	0	0	0	0	0

Table 20: IEEE 14 bus system cost and emission coefficients.

Gen no	1	2	3	4	5	6
Max MW	200	80	50	35	30	40
Min MW	50	20	15	10	10	12
γ	0.0126	0.02	0.027	0.0291	0.029	0.0271
β	-0.9	-0.1	-0.01	-0.005	-	-
α	22.983	25.313	25.505	24.9	24.7	25.3
a	0.00375	0.0175	0.0625	0.00834	0.025	0.025
b	2.0	1.7	1.0	3.25	3.0	3.0
c	0	0	0	0	0	0

Table 21: IEEE 30 bus system cost and emission coefficients.

increase in power generation. Thus, the feasible solutions fall apart causing increase in the spacing parameter.

7 Conclusion

A set of three multi-objective optimization algorithms have been applied to solve the EED problem for two test cases on three different load demands. The performance of the proposed MODE-RDC algorithm along with other two is assessed by considering three different performance metrics. The performances of these algorithms have been critically analyzed. The Pareto optimal fronts obtained by all the three algorithms incorporating the proposed recursive distributed constraint handling technique have sufficient diversity by exploiting the entire available range of search space. In particular, the Pareto front obtained by the multi-objective differential evolution with the recursive distributed constraint handling (MODE-RDC) approach has a better diversity in most cases. The spacing between the Pareto solutions has found to be increased with rise in the load demand on the system for all the three

algorithms. Moreover, the time requirement to achieve the Pareto front by applying the proposed recursive distributed constraint handling based technique is satisfactory.

Further research on this topic may include the inclusion of different evolutionary local search mechanisms into the approaches. It is expected to obtain lower computational speed and exploitation of the multi-dimensional search space. There is a need for further investigation to explore the strengths and weaknesses of the proposed algorithm, so that it can be applied to other multi-objective problems in power systems, such as management of voltage profiles, reactive power compensation etc. The performance of the proposed algorithm can also be investigated by considering other real world constraints like ramp rate limits, power loss etc.

Appendix

The standard test cases taken for the solution are IEEE 14 bus and IEEE 30 bus power systems. The parameters of the test cases have been adopted from standard sources. The cost and emission coefficients used for solution of the problem are shown in tables 20 and 21 below.

The values of B coefficients used for the IEEE 14 bus test case are

$$B = \begin{bmatrix} 0.0208 & 0.0090 & -0.0021 & 0.0024 & 0.0006 \\ 0.0090 & 0.0168 & -0.0028 & 0.0035 & 0.0000 \\ -0.0021 & -0.0028 & 0.0207 & -0.0152 & -0.0179 \\ 0.0024 & 0.0035 & -0.0152 & 0.0763 & -0.0103 \\ 0.0006 & 0.0000 & -0.0179 & -0.0103 & 0.0476 \end{bmatrix}$$

$$B0 = [-0.0001 \quad 0.0023 \quad -0.0012 \quad 0.0027 \quad 0.0011]$$

$$B00 = 3.1826 \times 10^{-4}$$

The values of B coefficients used for the IEEE 30 bus test case are

$$B = \begin{bmatrix} 0.0218 & 0.0103 & 0.0010 & -0.0025 & 0.0007 & 0.0033 \\ 0.0103 & 0.0233 & 0.0001 & -0.0043 & 0.0009 & 0.0032 \\ 0.0010 & 0.0001 & 0.0525 & -0.0380 & -0.0111 & -0.0066 \\ -0.0025 & -0.0043 & -0.0380 & -0.1011 & 0.0132 & 0.0045 \\ 0.0007 & 0.0009 & -0.0111 & 0.0132 & 0.0163 & -0.0001 \\ -0.0033 & 0.0032 & -0.0066 & 0.0045 & -0.0001 & 0.0270 \end{bmatrix}$$

$$B0 = [-0.0002 \quad 0.0029 \quad -0.0033 \quad 0.0035 \quad 0.00016 \quad 0.0048]$$

$$B00 = 0.0025$$

References

- [1] H.Saadat (2009). Power System Analysis. McGraw-Hill Publishing Company Limited.
- [2] A.J. Wood, B.F.Wollenberg (1998). Power Generation, Operation and Control. John Wiley & Sons, Inc.
- [3] A.Farag, Al-Baiyat, S.T.C. Cheng (1995) Economic load dispatch multiobjective optimization procedures using linear programming techniques.IEEE Transaction on Power System, pp.731–738
<https://doi.org/10.1109/59.387910>
- [4] L.F. Wang, C. Singh (2006). Multi-objective stochastic power dispatch through a modified particle swarm optimization algorithm. Special Session on Applications of Swarm Intelligence to Power Systems. Proceedings of IEEE Swarm Intelligence Symposium, pp.127–135
<https://doi.org/10.1109/pes.2006.1709288>
- [5] M.A.Abido (2003). Environmental/Economic Power Dispatch Using Multi-Objective Evolutionary Algorithms. IEEE Transactions on Power Systems 18, 4, pp. 1529–1537
<https://doi.org/10.1109/tpwrs.2003.818693>
- [6] C.L. Chiang, J. H.Liaw, C.T. Su (2005). New approach with a genetic algorithm framework to multi-objective generation dispatch problems. European Transactions on Electrical Power 15, pp.381–395
<https://doi.org/10.1002/etep.48>
- [7] K. Deb, A.Pratap, T.Meyarivan (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6, 2, pp. 182–197
<https://doi.org/10.1109/4235.996017>
- [8] R.T.F.A. King, H.C.S.Rughooputh, K. Deb (2004). Evolutionary Multi-Objective Environmental/Economic Dispatch: Stochastic vs. Deterministic Approaches. Kanpur GA Lab Report, Number 019
https://doi.org/10.1007/978-3-540-31880-4_47
- [9] Y.S.Brar, J.S.Dhillon, D.P. Kothari (2006). Multi objective Load Dispatch Based on Genetic-Fuzzy Technique, IEEE Conference on PSCE, pp. 931–937
<https://doi.org/10.1109/psce.2006.296438>
- [10] R.Muthuswamy, M. Krishnan, K. Subramanian, B. Subramanian (2015). Environmental and economic power dispatch of thermal generators using modified NSGA-II algorithm, International Transactions on Electrical Energy Systems 25, 1552–1569
<https://doi.org/10.1002/etep.1918>
- [11] K.Nayak, R.Krishnand, B.K.Panigrahi, P.K. Rout (2009). Application of Artificial Bee Colony to Economic Load Dispatch Problem with Ramp Rate Limits and Prohibited Operating Zones, IEEE World Congress on Nature & Biologically Inspired Computing, pp. 1237–1242
<https://doi.org/10.1109/nabic.2009.5393751>
- [12] R.H. Liang, C.Y. Wu, Y.T. Chen, W.T. Tseng (2016). Multi-objective dynamic optimal power flow using improved artificial bee colony algorithm based on Pareto optimization, International Transactions on Electrical Energy Systems 26, pp. 692–712
<https://doi.org/10.1002/etep.2101>
- [13] H. Mori, K. Okawa (2010). Advanced MOEPSO-based Multi-objective Environmental Economic Load Dispatching. IEEE Conference
<https://doi.org/10.1109/pes.2010.5590209>
- [14] B.Hadji, B.Mahdad, K.Srairi, N.Mancer (2015). Multi-objective PSO-TVAC for Environmental Economic Dispatch Problem, Intl. Conf. on Technologies and Materials for Renewable Energy

- and Sustainability TMREES15, Energy Procedia, Elsevier pp. 102–111
<https://doi.org/10.1016/j.egypro.2015.07.529>
- [15] R.Storn, K. Price (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, *Journal of Global Optimization* 11, 4, pp. 341-359
<https://doi.org/10.1023/a:1008202821328>
- [16] G.R. Meza, X.Blasco, J.Sanchis, M. Martinez (2010). Multiobjective optimization algorithm for solving constrained single objective problems. *IEEE Conference*
<https://doi.org/10.1109/cec.2010.5586408>
- [17] Y. Di, M. Fei, L. Wang, W. Wu (2014). Multi-objective optimization for economic emission dispatch using an improved binary differential evolution algorithm. *The 6th International Conference on Applied Energy –ICAE*, Energy Procedia, Elsevier
<https://doi.org/10.1016/j.egypro.2014.12.065>
- [18] R.C.Eberhart, J. Kennedy (1995). A New Optimiser using Particle Swarm Theory. *Sixth International Symposium on Micro-machine and Human Science*, pp. 39–43
<https://doi.org/10.1109/mhs.1995.494215>
- [19] V.Hosseinnezhad, M.Rafiee, Md.Ahmadian, Md.TaghiAmeli (2014). Species based Quantum Particle Swarm Optimization for economic load dispatch. *Electrical Power and Energy Systems*, Elsevier 63, pp. 311–332
<https://doi.org/10.1016/j.ijepes.2014.05.066>
- [20] S.K. Mishra, S.K. Mishra (2016). Solution of Constrained Economic Emission Dispatch Problem Using Multi-Objective Particle Swarm Optimisation. *International Journal of Control Theory and Applications* 9, 39, pp. 63–70, DOI: 10.5373/JARDCS/V12I3/20201167,
- [21] S.K. Mishra, G. Panda, R.A.Majhi (2014). Comparative Performance Assessment of a Set of Multi-objective Algorithms for Constrained Portfolio Assets Selection, *Swarm and Evolutionary Computing*, Elsevier 16, pp. 38–51
<https://doi.org/10.1016/j.swevo.2014.01.001>
- [22] S.P.Karthikeyan, K.Palaniswamy, C. Rani, I.J.Raglend, D.P. Kothari (2009). Security Constrained Unit Commitment Problem with Operational Power Flow and Environmental Constraints, *WSEAS Transactions on Power Systems* 4, 2, pp. 53–66
- [23] S.Varshney, M.Mehrotra (2018). A Hybrid Particle Swarm Optimization and Differential Evolution Based Test Data Generation Algorithm for Data-Flow Coverage Using Neighbourhood Search Strategy, *Informatica* 42 pp. 417-438
<https://doi.org/10.31449/inf.v42i3.1497>
- [24] S. Ma, Y. Wang, Y. Lv (2018). Multiobjective Environment/Economic Power Dispatch using Evolutionary Multiobjective Optimization, *IEEE Access*
<https://doi.org/10.1109/access.2018.2795702>
- [25] B. Dey, S. K. Roy, B. Bhattacharyya (2019). Solving multi-objective economic emission dispatch of a renewable integrated microgrid using latest bio-inspired algorithms, *Engineering Science and Technology, an International Journal*, Elsevier 22, pp. 55-66.
<https://doi.org/10.1016/j.jestch.2018.10.001>

Research on Data Transmission Optimization of Communication Network Based on Reliability Analysis

Hui Wang

Information College, Baoshan University, Baoshan, Yunnan 678000, China

E-mail: ynwvh@cumt.edu.cn

Keywords: vehicular ad-hoc network, ant colony algorithm, connectivity probability, routing algorithm

Received: August 17, 2020

Firstly, this paper briefly introduces the vehicular communication network, vehicular ad-hoc network (VANET), and the ant colony algorithm for path search in the network. Then the ant colony algorithm is improved. A simulation experiment is carried out on the improved ant colony algorithm on the simulation platform, and it is compared with the traditional ant colony algorithm and the geographical location-based routing algorithm. The results show that the network packet loss rate and time delay show a tendency of increasing first and then decreasing under the three algorithms with the increase of vehicle nodes in VANET; with the increase of nodes, the network reliability under the improved ant colony algorithm shows a tendency of increasing first and then being stable, and the network reliability under the other two algorithms shows a tendency of increasing first and then decreasing; under the same number of nodes, the packet loss rate and average delay are lower, and the reliability is higher under the improved ant colony algorithm.

Povzetek: Opisana je raziskava optimizacije prenosa podatkov po komunikacijskem omrežju na osnovi analize zanesljivosti prenosa.

1 Introduction

With the convenience of the wireless network, the Internet of things can connect different things anytime and anywhere to realize information interaction. The existence of the Internet of things has further facilitated our life [1]. In intelligent transportation, the interconnection of human-vehicle-road-environment is realized by the Internet of things to achieve the optimization of traffic resource allocation [2]. For the Internet of things, the data transmission quality of wireless communication network for information interaction can directly affect the service quality of the Internet of things. If the quality of data transmission in the communication network is poor, packet loss will often occur, or the data throughput will be small, so that the Internet of things cannot effectively carry out information exchange. More over due to the rule that data will be transmitted again if the transmission fails because of packet loss, the number of data transmission will increase, which will increase the energy consumption of the communication network [3]. Therefore, improving the quality of data transmission in the communication network is very important for the Internet of things. Kumar et al. [4] proposed a beacon independent geographic routing algorithm called BIIR which reduced the number of broadcasts to forward the data packets by making intelligent use of information collected by the vehicle during previous route discovery attempts for a destination and found through the simulation that the algorithm outperformed the existing beacon less routing protocols in terms of the average number of broadcasts per data packet forwarding, packet delivery ratio and end to end delay experienced by the data messages. Rahul et al. [5]

proposed a routing algorithm based on genetic algorithm. They found that the algorithm had a great improvement in the execution time compared with the traditional serial algorithm and could be expanded with vehicles on the road to reduce the roadside units and provide a more extensive coverage. Ning et al. [6] proposed a reliable opportunistic routing algorithm based on probability prediction. The utility of relay vehicles in the candidate set was confirmed by predicting the change of the signal to interference plus noise ratio of the receiving end and packet queue length, so as to determine the route. Through simulation, they found that the algorithm had better routing performance than SRPE, ExOR, and GPSR routing algorithms. This paper briefly introduced the vehicular communication network, vehicular ad-hoc network (VANET) and the ant colony algorithm for path search in the network, improved the ant colony algorithm, carried out a simulation experiment on the improved ant colony algorithm on the simulation platform, and compared it with the traditional ant colony algorithm and the geographical location-based routing algorithm.

2 Vehicle communication network

With the help of wireless networks, different things form the Internet of things through information interaction, which greatly facilitates people's daily life. The communication network used for data transmission in the Internet of things of the intelligent transportation system is a vehicle communication network, also known as VANET [7]. The basic principle of VANET is: within the scope of communication, vehicles are connected to realize

the interaction of the speed, location and detected environmental information to form a mobile network. The data transmission node in VANET is undertaken by vehicles, which makes the nodes in the network different from the network nodes in the general Internet of things. The nodes are not fixed but will constantly be moving and changing, which directly leads to the continuous change of the network topology structure of VANET, and the links between nodes may be disconnected at any time [8]. Moreover, as the node is the driving vehicle and the vehicle will be affected by the road environment in the process of driving, especially in the highway or urban road, the driving track of vehicles is limited, i.e., the moving track of network nodes are predictable.

The purpose of VANET is to realize the data interaction among vehicles, roads, and environment and make corresponding traffic resource allocation according to the collected data. However, the traffic environment in the road changes quickly because of the fast driving of vehicles. Therefore, VANET has strict requirements on the delay of data transmission. If data are lost because of the large time delay, the system cannot make a judgment in time. VANET needs to ensure the connectivity between nodes in data interaction. Nodes in the conventional Internet of things have fixed positions, and connectivity has been guaranteed in the design. However, most nodes in VANET are vehicles. Different vehicles have different speeds in the process of driving, and the connectivity between nodes will also change. Therefore, it is necessary to model and calculate the connectivity.

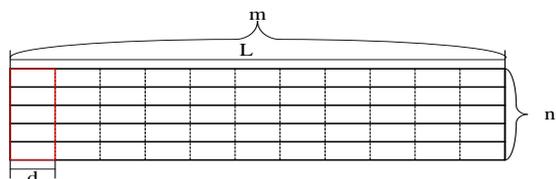


Figure 1: The schematic diagram of road division.

In this study, the connectivity model is built by dividing the road. As shown in Figure 1, the road is divided into m grids (the red box in Figure 1 is a grid), there are n lanes on the road, i.e., every grid has n lanes, and the width of every cell is d . It is assumed that there is only one vehicle on every lane in every grid, and the communication radius of the vehicle is set as $R = m_0 d$, where m_0 stands for the number of continuous grids.

The calculation formula of connectivity probability [9] is as follows:

$$\begin{cases} p_1 = 1 - p_2 \\ p_2 = \sum_{k=\max(m-o, m_0)}^{\max(m-[a/n], m_0)} p\{f(o, m) = k\} p\{g(m, k) > m_0\} \\ p\{f(o, m) = k\} = C_m^k \frac{C_{mn}^{(m-k)n}}{C_{mn}^o} \sum_{k=\max(m-o, m_0)}^m C_m^k (-1)^k \frac{C_{mn}^{(m-k)n}}{C_{mn}^o} \\ p\{g(m, k) > m_0\} = 1 - \frac{\sum_{i=k-m_0}^{\min\{k, (m-k)m_0\}} c[i]^{n-k}}{C_m^k} \end{cases} \quad (1)$$

where P_1 and P_2 are the connectivity probability and disconnectivity probability respectively, $p\{f(o, m) = k\}$ stands for the probability that there are no vehicles in k road grids, $p\{g(m, k) > m_0\}$ stands for the probability that the number of continuous grids between k road grids is larger than m_0 , and symbols in the form of C_a^b stand for the most probable number of selecting b objectives from a optional objectives for combination.

3 Data transmission optimization algorithm

VANET in the intelligent traffic system takes the vehicle as a node and transmits the information within the communication radius of the vehicle. The information interaction between vehicles constitutes the network. Naturally, for better relay transmission of information, the corresponding routing algorithm, or in other words, the data transmission path selection algorithm, is needed [10].

3.1 Ant colony algorithm

As an intelligent algorithm, the ant colony algorithm [11] is an imitation of the foraging phenomenon of an ant colony in nature, which is mostly used for issues such as TSP, scheduling optimization, and path selection. In this study, the data transmission optimization algorithm in VANET can be said to be a path selection problem, and moreover, the self-organization and dynamic variability of ant colony foraging is quite consistent with the characteristics of VANET. The ant colony algorithm is applied in VANET, and the process is as follows. Firstly, ants start from the source node and then select the next hop node from the neighbor candidate nodes according to the candidate probability. The calculation formula of candidate probability [12] is:

$$\begin{cases} p_{ij}^k(t) = \begin{cases} \frac{[\psi_{ij}(t)]^\alpha \cdot [f_{ij}(t)]^\beta \cdot [g_{ij}(t)]^\gamma \cdot [h_{ij}(t)]^\delta}{\sum_{s \in allowed_k} [\psi_{is}(t)]^\alpha \cdot [f_{is}(t)]^\beta \cdot [g_{is}(t)]^\gamma \cdot [h_{is}(t)]^\delta} & j \in allowed_k \\ 0 & otherwise \end{cases} \\ f_{ij}(t) = \frac{e_j(t)}{\sum_{s \in C(i)} e_s(t)} \\ g_{ij}(t) = \frac{(E_{\max} - e_j(t))^{-1}}{\sum_{s \in C(i)} (E_{\max} - e_s(t))^{-1}} \\ h_{ij}(t) = \frac{h_{jd}^{-1}}{\sum_{s \in C(i)} h_{sd}^{-1}} \end{cases} \quad (2)$$

where $p_{ij}^k(t)$ represents the probability of ant k jumping from node i to node j at the t -th time of traversal search, $\psi_{ij}(t)$ represents the pheromone concentration of the path between node i and j at the t -th time of traversal search, $f_{ij}(t)$ and $g_{ij}(t)$ are the energy metric parameters of node i and j at the t -th time of traversal search, $h_{ij}(t)$ represents the space parameter of node i and j at the t -th time of traversal search, $j \in allowed_k$

represents that node j belongs to the set that does not include the nodes that ant k has passed within the wireless communication range of node i , $s \in C(i)$ represents that node s belongs to the node set of node i within the wireless communication scope, $e_j(t)$ represents the energy that node j has at the t -th time of traversal search, E_{\max} represents the maximum energy that can be provided by node, h_{jd}^{-1} represents the rely hop count between node j and goal node d , and α, β, γ and δ are importance factors of corresponding parameters.

When all ants arrive at the goal node according to the selection probability, pheromones on all paths are updated. The update formula [13] is as follows:

$$\begin{cases} \psi_{ij}(t+1) = (1-\rho)\psi_{ij}(t) + \Delta\psi_{ij}(t) \\ \Delta\psi_{ij}(t) = \sum_{k=1}^n \frac{cE_{avg}^k (h_{\max} - h_k)}{wh_k} \end{cases}, (3)$$

where ρ is the pheromone volatilization coefficient, $\Delta\psi_{ij}(t)$ is the increment of pheromone between node i and j , c is a constant, w is a normal number, E_{avg}^k is the node average energy of ant k in the current path, h_{\max} is the maximum node hop count that ant in the network can realize, and h_k is the node hop count of ant k in the current path. After the pheromone is updated, the ant colony is asked to search the path again. The above steps are repeated until the termination condition is reached, and the path with the most pheromones is regarded as the optimal path.

3.2 Improved ant colony algorithm

In the process of calculating the selection probability of the next-hop node of an ant by the ant colony algorithm described above, although it takes into account the sharing of node energy efficiency so that the path can achieve the optimal energy efficiency, in the actual VANET, the premise of successful communication between nodes is that the nodes are connected, and the links between nodes may be broken as the vehicles acting as nodes are constantly changing in VANET. Only considering the energy efficiency, the nodes in the optimal path are likely to be disconnected. Therefore, this study improved the ant colony algorithm, predicted the connectivity by connectivity probability, put it into the calculation of node selection probability, and optimized the node searching of the ant colony. The process of the improved ant colony algorithm is shown in Figure 2.

① Parameters are initialized.

② Starting from the initial node, each ant chooses the next node according to the improved probability formula.

The formula of the improved selection probability is:

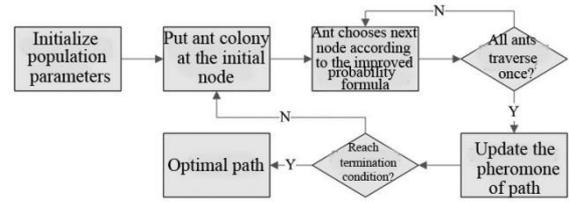


Figure 2: The process of the improved ant colony algorithm.

$$\begin{cases} p_{ij}^k(t) = \begin{cases} \frac{[\psi_{ij}(t)]^\alpha \cdot [\mu_{ij}(t)]^\beta \cdot [\eta_{ij}(t)]^\gamma \cdot [e_{ij}(t)]^\delta \cdot [C_{ij}(t)]^\lambda}{\sum_{s \in allowed_k} [\psi_{is}(t)]^\alpha \cdot [\mu_{is}(t)]^\beta \cdot [\eta_{is}(t)]^\gamma \cdot [e_{is}(t)]^\delta \cdot [C_{is}(t)]^\lambda} & j \in allowed_k \\ 0 & otherwise \end{cases} \\ C_{ij}(t) = \frac{P_{ij}(t)}{\sum_{s \in allowed_k} P_{is}(t)} \\ P_{ij}(t) = p_{con} \end{cases}, (4)$$

where $C_{ij}(t)$ represents for the metric parameter of connectivity between node i and j and λ is an importance factor of the corresponding parameter.

③ Whether all ants in the ant colony have traversed from the initial node to the target node once is determined. If not, step ② repeats; if they do, then the pheromone in the path that the ant passes through is updated, and the update formula is equation (3).

④ Whether the algorithm has reached the termination condition is determined. If it does, it outputs the optimal path; if not, it will return to step ②. The termination conditions include reaching the maximum number of iterations (taking all ants in the ant colony traversing the path once and updating the pheromone as one time of iteration) and the ant selection path converging to stability.

4 Simulation experiment

4.1 Experimental environment

In this study, the running track and running state of vehicles were simulated by VANET mobile simulator, and the network which was composed of vehicles and the routing algorithm proposed were simulated by MATLAB software [14]. VANET mobile simulator was a free software developed based on the Java platform, which can realize the macro and micro-movement simulation of vehicles, and it can make the movement of simulated vehicles closer to the actual situation. MATLAB software is a network simulation software, which builds the network based on the vehicle trajectory simulated by the VANET simulator and simulates the routing algorithm. The experiment was carried out in a server in a laboratory. The server configuration was Windows 7 system, i7 processor, and 16 G memory.

4.2 Experimental parameters

The relevant parameters of the simulation experiment are shown in Table 1, including vehicle simulation parameters and network simulation parameters. Vehicle simulation parameters included road topology size, number of vehicles, speed range, vehicle acceleration, and lane number. Network simulation parameters included simulation time, number of ant colonies, node communication radius, pheromone volatilization coefficient, minimum pheromone, E_{max} , node unit information transmitting and receiving energy and node communication protocol. Also, the Nakagami channel transmission model was adopted for the channel attenuation in the process of node communication [15]. To verify the performance of the improved algorithm, it is compared with the ant colony algorithm and geographic location-based routing algorithm.

4.3 Performance indicators

The indicators that can reflect the reliability of network data transmission include packet loss rate and time delay. Packet loss refers to the phenomenon that part of the data packets are lost due to the instability or disconnection of the node-link when the information is transmitted to other nodes in the form of data packets. The packet loss rate is the degree of packet loss. Time delay is the time taken for data transmission from one node to the goal node. For the node network, the lower the packet loss rate and time delay are, the faster and more stable data transmission is and the more reliable the whole network is. In this study, the network delay and packet loss rate were estimated by the time stamp method.

In addition to the above two indicators, the reliability of network transmission can also be measured by the statistical method. In this study, the VANET under three routing algorithms were simulated and analyzed by the Monte Carlo method many times, and the number of times that did not meet the requirements was counted to estimate the reliability of the network. The estimation formula is:

$$R_i = 1 - \frac{N_i}{i}, \quad (5)$$

where R_i stands for the estimated value of reliability after the i -th time of the simulation and N_i stands for the number of times that the network indicators did not meet the requirements after the i -th time of simulation. When the network transmits data, if the average packet loss rate was greater than 4% or the average delay was greater than 100 ms, it was considered that the network indicators of this simulation did not meet the requirements. In this study, when the Monte Carlo method was used, the precision interval was $\pm 3\%$, and 97% of them met the precision interval.

4.4 Experimental results

As shown in Figure 3, under the same number of vehicle nodes, the packet loss rate of the algorithm proposed in this study was the lowest, and the routing algorithm based

Vehicle simulation parameters	Road topology size	1000m×1000m
	Number of vehicles	10 ~ 100
	Speed range	10 ~ 30m/s
	Vehicle acceleration	2m/s ²
	Number of lanes	3
Network simulation parameters	Simulation time	900 s
	Number of ant colonies	30
	Node communication radius	150 m
	Pheromone volatilization coefficient	0.7
	Minimum pheromone	0.01
	E_{max}	1J
	Node unit information transmitting energy	4.3μJ/bit
	Node unit information receiving energy	2.4μJ/bit
	Node communication protocol	IEEE 802.11

Table 1: Related parameters of the simulation experiment.

on geographical location had the highest packet loss rate. It was seen from the curve variation that the network packet loss rate under the three routing algorithms showed a trend of decreasing first and then increasing with the increase of vehicle nodes in the network, and the number

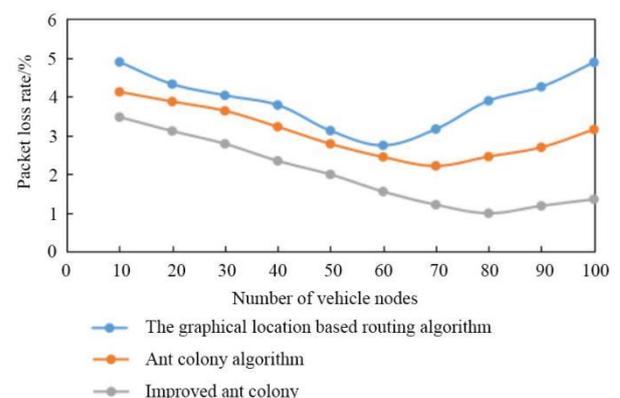


Figure 3: Relationship between network packet loss rate and nodes under three routing algorithms.

of vehicle nodes of the improved algorithm was the largest and that of the routing algorithm based on geographical location was the smallest when the packet loss rate was the lowest.

As shown in Figure 4, under the same number of vehicle nodes, the network average delay of the algorithm proposed in this study was the smallest, and that using of the routing algorithm based on geographical location was the largest. The average delay curve changes of the three routing algorithms clearly showed that the average delay of the network first decreased and then increased with the increase of the number of vehicle nodes, and the number of vehicle nodes of the ant colony algorithm and improved ant colony algorithm was the largest when the average time delay was the smallest and increased gently afterward.

The simulation model was simulated by the Monte Carlo method for many times, and the simulation times that did not meet the set indicators were counted to evaluate the reliability of the network. The results are shown in Figure 5. It was seen from Figure 5 that the network reliability of the improved ant colony algorithm was the highest, and the network reliability of the routing algorithm based on geographical location was the lowest under the same number of vehicle nodes. It was seen from the curve changes that the network reliability of the routing algorithm based on geographical location and ant colony algorithm increased first and then decreased, while the network reliability of the improved ant colony algorithm increased first and then basically stabilized at 0.99.

The above three results showed that the improved ant

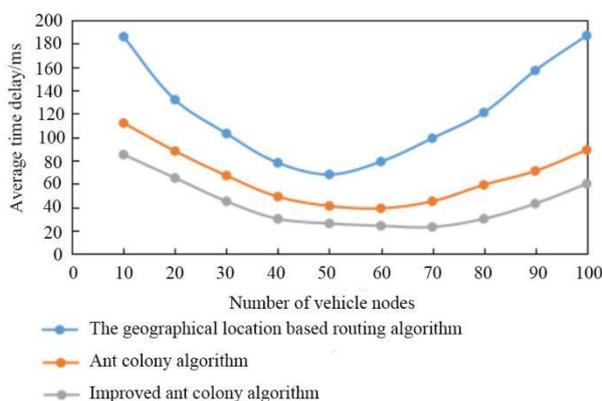


Figure 4: Relationship between average network delay and nodes under three routing algorithms.

colony algorithm could effectively reduce the packet loss rate and average delay in the data transmission between nodes in VANET and improve the reliability of network data transmission. Although the packet loss rate and average delay of the network increased when the number of vehicles in VANET increased to a certain extent, under the effect of the improved ant colony algorithm, the increase amplitude was smaller than the other two

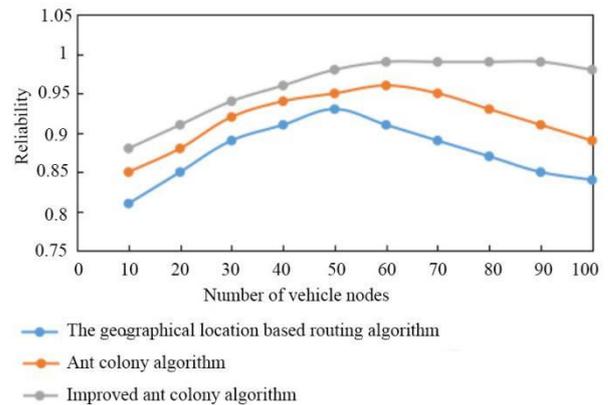


Figure 5: Relationship between network reliability and nodes under three routing algorithms.

algorithms, and the reliability of network transmission was maintained at a high level. The reason why the packet loss rate and time delay decreased and the reliability increased was that the number of nodes that could transmit smoothly around the nodes increased when the number of vehicle nodes in the network increased, which led to more choices of the data transmission path. The reason why the packet loss rate and time delay increased and the reliability decreased was that when the vehicle nodes increased to a certain extent, the interference of the surrounding node signals increased although the optional excellent path increased. However, the improved ant colony algorithm not only considered the energy efficiency but also considered the connectivity between vehicles, which made the path selection more inclined to the path with stronger connectivity to slow down the increase of packet loss rate and delay and maintain the stability of reliability.

5 Conclusion

This paper briefly introduced the vehicle communication network, VANET, and the ant colony algorithm for path search in the network, improved the ant colony algorithm, carried out a simulation experiment on the improved ant colony algorithm on the simulation platform, and compared it with the traditional ant colony algorithm and the geographical location-based routing algorithm. The results showed that: (1) with the increase of vehicle nodes in VANET, the packet loss rate presented a tendency of increasing first and then decreasing, the packet loss rate of the improved ant colony algorithm was always the lowest, and the packet loss rate of the geographical location-based routing algorithm was always the highest; (2) with the increase of vehicle nodes, the average delay in the network firstly decreased and then increased, the improved ant colony algorithm was always the smallest, and the geographical location based routing algorithm was always the largest; (3) the reliability of network data transmission increased and then decreased with the increase of nodes, but the reliability of the network under the improved ant colony algorithm rose first and then tended to be stable, the network reliability under the improved ant colony algorithm was always the highest and that of the routing

algorithm based on geographical location was always the lowest.

References

- [1] Bitam S, Mellouk A, Zeadally S (2015). VANET-cloud: a generic cloud computing model for vehicular Ad Hoc networks. *IEEE Wireless Communications*, 22(1), pp. 96-102.
- [2] Amoozadeh M, Deng H, Chuah CN, Zhang HM, Ghosal D (2015). Platoon management with cooperative adaptive cruise control enabled by VANET. *Vehicular Communications*, 2(2), pp. 110-123.
<https://doi.org/10.1016/j.vehcom.2015.03.004>
- [3] Melaouene N, Romadi R (2019). An enhanced routing algorithm using ant colony optimization and VANET infrastructure. *MATEC Web of Conferences*, 259(6), pp. 02009.
<https://doi.org/>
- [4] Kumar N, Dave M (2016). BIIR: A Beacon Information Independent VANET Routing Algorithm with Low Broadcast Overhead. *Wireless Personal Communications*, 87(3), pp. 869-895.
<https://doi.org/10.1007/s11277-015-2620-y>
- [5] Saxena R, Jain M, Sharma DP, Jaidka S, Thampi SM, El-Alfy EM (2019). A review on VANET routing protocols and proposing a parallelized genetic algorithm based heuristic modification to multicast routing for real time message passing. *Journal of Intelligent & Fuzzy Systems*, 36(3), pp. 2387-2398.
<https://doi.org/10.3233/jifs-169950>
- [6] Li N, Martinez-Ortega JF, Diaz VH, Fernandez JAS (2017). Probability Prediction based Reliable Opportunistic (PRO) Routing Algorithm for VANETs. *IEEE/ACM Transactions on Networking*, PP(99).
<https://doi.org/10.1109/TNET.2018.2852220>
- [7] Li AN, Zhao YM, Xie P, Wang Q (2016). Research on routing algorithm based on the VANET. *MATEC Web of Conferences*, 44, pp. 01085-.
<https://doi.org/10.1051/mateconf/20164401085>
- [8] Kumuthini C, Krishnakumari P (2015). An access point based routing algorithm (APBR) using greedy method towards improving quality of service for VANET. *International Journal of Applied Engineering Research*, 10(3), pp. 5489-5502.
- [9] Saravanan M, Ganeshkumar P, Kumar SM (2018). Survey on opportunistic routing algorithm for Vehicular Adhoc Network (VANET). *International Journal of Pure and Applied Mathematics*, 118(20), pp. 1735-1740.
- [10] Aadil F, Bajwa KB, Khan S, Chaudary NM, Akram A (2016). CACONET: Ant Colony Optimization (ACO) Based Clustering Algorithm for VANET. *Plos One*, 11(5), pp. e0154080.
<https://doi.org/10.1371/journal.pone.0154080>
- [11] Goudarzi F, Asgari H, Al-Raweshidy HS (2018). Traffic-aware VANET routing for city environments—A protocol based on ant colony optimization. *IEEE Systems Journal*, pp. 1-11.
- [12] Deepa Thilak K, Amuthan A (2017). Cellular Automata-based Improved Ant Colony-based Optimization Algorithm for mitigating DDoS attacks in VANETs. *Future Generation Computer Systems*, 82(MAY), pp. 304-314.
<https://doi.org/10.1016/j.future.2017.11.043>
- [13] Goudarzi F, Asgari H, Al-Raweshidy HS (2019). Traffic-Aware VANET Routing for City Environments—A Protocol Based on Ant Colony Optimization. *IEEE Systems Journal*, 2019, 13(1), pp. 571-581.
- [14] Jindal V, Bedi P (2017). Preemptive MACO (MACO-P) Algorithm for Reducing Travel Time in VANETs. *Applied Artificial Intelligence*, pp. 74-196.
<https://doi.org/10.1080/08839514.2017.1300017>
- [15] Lakshmanaprabu SK, Shankar K, Rani SS, Abdulhay E, Arunkumar N, Ramirez G, Uthayakumar J (2019). An effect of big data technology with ant colony optimization based routing in vehicular ad hoc networks: Towards smart cities. *Journal of Cleaner Production*, 2019, 217(APR.20), pp. 584-593.
<https://doi.org/10.1016/j.jclepro.2019.01.115>

Automatic Image Segmentation for Material Microstructure Characterization by Optical Microscopy

Naim Ramou, Nabil Chetih, Yamina Boutiche and Abdelkader Rabah
Research Center in Industrial Technologies CRTI, P.O. Box 64, Cheraga, 16014, Algiers, Algeria
E-mail: n.ramou@crti.dz

Keywords: level set, microstructure characterization, image segmentation

Received: January 13, 2020

This work shows the microstructure characterization utility for the analysis of material properties. To achieve this purpose, digital image segmentation is used on microscopic images of materials to extract the number of phases and their proportion present in the material to obtain a quantitative description of material properties and to better control product quality. In this way, we present here an automated method for segmenting the phases present in microscopic scanning images of metallographic samples using a multiphase level set with Mumford Shah formulation. Experience shows that the proposed model successfully detects phase regions for a variety of real micrographic images

Povzetek: Predstavljena je metoda za segmentiranje slik, pridobljenih z optičnim mikroskopom, za ugotavljanje lastnosti materiala..

1 Introduction

The rate of ferrite in steel has a direct effect on its functional properties (yield strength, toughness, hardness, corrosion resistance, weldability, mouldability, embrittlement, magnetism). Its control and measurement are consequently very important. For this microstructures are used to determine the properties of materials. A full description of microstructures means giving a description of the size, shape and distribution of grains and second phase particles and their composition. This is why it is essential to establish the link between phenomena occurring at the microstructural scale and the properties of the material [1-5]. The microstructures are made up of a set of elements organized in the microscopic scale. Their observation and thus their characterization require the use of microscopic techniques. The phases differ from each other by their crystalline, semi crystalline or amorphous structure. Morphologies are observed by the optical or electronic microscope.

The microstructures formed in the materials depend not only on the composition or on the chemical structure of the material but also on the existence of gradients of temperature or on concentration inside this one during its transformation. The microstructures are also strongly influenced by the energy needed for the creation of the new interfaces. Most of the microstructures which are formed during the solidification are crystalline in nature. The glass is always less stable than the crystal if it can form. In a number of cases, however, an amorphous (vitreous) structure appears during a fast cooling. Let us note that it is because of the absence of microstructure that lenses have their transparency. Some materials have a very irregular molecular structure and are not capable of developing a stable crystalline structure.

They exist, in the solid state, only under a glassy shape, whatever are the conditions of cooling (atactiques polymers). The microstructure of the organic polymers is largely controlled by their chemical structure. If macromolecules have a regular molecular structure, the crystallization occurs generally. Note that the crystallization of polymeric materials is never complete (semi-crystalline structure) and form spherulites.

The kinetic characteristics of phase transformations allow leading, by adequate heat treatments (tempering-annealing), very different microstructures, which have generally a nonhomogeneous composition and which are almost always metastable at the temperature of use. In the case of metals and their alloys, heat treatments combined with mechanical treatments, as the lamination, have reached a very high degree of sophistication. Ceramics are often obtained by sintering of powder which explains the presence of pores that are an important part of their microstructure.

A variation of the microstructure over time during use entrains an important modification of its properties (aging phenomenon). Several works have been done in this subject. Furthermore, a study has been made around the development of microstructure and grain border character distribution at different temperatures [6]. Moreover, studies were made in the domain of segmentation of the materials micrographic images for the extraction of grains/phases which are fundamental to realize the microstructure descriptions. An automated algorithm used for segmenting the phases present in the scanning electron microscopic images of dual phase steel [7-9]. A level set segmentation model was proposed to analysis the images of particle to overcome the drawback of sensitivity to weak boundaries and curve's initial position [10]. In this way, we thus propose a different approach for image segmentation which is based on using a multiphase

level set Mumford and Shah to segment the phases present in the scanning microscopic images of metallographic samples. To achieve this goal, we have to face in several complex problems such as the images quality and the presence of noises.

The paper is organized as follows: Section 2 presents the materials and methods; in this section we divide the work into two parts: the first one is devoted to the description the material studied with all the necessary steps of preparation and acquisition of images used in this work. In the second part we describe and analyze the Mumford and Shah model used for image segmentation (multiphase case). Section 3 shows the main contribution of this work which is the automatic distinguish between phases. We will illustrate the experimental results. Finally, conclusions are reported in Section 4.

2 Materials and methods

2.1 Sample preparation and image acquisition

The acquisition and the information on different micrographic images that we used in this work are obtained from the NIKON optical microscope (Eclipse I.V 100 ND) figure 1 with camera equipped by an acquisition system (LV-LH50PC 12V50W Precentered Lamphouse Bright/darkfield switch and linked aperture stop (centerable).), it is possible to visualize all types of surfaces with a magnification up to 1500 times. The base metal that we used for the realization of this work is a stainless steel two-phase austeno-ferritic (duplex) of nuance 2205.

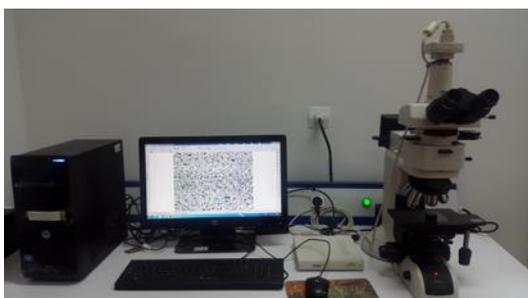


Figure 1: NIKON optical microscope (Eclipse I.V 100 ND).

Besides, the image processing which allows transforming easily the raw data into exploitable information, it is therefore possible to perform microscopic studies. For this a careful preparation are several successive steps for observation at the optical microscope: Starting with the coating of the samples in a hot phenolic resin, then the mechanical polishing under water and on an abrasive paper more and more fine (paper 320, 400, 500, 800, 1000, 1200, 2400, 4000), then the polishing of finishing by Felt sheet with use of diamond paste. Finally, after finishing of polishing is performed a chemical at- tack ($H_2O + HF$) of the surface to reveal the microstructure. In our work, we used the three micrographic images at 200

magnification of 2205 duplex stainless steel heat-treated at temperatures respectively: $800^\circ C$, $850^\circ C$ and $950^\circ C$. Clearly and obviously, after observing the Figure 1, we notice that the three micrographic images of steel contain three different zones: Ferrite, austenite and the third zone known as sigma phase [11-12]; with 30 atoms per mesh (Fig 2).

In addition to iron, the sigma phase contains

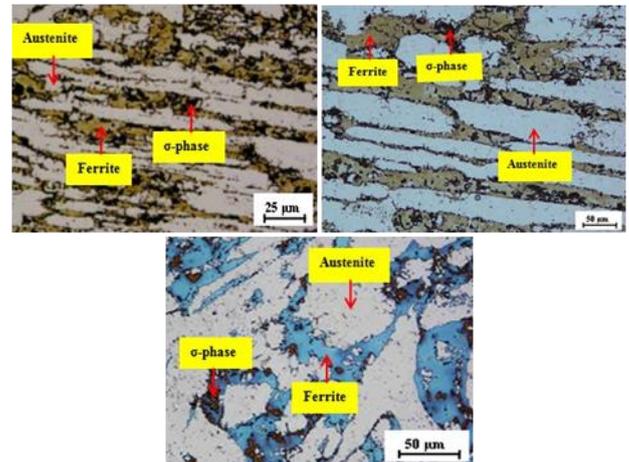


Figure 2: Optical micrograph of 2205 duplex stainless steel after treatment temperatures: (A) at $800^\circ C$; (203×288 pixels); (B) at $850^\circ C$; (236×315 pixels); (C) at $950^\circ C$; (203×288 pixels).

chromium and molybdenum which it draws from the matrix, thus causing a reduction in the corrosion resistance of Fe-Cr-Ni systems. Moreover, this phase, which is formed in the temperature range between 600 and $1000^\circ C$., causes a dramatic loss of the toughness of the stainless steels [13]. Precipitation of the sigma phase depends not only on the chemical composition of the steel [14]. Indeed, other factors influence its formation such as the size of the grains. Also, the sigma phase is more easily formed in high energy regions such as grain boundaries and interfaces. The solution temperature also affects precipitation in two ways:

- The high dissolving temperatures induce grain enlargement, which reduces the rate of sigma phase formed.
- On the other hand, at high temperatures, the ferrite content is increased, which at first glance encourages the precipitation of the sigma phase during the aging treatments.

The sigma phase appears preferentially in the Austenite/Ferrite phase joints, but it can also appear in the grain boundaries Ferrite/Ferrite and Austenite/Austenite. It germinates in the Ferrite/Austenite interfaces and then grows inside the ferritic grains. An illustrative diagram of the precipitation of σ is given in Fig. 2 and 3. The formation of the σ phase, which is rich in Ferrite elements, causes the adjacent ferritic regions to become depleted in these elements, leading to the transformation of this ferrite into the secondary austenite. In this case, formation of the sigma phase takes place in the vicinity of the chromium nitride

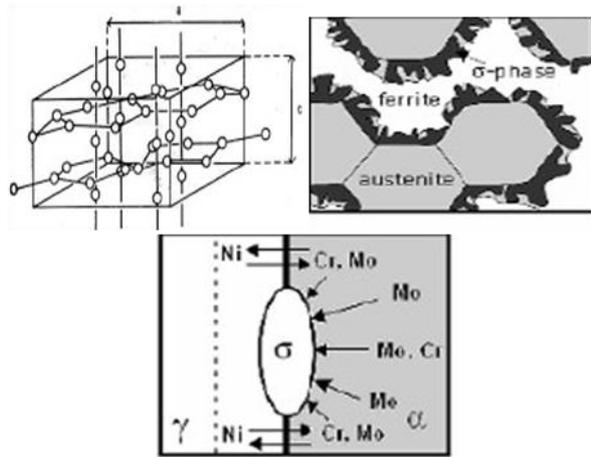


Figure 3: (a) Structure of the quadratic sigma phase with parameters: $a = 8.970$ and $c = 4.558$; (b) diffusion of the atoms of alpha-elements of the ferritic grain (α) towards austenite (γ) and σ ; (c) Illustrative schema of the germination of the sigma phase at the interface Austenite/Ferrite and its growth inside the ferrite [12].

particles in the austeno-ferritic stainless steels having a high content of nitrogen (about 1% by weight).

2.2 Level set formulation

Mumford and Shah have proposed a variational model that defines image segmentation as a problem of joint detection of homogeneous zones and contours [15]. The Mumford formulation model is based on the minimization of an energy function E where u represents our image with value bounded in Ω from \mathbb{R}^2 to \mathbb{R} and Γ_i the contour of each region R_i on which the image is approached by a function g_i ; minimizing functional can be written as:

$$E(\Gamma_i, g_i, u) = \lambda \iint_{\Omega} (u - g_i)^2 dx dy + \iint_{R_i} |\nabla g_i|^2 dx dy + \mu \int_{\Gamma_i} dl \quad (1)$$

Where λ and μ are positive real parameters for weighting the data fidelity term and long-term contours respectively. In the simple case where the functions g_i are constants, the solution always exists. It can then be shown that the value of the g_i on region R_i is the average denoted c_i of u restricted to R_i . In this framework, energy can therefore be rewritten as follows:

$$E(\Gamma_i, u) = \lambda \int_{\Omega} (u - c_i)^2 dx dy + \mu \int_{\Gamma_i} dl \quad (2)$$

With

$$c_i = \frac{\int_{R_i} u dx dy}{\int_{R_i} dx dy}$$

In this formalization, the curve C is represented by the level line zero of a lipschitzienne function Φ such as

$$\begin{aligned} \Phi &= 0 \\ \Phi &> 0 \text{ inside } C \\ \Phi &< 0 \text{ outside } C \end{aligned} \quad (3)$$

So in this approach the unknown is not any more C but Φ . If we introduce the function of Heaviside H and its derivative δ_0 in the sense of the distributions defined by

$$H(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases} \quad (4)$$

$$\delta_0 = \frac{d}{dz} H(z) \quad (5)$$

We have then:

$$\text{length } C = \iint_{\Omega} |\nabla H(\Phi)| dx dy = \int_{\Omega} \delta_0(\Phi) |\nabla(\Phi)| dx dy \quad (6)$$

And

$$\begin{aligned} \int_{\Phi > 0} |u - c_1|^2 dx dy &= \int_{\Omega} |u - c_1|^2 H(\Phi) dx dy \\ \int_{\Phi < 0} |u - c_2|^2 dx dy &= \int_{\Omega} |u - c_2|^2 (1 - H(\Phi)) dx dy \end{aligned}$$

From there, the energy functional can be written:

$$\begin{aligned} E(c_1, c_2, \Phi) &= \mu \int_{\Omega} \delta_0(\Phi) |\nabla(\Phi)| dx dy \\ &+ \lambda_1 \int_{\Omega} |u - c_1|^2 H(\Phi) dx dy \quad (7) \\ &+ \lambda_2 \int_{\Omega} |u - c_2|^2 (1 - H(\Phi)) dx dy \end{aligned}$$

To minimize $E(c_1, c_2, \Phi)$ we write the equation of Euler-Lagrange for Φ . This amounts to canceling the differential in Φ of energy. We obtain:

$$\delta_0(\Phi) [\mu \operatorname{div}(\frac{\nabla \Phi}{|\nabla \Phi|}) - \lambda_1 (u - c_1)^2 + \lambda_2 (u - c_2)^2] = 0 \quad (8)$$

If one then performs a gradient descent by introducing a parameter of time, we get an evolution of Φ which is given by the following equation:

$$\frac{\partial \Phi}{\partial t} = \delta_0(\Phi) [\mu \operatorname{div}(\frac{\nabla \Phi}{|\nabla \Phi|}) - \lambda_1 (u - c_1)^2 + \lambda_2 (u - c_2)^2] \quad (9)$$

2.3 Description of the Piecewise-Constant model (Multiphase case)

It is common to keep the model of Mumford-Shah piecewise constant with respect to the level set. In other words, the resulting image is approximated by a finite number of constants which represent the means of the intensity at the inside and the outside of each of the curves. Multiphase segmentation is better because it can characterize objects differently. It consists in making several level set functions. In Figure 4 by using two functions of level set [16-18], we can identify the following four regions:

$$\begin{aligned} c_{00} &= \{\text{mean}(u_0) \text{ in } (\Phi_1 > 0 \text{ and } \Phi_2 > 0)\} \\ c_{01} &= \{\text{mean}(u_0) \text{ in } (\Phi_1 > 0 \text{ and } \Phi_2 < 0)\} \\ c_{10} &= \{\text{mean}(u_0) \text{ in } (\Phi_1 < 0 \text{ and } \Phi_2 > 0)\} \\ c_{11} &= \{\text{mean}(u_0) \text{ in } (\Phi_1 < 0 \text{ and } \Phi_2 < 0)\} \end{aligned} \quad (10)$$

$$\begin{aligned} c_{00}(\Phi_1, \Phi_2) &= \text{mean}(u_0) \text{ in } \{x: \Phi_1 > 0, \Phi_2 > 0\} \\ &= \frac{\int_{\Omega} u_0 H(\Phi_1) H(\Phi_2) dx dy}{\int_{\Omega} H(\Phi_1) H(\Phi_2) dx dy} \end{aligned}$$

$$\begin{aligned} c_{01}(\Phi_1, \Phi_2) &= \text{mean}(u_0) \text{ in } \{x: \Phi_1 > 0, \Phi_2 < 0\} \\ &= \frac{\int_{\Omega} u_0 H(\Phi_1) (1 - H(\Phi_2)) dx dy}{\int_{\Omega} H(\Phi_1) (1 - H(\Phi_2)) dx dy} \end{aligned}$$

$$\begin{aligned} c_{10}(\Phi_1, \Phi_2) &= \text{mean}(u_0) \text{ in } \{x: \Phi_1 < 0, \Phi_2 > 0\} \\ &= \frac{\int_{\Omega} u_0 (1 - H(\Phi_1)) H(\Phi_2) dx dy}{\int_{\Omega} (1 - H(\Phi_1)) H(\Phi_2) dx dy} \end{aligned}$$

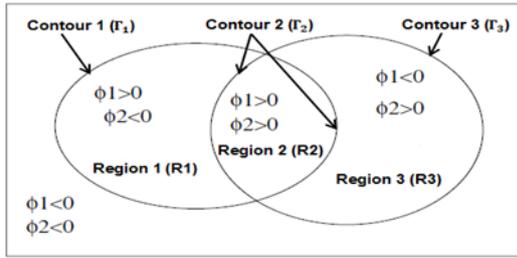


Figure 4: Two initial curves of evolution which divide the image in four regions.

$$c_{11}(\Phi_1, \Phi_2) = \text{mean}(u_0) \text{in}\{x: \Phi_1 < 0, \Phi_2 < 0\} \\ = \frac{\int_{\Omega} u_0 (1 - H(\Phi_1))(1 - H(\Phi_2)) dx dy}{\int_{\Omega} (1 - H(\Phi_1))(1 - H(\Phi_2)) dx dy}$$

The resolution by the associated Euler-Lagrange equation [17] leads to the evolution equations formulated by Φ_1, Φ_2 :

$$\frac{\partial \Phi_1}{\partial t} = \delta_{\epsilon}(\Phi_1) \{ v \nabla \left(\frac{\nabla \Phi_1}{|\nabla \Phi_1|} \right) - (|u_0 - c_{11}|^2 - |u_0 - c_{01}|^2) H(\Phi_2) \\ + (|u_0 - c_{10}|^2 - |u_0 - c_{00}|^2) (1 - H(\Phi_2)) \} \\ \frac{\partial \Phi_2}{\partial t} = \delta_{\epsilon}(\Phi_2) \{ v \nabla \left(\frac{\nabla \Phi_2}{|\nabla \Phi_2|} \right) - (|u_0 - c_{11}|^2 - |u_0 - c_{10}|^2) H(\Phi_2) \\ + (|u_0 - c_{01}|^2 - |u_0 - c_{00}|^2) (1 - H(\Phi_2)) \} \quad (11)$$

In Figs 5 and 6 present respectively the segmentation steps and automatic detection of phases of our proposed method for synthetic image

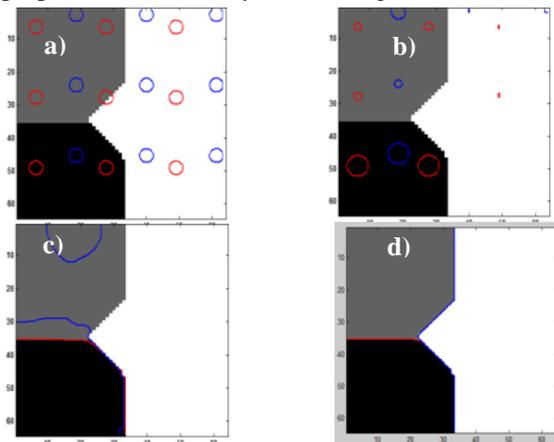


Fig.5. The segmentation steps of synthetic image : (a)Initial contour; (b) After 10 iterations; (c) After 30 iterations; (d) Finale contour.

3 Results

3.1 The conventional micrographic measurement approach

The conventional approach to measuring ferrite in steel is by counting on a micrographic cup. The sample, prior to measurement, requires prior preparation. First of all, it must be small in size (cut-out of the part). The sample must then be coated (to make it easier to hold and eliminate edge effects). Then, the specimen must be

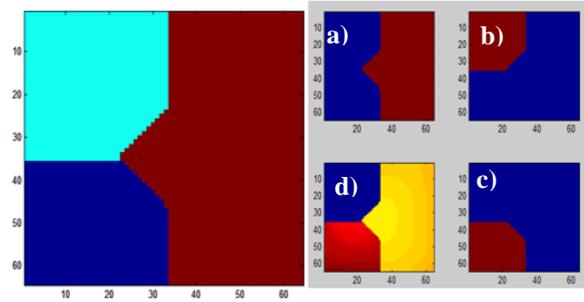


Fig. 6. Automatic detection of phases of synthetic image: a) First phase. b) Second phase. c) Third phase. d) Segmented image.

polished (mirror polished and free of scratches that could hinder observation). Finally, the preparation requires acid etching (to reveal ferritic and austenitic structures). Analyses are to be carried out according to ISO 9042 standards. For this norm, the method for determining the volume fraction of a constituent consists first of all in choosing the grid of points (dimensions and number of points) according to the constituent to be studied. Then, the chosen grid of points is superposed on the metallographic cup. An

Sample	Temperature /1h	Ferrite rate	Microscopic image
Sample 1	850°C	17%	
Sample 2	950°C	20%	

Table 1: Calculate of Ferrite rate.

enlargement will be chosen to show the delimitation between the phases. Next, the number of points of the grid included in the component whose rate must be determined is counted. Finally, its volume fraction is deduced (Table 1).

When counting, the values found in the fields are entered in an Excel table (which includes the calculations of the norm). When a field has been analyzed, the sample must be moved to observe another area. There are also other values calculated such as the 95% confidence interval. This method is relatively time-consuming because the sample has to be prepared before the analysis begins. In addition, it is destructive.

It also requires special attention on the part of the user. Indeed, a parallax error can lead to a bad count. The choice of magnification can also influence the measured phase rate. These errors can be limited by increasing the number of observed fields. Phase rate analysis by image analysis [18] can be used.

3.2 The proposed approach using the Piecewise-Constant model (Multiphase case)

Figure 7 illustrates the results of some of the steps to the segmentation of material microstructure images using a multi-circle type initialization of the initial contour with $\mu = 0.25 \cdot 10^{-5}$, $N = 50$ iterations and $O_t = 0.5$. To confirm the role of the segmentation step in the information analysis for the microstructure characterization, we have plot the histogram of micrographic image before segmentation step and after segmentation (figure 8). From this result we can say that after segmentation application we can used a simple thresholding to separate the phases in image, which he is not the case before segmentation step.

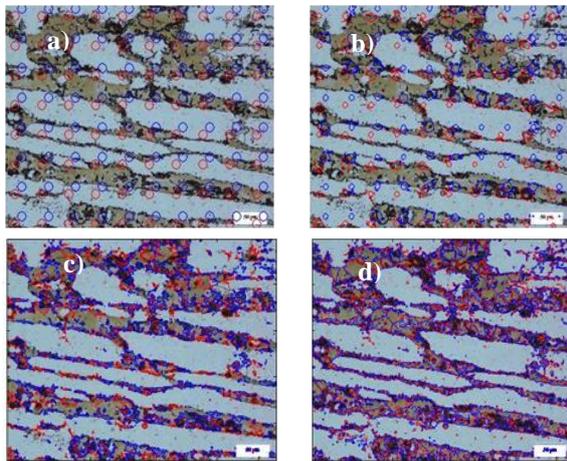


Fig. 7. The segmentation steps of micrographic image : (A) Initial contour; (B) After 10 iterations; (C) After 30 iterations; (D) Finale contour.

3.3 Phases rate calculation

For the characterization of the compositions of microstructure scale, it is really difficult to make the difference between the phases in order to calculate the phase rate using a micrographic image. In this work, we have proposed an automatic thresholding after segmentation application to detect the number of the phases as well as their percentage. The idea consists in the application of the segmentation multiphase follow-up by a histogram to separate the phases by a threshold

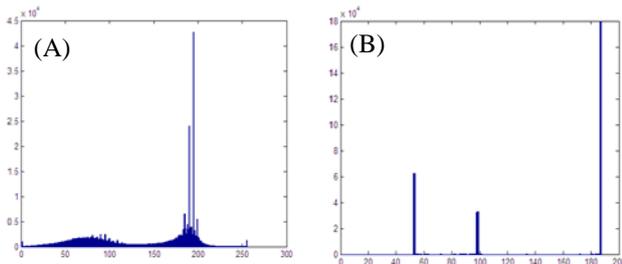


Fig. 8. Histogram of micrographic image: (A) before segmentation; (B) After segmentation, blue represent respectively the phases: austenite, ferrite and σ – phase.

which is the average value of every phase. To separate the phases we used an automatic thresholding, the idea is based on the calculation of the average between the maximum values of each phase in Figure 9:

$$\begin{cases} Phase1 < S_{m1} \\ S_{m1} < Phase2 < S_{m2} \\ Phase1 > S_{m3} \end{cases} \quad (12)$$

In the case of Fig 9 we have two average values S_{m1} and S_{m2} to separate the three phases in image as follow:

$$Pr = \frac{\text{the number of pixel of } (S_i < Phase < S_{i+1})}{\text{the number of pixel of the image}} \quad (13)$$

We present in Figures 9 and 10 the results that have been obtained by applying our proposed segmentation method on different micrograph images with no a priori knowledge about the number of phases of each image. This method makes the automatic detection of the number of phases of each image followed by a percentage of each phase representing the surface of the image. Three colours: yellow, green and blue represent respectively the phases: austenite, ferrite and σ phase.

4 Conclusion

Our contribution in this paper consists of determining automatically the number of phases and their proportion in a sample of metallic materials from a micrographic image. To reach this goal we used the variational approaches. The results obtained show that we have arrived to calculate the phase rates in an automatic way without the intervention of experts; we have applied this method on several images to validate the algorithm, and offer the expert a better micrographic image processing, which allows a reliable and reproducible results.

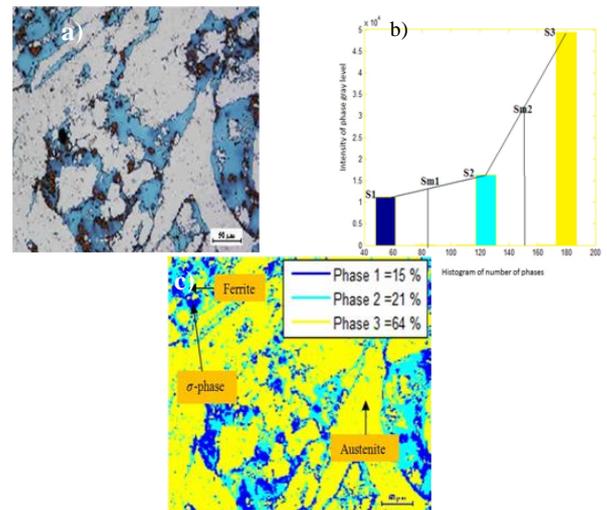


Fig. 9. Automatic detection of phases of micrographic image. (A) Original image; (B) Histogram of number of phase; (C) Segmented image.

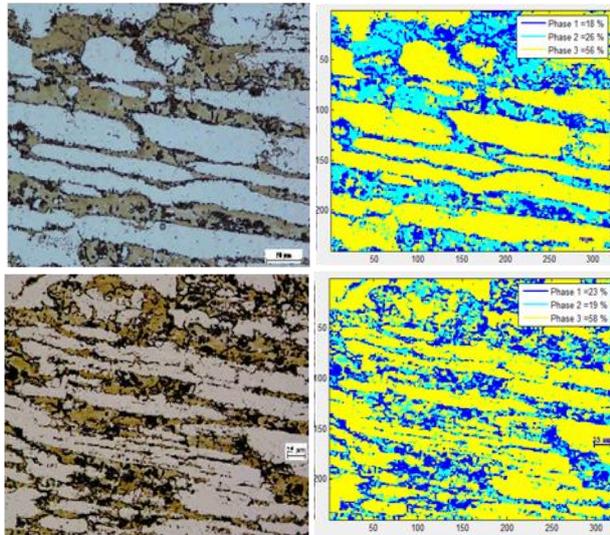


Figure 10: Results of segmentation micrographic images by proposed method.

References

- [1] Moghimi, M.K., Mohanna, F., (2019) A joint adaptive evolutionary model towards optical image contrast enhancement and geometrical reconstruction approach in underwater remote sensing. *SN Appl. Sci.* 1, 1242
<https://doi.org/10.1007/s42452-019-1255-0>
- [2] Joshua P. (2013) Mechanical Properties of Materials. *Solid Mechanics and Its Applications*.
- [3] Lovell M.C., Avery A.J., Vernon M.W. (1976) Physical Properties of Materials. *The Modern University Physics Series*.
- [4] Laszlo S., Totha b., Chengfan G.C. (2014) Ultrafine-grain metals by severe plastic deformation. *Microstructure characterization*. 92,1–14. [4]Sarkhawas G., Arti B. (2015) Particle Analysis Using Improved Adaptive Level Set Method Based Image Segmentation. *International Conference on Computing Communication Control and Automation*. Pune. India, 747– 751.
<https://doi.org/10.1109/iccubea.2015.149>
- [5] Mohammadi J., Behnamian Y., Mostafaei A., et al (2015) Friction stir welding joint of dissimilar materials between AZ31B magnesium and 6061 aluminum alloys: Microstructure studies and mechanical characterizations. *Microstructure characterization*. 101,189–207.
<https://doi.org/10.1016/j.matchar.2015.01.008>
- [6] Huang W.,Chai L.,Li Z.,Yang X., Guoc N.,Song B (2016) Evolution of microstructure and grain boundary character distribution of a tin bronze annealed at different temperatures'. *Microstructure characterization*. 114,204–210.
<https://doi.org/10.1016/j.matchar.2016.02.022>
- [7] Chatterjee O.,Das K.,Dutta S.,Datta S.,Saha S.K. (2010) Phase Extraction and Boundary Removal in Dual Phase Steel Micrographs. *IEEE India* <https://doi.org/10.1109/indcon.2010.5712693>
- [8] Murase K.,Sugal S. (2013) Segmentation of dual phase steel micrograph: An automated approach. *Measurement*. 46,2435–2440.
- [9] Choudhury, A., Pal, S., Naskar, R. and Basumallick, A. (2019), "Computer vision approach for phase identification from steel microstructure", *Engineering Computations*, Vol. 36 No. 6, pp. 1913-1933.
<https://doi.org/10.1108/ec-11-2018-0498>
- [10] Halimi, M. & Ramou, N. (2013) Extraction of weld defects dimension from radiographic images using the level set segmentation without reinitialization *Russ J Nondestruct Test* 49: 424.
<https://doi.org/10.1134/s1061830913070036>
- [11] Vanderschaeve E., Taillard R. and Foct J. (1994) Etude des phnomnes de prcipitation dans un acieraustnitique 19% de chrome et 19% de manganse, et ^rs forte teneur en azote. *J. Phys IV. Colloque C3, supplment au Journal de Physique III.4*. <https://doi.org/10.1051/jp4:1994312>
- [12] Zucatto I., Moreira M.C., Machado I.F., And Lebrao S.M.G. (2002) Microstructural Characterization and The Effect of Phase Transformations on Toughness of The UNS S31803 Duplex Stainless steel Aged treated at 850 C. *Materials Research*. 5(3),385-389.
<https://doi.org/10.1590/s1516-14392002000300026>
- [13] Lacombe P., Baroux B., Beranger G. (1990) Les aciers inoxydables.
- [14] Chen T.H., Weng, K.L. And Yang, J.R. (2002) The Effect Of High Temperature Exposure On The Microstructural Stability And Toughness Property In A 2205 Duplex Stainless Steel. *Materials Science And Engineering, A* 338,259-270.
[https://doi.org/10.1016/s0921-5093\(02\)00093-x](https://doi.org/10.1016/s0921-5093(02)00093-x)
- [15] D. Mumford and J. Shah. (1989) Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*. XLII, 577–685.
- [16] M. Rousson and R. Deriche. (2002) A Variational Framework for Active and Adaptative Segmentation of Vector Valued Images. *Technical Report*. 4515.Inria.France.
<https://doi.org/10.1109/motion.2002.1182214>
- [17] Chan T.F. and Vese L.A. (2000) Image segmentation using level sets and the piecewise constant Mumford-Shah model. *Tech.Rep. UCLA Dept. Math*, CAM 00-14.
- [18] Vese L.A. and Chan T.F. (2002) A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model. *International Journal of Computer Vision*. 50(3),271293.

Smart Design for Resources Allocation in IoT Application Service Based on Multi-agent System and DCSP

Mouadh Bali

LIMED Laboratory, Faculty of Exact Sciences, University of Bejaia, Algeria
Dept. Computer Science, Faculty of Exact Sciences, University of El Oued, Algeria
E-mail: bali-mouadh@univ-eloued.dz

Abdelkamel Tari

LIMED Laboratory, Faculty of Exact Sciences, University of Bejaia, Algeria
E-mail: tarikamel59@gmail.com

Abdallah Almutawakel and Okba Kazar

LINFI laboratory, Computer science department, University of Biskra, Algeria
E-mail: aboud.aboud2012@gmail.com, o.kazar@univ-biskra.dz

Keywords: IoT, IoT service, resource allocation, cloud computing, distributed constraints satisfaction problems, multi agent system

Received: September 21, 2019

In the present paper, we aim at solving two problems; the first problem occurring in the transformation of the IoT devices (sensors, actuators, ...) to cloud service. Therefore, we work on maintaining a smooth and efficient data transmission for the cloud and support customer applications like: data sharing, storage and processing. The second problem has two dimensions. In the first dimension, the problem is arisen in the submission of cloudlets (customer requested jobs) to Virtual Machines (VMs) in the hosts. To solve this problem, we propose scheduling algorithm for resource allocation according to the lowest cost and load. In the second dimension, the problem lies in the hosting of new VMs in the hosts. To overcome this problem, we need take into account the loads when housing new VMs in different datacenters. In this work, we suggest a resource allocation approach for services oriented IoT applications. The architecture of this approach is based on two technics: Multi Agent System (MAS) and Distributed Constraint Satisfaction Problems (DCSP). The MAS manages the physical resources, making decision and the communication between datacenters, while DCSP used to simplify the policy of the resources provisioning in Datacenters. Variables and constraints are distributed among multiple agents in different layers. The experimental results show that the efficiency of our approach is manifested in: Average System Load, Cost augmentation Rate and Available Mips.

Povzetek: Predlagan je način dodeljevanja virov za storitve v IoT aplikacijah na osnovi večagentnih sistemov (MAS) in zadovoljevanja porazdeljenih omejitev (DCSP).

1 Introduction

Internet of Things (IoT) and Cloud Computing are two paradigm technologies utilized for a wide range of application in our life. IoT is a smart system to connect physical objects with sensors to enable them to collect and share the data via the internet [18].

The cloud is type of parallel and distributed systems. It is described as a model for application execution and data storage [19],[2] Cloud infrastructure allows customers using a large number of resources such as: network, storage and applications [1]. The data centers have a large number of resources commonly known as RA [20]. In cloud computing, RA is an issue due several challenges such as complexity, heterogeneity of resource that resides in the datacenter, scheduling, virtualization, migration [2],[3].

The motivation for studying this problem comes from IoT limited properties including: limited storage capacity

and complicated processes (data analysis and a lot of heterogeneity in the devices) [18]. As result, we work on satisfying users' needs by providing resources allocation with lower cost. This cost is computed on the basis of smart solutions in datacenters (best host) according the resource constraints [8]. We provide a distributed resource allocation approach based on two technics: multi agent system (MAS) [17] and distributed constraint satisfaction problems (DCSP) [5], [10], [11], [23]. Overall, our main goal is to provide high performance services and minimize the costs of resources operating.

In this paper, we study two problems related to IoT applications deployment in cloud computing. The first problem (Service Providing) occurring in the transformation of the IoT devices (sensors, actuators, ...) to cloud service. Therefore, we work on a smooth and efficient data transmission for the cloud and support customer applications like data sharing, storage and processing. We suggest a number of functionalities for service providing: service creation, service publishing and

service search. The second problem (Service Consumption) lies in the selection and execution of the service of resources allocation in the infrastructure of cloud computing. It occurs in two levels. In the first level, the problem is arisen in the scheduling of tasks (service cloudlet) to assign (submit) the cloudlets to the appropriate VMs taking into consideration the service's functional requirements and minimization of resources exploitation cost. In the second level, the problem lies in the hosting of new VMs in the hosts of the different datacenters according to their loads. The hosting of virtual machines has become a difficult issue in the resource allocation systems because each virtual machine is associated to a physical host according its available resources [6]. In order to solve the problem in both levels; we suggest smart solutions that depend on two techniques: The Multi Agent System (MAS) and CSP. The MAS manages the physical resources, making decision and the communication between datacenters. On the other hand, DCSP is used to simplify the policy of the resources provisioning in Datacenters.

We organize the rest of the article as follows: Section 2 presents research works as related to the subject of this paper. Section 3 offers background and basic concepts. The developed mechanism and system architecture are defined in 4 section. Section 5 presents the main scenarios of interactions in the proposed system. Section 6 provides an illustrative example to clarify our approach. The experimental results are shown in section 7, the last section concludes the paper and presents the future perspectives.

2 Related works

Because of the increasing demand of customers in the field of IoT in cloud infrastructure, many researchers have developed a number of methods to meet customers' demands by taking into account the efficiency of resources and operating expenses. Here, we mention some of the work done in this regard.

Ghanbari *et al.* [9] proposed an analytics study for resource allocation mechanisms for IoT. The Authors of this paper seek to provide a model in the IoT resource allocation which aims at reducing load balancing, minimizing operational cost and power consuming. By reviewed and discussed the advantages and disadvantages of this mechanisms, they compared several parameters in different articles such as: availability, performance, bandwidth, cost, energy, QoS, SLA, throughput, etc. Besides, there are more service quality parameters to be studied such as: self-allocation features, self-adapting, modeling and earning from studies past and current behaviour.

Ma *et al.* [13] suggest a model for task scheduling of the workflow in the IoT infrastructure as a service (IaaS) based on deadline constraints and cost-aware genetic optimization algorithm. To their approach is distributed at different levels according to the characteristics of cloud infrastructure due to the important features of the cloud (on-demand acquisition, heterogeneous dynamics and performance variation of VMs) so that no dependency

exists between tasks at the same level. To demonstrate the feasibility of this approach, authors used the HEFT to generate individuals with the minimum completion time and cost.

Fayazi *et al.* [7] focus on two factors for resource allocation: the reliability and rapid implementation of the work. Therefore, they suggested cloud resource allocation based on auction mechanism. The increase and the decrease in the reliability are determined by the success or failure of the implementation. These solutions are checked by using imperialist competitive algorithm and cost function which is calculated by make span and reliability values. Beside of the diversity of the techniques used in this work, it needs more flexibility for the heterogeneous resources.

The work of Lu *et al.* [12] present a model to allocate the resources based on fairness evaluation framework by using two sub-models (Dynamic Demand Model (DDM) and Dynamic Node Model (DNM)) to describe the resource demand. The authors employ several typical algorithms in resource allocation like utility-based algorithm to prove their effectiveness. As strong point, this model supports the dynamic resources demands, but it does not take into account of the response time.

Mezache *et al.* [15] suggest a genetic algorithm for resource allocation with energy constraint in cloud computing. They focus on two levels of resource allocation: cloudlets to virtual machines and virtual machines to hosts. These levels allow adapting the resource allocation system and keeping the cloud resources updated by taking into account the current submitted cloudlets.

3 Background and basic concepts for IoT and cloud

In this section, we introduce some basic definitions and concepts as a background for our study.

3.1 Visions on integration internet of things and cloud computing

The hybridization (combination) between IoT and Cloud Computing generates synergy for both technologies and bring many benefits. Cloud infrastructure offers a clear advantage to IoT systems since its datacenters are able to calculate the users' needs of resources allocation efficiently. It, thus, shortens the execution time, reduces cost and speeds big data processing [16]. This combination between IoT and Cloud Computing allows to provide a number of technical benefits to users (for example, storage, optimization of resource utilization and energy efficiency) [4], [22]. Figure 1 describes the combination between IoT and Cloud Computing.

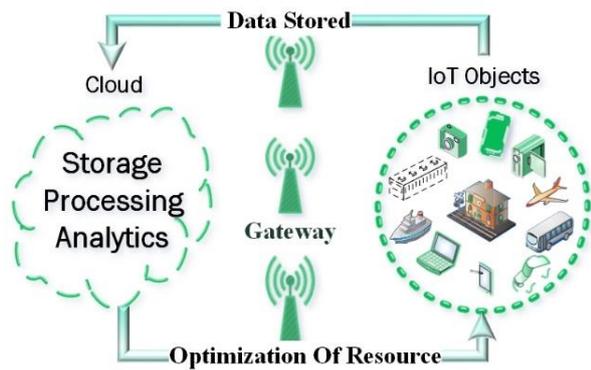


Figure 1: The combination between IoT and Cloud Computing.

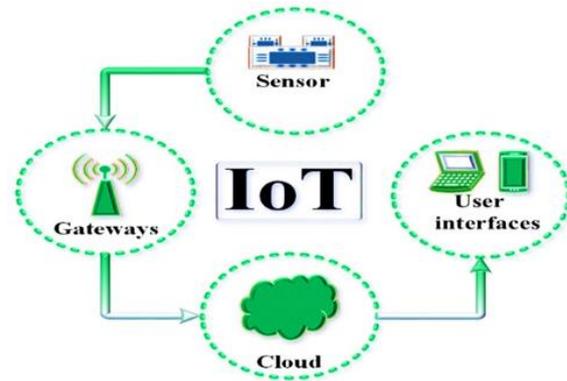


Figure 3: Components of IoT System.

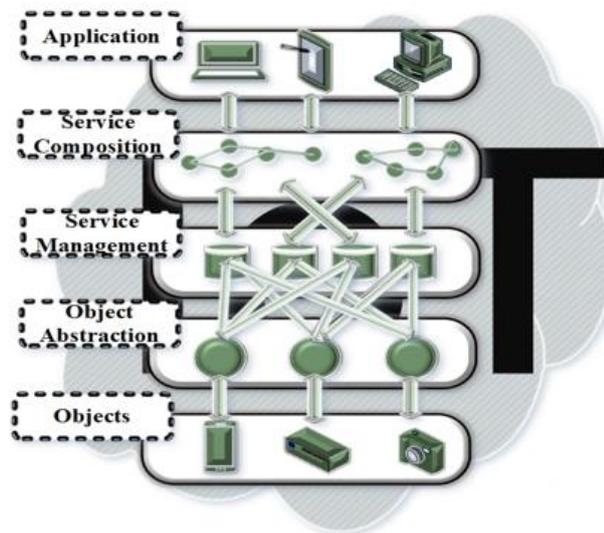


Figure 2: Architecture of IoT service-oriented.

3.2 IoT service-oriented architecture

The aim of service oriented architecture (Figure 2) is to take advantage of the infrastructure of things and the cloud resources for obtaining a better quality of service (reduce the computing costs and improve the overall performance) [24]. The IoT services and devices are usually heterogeneous, and its resources are limited (e.g., memory, processing, bandwidth and energy). To manage such constrained environments, we need to build up a flexible architecture that is capable of managing these resources.

3.3 Components of IoT system

In Figure 3, we present four fundamental components of IoT system (function and mechanism).

- **IoT devices and sensors:**

Sensor is one of IoT devices that has the capability to detect, measure and collect data from the physical environment such as: light, motion, heat, pressure or similar entities [9], [21].

- **IoT gateways:**

The IoT gateway is a bridge between sensor networks and cloud services. The role of gateway is processing the

collected data from sensors, then send it the cloud computing [21].

- **Cloud function:**

Cloud function facilitates the advanced analytics and the monitoring of IoT devices in order to shortening the execution time, reducing costs and reducing energy consumption.

- **User interfaces:**

User interfaces are the visible and tangible part of the IoT system. They enable users to contact and monitor their activities in services that they have already subscribed using IoT system.

3.4 IoT deployed applications

In 0 The deployment of IoT devices encounters number of challenges such as: heterogeneity, storage, bandwidth, implementation of management protocols. To overcome these challenges, researchers turn to the combination between IoT and Cloud Computing. This type of combination contributes in the deployment of high, smarter applications for smarter homes and offices, smarter transportation systems, smarter hospitals, smarter enterprises and factories [4], [25].

3.5 The internet of things and multi agent systems

Thanks its characteristics (intelligence, reactivity, autonomy, mobility and the ability to perform making decision). The MAS allows an efficient management for IoT applications in the physical cloud infrastructure such as: the heterogeneity, distribution and the data management In IoT applications. Briefly, MAS provides a decentralized smart solution to frame the new problems and their solutions in the resource allocation approach for services oriented IoT applications [22].

3.6 Cloud infrastructure and constraint satisfaction problem

The Constraint Satisfaction Problem technique is used to formulate and solve several artificial intelligence related problems such as: Scheduling and Optimization [14]. In the cloud Infrastructure, we use DCSP to simplify the policy of the resources provisioning in Datacenters. DCSP problem is formulated as a distributed Variables and

constraints to multiple agents. In MAS, each agent makes its proposal plan (solution) by using the distributed negotiation and satisfying its constraints. The various variables and constraints are identified, and the scenario of computing is painted accordingly.

4 Developing a new approach for RA in IoT

At this stage, we proposed a new RA in IoT service. Then, we discuss its System Objectives, architecture, layers, DCSP modelling and system scenario.

4.1 System objectives

This paper is interested mainly in the field of cloud of things. Particularly, it shows the importance of resource allocation in data centers. The aim of our approach is to ensure optimal management of resource allocation for service-oriented IoT applications based on decentralized intelligence in distributed computing. To achieve the stated goals i.e. load balancing (minimizing power consumption), efficiently exploiting resources and minimizing the execution time, we suggest:

1. Designing a system to manage the cloud infrastructure based on a multi-agent system for the allocation of resources in the cloud of things.
2. Developing a system to manage these resources by using two techniques: Multi-Agent System (SMA), Distributed Constraint Satisfaction Problems (DCSP).
3. Implementing and simulating the proposed system through a scenario that demonstrates the effectiveness of the proposed approach for the management of resources in the cloud of things.

In this concern, we introduce a number of concepts and rules for IoT service delivery system specifications and the resource allocation process in cloud computing as shown below:

-Concepts:

- 1- Cloud service contains a set of parameters (called nonfunctional parameters) such as:

Latency	Cost	Data-format	Availability
Real number	Real number	Real number	Real number

- 2- To execute cloud service, it requires a set of cloudlet’s resources (called functional parameters). The cloudlet is represented in term of (Ram, Storage, Cpu and Bandwidth).

RAM (MB)	Storage (GB)	CPU (mips)	Bandwidth (Gbit/s)
Real number	Real number	Real number	Real number

- 3- Submission of Cloudlet to VM: is the selection of the Virtual machines (VMs) that have enough available resources to run cloudlet according to its resource requirements.

- 4- The hosting of VMs in hosts: is the process of selecting the host that provides the least price, low load and the best resources available for this VM.

-Rules:

- 1- Every object can be linked to many services.
- 2- Each service has one cloudlet request.
- 3- Every Cloudlet should submitted to one VM.
- 4- Every VM can submit more than one cloudlet.
- 5- Every Host can host more than one VM.
- 6- Every Datacenter has two types of hosts: ON hosts and OFF hosts
- 7- Every host has special price.
- 8- The relationship between the price and the load of the host has a direct impact, where the augmentation in the load causes the increment of the price.

4.2 Smart design for resources allocation in IoT applications

In this section, we are mainly interested in introducing a System Architecture for IoT Resource Allocation, its functional aspect and various layers to provide a better understanding to: how it works, how it stores and how to access to the cloud. Figure 4 describes the proposal smart design.

Layer1 (customer): In this layer, the system focuses on customers and their requests.

The customer requests are presented in term of service name and characteristics.

Layer2 (IoT Service): This layer has a significant the role as mediator between Customer Layer and Broker Layer. It contains two agents:

1. Object agent (OA): is reactive agent that represents an IoT object (physical device). It enables to control exchange and collect data from this device in order to provide a set of services to customers.
2. Mediator agent (MA): is cognitive agent; its role is to manage the customers’ requests and the provided services. The main components of this agent are given below:

• **Service registry:** aims to allow the OA agents to publish the information about their services in term of performance and functionalities.

• **Service selection:** searches for a set of selected services in the registry that meet the customer request.

• **Service transfer** MA creates a list of requested cloudlets from the performance characteristics of the selected services. Then, it sends this list of cloudlets to Broker Agent in the next layer. Broker Agent, in turn, arranges this list of cloudlets and send it to Resources layer

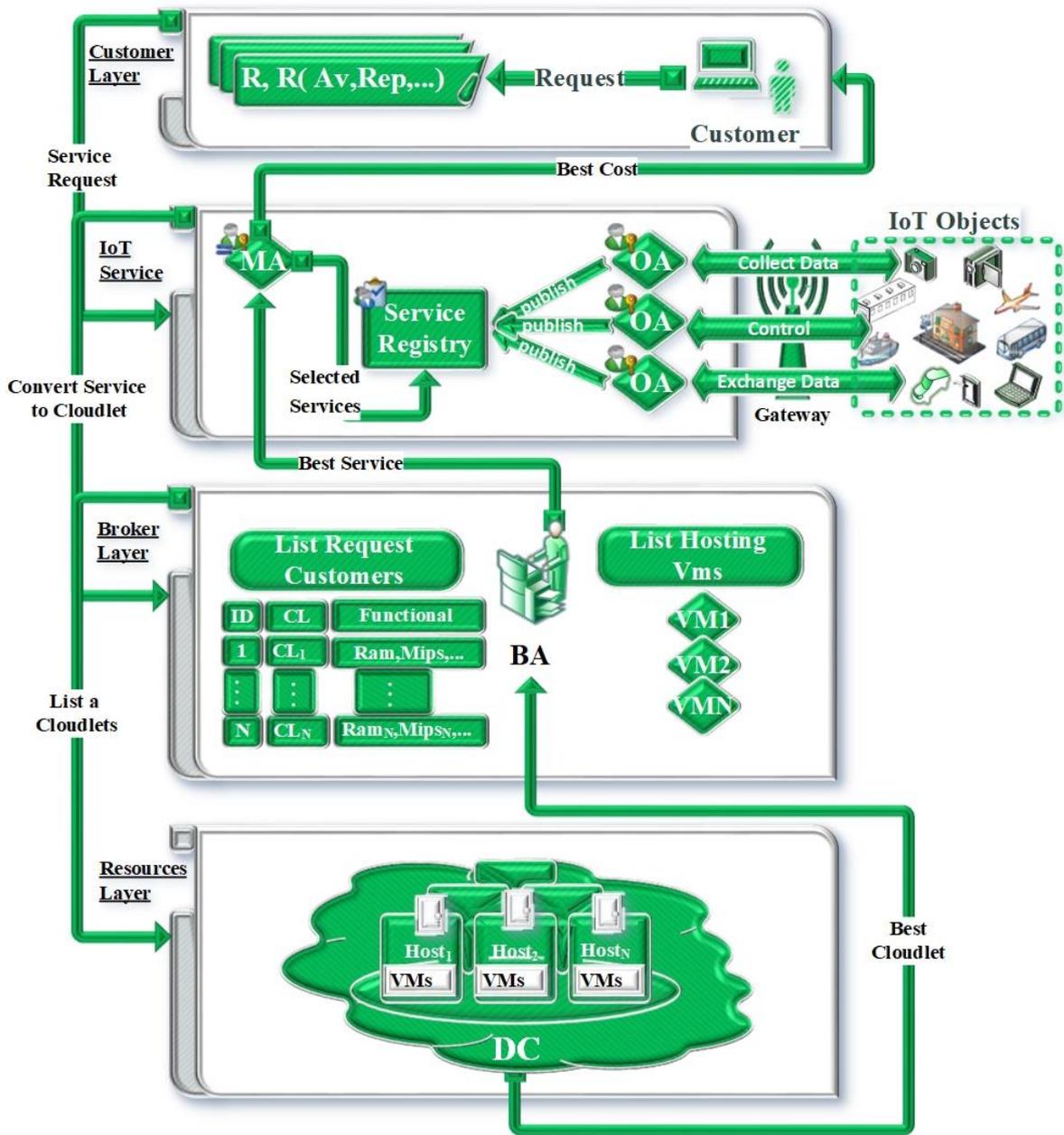


Figure 4: Smart Design for Resources Allocation in IoT Applications.

ID	CL	budget	Vm id	Size	RAM	Bandwidth	Mips/pe	Number of Pe
1	CPU, RAM, Bandwidth, Storage	\$	ID	GB	MB	Gbit/s	Mips	Real Number
		
2	CPU, RAM, Bandwidth, Storage	\$						
...						

Table 1: Example of Broker Agent components.

for the selection of the best cloudlet from this list by taking into account the resource allocation strategy in this layer.

•**Service Bind:** after the selection of the best cloudlet, the MA connects the customer with the provider of the service that is associated by selected cloudlet. It also allows the OA to execute this service through this cloudlet.

Layer3 (Broker Layer): The role of this layer is to manage the resources between IoT service and Resources layer. The broker agent (BA) manages the list of cloudlet requests, free VMs list, performance and delivery of cloud resources. The main role of this agent is to arrange a list of cloudlets, then send to Resources layer.

Layer4 (Resources layer): is the most important layer in the system due to its role in the managing, processing and selecting the best RA for the cloudlet in two levels: local level (between HA agents in the same datacenter) and the global level (between DCA agents of the cloud). This layer contains three types of agents. We introduce these agents and clarify the relationship between them by Figure 5.

Datacenter agent (DCA): communicates with BA and hosts agents in the same datacenter. It also negotiates with other DCA.

Host agent (HA): controls a host in state ON.

Host off agent HOffA: controls a host in state OFF.

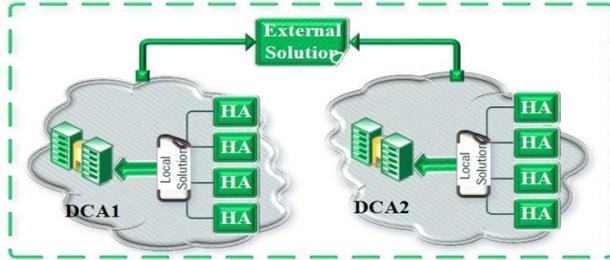


Figure 5: Relationships between DCA and HA agents.

4.3 Relationships between DCA and HA agents

DCSP problem is formulated as a distributed Variables and constraints to multiple agents. In MAS, each agent makes its proposal plan (solution) by using the distributed negotiation and satisfying its constraints. The various variables and constraints are identified, and the scenario of computing is painted accordingly.

4.3.1 Defining of the variables

In this section, we show the most important variables and their definitions Table 2.

4.3.2 Constraints

The aim of this section is to select the best solution for any task in DCSP systems. We thus need to define a set of constraints by using the previous defined variables that correspond to system requirements.

Constraint 1 (Service Usability): verifies a service S that meets customer request R ; it should satisfy the nonfunctional characteristics of the customer request according to the following constraint:

$$S \text{ meet } R : \begin{cases} R(Av) \leq S(Av) \\ \text{and} \\ R(Rep) \leq S(Rep) \end{cases} \quad (1)$$

Constraint 2 (Service Capacity): allows the service to handle new customer request. Before representing customer request in term of cloudlet, it should respect its capacity limitation:

$$S(hr) + 1 \leq S(Cap) \quad (2)$$

Constraint 3 (Cloudlet Submission ability): virtual machine VM_l has already a set of M cloudlets. In order to submit more cloudlet m' , this condition must be satisfied:

$$Cl_{m'}(length) + \sum_{m=1}^M (Cl_m(length) + Cl_m(outputsize)) \leq VM_l(ram) \quad (3)$$

$$Cl_{m'}(file\ size) + \sum_{m=1}^M Cl_m(file\ size) \leq VM_l(storage) \quad (4)$$

$$Cl_{m'}(bw) + \sum_{m=1}^M Cl_m(bw) \leq VM_l(bw) \quad (5)$$

$$Cl_{m'}(mips) + \sum_{m=1}^M Cl_m(mips) \leq VM_l(mips) \quad (6)$$

Where:

$$Cl_m(mips) = CL(length) * CL(nbr_pe) \quad (7)$$

$$VM_l(mips) = \sum_{k=1}^P PE_{kl}(mips) \quad (8)$$

$$\text{and } m' \neg \in [1, M] \quad (9)$$

Constraint 4 (VM Hosting ability): To allow a Host J hosting a new virtual machine l' (free or migrated VM), we must verify these conditions:

$$VM_{l'}(ram) + \sum_{l=1}^V VM_l(ram) \leq Host_j(ram) \quad (10)$$

$$VM_{l'}(storage) + \sum_{l=1}^V VM_l(storage) \leq Host_j(storage) \quad (11)$$

$$VM_{l'}(bw) + \sum_{l=1}^V VM_l(bw) \leq Host_j(bw) \quad (12)$$

$$VM_{l'}(mips) + \sum_{l=1}^V VM_l(mips) \leq Host_j(mips) \quad (13)$$

Where:

$$Host_j(mips) = \sum_{k=1}^P PE_{kj}(mips) \quad (14)$$

$$\text{and } l' \notin [1, V] \quad (15)$$

Constraint 5 (Ranking of Host Agents): The ranking Algorithm is based on **mipsPrice**. In case of finding two Hosts with the same price, then we must use **mipsLoad**:

$$\begin{aligned} \text{Best Host} = & \begin{cases} \min(Host_j(mipsPrice), Host_j(mipsPrice)) \\ \text{and} \\ \min(Host_j(mipsLoad), Host_j(mipsLoad)) \end{cases} \\ & \text{in case : } Host_j(mipsPrice) \equiv Host_j(mipsPrice) \end{aligned} \quad (16)$$

Where:

$$Host_j(mipsLoad) = \frac{Host_j(usedMips)}{Host_j(mips)} \quad (17)$$

$$Host_j(usedMips) = \sum_{l=1}^N VM_{jl}(mips) \quad (18)$$

Constraint 6 (Best VM Hosting Selection): The selection of the best host between different hosts j and j' for hosting VM, it is organized on the basis of Hosting Cost:

Variable	Description	Domain
R	The request of the customer	$\{R_1, \dots, R_v, \dots, R_s\}$
O	The abstract object, each object is connected to physical device (gateway, sensor, actuator...).	$\{O_1, \dots, O_w, \dots, O_t\}$
S	S_{xw} : The offered service x by the object w .	$\{S_{11}, \dots, S_{x w}, \dots, S_{ut}\}$
CL	$CL_{m x w}$: Cloudlet of the service x from the object w	$\{cl_{111}, \dots, cl_{m x w}, \dots, cl_{cut}\}$
R(Av)	The requested Availability.	Rate value (%)
R(Rep)	The requested Reputation.	Naturel number
S(Av)	The Availability of the service.	Rate value (%)
S(Cap)	The Capacity of the service: it is the number of requests can be handled per unit of time.	Naturel number/time
S(Rep)	The Reputation of the service.	Naturel number
S(hr)	The sum of current handled requests by the service.	Real number
Host	Physical host	$\{Host_1, \dots, Host_j, \dots, Host_h\}$
VM	Virtual machine	$\{vm_1, \dots, vm_l, \dots, vm_v\}$
Host(pe)	Processor in the host	$\{Pe_{11}, \dots, pe_{kj}, \dots, pe_{ph}\}$
Host (ram)	Size of host's ram	Naturel Number
Host(bw)	Bandwidth of the host	Real number
Host(Storage)	Size of the host's storage	Naturel Number
Host(mips)	Sum of Capacities of Processors in the host	Real number
Host(used_mips)	Sum of Capacities of the Processors used by virtual machines hosted in the host	Real number
Host(mips_load)	The energy of the host, the Capacity of Processors used in accordance to the total capacity of Processors in the host.	Real number (%)
Host(mips_price)	Unit price of mips in the host	Real number, obtained from proposed model for every host
Pe(mips)	Capacity of the Processor	Real number
VM(size)	VM's hard disc Size	Naturel Number
VM(ram)	Size of the ram	Naturel Number
VM(bw)	Bandwidth of the VM	Real number
VM(mips)	Sum of Capacities of the Processors of the VM	Real number
VM(Cost_j)	The hosting Cost of the VM in the host J	Real number (DA)
CL(length)	Size of the of CL.	Real number
CL(file size)	Total size of files of CL	Real number
CL(output size)	Size of the result of the execution of CL	Real number
CL(nbr_pe)	Max number of Processors of CL	Naturel Number
CL(mips)	Capacity of the Processors of CL	Real number
Cl(Cost_{l_j'})	it is the cost of resource exploitation of the submitted CL in the VM l which is hosted in the host j	Real number (DA)

Table 2: Defining of the Variables.

$$Best Vm Hosting = \min(Vm(Cost_j), Vm(Cost_{j'})) \quad Best CL = \min(Cl(Cost_{l_j}), Cl(Cost_{l'_j})) \quad (20)$$

(19)

Constraint 7(Best Cloudlet Selection): selection of the best Cloudlet (service) for customer request is based on resources exploitation Cost:

Where:

$Cl(Cost_{l_j})$: is the cost of resource exploitation of the submitted CL in the VM l which is hosted in the host j .

$Cl(Cost_{l'_j'})$: is the cost of resource exploitation of the submitted CL in the VM l' which is hosted in the host j' .

5 Scenario of interactions in the proposed system

In this section, we present the main scenarios to provide and select RA for IoT service in the proposed system. Also, we illustrate the interactions between Agents by sequence diagrams where there are two object agents (OA1, OA2) and two datacenters agents (DCA1, DCA2). Every datacenter has two Host Agents (HA1, HA2).

5.1 Global interaction

In this section, we explain the global interactions in the proposed system on three main levels: IoT Service request, Cloudlets Submission and Hosting Virtual Machines. The Search Algorithm and diagram in Figure 6 present the detailed descriptions for these interactions.

5.1.1 Search algorithm

1	Search Algorithm
2	Input
3	Request: customer request contains the nonfunctional characteristics (Av and Rep).
4	SR: Service Registry contains the services list and their Characteristics.
5	Output
6	SL: List of found Services
7	SL = ∅ for all S in SR do

8	if (Request, S) verify C1) then
9	if (S verifies C2) then
10	add S to SL
11	end if
12	end if
13	end for
14	Return SL
15	end.

Algorithm 1: Search Algorithm.

5.2 Cloudlets submission

The process of cloudlets submission in datacenter and their hosts is illustrated in Figure 7. In addition, the Planning Algorithm (Algorithm1) illustrates the process of cloudlets submission inside the Hots.

5.2.1 Planning algorithm

1	Algorithm Planning
2	Input
3	R: List of requested Cloudlets
4	Output
5	BCL: Best Cloudlet
6	BCL = 0
7	For all VM in this host do
8	While ∃ CL ∈ R and (VM, CL) verify C3 do
9	remove CL from R
10	if (BCL=0 or (CL(cost), BCL(cost)) verify C7) then
11	BCL = CL //the new CL is the best cloudlet
12	end if
13	end while
14	end for

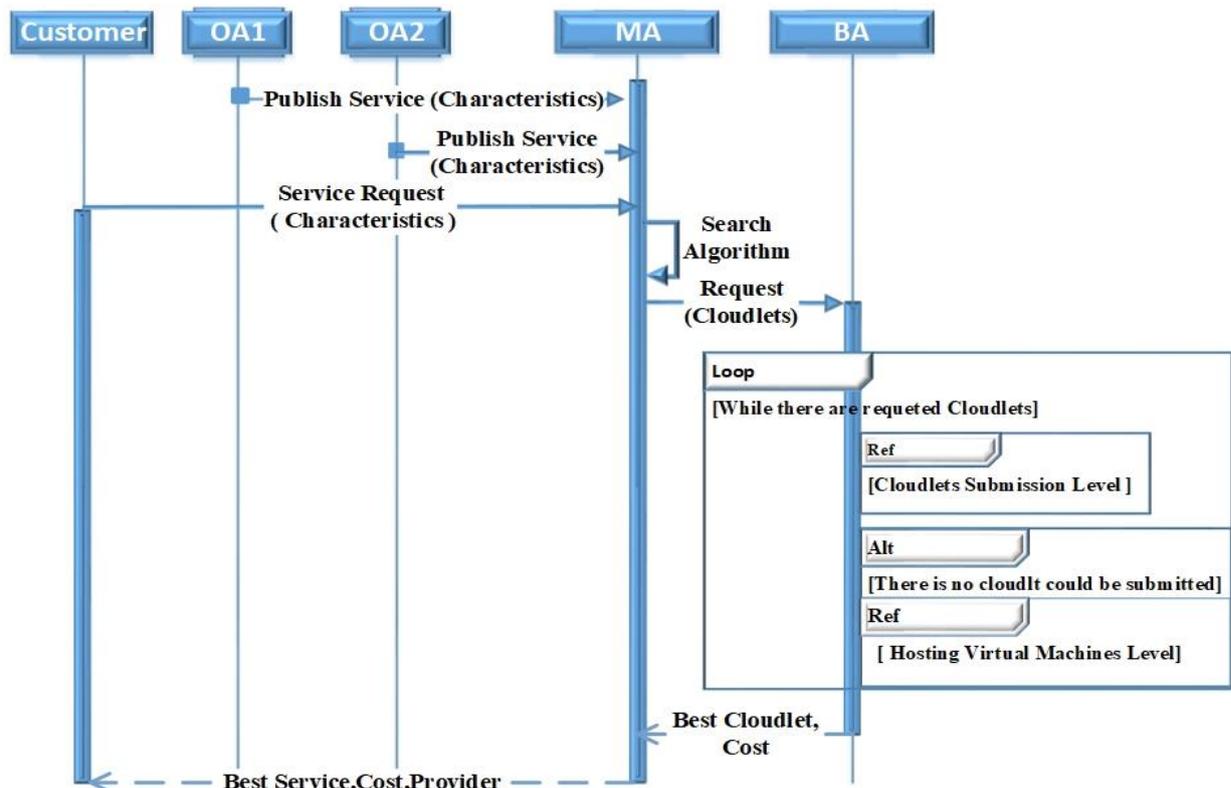


Figure 6: The global interactions in the system.

13	Return BCL
14	<u>end.</u>
15	

Algorithm 2: Planning Algorithm.

5.3 Hosting virtual machines

In a case where there is no VM resource available, we launch the Hosting Virtual Machines to submit the requested Cloudlets. The BA starts the process of hosting free virtual machines as illustrated in Figure.8.

6 Illustrative example

To illustrate our approach, we consider an example and discuss a case study of an IoT Application for smart transport system. We discuss this case study from two dimensions:

1. IoT service deployment

First dimension: we focus on the aspect of the defining, publishing and searching services in addition to the different characteristics of these services and the customers' requests. We show a scenario of using this dimension by the following steps:

Step 1: A company has IoT application for smart taxi. It provides the service of reservation of autonomous cars and tracking (monitoring program to be executed in the cloud) the car during the trip.

Step 2: Each autonomous car (physical IoT) is connected to an agent (object agent) in the cloud (IoT layer). This agent publishes information about his service in MA services registry. The Table 3 illustrates some characteristics of the service in term of functional and nonfunctional.

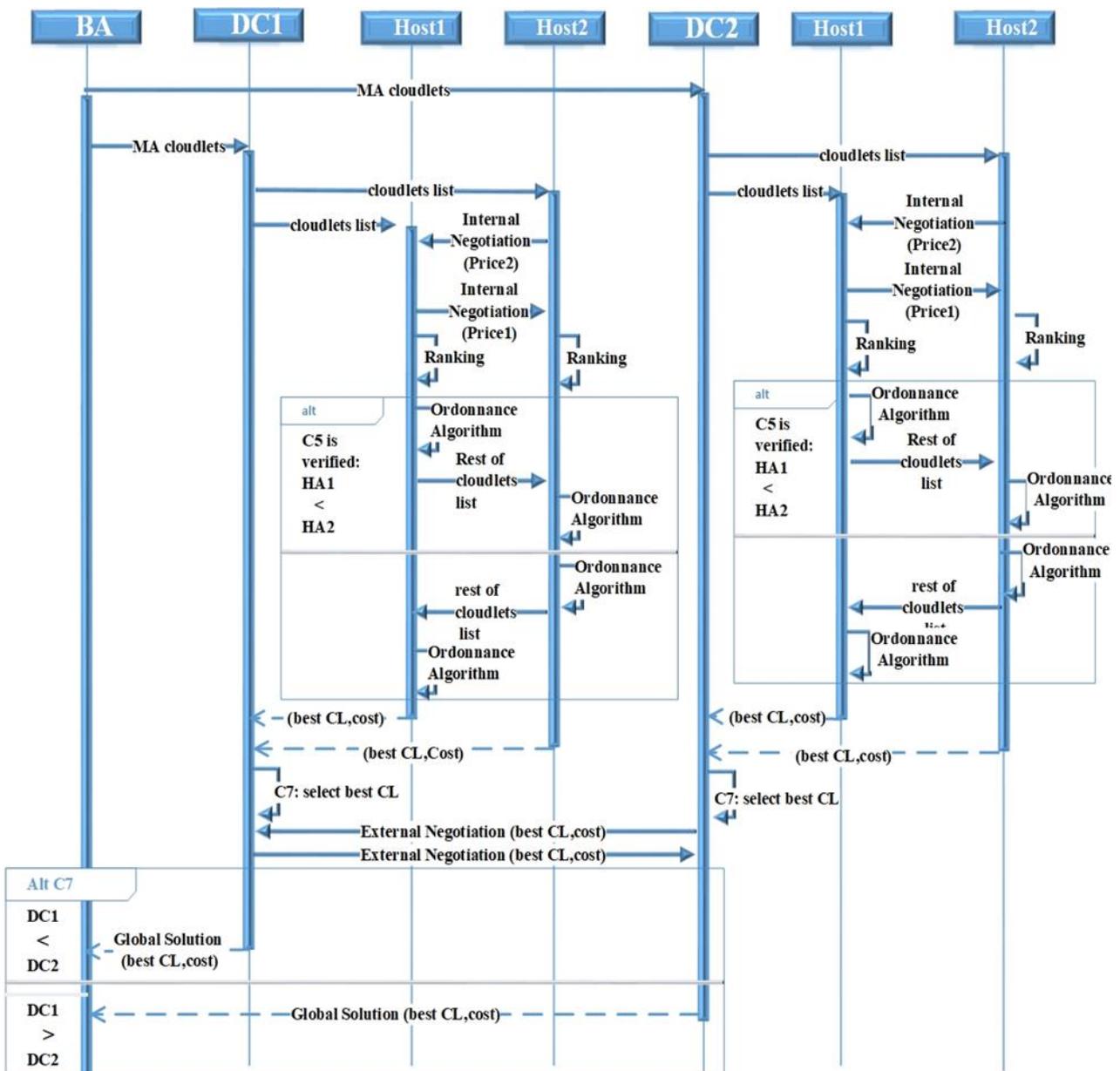


Figure 7: Cloudlets submission between datacenter and their hosts.

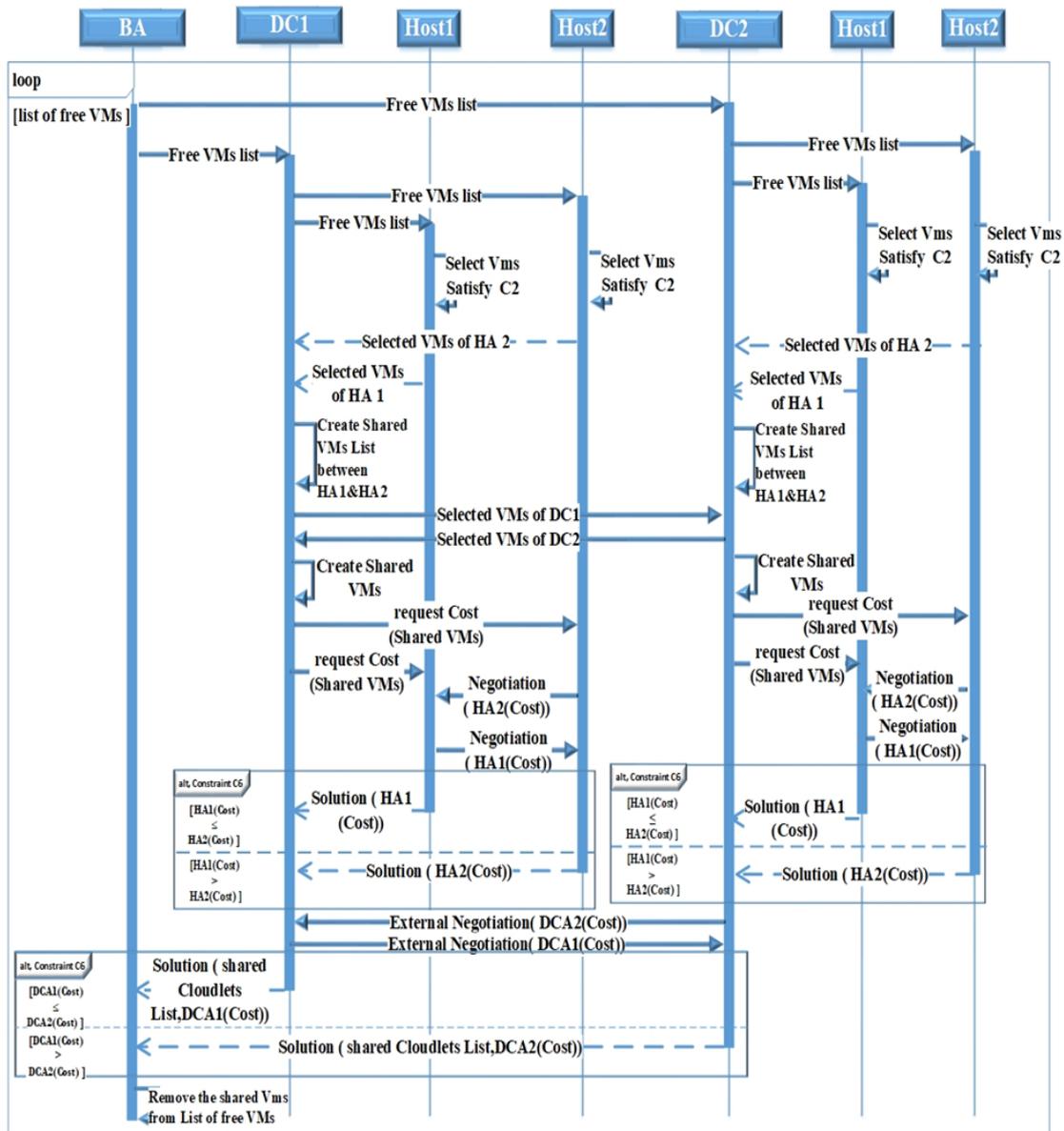


Figure 8: Hosting free virtual machines.

ID	Nonfunctional		Functional		
	Availability	Reputation	RAM(mb)	Storage (mb)	Cpu
OA1	80%	*****	300	500	2
OA2	65%	**	500	1024	3
....

Table 3: Services characteristics.

Step 3: The customer requests a car (service) via introducing the nonfunctional characteristics: availability, reputation and the type of desired trip.

Step 4: First, the MA searches in the registry the available services that meet the customer request. In order to select the best service from the found services, the MA converts these services into cloudlets by using resources requirements (from functional characteristics), and sends them to BA in the next layer.

2. Service selection in cloud computing (Planning procedure)

After obtaining the output (convert services to cloudlets) of the first dimension. We discuss how to the execution of the planning procedure in the second dimension in the cloud system functionality. We propose the cloud infrastructure that has two imaginary datacenters: where datacenter 1 has four hosts and three hosts for datacenter 2. In addition, there are eleven (11) Virtual machines (VMs) hosted in these different hosts. These VMs has already hosted thirty (30) Cloudlets, and BA needs to host seven (07) other requested cloudlets (CL31 ... CL37) in these Vms. In this case, the system looks forward to check the best resource allocation process for these cloudlets according to the cost and energy consumption as shown in the following steps.

Step 1 (requests):

BA distributes the received list of cloudlets to all DCA. As result, every DCA informs his HA agents who are in ON state to start the ranking process.

Cloudlet id	Length	File size	Outputs size	Number of Pe
31	10MB	2 MB	1MB	2
32	13MB	1 MB	1MB	1
33	5MB	3 MB	2MB	1
34	10MB	1 MB	1MB	1
35	5MB	1 MB	1MB	2
36	2MB	3 MB	3MB	1
37	4MB	2MB	1MB	2

Table 4: Cloudlets List Distribute.

Step 2 (Interne Negotiation 1 “Ranking process”):

After, the HA agents (in ON state) share their prices and rank themselves into ascending order by the price. As illustrated in Table 5 the ranking in Datacenter I is: H2, H1, H4 where H2 has the price (1.5 \$) which is the lowest price. And for Datacenter II: H1, H2 where H1 has the lowest price (1.4 \$).

At the end of the ranking, every first HA informs his DCA by the result of the ranking and asks him to send back the list of cloudlets.

	rank	Host id	price
DCA1	1	H2	1.5 \$
	2	H1	4 \$
	3	H4	7.8 \$
DCA2	1	H1	1.4 \$
	2	H2	5.5 \$

Table 5: Hosts ranking

Step 3 (Interne Negotiation 2 “Planning”):

After the ranking process, the first HA in the each DCA gets the list of cloudlets from his DCA, and starts the planning procedure by checking available resources in the hosted VMs of his Host and verifies the constraint C1. If there are Cloudlets and VMs that verify C1, then the first HA selects the best cloudlet that satisfies the constraint C7. The first HA sends the selected cloudlet to the DCA in the term of (cloudlet, VM, host, cost) as reply. At the end of his procedure, it sends the rest of cloudlets (they do not satisfy C1) to the next HA in the ranking list to consider them in his planning procedure. This process is repeated continuously until the last HA in the ranking list or there is no rest cloudlet.

Local solution of DCA1			
Cl id	cost	Host ID	Vm ID
<u>31</u>	39\$	H2	<u>Vm6</u>
32	43\$	H1	Vm3
33	41\$		Vm1
34	78\$		Vm2
35	78\$		Vm11
36	46\$		Vm3
37	40\$		Vm2

Otherwise, in case of there is no Cloudlet that satisfies C1 in any HA, this HA retransmits the whole of the list of cloudlets to the next HA in the ranking list to consider his planning procedure.

Step 4 (Local solution building):

After the planning, every DCA receives the solution from HA agents and selects the best solution, which satisfies C7, and consider it as his local solution. Table 6 illustrates the local solutions in DC1 and DC2 for CL31 and CL37.

Step 5 (External Negotiation) :

The DCA agents share their solutions and negotiate to select the best solution using the best price (to satisfy C7). The DCA that is the owner of the best solution sends his solution to BA to build the global solution as illustrated in Table7.

Cloudlet id	price	DCA id	Host id	Vm Id
33	37\$	DCA2	H1	Vm8

Table 7: Global solution for the Broker Agent

Step 6 (Show solutions and confirmation):

After building the global solution, BA agent sends the cloudlet to MA. As result, MA sends the associated service of the cloudlet as response for customer request, enables (confirms) OA to launch the tracking device of the car and allows the customer to use the car with the lowest cost.

7 Simulation experiments

To evaluate the performance of our approach, we used CloudSim [15] which is a Java based and extensible simulation framework for resource allocation algorithms. In this section, we discuss the experimental configuration and the results obtained by using our approach.

7.1 Experimental configuration

We define the different parameters in our experiments as follows: datacenters, hosts, virtual machines, Processors and cloudlet as shown in Table 8.

7.2 Simulation results

In this section, we present the experimental results and show the efficiency of our proposed approach by making a comparison between three solutions (First Fit algorithm (FF), the proposed Genetic Algorithm (GA) of Mezache et al. [15] and our algorithm (MD)). MD is built on MAS

Local solution of DCA2			
Cl id	cost	Host ID	Vm ID
31	80\$	H1	Vm10
32	72\$		Vm10
<u>33</u>	<u>37\$</u>		<u>Vm8</u>
34	44\$	H2	Vm7
35	54\$		Vm7
36	50\$		Vm8
37	93\$		Vm7

Table 6: Local solution for every Datacenter

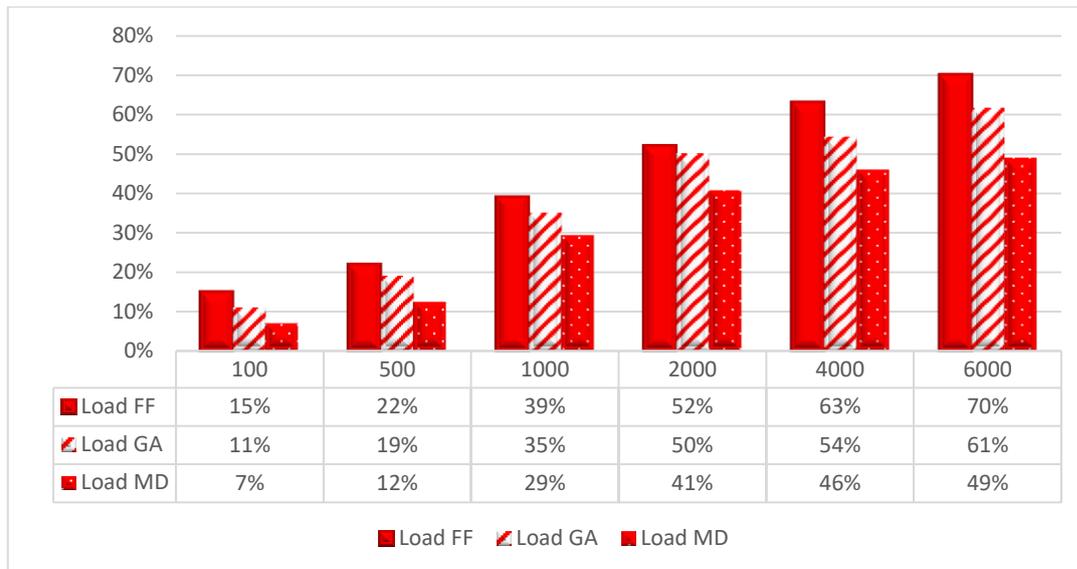


Figure 9: Average Load by number of requested cloudlets.

Parameters	Values
Max Length of cloudlet	50
Total number of cloudlets	500–3000
Total number of VMs	530
VM memory (RAM)	100–1000
Number of PEs requirements	500–1500
Number of datacenters	3
Number of hosts	47

Table 8: Values of experiments Parameters.

with DCSP. In addition, we have defined performance metrics for the evaluation of the three proposed solutions. These solutions have common characteristics (Average System Load (ASL), Cost augmentation Rate (CR) and Available Mips (AM)). In the experiments, the customer request (Service Request) will be submitted to the IoT system for processing this request. In this case, the proposed system converts this request to a list of cloudlets (network bandwidth, Storage, CUP and load consumed) in order to fulfill this request with lowest cost by using our algorithm (MD). The main goal of our algorithm (MD) is to balance between the cost and energy of Datacenters hosts. The obtained results show that this goal is achieved through the common characteristics (metrics) that are shown as follow:

a) Average System Load (ASL) This metric represents the energy consumption. The importance of this metric lies in specifying the datacenters status and reducing energy consumption in their hosts. Usually, the ideal system average load gives us a balance between the different hosts inside their datacenters. Figure 9 presents Average Load by the number of requested cloudlets (FF, GA, MD). The obtained results show the efficiency of our algorithms (MD) in getting a lower values of Average System Load (ASL) compared to FF and GA algorithms. The obtained (ASL) values after using our algorithm (MD) improves over the in terms of Average System Load, so that it does not exceed 50%.

b) Cost augmentation Rate (CR) This metric represents the Cost augmentation rate by cloudlets number. The importance of this metric is manifested in reduce the costs of resources exploitation. The values of (CR) in Figure 10 demonstrate the positive contribution of our algorithm (MD) on reducing cost with almost of all groups. Our algorithm (MD) maintains the augmentation rate (CR) between (105% - 190%) except for the first groups (500 and 1000) where the GA has lower values in (CR). This due to the efficiency of our algorithm (MD) with groups which have an important number of cloudlets (more than 1000).

c) Available Mips (AM) This metric represents the Available Mips by Cloudlets. The importance of this metric lies in measuring the computing performance and increasing Available Mips in datacenters. The more MIPS available for the datacenter, the lower cost of the resources exploitation. In Figure 11, we observe that the values of AM obtained by GA are bigger than the values of other algorithms in groups that have less than 1000 cloudlets. While, our algorithm (MD) has better values of AM when the number of cloudlets increases over 1000.

8 Conclusions and future work

In this paper, we addressed a new approach for Resource Allocation (RA) in Internet of Things. Our approach is based to decentralized intelligence into distributed computing by using two technics: MAS and DCSP. In this hybridization, variables are used to present the resources. While the rules and policies are presented by constraints. They are distributed among multiple agents in the different layers of the system. The experiments show that the use of DCSP beside MAS pave the way for new efficient paradigms in solving problems related not only to Resource Allocation but also to provide smart solutions which are helpful to synchronize the IoT application services with computing devices. The obtained results show that the efficiency of our approach is manifested in: (1) reducing energy consumption in datacenters by about

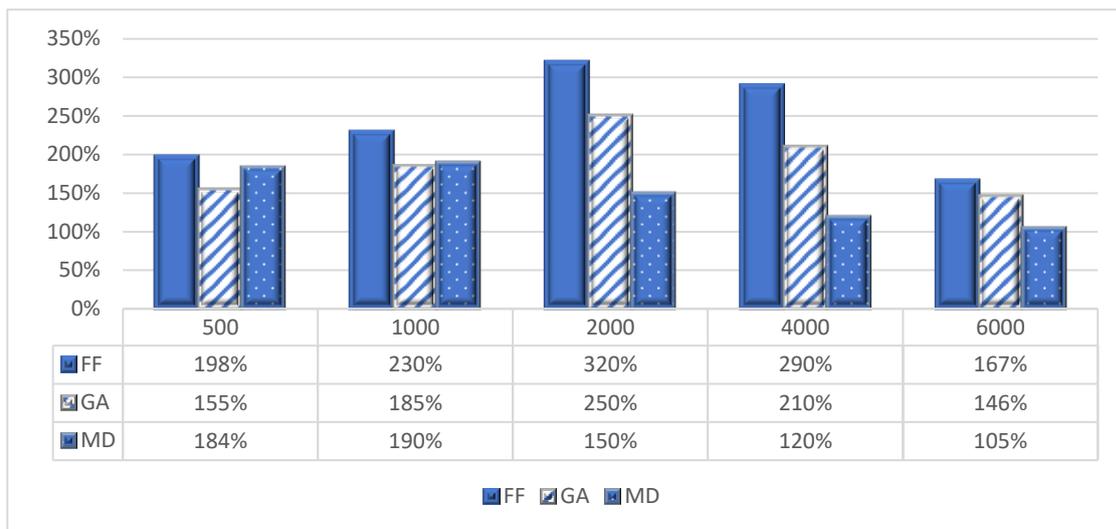


Figure 11: Cost augmentation rate by cloudlets number.

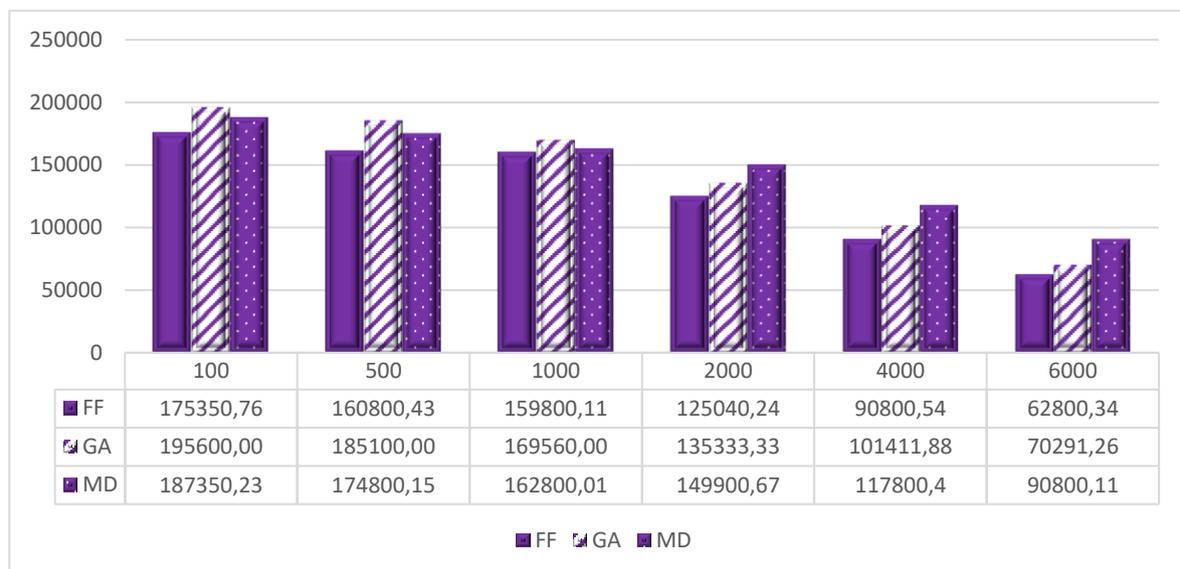


Figure 10: Available Mips by Cloudlet.

50 %, (2) reducing cost augmentation Rate between (105% - 190%) and (3) increasing Available Mips in datacenters.

Despite the provided advantages of our approach, we highlight the need of extending in its architecture to support other specific cases for IoT applications. Big data are generated day-to-day from the system, causing many challenges such as, the heterogeneity, scalability and simultaneous accessibility.

In future research, we are looking for enhancing our approach by using more techniques of resources in IoT application services and extending the procedures by exploiting other approaches as: Search Approximation Algorithms, Artificial Intelligence and Fog environments.

References

- [1] Anithakumari, S., Chandrasekaran, K., (2017). Interoperability based resource management in cloud computing by adaptive dimensional search, in: IEEE International Conference on Cloud Computing in Emerging Markets, CCEM. pp. 77-84. <https://doi.org/10.1109/CCEM.2017.23>
- [2] Artan, M., Minarolli, D., Bernd, F., (2017). Distributed Resource Allocation in Cloud Computing Using Multi-Agent Systems. Telfor 9, 110-115. <https://doi.org/10.5937/telfor1702110M>
- [3] Bajo, J., De la Prieta, F., Corchado, J.M., Rodríguez, S., (2016). A low-level resource allocation in an agent-based Cloud Computing platform. Appl. Soft Comput. 48, 716-728. <https://doi.org/10.1016/j.asoc.2016.05.056>
- [4] Botta, A., De Donato, W., Persico, V., Pescapé, A., (2016). Integration of Cloud computing and Internet of Things: A survey. Futur. Gener. Comput. Syst. 56, 684-700. <https://doi.org/10.1016/j.future.2015.09.021>
- [5] Chen, J., Han, X., Jiang, G., (2014). A Negotiation Model Based on Multi-agent System under Cloud

- Computing, in: In The Ninth International Multi-Conference on Computing in the Global Information Technology. pp. 157-164.
- [6] Ezugwu, A.E., Buhari, S.M., Junaidu, S.B., (2013). Virtual Machine Allocation in Cloud Computing Environment. *Int. J. Cloud Appl. Comput.* 3, 47-60. <https://doi.org/10.4018/ijcac.2013040105>
- [7] Fayazi, M., Reza, M., Enayatollah, S., (2016). Resource Allocation in Cloud Computing Using Imperialist Competitive Algorithm with Reliability Approach. *Int. J. Adv. Comput. Sci. Appl.* 7, 323-331. <https://doi.org/10.14569/IJACSA.2016.070346>
- [8] Gawanmeh, A., April, A., (2016). A Novel Algorithm for Optimizing Multiple Services Resource Allocation. *Int. J. Adv. Comput. Sci. Appl.* 7, 428-434. <https://doi.org/10.14569/IJACSA.2016.070655>
- [9] Ghanbari, Z., Jafari Navimipour, N., Hosseinzadeh, M., Darwesh, A., (2019). Resource allocation mechanisms and approaches on the Internet of Things. *Cluster Comput.* 22, 1253-1282. <https://doi.org/10.1007/s10586-019-02910-8>
- [10] Gutierrez-Garcia, J.O., Sim, K.M., (2011). Agents for cloud resource allocation: An amazon EC2 case study. *Commun. Comput. Inf. Sci.* 261 CCIS, 544-553. https://doi.org/10.1007/978-3-642-27180-9_66
- [11] Jing, L., Weicai, Z., Licheng, J., (2006). A multiagent evolutionary algorithm for constraint satisfaction problems. *IEEE Trans. Syst. Man Cybern. Part B* 36, 54-73. <https://doi.org/10.1109/TSMCB.2005.852980>
- [12] Lu, D., Ma, J., Xi, N., (2015). A universal fairness evaluation framework for resource allocation in cloud computing. *China Commun.* 12, 113-122. <https://doi.org/10.1109/CC.2015.7112034>
- [13] Ma, X., Gao, H., Xu, H., Bian, M., (2019). An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing. *Eurasip J. Wirel. Commun. Netw.* 2019. <https://doi.org/10.1186/s13638-019-1557-3>
- [14] Mataoui, M., Sebbak, F., Beghdad Bey, K., Benhammadi, F., (2015). CSP formulation for scheduling independent jobs in cloud computing. *CLOSER 2015 - 5th Int. Conf. Cloud Comput. Serv. Sci. Proc.* 105-112. <https://doi.org/10.5220/0005438801050112>
- [15] Mezache, C., Kazar, O., Bourekkache, S., (2016). A Genetic Algorithm for Resource Allocation with Energy Constraint in Cloud Computing, in: *International Conference on Image Processing, Production and Computer Science (ICIPCS'2016) London (UK), March 26-27, 2016* Pp.62-69 A. pp. 62-69. <https://doi.org/10.17758/UR.U0316020>
- [16] Mora, H., Signes-Pont, M.T., Gil, D., Johnsson, M., (2018). Collaborative working architecture for IoT-based applications. *Sensors (Switzerland)* 18. <https://doi.org/10.3390/s18061676>
- [17] Nair, A.S., Hossen, T., Campion, M., Selvaraj, D.F., Goveas, N., Kaabouch, N., Ranganathan, P., (2018). Multi-Agent Systems for Resource Allocation and Scheduling in a Smart Grid. *Technol. Econ. Smart Grids Sustain. Energy* 3, 1-15. <https://doi.org/10.1007/s40866-018-0052-y>
- [18] Rivera, W., (2017). Sustainable cloud and energy services: Principles and practice. *Sustain. Cloud Energy Serv. Princ. Pract.* 1-268. <https://doi.org/10.1007/978-3-319-62238-5>
- [19] Roogi, R.H., (2015). Big Data Solution by Divide and Conquer technique in Parallel Distribution System using Cloud Computing. *Orient. J. Comput. Sci. Technol.* 8, 9-12.
- [20] Shrimali, B., Bhadka, H., Patel, H., (2018). A fuzzy-based approach to evaluate multi-objective optimization for resource allocation in cloud. *Int. J. Adv. Technol. Eng. Explor.* 5, 140-150. <https://doi.org/10.19101/IJATEE.2018.542020>
- [21] Singh, A., Viniotis, Y., (2017). Resource allocation for IoT applications in cloud environments. *2017 Int. Conf. Comput. Netw. Commun. ICNC 2017* 719-723. <https://doi.org/10.1109/ICCNC.2017.7876218>
- [22] Singh, M.P., Chopra, A.K., (2017). The Internet of Things and Multiagent Systems: Decentralized Intelligence in Distributed Computing. *Proc. - Int. Conf. Distrib. Comput. Syst.* 1738-1747. <https://doi.org/10.1109/ICDCS.2017.304>
- [23] Son, S., Sim, K.M., (2012). A price-and-time-slot-negotiation mechanism for cloud service reservations. *IEEE Trans. Syst. Man, Cybern. Part B Cybern.* 42, 713-728. <https://doi.org/10.1109/TSMCB.2011.2174355>
- [24] Suci, G., Suci, V., Martian, A., Craciunescu, R., Vulpe, A., Marcu, I., Halunga, S., Fratu, O., (2015). Big Data, Internet of Things and Cloud Convergence - An Architecture for Secure E-Health Applications. *J. Med. Syst.* 39. <https://doi.org/10.1007/s10916-015-0327-y>
- [25] Zahoor, S., Mir, R.N., (2018). Resource management in pervasive Internet of Things: A survey. *J. King Saud Univ. Inf. Sci.* <https://doi.org/10.1016/j.jksuci.2018.08.014>

How to Define *Co-occurrence* in a Multidisciplinary Context?

Mathieu Roche

CIRAD, TETIS, F-34398 Montpellier, France

TETIS, Univ. Montpellier, AgroParisTech, CIRAD, CNRS, INRAE, Montpellier, France

E-mail: mathieu.roche@cirad.fr, <http://textmining.biz/Staff/Roche>

Position paper

Keywords: co-occurrence, collocation, phrase, n -gram, skyp- n -gram, association rule, sequential pattern

Received: October 28, 2019

This position paper presents a comparative study of co-occurrences. Some similarities and differences in the definition exist depending on the research domain (e.g. linguistics, natural language processing, computer science). This paper discusses these points and deals with the methodological aspects in order to identify co-occurrences in a multidisciplinary paradigm.

Povzetek: Predstavljena je analiza sočasnosti.

1 Introduction

Determining *co-occurrences* in corpora is challenging for different applications such as classification, translation, terminology building, etc. More generally, *co-occurrences* can be identified with all types of data, e.g. databases [8], texts [30], images [38], music [15], video [19], etc.

The *co-occurrence* concept has different definitions depending on the research domain (i.e. linguistics, natural language processing (NLP), computer science, biology, etc.). This position paper reviews the main definitions in the literature and discusses similarities and differences according to the domains. This type of study can be crucial in the context of data science, which is geared towards developing a multidisciplinary paradigm for data processing and analysis, especially textual data.

Here the *co-occurrence* concept related to textual data is discussed. Note that before their validation by an expert, co-occurrences of words are often considered as *candidate terms*.

First, Section 2 of this paper details the different definitions of *co-occurrence* according to the studied domains. Section 3 discusses and compares these different aspects based on their intrinsic definition but also on the associated methodologies in order to identify them. Finally, Section 4 lists some perspectives.

2 *Co-occurrence* in a multidisciplinary context

2.1 Linguistic viewpoint

In linguistics, one notion that is broadly used to define the term is called *lexical unit* [23] and *polylexical expression* [16]. The latter represents a set of words having an au-

tonomous existence, which is also called *multi-word expression* [33].

In addition, several linguistics studies use the *collocation* notion. [10] gives two properties defining a collocation. First, collocation is defined as a group of words having an overall meaning that is deducible from the units (words). For example, *climate change* is considered as a collocation because the overall meaning of this group of words can be deduced from both words *climate* and *change*. On the other hand, the expression *to rain cats and dogs* is not a collocation because its meaning cannot be deduced from each of the words; this is called a *fixed expression* or an *idiom*.

A second property is added by [10] to define a collocation. The meaning of the words that make up the collocation must be limited. For example, *buy a dog* is not a collocation because the meaning of *buy* is not limited.

2.2 NLP viewpoint

In the natural language processing (NLP) domain, the *co-occurrence* notion refers to the general phenomenon where words are present together in the same context. More precisely, several principles are used that take contextual criteria into account.

First, the terms or phrases [6, 11] can respect syntactic patterns (e.g. adjective noun, noun noun, noun preposition noun, etc.). Some examples of extracted phrases (i.e. *syntactic co-occurrences*) are given in Table 1.

In addition, the methods without linguistic filtering are also conventionally used in the NLP domain by extracting n -grams of words (i.e. *lexical co-occurrences*) [25, 35]. n -grams are contiguous sequences of n words extracted from a given sequence of text (e.g. the bi-grams¹ $x y$ and $y z$ are associated with the text $x y z$). n -grams that allow gaps

¹ n -grams with $n = 2$.

are called skip- n -grams (e.g. the skip-bi-grams $x y$, $x z$, $y z$ are related to the text $x y z$). Skip-gram model is an efficient method for learning high-quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships [27]. Some examples of n -grams and skip- n -grams are given in Table 1.

After summarizing the term notion in the NLP domain, the following section discusses these aspects in the computer science context, particularly in data mining. Note that the NLP domain may be considered as being located at the linguistics and computer science interface.

2.3 Computer science viewpoint

In the data mining domain, co-occurring items are called *association rules* [1, 39] and they could be candidates for construction or enrichment of terminologies [12].

In the data mining context, the list of items corresponds to the set of available articles. With textual data, items may represent the words present in sentences, paragraphs, or documents [2, 29]. A transaction is a set of items. A set of transactions is a learning set used to determine association rules.

Some extensions of association rules are called *sequential patterns*. They take into account a certain order of extracted elements [18, 34] with an enriched representation related to textual data as follows:

- *objects* represent texts or pieces of texts,
- *items* are the words of a text,
- *itemsets* represent sets of words present together within a sentence, paragraph or document,
- *dates* highlight the order of sentences within a text.

There are several algorithms for discovering association rules and sequential patterns. One of the most popular is Apriori, which is used to extract frequent itemsets from large databases. The Apriori algorithm [1] finds frequent itemsets where k -itemsets are used to generate $k + 1$ -itemsets.

Association rules and sequential patterns of words are often used in text mining for different applications, e.g. terminology enrichment [12], association of concept instances [5, 29], classification [18, 34], etc.

3 Discussion: comparative study of definitions and approaches

This section proposes a comparison of: (i) *co-occurrence* definitions (see Section 3.1), (ii) automatic methods in order to identify them (see Section 3.2). This section highlights some similarities and differences between domains.

3.1 Co-occurrence extraction

The general definition of *co-occurrence* is finally close to *association rules* in data mining domain. Note that the integration of windows² in the association rule or sequential pattern extraction process enables us to have similarity with skip- n -gram extraction.

The integration of syntactic criteria makes it possible to extract more relevant candidate terms (see Table 1). Such information is typically taken into account in NLP to extract terms from general or specialized domains [20, 24, 28, 32].

Table 1 highlights relevant terms extracted using linguistic patterns (e.g. *climate change*, *water cycle*, *significant change*). The use of linguistic patterns tends to improve precision values. Generally other methods such as skip-bi-grams return lower precision, i.e. many extracted candidates are irrelevant (e.g. *climate the*). But this kind of method enables extraction of some relevant terms not found with linguistic patterns (e.g. *cycle expected*); then the recall can be improved.

Table 2 presents research domains related to different types of candidates, i.e. collocations, polylexical expressions, phrases, n -grams, association rules, sequential patterns.

Table 3 summarizes the main criteria described in the literature. Note that the extraction is more flexible and automatic when there are fewer criteria. In this table, two types of information are associated with the different criteria. The first one (marked with ✓) designates the characteristics given by the *co-occurrence* definitions. The second type of information (marked with ★) represents characteristics that are implemented in many extensions of the state-of-the-art.

Table 3 shows that the semantic criterion is seldom associated with *co-occurrence* definitions. This criterion is however taken into account in linguistics. For example, semantic aspects are taken into account in several studies [17, 22, 26]. In this context [26] introduced *lexical functions* rely on semantic criteria to define the relationships between collocation units. For instance, a given relation can be expressed in various ways between the arguments and their values, like *Centr* (*the center, culmination of*) that returns different meanings³:

- *Centr*(*crisis*) = *the peak*
- *Centr*(*desert*) = *the heart*
- *Centr*(*forest*) = *the thick*
- *Centr*(*glory*) = *summit*
- *Centr*(*life*) = *prime*

In the data mining domain, semantic information is used in two main directions. The first one involves filtering the

²Association Rule with Time-Windows (ARTW) [39].

³http://people.brandeis.edu/~smalamud/ling130/lex_functions.pdf

Sentence (input)	
<i>With climate change the water cycle is expected to undergo significant change.</i>	
Candidates (output)	
Phrases (noun noun, adjective noun)	<i>climate change water cycle, significant change</i>
bi-grams of words	<i>With climate, climate change, change the, the water, water cycle, cycle is, is expected, expected to, to undergo, undergo significant, significant change</i>
2-skip-bi-grams	<i>With climate, With change, With the, climate change, climate the, climate water, change the, change water, change cycle, the water, the cycle, the is, water cycle, water is, water expected, cycle is, cycle expected, cycle to, is expected, is to, is undergo, expected to, expected undergo, expected significant, to undergo, to significant, to change, undergo significant, undergo change, significant change</i>

Table 1: Examples of candidates extracted with different NLP techniques.

Definitions	Domains
Collocations	L
Polylexical expressions	L + NLP
Phrases	NLP
n-grams	NLP + CS
Association rules	CS
Sequential patterns	CS

Table 2: Summary of the main domains associated with expressions (L: linguistics, NLP: natural language processing, CS: computer science).

results if they respect certain semantic information (e.g. phrases or patterns where a word is an instance of a semantic resource). Other methods involve semantic resources in the knowledge discovery process, i.e. the extraction is driven by semantic information [5].

In recent studies in the NLP domain, the semantic aspects are based on word embedding, which provides a dense representation of words and their relative meanings [14, 40].

Finally, note that several types of *co-occurrence* are often used in different domains. For example, polylexical expressions are commonly used in NLP and also in linguistics. In addition, *n*-grams is currently used in NLP and computer science domains. For example, *n*-grams of

words are often used to build terminologies (NLP domain) but also as features for machine learning algorithms (computer science domain) [35].

Table 4 summarizes the main types of criteria (i.e. statistical, morpho-syntactic, and semantic) used for extracting *co-occurrences* according to the research domains considered in this paper.

After presenting the characteristics associated with the *co-occurrence* notion in a multidisciplinary context, the following section compares the methodological viewpoints to identify these elements according to the domains.

3.2 Ranking of *co-occurrences*

Co-occurrence identification by automatic systems is generally based on the use of quality measures and/or algorithms. This section provides two illustrative examples that show similarities between approaches according to the domains.

3.2.1 Mutual Information and Lift measure

Firstly the use of specific statistical measures from different domains is highlighted. This subsection focuses on the study of Mutual Information (MI). This measure is often used in the NLP domain to measure the association between words [9]. MI (see formula (3.1)) compares the probability of observing *x* and *y* together (joint probabil-

	Ordered sequences	Sequences with gaps	Morpho-syntactic information	Semantic information
Collocations	✓		✓	★
Polylexical expressions	✓		✓	
Phrases	✓		✓	
<i>n</i> -grams	✓	★		
Association rules		✓		
Sequential patterns	✓	✓		

Table 3: Summary of the main criteria associated with *co-occurrence* identification. ✓ represents the respect of the criterion by definition. ★ is present when extensions are currently used in the state-of-the-art.

	Statistic information	Morpho-syntactic information	Semantic information
Linguistics		✓	★
NLP	✓	✓	★
Data mining	✓	★	★

Table 4: Summary of the main criteria associated with research domains. ✓ represents the respect of the criterion for extracting *co-occurrences* from textual data. ★ is present when extensions are currently used in the state-of-the-art.

ity) with the probability of observing x and y independently (chance) [9].

$$I(x) = \log_2 \frac{P(x, y)}{P(x)P(y)} \tag{3.1}$$

In general, word probabilities $P(x)$ and $P(y)$ correspond to the number of observations of x and y in a corpus normalized by the size of the corpus. Some extensions of *MI* are also proposed. The algorithm PMI-IR (Pointwise Mutual Information and Information Retrieval) described in [36] queries the Web via the AltaVista search engine to determine appropriate synonyms for a given query. For a given word, denoted x , PMI-IR chooses a synonym among a given list. These selected terms, denoted $y_i, i \in [1, n]$, correspond to TOEFL questions. The aim is to compute the y_i synonym that gives the best score. To obtain scores, PMI-IR uses several measures based on the proportion of documents where both terms are present. Turney’s formula is given below (3.2): It is one of the basic measures used in [36]. It is inspired from *MI* described in [9]. With this formula (3.2), the proportion of documents containing both x and y_i (within a 10 word window) is calculated and compared with the number of documents containing the word y_i . The higher this proportion, the more x and y_i are seen as synonyms.

$$score(y_i) = \frac{nb(x \text{ NEAR } y_i)}{nb(y_i)} \tag{3.2}$$

- $nb(x)$ computes the number of documents containing the word x (i.e. nb corresponds to number of web-pages returned by search engines),
- *NEAR* (used in the ‘advanced research’ field of AltaVista) is an operator that identifies if two words are

present in a 10 word wide window.

This kind of web mining approach is also used in many NLP applications, e.g. (i) computing the relationship between *host* and *clinical sign* for an epidemiology surveillance system [3], (ii) computing the dependency of words of acronym definitions for word-sense disambiguation tasks [31].

The probabilities are generally symmetric (i.e. $P(x, y) = P(y, x)$), while the original *MI* measure is also symmetric. But the association ratio applied in the NLP domain is not symmetric, i.e. the occurrence number of pairs of words " $x y$ " and " $y x$ " generally differ. Moreover the meaning and relevance of phrases should differ according to the word order in a text, e.g. *first lady* and *lady first*.

Finally, *MI* is very close to the *lift* measure [7, 37, 4] in data mining. This measure identifies relevant association rules (see formula (3.3)). The *lift* measure evaluates the relevance of co-occurrences only (not implication) and how x and y are independent [4].

$$lift(x \rightarrow y) = \frac{conf(x \rightarrow y)}{sup(y)} \tag{3.3}$$

This measure is based on both *confidence* and *support* criteria, which in turn are based on association rule ($x \rightarrow y$) identification. Support is an indication of how frequently the itemset appears in the dataset. Confidence is a standard measure that estimates the probability of observing y given x (see formula 3.4)

$$conf(x \rightarrow y) = \frac{sup(x \cup y)}{sup(x)} \tag{3.4}$$

Note that other quality measures of the data mining domain, such as *Least contradiction* or *Conviction* [21], could

be tailored to deal with textual data.

3.2.2 C-value and closed itemset

Another example is the methodological similarities associated with different approaches. For example, the C-value approach [13] used in the NLP domain [24, 20] favors terms that do not appear to a significant extent in longer terms. For example, in a specialized corpus related to ophthalmology, the work of [13] shows that a more general term such as *soft contact* is irrelevant, whereas a longer and therefore more specific term such as *soft contact lens* is relevant. This kind of measure is particularly relevant in the biology domain [24, 20].

In addition, in the computer science domain (i.e. data mining), the notion of *closed itemset* is finally very close to the C-value approach. In this context, a frequent itemset is considered as closed if none of its supersets⁴ has the same support (i.e. frequency).

This section and both illustrative examples confirm the importance of having a real multidisciplinary viewpoint on the methodological aspects in order to build scientific bridges and thus contribute to the development of the emerging data science domain.

4 Conclusion and Future Work

This position paper proposes a discussion on similarities as well as differences in the definition of *co-occurrence* according to research domains (i.e. linguistics, NLP, computer science). The aim of this position paper is to show the bridges that exist between different domains.

In addition, this paper highlights some similarities in the methodologies used in order to identify *co-occurrences* in different domains. We could extend the discussion to other domains. For example, methodological transfers are currently applied between bioinformatics and NLP. For example, the use of edition measures (e.g. Levenshtein distance) for sequence alignment tasks (bioinformatics) v.s. string comparison (NLP).

Acknowledgments

This work is funded by the SONGES project (Occitanie and FEDER) – Heterogeneous Data Science (<http://textmining.biz/Projects/Songes>).

References

- [1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94*,

pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
<http://dl.acm.org/citation.cfm?id=645920.672836>.

- [2] Amihood Amir, Yonatan Aumann, Ronen Feldman, and Moshe Fresko. Maximal association rules: A tool for mining associations in text. *Journal of Intelligent Information Systems*, 25(3):333–345, Nov 2005.
<https://doi.org/10.1007/s10844-005-0196-9>.
- [3] Elena Arsevska, Mathieu Roche, Pascal Hendriks, David Chavernac, Sylvain Falala, Renaud Lancelot, and Barbara Dufour. Identification of associations between clinical signs and hosts to monitor the web for detection of animal disease outbreaks. *International Journal of Agricultural and Environmental Information Systems*, 7(3):1–20, 2016.
<https://doi.org/10.4018/IJAEIS.2016070101>.
- [4] Paulo J. Azevedo and Alípio M. Jorge. Comparing rule measures for predictive association rules. In *Proceedings of the 18th European Conference on Machine Learning, ECML '07*, pages 510–517, Berlin, Heidelberg, 2007. Springer-Verlag.
http://dx.doi.org/10.1007/978-3-540-74958-5_47.
- [5] Soumia Lilia Berrahou, Patrice Buche, Juliette Dibie, and Mathieu Roche. Xart: Discovery of correlated arguments of n-ary relations in text. *Expert Systems with Applications*, 73(Supplement C):115 – 124, 2017.
<https://doi.org/10.1016/j.eswa.2016.12.028>.
- [6] Didier Bourigault. Surface grammatical analysis for the extraction of terminological noun phrases. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 3, COLING '92*, pages 977–981, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.
<http://dx.doi.org/10.3115/992383.992415>.
- [7] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, SIGMOD '97*, pages 265–276, New York, NY, USA, 1997. ACM.
<http://doi.acm.org/10.1145/253260.253327>.
- [8] Hui Cao, George Hripsak, and Marianthi Markatou. A statistical methodology for analyzing co-occurrence data from a large sample. *Journal of*

⁴A superset is defined with respect to another itemset, for example {M1, M2, M3} is a superset of {M1, M2}. B is superset of A if $\text{card}(A) < \text{card}(B)$ and $A \subset B$.

- Biomedical Informatics*, 40(3):343–352, 2007.
<https://doi.org/10.1016/j.jbi.2006.11.003>.
- [9] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March 1990.
<http://dl.acm.org/citation.cfm?id=89086.89095>.
- [10] André Clas. Collocations et langues de spécialité. *Meta*, 39(4):576–580, 1994.
<https://doi.org/10.7202/002327ar>.
- [11] Béatrice Daille, Éric Gaussier, and Jean-Marc Langé. Towards automatic extraction of monolingual and bilingual terminology. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1, COLING '94*, pages 515–521, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.
<https://doi.org/10.3115/991886.991975>.
- [12] Lisa Di-Jorio, Sandra Bringay, Céline Fiot, Anne Laurent, and Maguelonne Teisseire. Sequential patterns for maintaining ontologies over time. In *On the Move to Meaningful Internet Systems: OTM 2008, OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 9-14, 2008, Proceedings, Part II*, pages 1385–1403, 2008.
https://doi.org/10.1007/978-3-540-88873-4_32.
- [13] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms: the C-value/NC-value method. *International Journal on Digital Libraries*, 3(2):115–130, Aug 2000.
<https://doi.org/10.1007/s007999900023>.
- [14] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth J.F. Jones. Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15*, pages 795–798, New York, NY, USA, 2015. ACM.
<http://doi.acm.org/10.1145/2766462.2767780>.
- [15] Arijit Ghosal, Rudrasis Chakraborty, Bibhas Chandra Dhara, and Sanjoy Kumar Saha. *Song Classification: Classical and Non-classical Discrimination Using MFCC Co-occurrence Based Features*, pages 179–185. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
https://doi.org/10.1007/978-3-642-27183-0_19.
- [16] Gaston Gross. *Les expressions figées en français*. Ophrys, 1996.
- [17] Ulrich Heid. Towards a corpus-based dictionary of german noun-verb collocations. In *Proceedings of the Euralex International Congress*, pages 301–312, 1998.
- [18] Simon Jaillet, Anne Laurent, and Maguelonne Teisseire. Sequential patterns for text categorization. *Intelligent Data Analysis*, 10(3):199–214, May 2006.
<https://doi.org/10.3233/IDA-2006-10302>.
- [19] Hyun-Ho Jeon, Andrea Basso, and Peter F. Driessen. *Camera Motion Detection in Video Sequences Using Motion Cooccurrences*, pages 524–534. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
https://doi.org/10.1007/11581772_46.
- [20] Min Jiang, Joshua C. Denny, Buzhou Tang, Hongxin Cao, and Hua Xu. Extracting semantic lexicons from discharge summaries using machine learning and the c-value method. In *AMIA 2012, American Medical Informatics Association Annual Symposium, Chicago, Illinois, USA, November 3-7, 2012*, 2012.
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3540581/>.
- [21] Stephane Lallich, Olivier Teytaud, and Elie Prudhomme. *Association Rule Interestingness: Measure and Statistical Validation*, pages 251–275. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
https://doi.org/10.1007/978-3-540-44918-8_11.
- [22] Marleen Laurens. La description des collocations et leur traitement dans les dictionnaires. *Romanesque*, 4:44–51, 1999.
<http://www.vlrom.be/pdf/994colloc.pdf>.
- [23] Carmen Lederer. La notion d'unité lexicale et l'enseignement du lexique. *The French Review*, 43(1):96–98, 1969.
<https://www.jstor.org/stable/386736>.
- [24] Juan Antonio Lossio-Ventura, Clement Jonquet, Mathieu Roche, and Maguelonne Teisseire. Biomedical term extraction: Overview and a new methodology. *Information Retrieval Journal*, 19(1-2):59–99, April 2016.
<http://dx.doi.org/10.1007/s10791-015-9262-2>.
- [25] Sean Massung and Chengxiang Zhai. Non-native text analysis: A survey. *Natural Language Engineering*, 22(2):163–186, 2016.
<https://doi.org/10.1017/S1351324915000303>.

- [26] Igor A. Mel'čuk, Nadia Arbatchewsky-Jumarie, Léo Elnitsky, and Adèle Lessard. *Dictionnaire explicatif et combinatoire du français contemporain*. Presses de l'Université de Montréal, Montréal, Canada, 1984,1988,1992,1999. Volume 1, 2, 3, 4.
- [27] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc.
<http://dl.acm.org/citation.cfm?id=2999792.2999959>.
- [28] Goran Nenadić, Irena Spasić, and Sophia Ananiadou. Terminology-driven mining of biomedical literature. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, SAC '03, pages 83–87, New York, NY, USA, 2003. ACM.
<http://doi.acm.org/10.1145/952532.952553>.
- [29] Julien Rabatel, Yuan Lin, Yoann Pitarch, Hassan Saneifar, Claire Serp, Mathieu Roche, and Anne Laurent. Visualisation des motifs séquentiels extraits à partir d'un corpus en ancien français. In *Extraction et gestion des connaissances (EGC'2008)*, pages 237–238, 2008.
<https://editions-rnti.fr/?inprocid=1000605>.
- [30] Mathieu Roche, Jérôme Azé, Oriane Matte-Tailliez, and Yves Kodratoff. Mining texts by association rules discovery in a technical corpus. In *Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'04 Conference held in Zakopane, Poland, May 17-20, 2004*, pages 89–98, 2004.
https://link.springer.com/chapter/10.1007/978-3-540-39985-8_10.
- [31] Mathieu Roche and Violaine Prince. A web-mining approach to disambiguate biomedical acronym expansions. *Informatica (Slovenia)*, 34(2):243–253, 2010.
<http://www.informatica.si/index.php/informatica/article/view/296>.
- [32] Mathieu Roche, Maguelonne Teisseire, and Gaurav Shrivastava. Valorcan-TETIS: Terms extracted with Biotex [dataset]. *CIRAD Dataverse*, 2017.
<http://dx.doi.org/10.18167/DVN1/PQQQL>.
- [33] Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, CICLing '02*, pages 1–15, London, UK, UK, 2002. Springer-Verlag.
<http://dl.acm.org/citation.cfm?id=647344.724004>.
- [34] Claire Serp, Anne Laurent, Mathieu Roche, and Maguelonne Teisseire. La quête du graal et la réalité numérique. *Corpus*, 7, 2008.
<https://doi.org/10.4000/corpus.1512>.
- [35] Piyoros Tungthamthiti, Kiyooki Shirai, and Masnizah Mohd. *Recognition of sarcasm in tweets based on concept level sentiment analysis and supervised learning approaches*, pages 404–413. Faculty of Pharmaceutical Sciences, Chulalongkorn University, 2014.
<https://www.aclweb.org/anthology/Y14-1047>.
- [36] Peter D. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning, ECML '01*, pages 491–502, London, UK, UK, 2001. Springer-Verlag.
<http://dl.acm.org/citation.cfm?id=645328.650004>.
- [37] Sebastián Ventura and José María Luna. *Quality Measures in Pattern Mining*, pages 27–44. Springer International Publishing, Cham, 2016.
https://doi.org/10.1007/978-3-319-33858-3_2.
- [38] Manisha Verma, Balasubramanian Raman, and Subrahmanyam Murala. Local extrema co-occurrence pattern for color and texture image retrieval. *Neurocomput.*, 165(C):255–269, October 2015.
<http://dx.doi.org/10.1016/j.neucom.2015.03.015>.
- [39] Yong Yin, Ikou Kaku, Jiafu Tang, and JianMing Zhu. *Association Rules Mining in Inventory Database*, pages 9–23. Springer London, London, 2011.
https://doi.org/10.1007/978-1-84996-338-1_2.
- [40] Hamed Zamani and W. Bruce Croft. Relevance-based word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17*, pages 505–514, New York, NY, USA, 2017. ACM.
<http://doi.acm.org/10.1145/3077136.3080831>.

Association Rule Model of On-demand Lending Recommendation for University Library

Shixin Xu

Huaiyin Institute of Technology, Huaian, Jiangsu 223003, China

E-mail: xusx@hyit.edu.cn

Student paper

Keywords: library, recommendation, association rules, Bayes

Received: August 31, 2020

University library that is connected to the Internet is more convenient to search, but the huge amount of data is not convenient for users who lack a precise target. In this study, the traditional association rule algorithm was improved by a Bayesian algorithm, and then simulation experiment was carried out taking borrowing records of 1000 students as examples. In order to verify the effectiveness of the improved algorithm, it was compared with the traditional association rule algorithm and collaborative filtering algorithm. The results showed that the recommendation results of the improved association rule recommendation algorithm were more relevant to students' majors, and the coincidence degree of different students was low. In the objective evaluation of the performance of the algorithm, the accuracy, recall rate and F value showed that the personalized recommendation performance of the improved association rule algorithm was better and the improved association rule algorithm could recommend users with the book type that they need.

Povzetek: Opisan je asociativni algoritem z dodanim Bayesovim klasifikatorjem za iskanje po univerzitetni knjižnici.

1 Introduction

The arrival of the Internet era has made great changes in our lives, the most intuitive expression of which is that the amount of information that can be obtained far exceeds the era before the emergence of the Internet [1]. Although the Internet with a large amount of information greatly facilitates people's lives, the huge amount of information also greatly increases the difficulty of people's retrieval of effective information. The same is true in the university libraries. Generally, the number of books in a university library is very large. In order to meet the needs of university teachers and students, the selection of books is often very rich [2]. According to the method of field search, it takes time and effort to browse the bookshelves one by one. When the Internet is combined with the university library, the book information in the university library is uploaded to the Internet, and the university teachers and students can simply retrieve the desired book information by using the Internet [3]. However, similar to the Internet described before, although the amount of book information in university library cannot be compared with the amount of data in the whole Internet, it is still a huge amount of data for university teachers and students. If there is a clear goal, it can be accurately retrieved, but if there is only a vague demand range, it is difficult to accurately retrieve the required information. Zhang [4] proposed a personalized book recommendation algorithm based on time series collaborative filtering recommendation and found through experiment that the book recommendation algorithm met the professional

learning needs of college students. Sohail et al. [5] put forward an opinion mining based recommendation technology which provided college students with promising books in the syllabus and found through experiment that the accuracy of this method improved by 55% and it could be applied to the recommendation of other products. Chahinez et al. [6] proposed a book recommendation method based on complex user query, and the experimental results showed that the combination with retrieval model could significantly improve the standard ranked retrieval metrics. In this study, the traditional association rule algorithm was improved by a Bayesian algorithm, and then simulation experiment was carried out taking borrowing records of 1000 students as examples. In order to verify the effectiveness of the improved algorithm, it was compared with the traditional association rule algorithm and collaborative filtering algorithm.

2 Book recommendation algorithm based on association rules

2.1 Association rule algorithm

Association rule recommendation algorithm [7] is to find the connection between different project elements from a large data set and regard the connection whose degree exceeds the set threshold as a strong association rule to guide the recommendation of books. The key point of the

association rule recommendation algorithm is to find the strong rule in the database. The algorithm is generally divided into two steps: ① search frequent sets in the database; ② search strong rules in the frequent set.

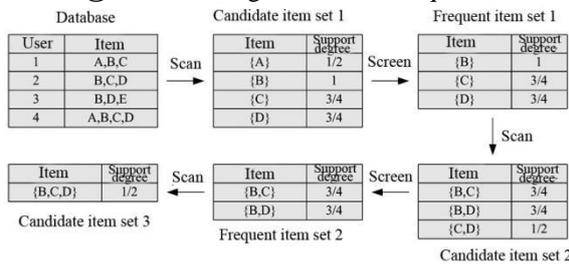


Figure 1: The basic diagram of association rule algorithm flow.

For convenience, as shown in Figure 1, numbers represent users, letters represent the names of books borrowed by users, and the number of records is reduced to 4 users and 4 books. Firstly, a book is taken as the candidate item, and then the support degree of each item in candidate project set 1 is calculated [8] using the following formula:

$$SUP = \frac{n}{N}, \quad (1)$$

where SUP is the support degree of item, N is the total number of records in the database, for example, there are 4 borrowing records of users in Figure 1, and n is the number of records containing the item. According to the set support degree threshold, frequent set 1 is filtered out, then frequent set items are combined to form a new candidate set, and the new frequent set is filtered out; the above operation is repeated until no candidate set can be obtained.

In addition to calculating the support degree, the confidence degree should also be calculated for searching the strong rule in frequent item set. Taking $\{B, C\}$ item in frequent item set 2 in Figure 1 as an example, a strong association rule may produce between its non-empty subset and the set of remaining elements, then the possible strong rule is $\{B\} \Rightarrow \{C\}$ and $\{C\} \Rightarrow \{B\}$. For association item set $X \Rightarrow Y$, the calculation formula of confidence degree [9] is:

$$CON = \frac{|X \cap Y|}{|X|}, \quad (2)$$

where CON stands for the confidence level of the association term set, $|X \cap Y|$ stands for the number of records containing two items at the same time, and $|X|$ is the number of records containing the item. If the confidence level of the association item set exceeds the

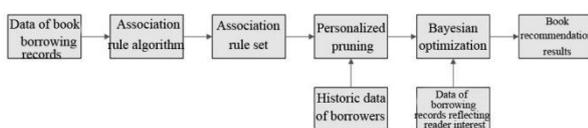


Figure 2: The process of the association rule recommendation algorithm improved by Bayesian network.

threshold, the association item set is considered as a strong rule. The confidence degree of all the items in the frequent item set are calculated as above, and the strong association rule is selected as the reference of book recommendation.

2.2 Improvement of association rule algorithm by Bayesian network

In order to make up for the shortcomings of the association rule algorithm, the association rule algorithm was improved by Bayesian algorithm [10]. The basic steps are as follows. Firstly, a training set is established, and the conditional probability estimation of different characteristic attributes of items to be classified in every classification is counted. Secondly, the probability of belonging to a classification is calculated according to characteristic attributes of the item to be classified [11]:

$$P(Y_i|X) = \frac{P(X|Y_i)P(Y_i)}{P(X)}, \quad (3)$$

where $P(Y_i|X)$ stands for the probability of item X to be classified belonging Y_i , X represents a set of some borrowed books in the historic record of borrows, Y_i indicates the set of some kind of recommended books obtained according to X , i.e., the probability of book X being classified to book Y_i or the establishment probability of association item set $X \Rightarrow Y_i$ after Bayesian calibration, and $P(X|Y_i)$ stands for the distribution probability of X in Y_i , whose value is obtained by estimating the conditional probability of X by the training set.

Thirdly, the probability of X belonging to Y_i is calculated using equation (3), and the set with the largest probability is the most possible association item set.

The association rule set which is calculated by the association rule algorithm is optimized by Bayesian algorithm, and the book recommendation result is obtained according to its probability. The basic flow is shown in Figure 2.

Firstly, the data of book borrowing records in the library are input, and then the association rule set is summarized using the association rule algorithm described above from the borrowing records.

② After obtaining the association rule set, in order to obtain the personalized book recommendation, the association rule set is pruned based on the historical data of the borrower [12]: the items in the association rule set are compared with the items in the historical data, and the record is deleted if the difference is smaller than the set threshold value. The calculation formula of the threshold value is:

$$N = \frac{\text{count}(S_i)}{\text{count}(H_{user})}, \quad (4)$$

where N is the set threshold and $\text{count}(S_i)$ and $\text{count}(H_{user})$ are the number of items in the frequent item set and the number of borrowing records.

③ Through practical investigation, the interest tendency of borrowers to different books in the borrowing records are confirmed, so as to build the borrowing record database [13] which reflects the interest of readers, i.e., the training set of Bayesian algorithm. After the training of

Bayesian algorithm, the association rule set is calibrated after personalized pruning. Finally, the book is recommended according to the probability obtained after the calibration of Bayesian algorithm.

3 Simulation experiment

3.1 Experimental environment

In this study, the above recommended algorithm was simulated using MATLAB software [14]. The experiment was carried out in a laboratory server. The configuration of the server was Windows 7 operating system, 16 G memory and Core i7 processor.

3.2 Experimental setup

First of all, the experimental data used for the simulation experiment came from the book borrowing management system of a university library. Taking 1000 students as subjects, the borrowing records of them from freshmen year to senior year were collected, and then the preliminary processing was carried out, including deleting the records with less than 7 books borrowed (it will reduce the amount of samples, leading to a large contingency in the association rule summarized by the algorithm, deleting the useless fields in the records, such as name and gender of borrows, book author, etc., deleting the invalid data records. There were 26525 borrowing records after final processing, and some records after pretreatment are shown in Table 1.

Library card No.	Borrowing grade	Major disciplines	Book type
A12045	Freshman	Law	D90; D92; D923; D924;
B21541	Sophomore	Economics	F03; F05; G114; G411;
B22548	Junior	Mathematics	O1; O4; P3; Q2;.....
A12365	Senior	Medicine	R4; R75; Q3;.....
.....

Table 1: Some book borrowing records after pretreatment.

Some of the borrowing records after pre-processing are shown in Table 1. Only the library card number which represents the identity of the borrower, the borrowing grade which represents the borrowing time, the major of the borrower and the type of books borrowed by the borrower were left in the borrowing records. Taking the record in the first row of Table 1 as an example, a student whose library card number was A12045 and who was major in law borrowed “D90;D92;D923;D924” books, and most of the books was about law.

In the process of iterative induction of frequent itemsets, the support and confidence degrees of the traditional and improved association rule recommendation algorithms were set as 0.1 and 0.5 respectively, and the

final number of recommended books was set as 5; the data set which was used for training Bayesian algorithm the improved association rule algorithm was the borrowing record which was constructed after investigation and could reflect the interest of readers.

3.3 Evaluating indicator

In this study, the recommendation effect of the recommendation algorithm was evaluated by the accuracy, recall and F value, and their formulas are:

$$\begin{cases} P = \frac{\sum_{i=1}^M L_i}{M \cdot N} \\ R = \frac{\sum_{i=1}^M \frac{L_i}{M \cdot P_i}}{R+P} \\ F = \frac{2RP}{R+P} \end{cases} \quad (5)$$

where L_i is the number of recommended books in line with readers’ interests, M is the number of readers, N is the total number of recommended books, and P_i is the number of books that the reader is interested in.

3.4 Experimental results

The borrowing records obtained after pre-processing were calculated using three algorithms, and finally the book recommendation results of different people were obtained. Limited by the length, this paper only shows some recommendation results, as shown in Table 2. It was seen from Table 2 that the recommendation results of the three algorithms were different for the same person. According to the book classification number, it was found that the books recommended by the collaborative filtering recommendation algorithm were mostly irrelevant, although there were books related to the major; only one or two books recommended by the traditional association rule algorithm were irrelevant; the books recommended by the improved association rule algorithm were basically relevant to the major. The vertical comparison of the recommendation results of different people under the same algorithm showed that the result types under the collaborative filtering algorithm were messy and nearly involved all the majors; the results under the traditional association rule algorithm had overlapping, i.e., high similarity; the results under the improved association rule algorithm involved different types, but different from the collaborative filtering algorithm, they were relevant to the major of the borrower.

The recommendation results of the three recommended algorithms were counted and checked with the corresponding borrower to see if the book was what he was interested in or needed. The final results of the performance of the algorithms are shown in Figure 3. The accuracy of the collaborative filtering algorithm was 67.3%, the recall rate was 72.1%, and the F value was 69.6%; the accuracy of the traditional association rule algorithm was 89.6%, the recall rate was 89.1%, and the F value was 89.3%; the accuracy of the improved association rule algorithm was 98.2%, the recall was 98.3%, and the F value was 98.2%. It was seen from Figure 3 that the improved association rule algorithm had the highest accuracy rate and recall rate, followed by the

traditional association rule algorithm and the collaborative filtering algorithm, indicating that the improved association rule algorithm could provide users with more accurate recommended books; the improved association rule algorithm also had the largest F value, followed by the traditional association rule algorithm and collaborative filtering algorithm. F value is the combination of accuracy and recall rate, which can reflect the personalized recommendation level of the algorithm to different users. The traditional association rule algorithm started from the connection between different book items and used the connection to speculate users' needs. Although personalized pruning was applied, the traditional association rule algorithm was also based on the whole borrowing record, and the strong rule still reflected the overall trend; the improved association rule algorithm used the trained Bayesian algorithm for calibration and optimization to further reflect the demand tendency of different people, therefore the accuracy, recall rate and F value of its personalized recommendation results were larger.

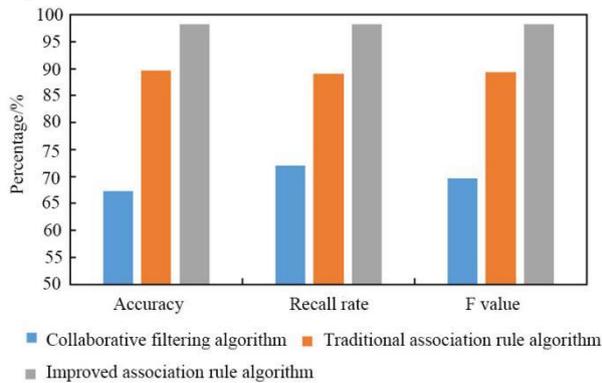


Figure 3: The performance of three recommended algorithms.

4 Conclusion

This paper introduced a recommendation algorithm which mined association rules in borrowing records and improved it with a Bayesian algorithm. Then, borrowing records of 1000 students in the library management system of a university were simulated using MATLAB software. The results are as follows. (1) For the same person, the recommendation results of three algorithms were different: there were many kinds of recommendation results under the collaborative filtering algorithm, only one or two of which were related to the major; there were many kinds of recommendation results under the traditional association rule algorithm, but most of them were related to the major; the recommendation results under the improved association rule algorithm were basically related to the major. (2) Under the same algorithm, the recommendation results for different people were also different: under the collaborative filtering algorithm, the types of recommendation books for different people were diverse; under the traditional association rule algorithm, the types of recommendation books for different people overlapped to a certain extent; under the improved association rule algorithm, the

Library card No.	Recommendation results of collaborative filtering	Results of traditional association rule	Results of improved association rule
A12045	D92;D923;D924;O1;I253.1	D92;D923;D924;I253.1;H1	D923;D924;D923.6;D99;D90
B21541	F05;D99;G20;F12;G114	F05;G114;D923;D924;F12	F03;F05;G114;G411;F12
B22548	O1;O4;P3;F12;H1	O4;P3;G114;D923;D92	O1;O4;P3;Q2;P2
A12365	G20;F12;D923;G114;I253.1	R4;R75;D92;D923;G114	R4;R75;Q3;R8;Q5

Table 2: Some recommendation results of three algorithms.

recommendation books for different people were related to their respective majors, with a low degree of overlap. (3) The results of the objective evaluation showed that the improved association rule algorithm had the largest accuracy, recall rate and F value, followed by the traditional association rule algorithm.

5 References

- [1] Zhou Y (2020). Design and Implementation of Book Recommendation Management System Based on Improved Apriori Algorithm. *Intelligent Information Management*, 12(3), pp. 75-87. <https://doi.org/10.4236/iim.2020.123006>
- [2] Zhang FL (2016). A Personalized Time-Sequence-based Book Recommendation Algorithm for Digital Libraries. *IEEE Access*, pp. 1-1. <https://doi.org/10.1109/ACCESS.2016.2564997>
- [3] Kim JY (2015). A Comparative Study of Pre-service Teachers with Korean Language Education Majors in Book Recommendation Criteria for the Middle and High School Students. *journal of research in reading*, 36, pp. 201-234.
- [4] Zhang F (2016). A Personalized Time-Sequence-Based Book Recommendation Algorithm for Digital Libraries. *IEEE Access*, 4, pp. 1-1. <https://doi.org/10.1109/ACCESS.2016.2564997>
- [5] Sohail SS, Siddiqui J, Ali R (2018). Feature-Based Opinion Mining Approach (FOMA) for Improved Book Recommendation. *Arabian Journal for Science & Engineering*, (2), pp. 1-20. <https://doi.org/10.1007/s13369-018-3282-3>
- [6] Chahinez B, Patrice B (2015). Information Retrieval and Graph Analysis Approaches for Book Recommendation. *Scientific World Journal*, 2015, pp. 1-8. <https://doi.org/10.1155/2015/926418>
- [7] Joa JH, Bangb SW, Parka GD (2016). Implementation of a Recommendation System Using Association Rules and Collaborative Filtering. *Procedia Computer Science*, 91, pp. 944-952. <https://doi.org/10.1016/j.procs.2016.07.115>

- [8] Ping H (2015). The Research on Personalized Recommendation Algorithm of Library Based on Big Data and Association Rules. *Open Cybernetics & Systemics Journal*, 9(1), pp. 2554-2558. <https://doi.org/10.2174/1874110X01509012554>
- [9] Gabroveanu M (2015). Recommendation System Based On Association Rules For Distributed E-Learning Management Systems. *Acta Universitatis Cibiniensis*, 67(1). <https://doi.org/10.1515/aucts-2015-0072>
- [10] dos Santos FF, Domingues MA, Sundermann CV, de Carvalho VO, Moura MF, Rezende SO (2018). Latent association rule cluster based model to extract topics for classification and recommendation applications. *Expert Systems with Application*, 112(DEC.), pp. 34-60. <https://doi.org/10.1016/j.eswa.2018.06.021>
- [11] Gao Y, Xu A, Hu JH, Cheng TH (2017). Incorporating association rule networks in feature category-weighted naive Bayes model to support weaning decision making. *Decision Support Systems*, 96, pp. 27-38.
- [12] Xiao S, Hu Y, Han J, Zhou R, Wen JQ (2016). Bayesian Networks-based Association Rules and Knowledge Reuse in Maintenance Decision-Making of Industrial Product-Service Systems. *Procedia CIRP*, 47, pp. 198-203. <https://doi.org/10.1016/j.procir.2016.03.046>
- [13] Rao W, Zhu L, Pan S, Yang P, Qiao J (2019). Bayesian Network and association rules-based transformer oil temperature prediction. *Journal of Physics Conference*, 1314, pp. 012066. <https://doi.org/10.1088/1742-6596/1314/1/012066>
- [14] Siddiquee MR, Rahman S, Chowdhury SUI, Rahman MR (2016). Association rule mining and audio signal processing for music discovery and recommendation. *International Journal of Software Innovation*, 4(2), pp. 71-87. <https://doi.org/10.4018/IJSI.2016040105>

Designing Hybrid Intelligence Based Recommendation Algorithms: An Experience Through Machine Learning Metaphor

Arup Roy

Birla Institute of Technology, Mesra, India

E-mail: aruproy.cse@gmail.com

Thesis summary

Keywords: recommendation system, hybrid intelligent system, optimization, machine learning

Received: August 12, 2019

This article presents a summarization of the doctoral thesis, which proposes efficient hybrid intelligent algorithms in recommendation systems. The development of effective recommendation algorithms for ensuring quality recommendation in a timely manner is a tricky task. Moreover, the traditional recommendation system is inadequate to cope up with the new technological trends. To overcome these issues, a batch of sophisticated recommendation systems has been discovered e.g. contextual recommendation, group recommendation, and social recommendation. The research work investigates and analyzes new genres of recommenders using nature-inspired algorithms, evolutionary algorithms, swarm intelligence algorithms, and machine learning techniques. The algorithms resolve some crucial problems of these recommenders. As a result, the more personalized recommendation is ensured.

Povzetek: Povzetek doktorske disertacije, ki predlaga učinkovite hibridne inteligentne algoritme v priporočenih sistemih, raziskuje in analizira nove zvrsti priporočil z uporabo algoritmov po naravnih vzorih, evlucijskih algoritmov, algoritmov z roji in tehnik strojnega učenja.

1 Introduction

Recommendation from known sources assist to achieve unknown tasks, e.g. purchasing of products, making plans for vacation, etc. However, verbal assurance often lacks real-time information and consequences contradicting opinions. Consequently, users are overwhelmed by the voluminous information, and the possibility of opting wrong products could increase. *Recommendation System* (RS) becomes functional in such situations, e.g. movie recommendation of movielens.org, music recommendation of last.fm, product recommendation of amazon.com [1][2]. An RS lessens, the “information overload” problem as well as provide quick *personalized* recommendations [3]. Technically, the recommendation process consists of collecting user preferences, tracking the relevant data, and executing the recommendation algorithms [4].

The thesis presents intelligent recommendation models considering real-life applications viz., movies, e-commerce, restaurants, hotels, and matrimonial sites using learning algorithms, nature-inspired algorithms and meta-heuristics optimizations. The initial chapters of the thesis propose a new group and contextual recommendation algorithms. The final few chapters depict performance optimization algorithms.

2 Methodology

The thesis deals with designing hybrid intelligent algorithms using soft computing techniques, bio-inspired algorithms, and probabilistic models. Specifically, the

research introduces (a) **Crowd-Sourcing based Group Recommendation Framework:** a modified termite colony based hybrid movie recommendation framework is introduced to minimize the scalability problem, recommendation of high quality products, and minimization of the recommendation time[5] (b) **Trusted Contextual Recommendation Framework:** a fish school search algorithm based model is proposed to ensure the recommendation from reputed users, and a reduction of the recommendation hazards using artificial bee colony based simulated annealing algorithm[6] (c) **Functional Retail Recommendation Framework:** a termite colony based optimized model is introduced for product recommendation, predicted of stocks based on product consumption pattern, and prediction to increase the overall selling[7] (d) **New Collaborative Filtering Framework:** a rough-dragonfly hybrid is proposed to find the optimal neighbors of the active user, accurate rating prediction, and removal of data sparsity issue[8] (e) **New Vista in Demographic Filtering Framework:** a K-means-ant colony hybrid is introduced to recommend the best partners in matrimonial sites, intelligent noisy data removal mechanism prior to recommendation, and intelligent classification of the significant attributes[9].

3 Results

In (a) **Crowd-Sourcing based Group Recommendation Framework:** The well-known *Movie-Lens* dataset is used in the experimentation purpose. The metrics such as *Mean Absolute Error* and *Root Mean Squared Error* have been used to test the error in the predicted rating.

Moreover, the proposed content based filtering has been compared with the *Jaccard*, *Tanimoto*, and *Binary Cosine* techniques. The experimental results show promising results compared to these techniques. In **(b) *Trusted Contextual Recommendation Framework***: The Irish *Trip-Advisor* dataset is used in the experimentation. Moreover, the AOL data set is used for the verification of proper access operations. Particularly, location and time are considered as contextual features. The parameters *reputation of a user*, *recommendations to a user*, *degree of impact*, and *fitness* are considered to demonstrate the effectiveness of the proposed algorithm. In **(c) *Functional Retail Recommendation Framework***: The transactional dataset (13 distinct values) from a UK-based online retail store is used in experimentation purpose. The parameters *frequency of selling*, *repeat purchase*, *number of purchases*, and *total selling frequency* assisting to predict the pattern of the stock in the near future. In **(d) *New Collaborative Filtering Framework***: The model is trained using *Restaurant and Consumer data of the Recommender Systems Domain*. Subsequently, metrics such as *Coverage*, *Root Mean Squared Error*, *Precision*, *F-Measure*, and *Reliability* demonstrates the effectiveness of the proposed model. In **(e) *New Vista in Demographic Filtering Framework***: The first 100 demographic profiles of the prospective brides and grooms from the popular Indian matchmaking website *SimplyMarry.com* is considered for the validation purpose. The metrics such as *Success Rate* and *Recall* depicts the efficiency of the proposed algorithm.

4 Conclusion and Future Work

The thesis proposes some novel ideas and implementations to envisage recommendations. To achieve this, the research exhibit intelligent recommendations through learning. It successfully gets rid of some inherent limitations such as detection of intruders, recommendation generation, rating prediction, neighbor selection, and matching of the user profiles. The frameworks have been proposed in view of some real-life applications such as movies, e-commerce, restaurants, hotels, and matrimonial sites. As a result, researches become more vibrant and exciting. Moreover, the models could be easily plugged into commercial recommenders. Although, the researches show promising results, some improvements need to be taken care of such as utility-based recommendations, good consensus functions for group recommenders, management of big data, robust algorithms to efficiently deal with the fuzzy, ambiguous, and non-deterministic information.

5 References

- [1] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas, Recommending and evaluating choices in a virtual community of use, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 194-201, 1995.
<https://doi.org/10.1145/223904.223929>
- [2] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl, GroupLens: An open architecture for collaborative filtering of netnews, in: Proceedings of the ACM Conference on Computer Supported Cooperative Work, 175-186, 1994.
<https://doi.org/10.1145/192844.192905>
- [3] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry, Using collaborative filtering to weave an information, Communications of the ACM, 35(12): 61-70, 1992.
<https://dl.acm.org/doi/10.1145/138859.138867>
- [4] Keunho Choi, Donghee Yoo, Gunwoo Kim, and Yongmoo Suh, A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis, Electronic Commerce Research and Applications, 11(4): 309-317, 2012.
<https://doi.org/10.1016/j.elerap.2012.02.004>
- [5] Arup Roy, Soumya Banerjee, Chintan Bhatt, Youakim Badr, Sourav Mallik, Hybrid group recommendation using modified termite colony algorithm: A context towards big data, Journal of Information and Knowledge Management, 17(2), 2018.
<https://doi.org/10.1142/S0219649218500193>
- [6] Arup Roy, Madjid Tavana, Soumya Banerjee, Debora Di Caprio, A secured context aware tourism recommender system using artificial bee colony and simulated annealing, International Journal of Applied Management Science, 8(2): 93-113, 2016.
<https://dx.doi.org/10.1504/IJAMS.2016.077014>
- [7] Soumya Banerjee, Neveen I. Ghali, Arup Roy, Aboul Ella Hassanein, A bio-inspired perspective towards retail recommender system: Investigating optimization in retail inventory, in: Proceedings of the IEEE International Conference on Intelligent Systems Design and Applications, 161-165, 2012.
<https://doi.org/10.1109/ISDA.2012.6416530>
- [8] Arup Roy, Soumya Banerjee, Manash Sarkar, Ashraf Darwish, Mohamed Elhoseny, Aboul Ella Hassanein, Exploring new vista of intelligent collaborative filtering: A restaurant recommendation paradigm, Journal of Computational Science, 27(1): 168-182, 2018.
<https://doi.org/10.1016/j.jocs.2018.05.012>
- [9] Arup Roy, Soumya Banerjee, Who will be my dearest one? An expert decision, International Journal of Advanced Intelligence Paradigm (In Press).

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 900 staff, has 700 researchers, about 250 of whom are postgraduates, around 500 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of Slovenia (or S^onia). The capital today is considered a crossroad between East, West and Mediter-

anean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology park “Ljubljana” has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park “Ljubljana”. The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.: +386 1 4773 900, Fax.: +386 1 251 93 85
WWW: <http://www.ijs.si>
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please register as an author and submit a manuscript at: <http://www.informatica.si>. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible from typing errors to global philosophical disagreements. The chosen editor will send the author the obtained reviews. If the paper is accepted, the editor will also send an email to the managing editor. The executive board will inform the author that the paper has been accepted, and the author will send the paper to the managing editor. The paper will be published within one year of receipt of email with the text in Informatica MS Word format or Informatica L^AT_EX format and figures in .eps format. Style and examples of papers can be obtained from <http://www.informatica.si>. Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the managing editor.

SUBSCRIPTION

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia. E-mail: drago.torkar@ijs.si

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than twenty-six years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica web edition is free of charge and accessible at <http://www.informatica.si>.

Informatica print edition is free of charge for major scientific, educational and governmental institutions. Others should subscribe.

Informatica WWW:

<http://www.informatica.si/>

Referees from 2008 on:

A. Abraham, S. Abraham, R. Accornero, A. Adhikari, R. Ahmad, G. Alvarez, N. Anciaux, R. Arora, I. Awan, J. Azimi, C. Badica, Z. Balogh, S. Banerjee, G. Barbier, A. Baruzzo, B. Batagelj, T. Beaubouef, N. Beaulieu, M. ter Beek, P. Bellavista, K. Bilal, S. Bishop, J. Bodlaj, M. Bohanec, D. Bolme, Z. Bonikowski, B. Bošković, M. Botta, P. Brazdil, J. Brest, J. Brichau, A. Brodnik, D. Brown, I. Bruha, M. Bruynooghe, W. Buntine, D.D. Burdescu, J. Buys, X. Cai, Y. Cai, J.C. Cano, T. Cao, J.-V. Capella-Hernández, N. Carver, M. Cavazza, R. Ceylan, A. Chebotko, I. Chekalov, J. Chen, L.-M. Cheng, G. Chiola, Y.-C. Chiou, I. Chorbev, S.R. Choudhary, S.S.M. Chow, K.R. Chowdhury, V. Christlein, W. Chu, L. Chung, M. Cigliarić, J.-N. Colin, V. Cortellessa, J. Cui, P. Cui, Z. Cui, D. Cutting, A. Cuzzocrea, V. Cvjetkovic, J. Cyprianski, L. Čehovin, D. Čerepnalkoski, I. Čosić, G. Daniele, G. Danoy, M. Dash, S. Datt, A. Datta, M.-Y. Day, F. Debili, C.J. Debono, J. Dedič, P. Degano, A. Dekdouk, H. Demirel, B. Demoen, S. Dendamrongvit, T. Deng, A. Derezsinska, J. Dezert, G. Dias, I. Dimitrovski, S. Dobrišek, Q. Dou, J. Doumen, E. Dovgan, B. Dragovich, D. Dragic, O. Drbohlav, M. Drole, J. Dujmović, O. Ebers, J. Eder, S. Elaluf-Calderwood, E. Engström, U. riza Erturk, A. Farago, C. Fei, L. Feng, Y.X. Feng, B. Filipič, I. Fister, I. Fister Jr., D. Fišer, A. Flores, V.A. Fomichov, S. Forli, A. Freitas, J. Fridrich, S. Friedman, C. Fu, X. Fu, T. Fujimoto, G. Fung, S. Gabrielli, D. Galindo, A. Gambarara, M. Gams, M. Ganzha, J. Garbajosa, R. Gennari, G. Georgeson, N. Gligorić, S. Goel, G.H. Gonnet, D.S. Goodsell, S. Gordillo, J. Gore, M. Grčar, M. Grgurović, D. Grosse, Z.-H. Guan, D. Gubiani, M. Guid, C. Guo, B. Gupta, M. Gusev, M. Hahsler, Z. Haiping, A. Hameed, C. Hamzaçebi, Q.-L. Han, H. Hanping, T. Härder, J.N. Hatzopoulos, S. Hazelhurst, K. Hempstalk, J.M.G. Hidalgo, J. Hodgson, M. Holbl, M.P. Hong, G. Howells, M. Hu, J. Hyvärinen, D. Ienco, B. Ionescu, R. Irfan, N. Jaisankar, D. Jakobović, K. Jassem, I. Jawhar, Y. Jia, T. Jin, I. Jureta, Đ. Juričić, S. K, S. Kalajdziski, Y. Kalantidis, B. Kaluža, D. Kanellopoulos, R. Kapoor, D. Karapetyan, A. Kassler, D.S. Katz, A. Kaveh, S.U. Khan, M. Khattak, V. Khomenko, E.S. Khorasani, I. Kitanovski, D. Kocev, J. Kocijan, J. Kollár, A. Kontostathis, P. Korošec, A. Koschmider, D. Košir, J. Kovač, A. Krajnc, M. Krevs, J. Krogstie, P. Krsek, M. Kubat, M. Kukar, A. Kulis, A.P.S. Kumar, H. Kwašnicka, W.K. Lai, C.-S. Lai, K.-Y. Lam, N. Landwehr, J. Lanir, A. Lavrov, M. Layouni, G. Leban, A. Lee, Y.-C. Lee, U. Legat, A. Leonardis, G. Li, G.-Z. Li, J. Li, X. Li, X. Li, Y. Li, Y. Li, S. Lian, L. Liao, C. Lim, J.-C. Lin, H. Liu, J. Liu, P. Liu, X. Liu, X. Liu, F. Logist, S. Loskovska, H. Lu, Z. Lu, X. Luo, M. Luštrek, I.V. Lyustig, S.A. Madani, M. Mahoney, S.U.R. Malik, Y. Marinakis, D. Marinčič, J. Marques-Silva, A. Martin, D. Marwede, M. Matijašević, T. Matsui, L. McMillan, A. McPherson, A. McPherson, Z. Meng, M.C. Mihaescu, V. Milea, N. Min-Allah, E. Minisci, V. Mišić, A.-H. Mogos, P. Mohapatra, D.D. Monica, A. Montanari, A. Moroni, J. Mosegaard, M. Moškon, L. de M. Mourelle, H. Moustafa, M. Možina, M. Mrak, Y. Mu, J. Mula, D. Nagamalai, M. Di Natale, A. Navarra, P. Navrat, N. Nedjah, R. Nejabat, W. Ng, Z. Ni, E.S. Nielsen, O. Nouali, F. Novak, B. Novikov, P. Nurmi, D. Obrul, B. Oliboni, X. Pan, M. Pančur, W. Pang, G. Papa, M. Paprzycki, M. Paralič, B.-K. Park, P. Patel, T.B. Pedersen, Z. Peng, R.G. Pensa, J. Perš, D. Petcu, B. Petelin, M. Petkovšek, D. Pevec, M. Pičulin, R. Piltaver, E. Pirogova, V. Podpečan, M. Polo, V. Pomponiu, E. Popescu, D. Poshyvanik, B. Potočnik, R.J. Povinelli, S.R.M. Prasanna, K. Pripuzić, G. Puppis, H. Qian, Y. Qian, L. Qiao, C. Qin, J. Que, J.-J. Quisquater, C. Rafe, S. Rahimi, V. Rajković, D. Raković, J. Ramaekers, J. Ramon, R. Ravnik, Y. Reddy, W. Reimche, H. Rezankova, D. Rispoli, B. Ristevski, B. Robič, J.A. Rodriguez-Aguilar, P. Rohatgi, W. Rossak, I. Rožanc, J. Rupnik, S.B. Sadek, K. Saeed, M. Saeki, K.S.M. Sahari, C. Sakharwade, E. Sakkopoulos, P. Sala, M.H. Samadzadeh, J.S. Sandhu, P. Scaglioso, V. Schau, W. Schempp, J. Seberry, A. Senanayake, M. Senobari, T.C. Seong, S. Shamala, c. shi, Z. Shi, L. Shiguo, N. Shilov, Z.-E.H. Slimane, F. Smith, H. Sneed, P. Sokolowski, T. Song, A. Soppera, A. Sornioti, M. Stajdohar, L. Stanescu, D. Strnad, X. Sun, L. Šajn, R. Šenkeřík, M.R. Šikonja, J. Šilc, I. Škrjanc, T. Štajner, B. Šter, V. Štruc, H. Takizawa, C. Talcott, N. Tomasev, D. Torkar, S. Torrente, M. Trampuš, C. Tranoris, K. Trojancanec, M. Tschierschke, F. De Turck, J. Twycross, N. Tziritas, W. Vanhoof, P. Vateekul, L.A. Vese, A. Visconti, B. Vlaovič, V. Vojisavljević, M. Vozalis, P. Vračar, V. Vranić, C.-H. Wang, H. Wang, H. Wang, H. Wang, S. Wang, X.-F. Wang, X. Wang, Y. Wang, A. Wasilewska, S. Wenzel, V. Wickramasinghe, J. Wong, S. Wrobel, K. Wrona, B. Wu, L. Xiang, Y. Xiang, D. Xiao, F. Xie, L. Xie, Z. Xing, H. Yang, X. Yang, N.Y. Yen, C. Yong-Sheng, J.J. You, G. Yu, X. Zabulis, A. Zainal, A. Zamuda, M. Zand, Z. Zhang, Z. Zhao, D. Zheng, J. Zheng, X. Zheng, Z.-H. Zhou, F. Zhuang, A. Zimmermann, M.J. Zuo, B. Zupan, M. Zuqiang, B. Žalik, J. Žižka,

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: <http://www.informatica.si>.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Litostrojska cesta 54, 1000 Ljubljana, Slovenia.

The subscription rate for 2020 (Volume 44) is

- 60 EUR for institutions,
- 30 EUR for individuals, and
- 15 EUR for students

Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Borut Žnidar, borut.znidar@gmail.com.

Printing: ABO grafika d.o.o., Ob železnici 16, 1000 Ljubljana.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) in cooperation with the following societies (and contact persons):

Slovene Society for Pattern Recognition (Vitimir Štruc)

Slovenian Artificial Intelligence Society (Sašo Džeroski)

Cognitive Science Society (Olga Markič)

Slovenian Society of Mathematicians, Physicists and Astronomers (Dragan Mihailović)

Automatic Control Society of Slovenia (Giovanni Godena)

Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Mark Pleško)

ACM Slovenia (Nikolaj Zimic)

Informatica is financially supported by the Slovenian research agency from the Call for co-financing of scientific periodical publications.

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math

Informatica

An International Journal of Computing and Informatics

Reminder of the First Paper on Transfer Learning in Neural Networks, 1976	S. Bozinovski	291
Minimum Flows in Parametric Dynamic Networks the Static Approach	N. Grigoras	303
Investigating Algorithmic Stock Market Trading Using Ensemble Machine Learning Methods	R. Saifan, K. Sharif, M. Abu-ghazaleh, M. Abdel-majeed	311
Increasing the Engagement Level in Algorithms and Data Structures Course by Driving Algorithm Visualizations	S. Simonak	327
Similarity Measure of Multiple Sets and its Application to Pattern Recognition	V. Shijina, U. Adithya, J.J. Sunil	335
Performance Assessment of a Set of Multi-Objective Optimization Algorithms for Solution of Economic Emission Dispatch Problem	S. Mishra, S.K. Mishra	349
Research on Data Transmission Optimization Of Communication Network Based on Reliability Analysis	H. Wang	361
Automatic Image Segmentation for Material Microstructure Characterization by Optical Microscopy	N. Ramou, N. Chetih, Y. Boutiche, R. Abdelkader	367
Smart Design for Resources Allocation in IoT Application Service Based on Multi-agent System and CSP	M. Bali, A. Tari, A. Almutawakel, O. Kazar	373
How to Define Co-occurrence in a Multidisciplinary Context?	M. Roche	387
Association Rule Model of On-demand Lending Recommendation for University Library	S. Xu	395
Designing Hybrid Intelligence Based Recommendation Algorithms: An Experience Through Machine Learning Metaphor	A. Roy	401

