ISSN 2395-8618

Editorial

MANY readers have noticed that Polibits is now indexed in the **DBLP** computer science bibliography, the most important and most widely used international indexing service in the area of computer science. This is a seal of high international recognition of the quality of a journal and impact of the papers published in it. For us, European computer scientists, Polibits, along with another excellent Mexican journal called Computación y Sistemas, represents the best and brightest of Mexican computer science and, more generally, the best and brightest of Latin American computer science.

Starting from this issue, the journal uses electronic **ISSN** different from its print ISSN, and **DOI** references are assigned to all papers and indicated in both the printed and electronic versions. While the journal has been published electronically and with open access since 2008 and now provides electronic open access to its back issues dating many years back, now the electronic version is recognized via a separate ISSN and assignment of DOIs. Yet another achievement of the journal is its inclusion in the **Redalyc** index.

This issue of the journal Polibits includes ten papers by authors from 12 different countries: Argentina, Brazil, Chile, Colombia, Ecuador, France, Mexico, Morocco, Panama, Spain, Turkey, and USA. The papers included in this issue are devoted to such topics as computer security, computer vision, database technologies, city traffic modeling, robotics, optimization, software technology, soft computing, and natural language processing.

M. A. Garcia and T. Trinh from **USA** in their paper "Detecting Simulated Attacks in Computer Networks Using Resilient Propagation Artificial Neural Networks" describe the use of resilient propagation neural networks in the field of computer security. They train the neural network to recognize simulated attacks, for which there is complete information available that permits evaluation of the performance of the suggested approach and compare it with existing approaches. The authors demonstrate that this kind of neural network is a promising mechanism for detecting intrusion in large computer networks.

F. Dornaika et al. from **Spain, Morocco, and France** in their paper "Object Classification using Hybrid Holistic Descriptors: Application to Building Detection in Aerial Orthophotos" present a framework for detection of objects in images, based on the use of hybrid image descriptors. Specifically, they show the advantages of their method on the task of detection of buildings in aerial images. The advantages of the proposed method include its better applicability, suitability, and simplicity, as well as better performance. A hybrid descriptor combines color histograms with a number of local binary patterns. Such descriptors, applied to regions of a segmented image, permit to use supervised machine learning methods for classification that results in detection of the objects of the desired type, building roofs in the case study performed by the authors.

B. Velázquez Ordoñez et al. from **Mexico** in their paper "Integración de fuentes heterogéneas de datos textuales" ("Integration of Heterogeneous Textual Data Sources") show how to improve the results of combining data from different databases. The problem arises because when heterogeneous data from different databases are combined, the merging process can generate inconsistencies, the schema can lack entities to store information of the corresponding kind, different databases may use different languages for data representation or different measures for the values, etc. The authors' approach is based on object-oriented model, which facilitates class extension and reuse.

C. S. G. Pires et al. from **Brazil** in their paper "Mobile ACORoute—Route Recommendation Based on Communication by Pheromones" address the problem of congestion in urban transportation in large cities. Existing major attempts at solution include intelligent transportation systems, and route recommendation systems, based on artificial intelligence techniques. The authors have developed a route recommendation system based on the principles of artificial ant colony optimization, namely, on the pheromone-based communication, combined with A* technique. Their system is capable of real-time modeling of traffic situation, including dynamics of cars and passengers. The system is being implemented in the form of an Android application that will help the users to avoid areas with heavy traffic.

L. Barba and N. Rodríguez from **Ecuador and Chile** in their paper "Traffic Accidents Forecasting using Singular Value Decomposition and an Autoregressive Neural Network Based on PSO" continue the topic of city traffic problems, in this case, traffic accidents. Their case study is the city of Concepción, Chile. They apply artificial intelligence methods for forecasting of the number of traffic accidents in this city. The technique include four main phases: embedding of the input time series using the Hankel matrix, decomposition with the singular value decomposition method, estimation using an autoregressive neural network based on particle swarm optimization, and, finally, recomposition. The proposed strategy shows accuracy superior to that of the existing forecasting techniques.

M. G. Villarreal-Cervantes et al. from **Mexico** in their paper "Influence of the Binomial Crossover in the DE Variants

ISSN 2395-8618

Based on the Robot Design with Optimum Mechanical Energy" discuss the problems that arise in the choice of the binominal crossover parameter for differential evolution. The choice of this parameter depends on the specific problem, and it is difficult to formulate general recommendation for it. Accordingly, the authors investigate the effect of the choice of this parameter in their case study, which is the design of a parallel robot. The goal of this design is to minimize the energy consumed by the robot. The authors show that a correct choice of the crossover parameter improves the energy-related characteristics of the robot.

L. André and R. Stubs Parpinelli from **Brazil** in their paper "The Multiple Knapsack Problem Approached by a Binary Differential Evolution Algorithm with Adaptive Parameters" continue with the topic of optimization techniques based on differential evolution, in this case an adaptive binary differential evolution. They apply this technique to the solution of the 0-1 multiple knapsack problem, which is an NP-hard optimization problem. The authors compare their solution with a number of conventional optimization techniques, such as conventional (not adaptive) binary differential evolution, genetic algorithms, adaptive genetic algorithms, island-inspired genetic algorithm, and adaptive island-inspired genetic algorithm. The proposed technique shows results better than those of the conventional techniques.

A. Castro-Hernández et al. from **USA, Turkey, and Panama** in their paper "Classification of Group Potency Levels of Software Development Student Teams" research into the area of human factors in software engineering. The authors collected various collaboration measures from software development teams in USA, Turkey, and Panama. They used these measures to predict how successful a group that shows particular characteristics can be. They show that simplistic methods are not suitable for the task; however, advanced machine-learning methods allow for good prediction accuracy. Analysis of the factors that correlate with a group's success will be useful to both leaders of the groups and managers at software development companies.

S. Jimenez et al. from **Colombia and Mexico** in their paper "Soft cardinality in Semantic Text Processing: Experience of the SemEval International Competitions" analyze the success factors of the application of soft cardinality measure, which they have developed in their previous work, in semantic text processing. Soft cardinality is a very promising novel general measure of the "size", or diversity, of a set, i.e., the "number" of different elements in the set, which, unlike conventional cardinality, can be non-integer if some of the elements are similar: a set can contain "one and a half" different elements if they are half-similar. This definition of the "size" of sets results in a more accurate idea of intersection or union of sets in a great number of tasks of different nature where things can be partially similar. In particular, with this technique the authors obtained good results during several years at the main international competition on semantic text processing, SemEval. The paper analyzes in detail what properties of the soft cardinality technique defined its success.

M. G. Armentano et al. from **Argentina** in their paper "Applying the Technology Acceptance Model to Evaluation of Recommender Systems" continue with the topic of human factors in software development. When evaluating recommender systems, most researchers concentrate on technical measures such as accuracy of recommendations. However, the impact of a recommender system depends not only on the technical quality of the results but also on its acceptance by the users. The user acceptance may depend on factors of a completely different nature, such as attractiveness of the user interface. A novel technique used by the authors for analysis of the factors of user acceptance includes self-assessment of the user's skills. The study shows that perceived ease of use of the system depends to a large degree on the skill level of the users.

This issue of the journal will be useful to researchers, students, and practitioners working in the corresponding areas, as well as to general public interested in advances in computer science, computer engineering, and artificial intelligence.

Dr. Marta Ruiz Costa-jussà

Universitat Politècnica de Catalunya, Barcelona, Spain
Guest Editor

ISSN 2395-8618

Detecting Simulated Attacks in Computer Networks Using Resilient Propagation Artificial Neural Networks

Mario A. Garcia and Tung Trinh

***Abstract*—In a large network, it is extremely difficult for an administrator or security personnel to detect which computers are being attacked and from where intrusions come. Intrusion detection systems using neural networks have been deemed a promising solution to detect such attacks. The reason is that neural networks have some advantages such as learning from training and being able to categorize data. Many studies have been done on applying neural networks in intrusion detection systems. This work presents a study of applying resilient propagation neural networks to detect simulated attacks. The approach includes two main components: the Data Pre-processing module and the Neural Network. The Data Pre-processing module performs normalizing data function while the Neural Network processes and categorizes each connection to find out attacks. The results produced by this approach are compared with present approaches.**

***Index Terms*—Computer security, artificial neural network, resilient propagation.**

I. Introduction

The number of web attacks in the United States was 413,622,456, which shows the importance of detecting and preventing intrusions [1]. Moreover, according to Shum and Malki [2], from July 2004 to August 2004, the number of network attacks increased 55%. Those statistics alarm network security communities to develop more secured solutions that could protect the tenets of information security: confidentiality, integrity, and availability [3]. There are many proposed methods to develop an intrusion detection system; however, the neural network is considered as an alternative solution to detect zero day attacks. An advantage of neural networks for intrusion detection is that they can "acquire knowledge through learning and store it in inter-neuron connections known as synaptic weights" [4]. In other words, neural networks can detect attacks after they were trained with a subset of network traffic representing the signatures of the attacks to be detected. This method is called "Supervised Learning" because the neural network needs to be trained. (Unsupervised Neural Network can detect attacks without training.) There are several training algorithms to train neural networks such as back propagation, the Manhattan update rule, Quick propagation, or Resilient propagation.

This research describes a solution of applying resilient propagation artificial neural networks to detect simulated attacks in computer networks. The resilient propagation is a supervised training algorithm. The term "supervised" indicates that the neural networks are trained with expected output. The resilient propagation algorithm is considered an efficient training algorithm because it does not require any parameter setting before being used [14]. In other words, learning rates or update constants do not need to be computed. The approach is tested on eight neural network configurations and the results are compared with other approaches found in the literature.

A. Neural Networks and Intrusion Detection

An intrusion is defined as "an attempt to gain unauthorized accesses to network resources" [5]. External people or internal users of networks can be responsible for an intrusion. There are many types of intrusions that are being used by hackers such as viruses, Trojan, attempt break in, successful break in, and Denial-of-Service [6]. An intrusion detection system is software and hardware components to perform three network defense functions: prevention, detection, and response. There are two main criteria to classify intrusion detection systems: the trigger and the source of data used by intrusion detection systems [7]. Based on the trigger, intrusion detection systems can be divided into two types: misuse detection and anomaly detection. Misuse detection is a method of using attack databases to identify attacks. Every activity is compared with known attacks to figure out if that activity is an attack or not. In contrast, anomaly detection finds intrusions by keeping track of characteristics of profiles of authorized users or groups in the network and alerting on discrepancies. Intrusion detection systems use alarms to label those profiles.

A neural network is "an information processing system that is inspired by the way biological nervous systems, such as the brain, process information" [8]. In other words, a neural network consists of a number of elements which work together to solve a given problem. In additional, a neural

Manuscript received on January 15, 2015, accepted for publication on May 10, 2015, published on June 15, 2015.

The authors are with Texas A&M University–Corpus Christi, Computer Science, 6300 Ocean Dr., Corpus Christi, TX, USA (e-mail: {mario.garcia, tung.trinh}@tamucc.edu).

ISSN 2395-8618

network also can be trained to gain knowledge before being used. A neural network contains two main components: the input layer and the output layer. Depending on the complexity of the problem, a neural network can have several hidden layers between the input layer and the output layer. The number of neurons in the input layer should match the size of the input. Normally, the neuron in the output layer is one. The hidden layer plays a role of a data processing station. These layers handle data from the input layer and transfer processed data to the output layer. Neurons in two adjacent layers are connected by the weights. These weights are used to compute the output and to minimize the error produced by the neural networks.

There are two algorithms used to define the structure of a neural network: the topology algorithm and the training algorithm. The topology algorithm refers to the way neurons are connected and how data is transferred between neurons, while the training algorithm denotes the method to adjust weights between neurons to produce accurate output and minimal error. The function to calculate the output is the activation function attached in hidden layers and the output layer. Equation 2.1 shows how to compute the output from a node *j*.

$$output_j = f(x_i) = f\left(\sum_{i=1}^{n} o_i w_{ij}\right) \qquad (2.1)$$

where output *j* is the output of node *j*, x_j is the data of node *j*, o_i is the output of node *i* connected to *j* with the corresponding weight w_{ij}, and $f()$ is the activation function. There are three main activation functions used in neural networks: linear, sigmoid, and hyperbolic tangent. Each activation function scales output in a specific range. Input used in neural networks should be normalized into numbers in that range.

II. State of the Art

In last few years, networking researchers have developed intrusion detection systems using various neural network types. Shum and Malki [2] described a feedforward neural network using the back propagation algorithm implemented with three layers: an input layer, a hidden layer, and an output layer. Similarly, Poojitha, Naveen kumar, and JayaramiReddy [6] introduced an intrusion detection system using an artificial neural network using the back propagation algorithm. This proposed approach uses two phases, training and testing, to detect intrusion activities. First, the intrusion detection system is trained to "capture the underlying relationship between the chosen inputs and outputs" [6].

After that, the system is tested with an available data set. Mukhopadhyay, Chakraborty, Chakrabarti, and Chatterjee [9] presented a study of applying the back propagation algorithm in intrusion detection systems. This approach detects intrusions in four steps: collect data, convert data into MATLAB format, convert data into double data type. This data is used as input to the neural network. Yao [10] expressed a combination of the back propagation neural network and the genetic algorithm. This intrusion detection system has eight modules including: a network packet capture device, the preprocessing module (a), the normal data detection module, the misuse detection module, a statistical module, the preprocessing module (b), the abnormal data detection module, and the alert response module. This approach is proposed to "overcome the blindness of optimization" and "avoid occurring local convergence". Jiang, Yang, and Xia [4] introduced an intrusion detection system based on the improvement of the Self-Organizing Maps algorithm. This approach can "increase detection rate and improve the stability of intrusion detection" by modifying the strategy of "winner-take-all" and using interaction weight which is the effect between each neuron in the output layer [4].

Han [11] proposed an improved model of the Adaptive Resonance Theory 2-A neural network which can "handle data directly". This implementation consists of three layers: F0, F1, and F2. The F0 layer takes input data and transfer to the layer F1 which "performs a Euclidean normalization" to filter only acceptable data to send to the F3 layer. The F3 layer then computes the activation value and labels the winning node as "normal" or "one of the 22 attack types" based on the classification of the data [11]. Bashah, Shanmugam, and Ahmed [7] presented a host-based intrusion detection system using both anomaly detection and misuse detection trigger with the SOM algorithm. Ahmad, Abdullah, and Alghamdi [5] described another proposed intrusion detection system. This system uses resilient back propagation algorithm to compute weights between neural neurons.

A. The Knowledge Data Discover KDD Cup 1999 Data Set

Intrusion detection systems have been grabbing attention of computer science researchers in recent years. There are many approaches have been proposed and presented to network security communities. Instead of using real data, most of proposed approaches use either the DARPA 1998 data set or the KDD Cup 1999 data set or both as the input. The KDD Cup 1999 data set provide a completed source for implementing and testing intrusion detection systems. This database contains 22 different attacks and normal connections [12]. The data set contains around five million TCP/IP connections which are labeled as normal or attacks connections. Each connection record contains 41 features in a TCP/IP packet and the "Label" feature denoting the category into that the connection falls. Table I shows 22 intrusion categories included in the KDD Cup 1999 data set. Besides, the "normal" value is assigned to normal connections. Each attack is classified in one of four groups: DoS, U2R, R2L, and Probe. According to [12], these groups are described as follows: DoS: denial-of-service; U2R: unauthorized access to local root privileges; R2L: unauthorized access from remote machines; Probe: surveillance or other probing.

ISSN 2395-8618

TABLE I.
ATTACKS IN THE KDD CUP 1999 DATA SET

Group	Attack Names
DoS	Back, Land, Neptune, Pod, Smurf, Teardro
U2R	Buffer_overflow, loadmodule,
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient warezmaster,
Probe	ipsweep, nmap, portsweep, satan,

B. The Encog Framework

In 2008, a neural network and machine learning framework named Encog was published and developed for C/C++, Java and .NET programming languages by Heaton Research, Inc. [13]. This framework provides the library for creating neural networks and normalizing data. Developers can implement various types of neural network such as feedforward neural networks, adaptive resonance theory 1 neural networks, and self-organizing map neural networks. Moreover, Encog also contains several training techniques like backpropagation, genetic algorithm, Manhattan update rule propagation, and resilient propagation. In addition, multiple activation functions are included in Encog such as Bipolar function, linear function, sigmoid function, and hyperbolic tangent function. In this research, the Encog is used to build the neural network in .NET framework.

III. NEURAL NETWORK – INTRUSION DETECTION SYSTEM DESIGN

According to Heaton [14], neurons in a feedforward neural network are connected forward. In essence, data is transferred from the input layer to the hidden layer 1 and so on, but there are no backward connections. The weight connecting two neurons in two adjacent layers is calculated randomly in the initialization of the neural network. Then this weight is adjusted during the training process. The computation of the output in a Feedforward neural network is described as follows. First, input from the input layer is transferred to each neuron of the hidden layer 1. Then by using Equation 2.1, output of neurons in the hidden layer 1 is transferred to neurons in the hidden layer 2. Similarly, each neuron in this layer generates and then distributes its output to the output layer. Finally, the neuron in the output layer computes the final output.

The resilient propagation is a supervised training algorithm. The term "supervised" indicates that the neural networks are trained with expected output. The resilient propagation algorithm is considered the most efficient training algorithm because it does not require any parameter setting before being used [14]. In other words, learning rates or update constants do not need to be computed. As it was previously discussed, the training algorithm is used to adjust weights to produce accurate output and minimal error rates. The change in weight between two neurons is calculated using Equation 3.1:

$$\Delta w_{ij}^{k} = \begin{cases} -\Delta_{ij}^{k} \; if \; \dfrac{\partial E(w^{k})}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{k} \; if \; \dfrac{\partial E(w^{k})}{\partial w_{ij}} < 0 \\ 0 \text{ otherwise} \end{cases} \tag{3.1}$$

where $\partial E(w^k)/\partial w_{ij}$ refers to the partial derivative of the error with respect to the weight w, w_{ij} is the weight of neurons i and j, Δ_{ij}^{k} is the update value, and k expresses the index of iteration [16]. The update value Δ_{ij}^{k} is updated using Equation 3.2:

$$\Delta_{ij}^{k} = \begin{cases} \eta^{+}\,\Delta_{ij}^{k-1} \text{if } \dfrac{\partial E(w^{k-1})}{\partial w_{ij}} \dfrac{\partial E(w^{k})}{\partial w_{ij}} > 0 \\ \eta^{-}\,\Delta_{ij}^{k-1} \text{if } \dfrac{\partial E(w^{k-1})}{\partial w_{ij}} \dfrac{\partial E(w^{k})}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{k-1} \text{ otherwise,} \end{cases} \tag{3.2}$$

where $0 < \eta^{-} < 1 < \eta^{+}$, $\eta^{+}$is the increase factor and $\eta^{-}$is the decrease factor. In default, η^{+} is equal to 1.2 while η^{-} is equal to 0.5 [14]. If this result is greater than 0, it means the sign has not changed, so the update value Δ is increased by multiplying the previous update value with the increase factor. Nevertheless, if the result is smaller than 0, it means the sign has changed and the previous Δ is too large. Hence, the update value is decreased by multiplying the previous update value with the decrease factor. In order to evaluate how well the neural network is trained, the mean square error method is applied to calculate the error between actual output and expected output. Equation 3.3 describes the mean square error:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}\left(actual_{t} - ideal_{t}\right)^{2} \tag{3.3}$$

where MSE is the mean square error value, n is the size of actual output, actual t is the t^{th} actual output and the ideal t is the corresponding ideal output.

The activation function used in the neural network is the hyperbolic tangent function, which is shown in Equation 3.4.

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{3.4}$$

A. The Data Pre-processing Module

As it was mentioned, neural networks are trained using the activation function. Depending on the activation function used in the neural network, data is normalized in different ranges which can be (0, 1) or (–1, 1). Because the activation function used in the neural network in this research produces output between -1 and 1, training data and testing data are normalized to values in the range from –1 to 1. In the KDD Cup 1999 data set, there are four features in an individual TCP connection in text format: protocol_type, service, flag, and label. Because each of these features has various groups of values, each group is converted into a number between –1

ISSN 2395-8618

and 1. Other features are in numeral format, so they are normalized by Equation 3.5:

$$f(x) = \frac{(x - dL)(nH - nL)}{dH - dL} + nL \qquad (3.5)$$

where *x*: value needed to be normalized; *dL*: the lowest value of the data; *dH*: the highest value of the data; n_H: the highest value of the normalization range; *nL*: the lowest value of the normalization range.

IV. Neural Network Intrusion Detection System Testing

Before describing the evaluation plan used in this thesis, a brief review of evaluations of previous and related work is provided. The KDD Cup 1999 data set is used for creating training and testing data sets in every approach. In [2], the training set contains 196 records while three testing sets are the normal traffic set, the known attack set, and the unknown attack set which consist of 50, 25, and 25 records respectively. The records in normal traffic and known attack testing set are extracted from the training set while the unknown attack set contains different data. The results of the evaluation are: 100% of normal traffic and know attacks are detected only 76% of unknown attacks are classified. The results seem very accurate, but the detection rate of new attack is quite low, only 76%. Moreover, the detection rate could increase if the size of testing data increments.

In [4], the approach uses the KDD Cup 1999 data set for training and testing data sets. Each category of attacks is trained and tested individually [4]. Detection rates are only good in three types, normal, dos, and probe, while the other two groups have very low accuracy. Furthermore, most of the testing sets have smaller sizes than their corresponding training sets. In general, the proposed solution does not perform well in detecting all types of attacks. The results are quite better than the results computed in [4]: 99.76% of normal, 100% of DoS and Probe, 67.77% of U2R and 36.84% of R2L. Nevertheless, the classifier accuracies of U2R and R2L are not acceptable.

Another similar evaluation method is described in [11]. The system is trained 10 times with the same 10% subset of the entire KDD Cup 1999 data set and then is tested with the entire data set. In other words, the number of training record is equal to the number of testing records. The lowest detection rate is 90.385% while the highest is 99.946%. The results are accurate, but in order to get that achievement, the neural network is trained with the same amount of data of the testing data. Hence, they do not fully express the capability of neural networks that is detecting new objects based on training objects. In [5], the work focuses on detection of DoS attacks.

The proposed system produces 96.16% detection rate in case of attack detection with the highest ratio is 100% and the lowest scale is 79%. However, the testing data set is extremely small, 11 back attacks and 5 normal packets while the numbers of records of other attacks are not mentioned. Meanwhile, the training data set contains full feature packet of DoS attacks of the KDD Cup 1999 data set [5]. Hence, it is hard to estimate the performance of this system. The method proposed in [9] uses a different method to evaluate neural networks: 2 testing levels. At level 1, the system is trained and then tested with the same data set. Consequently, the success rate is very high: 95.6%. However, at level 2, when the system is tested with the new data set, the success rate reduces to 73.9% [9]. Therefore, this proposed method is considered not accurate.

A. Evaluation Plan

As discussed in the previous section, some prior studies do not produce accurate results. In some cases, the results are lower than 70% which is unacceptable. Only the approach presented in [11] can generate over a 90% detection rate. However, this method uses the same amount of data for training and testing the neural network which may not be practical. Therefore, this study focuses on producing accurate testing results using a small amount of training data. In order to evaluate this approach, several neural networks are tested with the same training and testing data for comparing the efficiency of different structures of neural networks. All neural networks have the same input layer and output layer, resilient propagation training algorithm, and hyperbolic tangent activation function. The only different configuration is the number of neurons in hidden layers. The first number in the Structure column refers to the number of neurons of the hidden layer 1 while the second number is the number of neurons of the hidden layer 2. For example, the neural network NN1 has 9 neurons in the hidden layer 1 and 10 neurons in hidden layer 2. Although there are more choices, due to the limitation of time, only 8 neural networks are trained and tested.

The training data set contains 73,249 records extracted from the entire KDD Cup 1999 data set. The training set is then normalized using the Data Pre-processing module. There are 37,860 records of normal traffic included in the training data. At an individual iteration of training process, the input of the neural network is the set of 41 TCP/IP features of a particular record while the ideal output is the "Label" feature of that record as well. Each element of the input is fed into each neuron of the input layer respectively. Then the training process continues until the end of the training set is reached.

This is to ensure that the neural networks have knowledge of all attacks in the training data. If the neural networks are trained using an error rate, the training process stops immediately after that error rate is achieved. Hence, the neural networks may not be trained with attacks in the rest of the training data set. After being trained, each neural network is tested with several testing data sets. Unlike previous works, in this study, the overall detection rates are considered instead of individual attacks' detection rates. First, the neural networks

ISSN 2395-8618

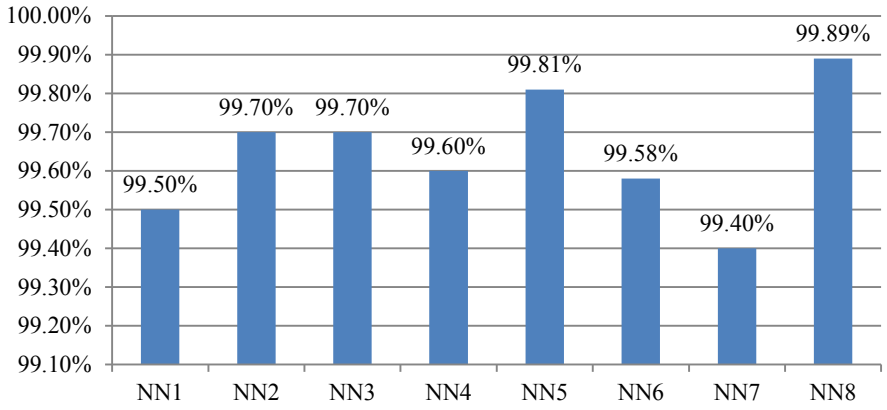


Fig. 1. Training detection rates

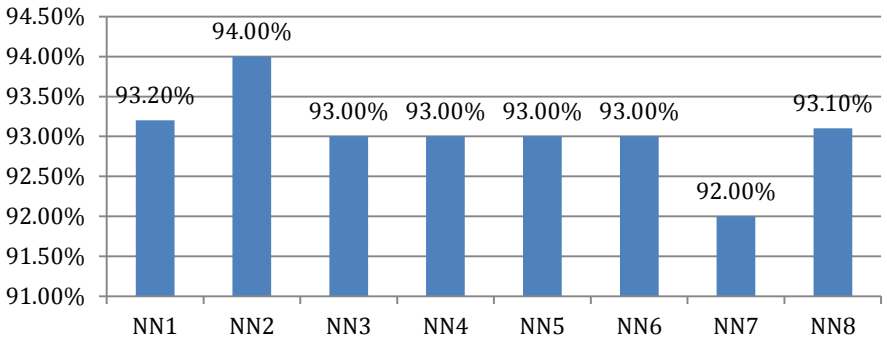


Fig. 2. Detection rates in 10-percent subset test

are tested with the same training set to verify they can detect records used to train them. Then, they are tested against a normal data set, which consists of 83,538 records.

This is to verify that the neural networks can detect normal traffic correctly to prevent false positives. After that, each neural network is tested with the 10-percent subset of the KDD Cup 1999 data set. This subset contains 494,021 records. The input and the ideal output in each test case are computed similarly as presented in the training process. The detection is considered correct if the absolute value of the actual output and the expected output is less than 0.04. Otherwise, it is counted as an incorrect detection. Similarly, the output is compared with each individual connection, including both attack values and normal values, in succession.

B. Results

In this study, the training data set contains more than 70,000 records and 23 different types of output. Hence, the smaller the error is, the better trained neural networks are. The best training error rates among all neural networks was 0.00005 and the worst was 0.00029. The average time it took to train each neural network was 12 hours. The obtained error rates denote that the actual output is very close to the ideal output.

After being trained, the neural networks were tested against the training data set itself. The best result was 99.89% generated by the Neural Network 8, while the worst was 99.4% computed by the Neural Network 7. After being trained and tested with the training data set, each neural network was tested using the normal traffic data set extracted from the KDD Cup 1999 data set. Figure 1 shows the results of this test.

The worst result is 95% produced by the Neural Network 2 while the Neural Network 8 once again produces the best result. In essence, the Neural Network 8 can detect up to 99.99912% normal traffic. Finally, the neural networks are tested with the 10-percent subset of the KDD Cup 1999 data set. This subset consists of 494,021 records, which is equal to 6.7 times of the training data set. There are 8 types of intrusions in the 10-percent subset have similar numbers of records in the training data set. Meanwhile, the rest has the big differences with the training data set.

As an illustration, the neptune attack type has only 20,482 records in the training data set, but it has 10,7201 records in the 10-percent subset. Figure 2 shows the results after testing eight neural networks with the 10-percent subset. The differences between the testing neural networks are really small. The best result is 0.06% and the worst one is 0.08%. Hence, the average result is about 93%. This result is acceptable because the training data set is much smaller than the testing data set. Moreover, the result also shows what is happening in real life where the network traffic is much larger

ISSN 2395-8618

than any data sets. After several tests, the Neural Network 8, which contains 14 neurons in each hidden layer shows the outstanding performance among 8 neural networks. Testing results generated from this neural network are used to compare with other studies.

V. Conclusion

This study already proves a reliable and efficient solution for detecting simulated attacks in computer networks. The system includes two components: the Data Pre-Processing module and the Neural Network. The Data Pre-processing module plays a role of processing data in the KDD Cup 1999 data set before data is used in the Neural Network. Meanwhile, the Neural Network is to detect simulated attacks. There are total eight different structures used to evaluate the Neural Network. These structures are the combinations of different numbers of neurons in two hidden layers of the Neural Network. These eight neural networks are built using the feedforward algorithm and trained using the resilient propagation algorithm. Each neural network is trained with a 73,249-records training data set. Then it is tested against three different testing data sets: the training data set, the normal traffic data set, and the 10-percent subset of the KDD Cup 1999 data set. The detection rates are 99.89%, 99.9%, and 93% respectively. The proposed approach produces highly accurate results compared with other approaches. However, while most previous studies use large training data and small testing data, the ratio of training data and testing data in this study is 0.15.

VI. Future Work

Training a neural network is a time consuming process. In this research it took twelve hours to train the resilient back-propagation neural network. The amount of traffic generated in a real computer network can be extremely large. More efficient techniques such as parallel computing may be applied to speed the training.

In order to enhance the detection rate and efficiency of the neural network, different configurations of neural networks such as Back-propagation, Feed-forward, Redial-base neural networks, etc., need to be implemented and analyzed.

An enhanced version of the KDD Cup 1999 data set, NSL-KDD [16] could be used as a training dataset. This data set removes duplicate records in the original KDD Cup 1999. Even better, a huge contribution to the intrusion detection community will be the creation of a new training dataset that includes the most recent attacks.

Another task will be to evaluate the combination of this approach with other traditional methods used in intrusion detection systems including statistical and mathematical models.

One of the most relevant tasks to be performed is the application of this methodology to a real computer network in order to make the research more practical.

Acknowledgments

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Visiting Faculty Program (VFP).

References

[1] M. Baldonado, C.-C.K. Chang, L. Gravano, and A. Paepcke, “The Stanford Digital Library Metadata Architecture,” Int. J. Digit. Libr. 1 (1997) 108–121

[2] J. Shum and H. A. Malki, “Network intrusion detection system using neural networks,” *Fourth International Conference on Natural Computation,* vol. 5, p. 242-246, Oct. 2008

[3] R. Weaver, “Guide to network defense and countermeasures,” Jan. 2006.

[4] D. Jiang, Y. Yang, and M. Xia, “Research on intrusion detection based on an improved SOM neural network,” *Fifth International Conference on Information Assurance and Security*, vol. 1, p. 400-403, Aug. 2009.

[5] I. Ahmad, A.B. Abdullah and A.S. Alghamdi, “Application of artificial neural network in detection of DOS attacks,” *2nd International Conference on Security of Information and Networks,* 2009.

[6] G. Poojitha, K. Naveen kumar and P. JayaramiReddy, “Intrusion detection using artificial neural network,” *International Conference on Computing Communication and Networking Technologies,* p. 1–7, Jul. 2010.

[7] N. Bashah, I. B. Shanmugam, and A.M. Ahmed, “Hybrid intelligent intrusion detection system,” *World Academy of Science, Engineering and Technology*, 2005.

[8] R. Beghdad, “Critical study of neural networks in detecting intrusions,” *Computers and Security,* p. 168-175, Jun. 2008.

[9] I. Mukhopadhyay, M, Chakraborty, S. Chakrabarti, and T. Chatterjee, “Back propagation neural network approach to intrusion detection system,” *International Conference on Recent Trends in Information Systems,* p. 303-308, Dec. 2011.

[10] X. Yao, “A network intrusion detection approach combined with genetic algorithm and back propagation neural network,” *International Conference on E-Heath Networking, Digital Ecosystems and Technologies,* p. 402–405, Apr. 2010.

[11] X. Han, “An improved intrusion detection system based on neural network,” *International Conference on Intelligent Computing and Intelligent Systems,* p. 887–890, Nov. 2009.

[12] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[13] J. Heaton, “Programming neural networks with Encog 3,” Oct. 2011.

[14] J. Heaton, “Introduction to neural networks for C#,” 2008.

[15] A. D. Amastasiadis, G. D. Magoulas, and M. N. Vrahatis,”New globally convergent training scheme based on the resilient propagation algorithm," *Neurocomputing,* vol. 64, p. 253–270, 2005.

[16] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani , "A detailed analysis of the KDD CUP 99 data set," *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on* , p. 1,6, 8-10, July 2009

[17] https://www.securelist.com/en/analysis/204792255/Kaspersky_Security_Bulletin_2012_The_overall_statistics_for_2012#6

ISSN 2395-8618

Object Classification using Hybrid Holistic Descriptors: Application to Building Detection in Aerial Orthophotos

Fadi Dornaika, Abdelmalik Moujahid, Alireza Bosaghzadeh, Youssef El Merabet, and Yassine Ruichek

***Abstract*—We present a framework for automatic and accurate multiple detection of objects of interest from images using hybrid image descriptors. The proposed framework combines a powerful segmentation algorithm with a hybrid descriptor. The hybrid descriptor is composed by color histograms and several Local Binary Patterns based descriptors. The proposed framework involves two main steps. The first one consists in segmenting the image into homogeneous regions. In the second step, in order to separate the objects of interest and the image background, the hybrid descriptor of each region is classified using machine learning tools and a gallery of training descriptors. To show its performance, the method is applied to extract building roofs from orthophotos. We provide evaluation performances over 100 buildings. The proposed approach presents several advantages in terms of applicability, suitability and simplicity. We also show that the use of hybrid descriptors lead to an enhanced performance.**

***Index Terms*—Automatic building detection and delineation, classification, supervised learning, image descriptors, orthophoto.**

I. Introduction

AUTOMATIC objects recognition has become a topic of growing interest for computer vision community. In the last two decades, machine vision techniques were more and more used in order to assist the whole process of Geographical Information Systems (GIS), cultural heritage preservation, risk management, and monitoring of urban regions. For instance, automatic extraction of man-made objects such as buildings and roads has gain significant attention over the last decade. Aerial data are very useful for the coverage of large areas such as cities and several aerial-based approaches are proposed for the extraction of buildings. More precisely, the data essentially employed as input to these approaches are either optical aerial images and derived Digital Surface Model (e.g., [1]) or aerial LiDAR 3D point clouds (e.g., [2]). It is well-known that segmenting buildings in aerial images is a challenging task. This problem is generally considered when we talk about high-level image processing in order to produce numerical or symbolic information. In this context, several methods have been proposed in the literature. Among the techniques most frequently used, one can cite semi-automatic methods that need user interaction in order to extract desired targets or objects of interest from images. Generally, this category of methods has been introduced to alleviate the problems inherent to fully automatic segmentation which seems to never be perfect. It consists to divide an image into two segments: "object" and "background." The interactivity consists in imposing certain hard constraints for segmentation by indicating certain pixels (seeds) that absolutely have to be part of the object and certain pixels that have to be part of the background. Rother et al. [3] presented an iterative algorithm called GrabCut by simplifying user interaction. Their method combines image segmentation using graph cut and GMMs (Gaussian Mixture Models) based statistical models (using the Orchard-Bouman clustering algorithm) of foreground and background structures in color space. A very useful segmentation benchmark, with a platform implementing important algorithms, has recently been proposed by McGuinness and Connor [4]. The authors compared important algorithms such as IGC [5], seeded region growing (SRG) [6], simple interactive object extraction (SIOX) [7]. The SIOX [7] algorithm is also based on color information and has recently been integrated into the popular imaging program GIMP as the "Foreground Selection Tool."

From the point of view of machine learning paradigms, it is desirable to keep the user interaction at the training phase only and to fully automate the detection and recognition at the test phase. In this paper, we propose an image-based approach for object detection and classification namely, detecting roof building in orthophotos. We use a Statistical Region Merging (SRM) regions to get an over-segmented image. The obtained regions are then described by holistic and hybrid descriptors for detection of roof building in orthophotos. First, an over-segmentation is applied on the orthophoto using the SRM algorithm. This over-segmentation is applied on both the training and test images. Second, holistic descriptors including color and Local Binary Patterns are fused in order to get the feature descriptor of a given region. Third, the SRM regions

Manuscript received on May 10, 2015, accepted for publication on June 5, 2015, published on June 15, 2015.

Fadi Dornaika is with the University of the Basque Country (UPV/EHU) and IKERBASQUE, Basque Foundation for Science, Spain (e-mail: fdornaika@hotmail.fr).

Abdelmalik Moujahid, Alireza Bosaghzadeh are with the University of the Basque Country (UPV/EHU), Spain (e-mail: jibmomoa@gmail.com, alireza.bossaghzadeh@gmail.com).

Youssef El Merabet is with Université Ibn Tofail, Kenitra, Morocco (e-mail: elmerabet113@gmail.com).

Yassine Ruichek is with IRTES-SeT, UTBM, Belfort, France (e-mail: yassine.ruichek@utbm.fr).

ISSN 2395-8618

in a test image are then classified using machine learning tools. We argue that the use of color and LBP descriptors will lead to better performance than relying on a single type of descriptors. We provide a performance study on classifiers whose role is to decide if any arbitrary region is a building or not. Furthermore, we provide performance evaluation at pixel level. This evaluation quantifies both the quality of the segmentation and the classification.

II. Proposed Method

The general flowchart of the proposed building-detection method is illustrated in Figure 1. It should be noticed that the training set is formed by a set of labeled regions together with their image descriptor.

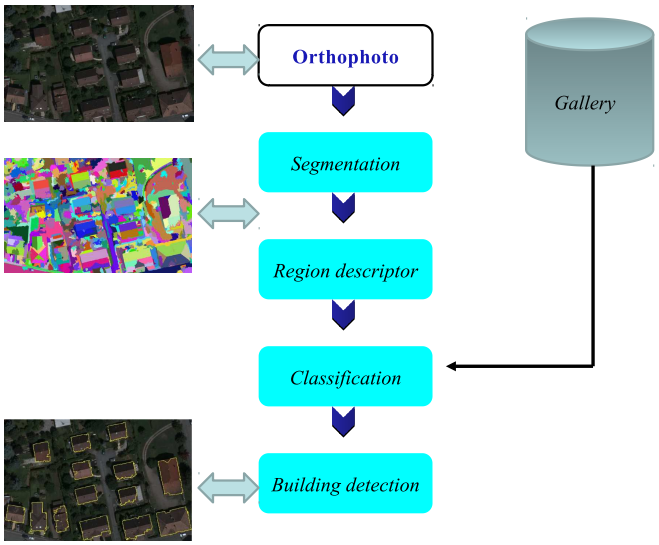


Fig. 1. General flowchart of the proposed building-detection method.

A. Initial Segmentation using Statistical Region Merging

The low-level processing step consists in over-segmenting the input image into many small and homogeneous regions with the same properties. The goal of this initial segmentation, is to avoid the under-segmentation problem and thus correctly extract all significant regions where boundaries coincide as closely as possible with the significant edges characterizing the image. Of course, there are many low level segmentation methods in the literature which can do the job. One can cite Mean shift, Jseg unsupervised segmentation algorithm, watershed, Turbopixels, Statistical Region Merging (SRM), etc. In this work, we have used SRM algorithm to obtain the initial segmentation of the input image. Particular advantages of using this algorithm for dealing with large images are that it dispenses dynamical maintenance of a region adjacency graph, allows defining a hierarchy of partitions. In addition, the SRM segmentation method not only considers spectral, shape, scale information, but also has the ability to cope with significant noise corruption, handle occlusions.

B. Region Representation

In this stage of our method, we dispose of a segmented image obtained via the SRM algorithm. It is still a challenging problem to extract accurately the object contours from this image because only the segmented regions are calculated and no information estimation on their content necessary for the extraction process, is yet done. Our main goal consists in classifying each segmented region as target object or background. For this purpose, we need to characterize these regions using some suitable descriptors. It appears from the literature that there are several aspects that could be considered to represent a region such as the color, edge, texture, shape and size of the region. In our particular context, we believe that color and texture information are the most useful information.

1) Color Histograms: Color histograms were common image descriptors that can describe an object. Note that the region histograms are local histograms and they represent local features of images, and hence the regional color mean value or color histogram are effective parameters to describe statistical information of the object's color distribution. Therefore, we use the color histogram to represent all regions of the segmented image. First we uniformly quantize each color channel into $l = 16$ levels and then the color histogram of each region is calculated in the feature space of l6 x 16 x 16 = 4096 bins. Obviously, quantization reduces the information regarding the content of regions and it is used as trade-off when one wants to reduce processing time. The RGB color space is used in order to calculate the color histogram. Obviously, other color space can be used. Figure 2 illustrates this process on two segmented regions, each belongs to a class.

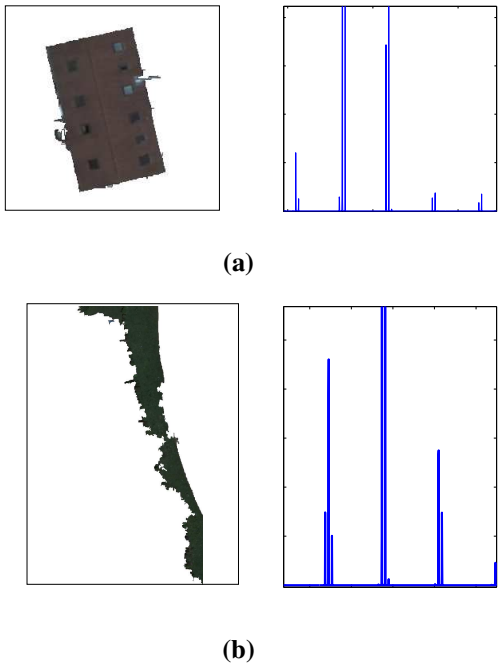


Fig. 2. **(a)** A segmented roof region and its color histogram. **(b)** A segmented background region and its color histogram.

 http://dx.doi.org/10.17562/PB-51-2

ISSN 2395-8618

2) Local Binary Patterns: Local Binary Patterns have proved to be a good texture descriptor. The original LBP operator labels the pixels of an image with decimal numbers, which are called LBPs or LBP codes that encode the local structure around each pixel [8], [9], [10]. It proceeds thus, as illustrated in Figure 3: Each pixel is compared with its eight neighbors in a neighborhood by subtracting the center pixel value; the resulting strictly negative values are encoded with 0, and the others with 1. For each given pixel, a binary number is obtained by concatenating all these binary values in a clockwise direction, which starts from the one of its top-left neighbor. The corresponding decimal value of the generated binary number is then used for labeling the given pixel. The histogram of LBP labels (the frequency of occurrence of each code) calculated over a region or an image can be used as a texture descriptor.

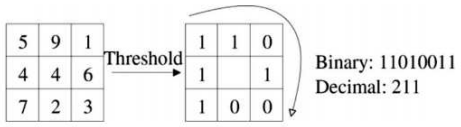
The size of the histogram is 2^P since the operator $LBP(P, R)$ produces 2^P different output values, corresponding to 2^P different binary patterns formed by P pixels in the neighborhood. Several LBP variants have been developed recently to improve performance in different applications [11], [12], [13]. These variants focus on different aspects of the original LBP operator.

For describing a segmented region, we use eight points ($P = 8$) with three radii ($R = 1$, $R = 2$, $R = 3$) each with three modes (uniform, rotation invariant, uniform and rotation invariant). Thus, there are nine LPB descriptors. The final descriptor is given by the concatenation of all. It is worth noting that despite the use of nine LBP descriptors the final one is described by $3 \times (59 + 36 + 10) = 315$ variables only.

3) Hybrid Descriptors: We propose to combine color and texture information in our region descriptor. This is done by simply concatenating the color descriptor and the LBP descriptor. Once the descriptor is computed we apply the square root on all its elements. The motivation for using the square root is that descriptor vectors consist of histograms, and applying square root prior to the distance calculations corresponds to the Hellinger distance between probabilities [14]. Moreover, some recent papers in face recognition literature has shown that the use of the square root of LBP histograms can enhance the recognition performance.

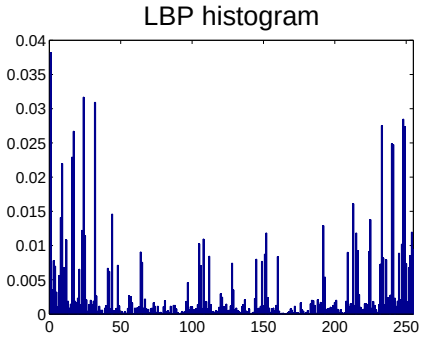
III. Performance Evaluation

In this section, we evaluate several classifiers on the detected regions. This aims at studying the performance of binary classifications on the segmented regions. We stress that the evaluation addresses the overall framework (segmentation and classification) for the problem at hand, namely detecting the building regions in an orthophoto. Firstly, we briefly describe the classifiers used. Secondly, we present the performance of the system for classifying the segmented regions. Thirdly, we present the performance of the system for building detection at pixel level using manually delineated building footprints. We consider six orthophotos depicting one hundred buildings.



(a)

(b)



(c)

Fig. 3. **(a)** Example of basic LBP operator. **(b)** Example of original image. **(c)** Example of LBP descriptor.

A. Machine Learning Approaches

Classification is the sub-field of supervised learning which is concerned with the prediction of the category of a given input. The classification model or classifier is trained using a labelled training set (i.e. a data set containing observations whose category membership is known). Each observation in the data set is a n-dimensional vector, and each element of the vector is called a *feature* (also *attribute* or *variable*). We have used four classifiers: K Nearest Neighbor (K-NN) with (K=1 and K=3), Support Vector Machines (SVM), and Classification Trees (C4.5). A brief description of all of them is included below.

a) Instance Based Learning: Instance Based Learning (IBL) belongs to the K-NN paradigm, a distance based classifier. It computes the distance of a new case to be classified to each of the observations in the database it uses as model and decides the class it will assign based on the K nearest cases. We have used the IBL algorithm described in [15], [16].

ISSN 2395-8618

b) Classification Trees: A Classification Tree is a classifier composed by nodes and branches which break the set of samples into a set of covering decision rules. In each node, a single test is made to obtain the partition. The starting node is called the root of the tree. In the final nodes or leaves, a decision about the classification of the case is made. In this work, we have used the C4.5 algorithm [17]. Note that C4.5 algorithm is also called J48.

c) Support vector machines (SVMs): SVMs are a set of related supervised learning methods used for classification and regression. In a bi-class problem, SVM views the input data as two sets of vectors (one set per class) in a n-dimensional space. The SVM will construct a separating hyperplane in that space, one which maximizes the margin between the two data sets. To calculate the margin, two parallel hyperplanes are constructed, one on each side of the separating hyperplane, which are "pushed up against" the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighboring datapoints of both classes since, in general, the larger the margin the lower the generalization error of the classifier [18]. SVMs were extended to classify data sets that are not linearly separable through the use of non-linear kernels. In our work, we use non-linear SVMs with radial kernel.

d) Partial Least Square (PLS): The Partial Least Squares (PLS) classifier or regressor [19] is a statistical method that retrieves relations between groups of observed variables X and Y through the use of latent variables. It is a powerful statistical tool which can simultaneously perform dimensionality reduction and classification/regression. It estimates new predictor variables, known as components, as linear combinations of the original variables, with consideration of the observed output values. In our work, in both types of PLS, the number of latent components is fixed to 50.

B. Training

In order to get a training set which contains regions belonging to two classes (background and building) with ground-truth labels, we proceed as follows. The buildings are first manually delineated in each orthophoto. Each such ground-truth map is then intersected with the corresponding automatically over-segmented orthophoto. The label of any segmented region can be inferred by using the size of the overlap with the ground-truth building region. Any segmented region whose overlap with a building region exceeds 90% of its size will be labeled as building. Any segmented region whose overlap is below 3% of its size will have the non-building label. The regions that do not meet any of the two conditions are discarded and will not be used in as a training sample. The reason behind using these thresholds is the fact that an automatically segmented region may be shared by a building region and a background region. So it would be advantageous to use only quasi pure regions in the training set.

C. Region classification performance

It would be interesting to study the ability of descriptors and classifiers for recognizing the label of a given region. To this end, we collect a large number of labeled segmented regions, each is assumed to be a region that either belongs to the building category or to the background category. To achieve that we adopt the filtering process explained above. By adopting this filtering scheme, we collect 5656 regions with known labels. We then apply on them the 10-fold-cross validation scheme using the 1-NN, 3-NN, J48 and SVM classifiers. The obtained results are summarized in Tables I and II.

Table I depicts the number of misclassified regions and the rate of correct classification for three types of descriptors (color descriptor, LBP descriptor, and hybrid descriptor) and for four classifiers. The color descriptor is described by 805 features, the LBP descriptor by 315 features and the hybrid descriptor by 1120 features. In this evaluation, the use of hybrid descriptor has not improved the region classification over the color descriptor. However, the hybrid descriptor will improve the pixel classification as will be shown in the sequel. The main reason behind the difference in the obtained performance for regions and pixels is the fact that the SRM segmentation algorithm provides regions whose sizes (number of pixels) vary a lot. In other words, the region misclassification occurs mostly with small regions.

TABLE I
OVERALL REGION CLASSIFICATION RESULTS OBTAINED WITH 10-FOLD-CROSS VALIDATION.

		Color (805)		*LBP (315)*		*Hybrid (1120)*	
Classifier	Regions	Err.	Acc.	Err.	Acc.	Err.	Acc.
1-NN	5656	165	97.08	487	91.38	205	96.37%
3-NN	5656	144	97.45	424	92.50	205	96.37%
J48	5656	205	96.37	653	88.45	228	95.96%
SVM	5656	129	97.71	396	92.99	149	97.36%

Table II depicts the Recall, Precision, and F1 measure for building and background categories. From these two tables, we can observe that the non-linear SVM classifier has provided the best performance. We can also observe that the ability of all classifiers to discriminate background regions was better than that associated with building regions.

TABLE II
RECALL, PRECISION AND F1 FOR BACKGROUND AND BUILDING AND FOR ALL CLASSIFIERS.

	Background			*Building*		
Classifier	Recall	Precision	F1	Recall	Precision	F1
1-NN	98.1%	98.7%	98.4%	88.9%	84.5%	86.7%
3-NN	98.4%	98.8%	98.6%	89.9%	86.7%	88.3%
NB	88.9%	99.1%	93.7%	93.0%	50.0%	65.0%
J48	98.0%	98.0%	98.0%	83.1%	82.9%	83.0%
SVM	98.7%	98.8%	98.7%	89.7%	89.0%	89.3%

 http://dx.doi.org/10.17562/PB-51-2

ISSN 2395-8618

D. Segmentation and Classification Performance

In this section, we study the performance of the segmentation and classification at pixel level. Before presenting the quantitative evaluation, we first present in Figure 4 the results of building detection on the set of processed images using hybrid descriptors and SVM classifier. In each row of this figure, we show the initial orthophoto, the segmented image and the corresponding building roofs extraction where the final detected building boundaries are shown superimposed on the original orthophoto. Based on the visual evaluation of the results, we can state that the developed approach demonstrates excellent accuracy in terms of building boundary extraction, i.e., the majority of the building roofs present in the image are detected with good boundary delineation. Indeed, our method gives reliable results for complex environments having buildings with red and non-red rooftop buildings and/or buildings with the same color and texture with road areas.

In order to get a quantitative evaluation, we use the ground-truth building maps. The manually delineated buildings were used as a reference building set to assess the automated building-extraction accuracy. The extracted buildings and the manually delineated buildings are compared pixel-by-pixel. All pixels in the test image are categorized into four types.

1) True positive (TP). Both manual and automated methods label the pixel belonging to the buildings.
2) True negative (TN). Both manual and automated methods label the pixel belonging to the background.
3) False positive (FP). The automated method incorrectly labels the pixel as belonging to a building.
4) False negative (FN). The automated method does not correctly label a pixel truly belonging to a building.

From these measures it is straightforward to compute the following scores associated with the building regions in the test image: recall, precision, F1 measure, accuracy, and Matthews correlation coefficient (MCC). The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and -1 indicates total disagreement between prediction and observation.

Table III (upper part) illustrates the above scores when color descriptors are used. The classifier used is the non-linear SVM. Table III (lower part) illustrates the above scores when the hybrid descriptors are used. In this table, the evaluation adopted a similar protocol to the 6-fold cross validation in the sense that each orthophoto is used as a test set and the remaining orthophotos (i.e., their descriptors retained in the training set) are used as training samples.

Tables IV, V, VI, and VII illustrate the same evaluation obtained with 1-NN, 3-NN, tree, and PLS classifiers, respectively. We can observe that the use of the hybrid descriptor has improved the average performance of building detection. This is true for all the classifiers used. For example, let's consider SVM classifier and **orthohpoto5**. The rate of correct classification of its pixels is 89.6 % when only color information is used. This rate becomes 93.9 % when the hybrid descriptor is used. Since the size of **orthophoto5** is 652392 pixels this means that with the hybrid descriptor 28053 more pixels are correctly classified. We can also observe that the non-linear SVM and 3-NN classifiers adopting the hybrid descriptor give the best performances. It should be noticed that all results are obtained by using a binary classification without any post-processing.

TABLE III
RECALL, PRECISION, F1, AND MATTHEWS CORRELATION COEFFICIENT (MCC) CORRESPONDING TO A BINARY CLASSIFICATION (PIXEL LEVEL) USING BOTH COLOR AND HYBRID DESCRIPTORS (COLOR HISTOGRAMS WITH LBPS). THE RESULTS ARE OBTAINED WITH SVM.

Image	Recall (%)	Precision (%)	F1-measure (%)	Accuracy (%)	MCC
Color descriptor					
Orthophoto1	78.1	89.2	83.3	96.8	0.82
Orthophoto2	88.7	94.0	91.2	95.4	0.88
Orthophoto3	80.6	85.3	82.9	92.3	0.78
Orthophoto4	92.4	87.8	90.0	96.5	0.88
Orthophoto5	77.4	82.4	79.8	89.6	0.73
Orthophoto6	95.7	76.4	84.9	94.5	0.82
Average	85.5	85.8	85.4	94.2	0.82
Hybrid descriptor					
Orthophoto1	88.1	90.0	89.0	97.8	0.88
Orthophoto2	93.0	95.8	94.4	97.0	0.92
Orthophoto3	83.9	91.9	87.7	94.5	0.84
Orthophoto4	93.6	93.8	93.7	97.8	0.92
Orthophoto5	85.7	91.0	88.3	93.9	0.84
Orthophoto6	95.8	87.4	91.4	97.1	0.90
Average	90.0	91.6	90.8	96.4	0.88

TABLE IV
RECALL, PRECISION, F1, AND MATTHEWS CORRELATION COEFFICIENT (MCC) CORRESPONDING TO A BINARY CLASSIFICATION (PIXEL LEVEL) USING BOTH COLOR AND HYBRID DESCRIPTORS (COLOR HISTOGRAMS WITH LBPS). THE RESULTS ARE OBTAINED WITH 1-NN.

Image	Recall (%)	Precision (%)	F1-measure (%)	Accuracy (%)	MCC
Color descriptor					
Orthophoto1	74.4	86.1	79.8	96.1	0.78
Orthophoto2	89.0	93.0	91.0	95.3	0.88
Orthophoto3	95.5	84.7	89.8	94.9	0.87
Orthophoto4	92.5	90.7	91.6	97.1	0.90
Orthophoto5	76.7	84.0	80.2	89.9	0.74
Orthophoto6	89.6	86.6	88.1	96.1	0.86
Average	86.3	87.5	86.7	94.9	0.84
Hybrid descriptor					
Orthophoto1	93.9	86.8	90.2	97.9	0.89
Orthophoto2	95.4	92.0	93.7	96.5	0.91
Orthophoto3	98.0	87.2	92.3	96.2	0.90
Orthophoto4	93.9	90.8	92.3	97.3	0.91
Orthophoto5	86.5	86.5	86.5	92.8	0.82
Orthophoto6	92.3	86.0	89.1	96.3	0.87
Average	93.3	88.2	90.7	96.2	0.88

IV. CONCLUSION

In this paper, we have introduced a method which accounts for automatic and accurate multiple objects recognition

ISSN 2395-8618

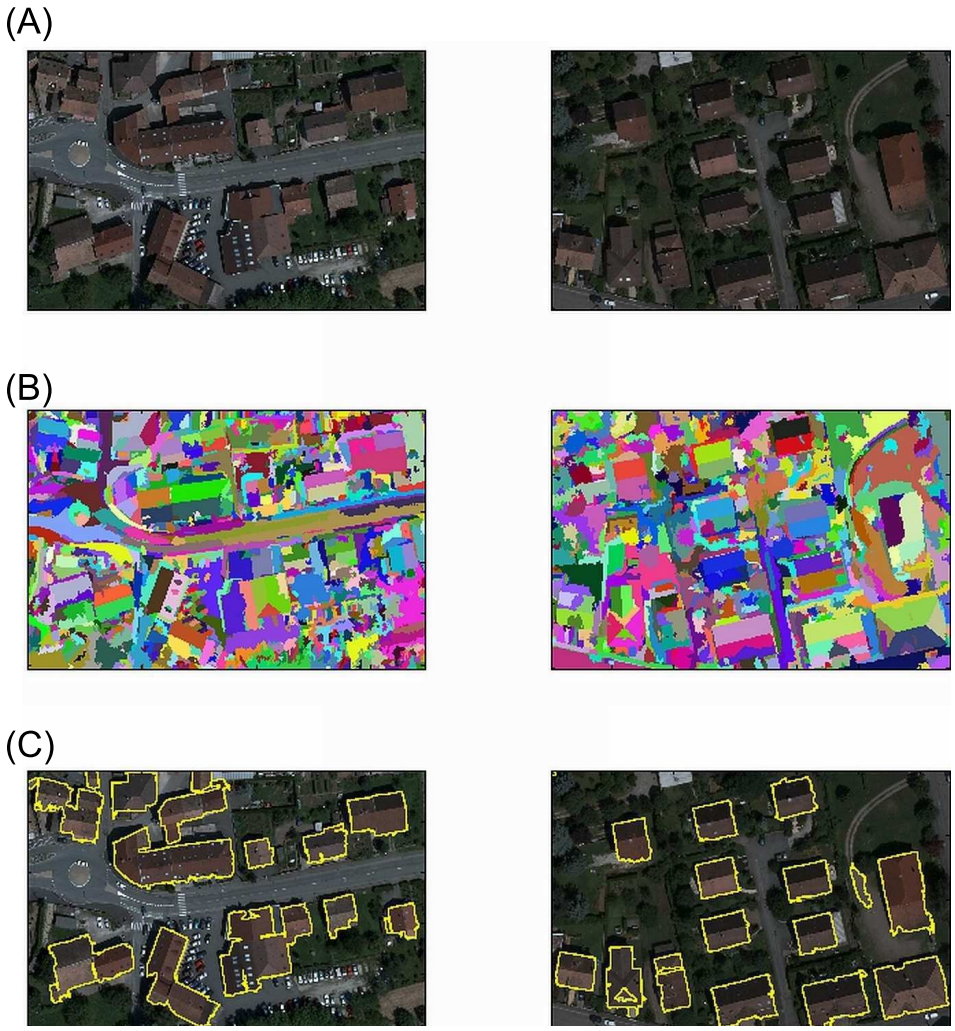


Fig. 4. **(A)** Orthophotos. **(B)** Segmented orthophotos. **(C)** Countours of detected roof regions.

TABLE V
RECALL, PRECISION, F1, AND MATTHEWS CORRELATION COEFFICIENT (MCC) CORRESPONDING TO A BINARY CLASSIFICATION (PIXEL LEVEL) USING BOTH COLOR AND HYBRID DESCRIPTORS (COLOR HISTOGRAMS WITH LBPS). THE RESULTS ARE OBTAINED WITH 3-NN.

Image	Recall (%)	Precision (%)	F1-measure (%)	Accuracy (%)	MCC
Color descriptor					
Orthophoto1	79.7	87.1	83.3	96.7	0.82
Orthophoto2	89.4	95.5	92.3	96.0	0.90
Orthophoto3	98.3	85.6	91.5	95.8	0.89
Orthophoto4	93.8	93.9	93.8	97.9	0.93
Orthophoto5	75.8	82.5	79.0	89.3	0.72
Orthophoto6	87.2	86.7	87.0	95.8	0.84
Average	87.4	88.5	87.8	95.2	0.85
Hybrid descriptor					
Orthophoto1	94.3	86.4	90.2	97.9	0.89
Orthophoto2	93.9	92.4	93.1	96.3	0.91
Orthophoto3	98.1	86.4	91.9	96.0	0.89
Orthophoto4	95.8	94.3	95.0	98.3	0.94
Orthophoto5	85.9	87.0	86.4	92.8	0.82
Orthophoto6	93.4	86.3	89.7	96.5	0.88
Average	93.6	88.8	91.1	96.3	0.89

TABLE VI
RECALL, PRECISION, F1, AND MATTHEWS CORRELATION COEFFICIENT (MCC) CORRESPONDING TO A BINARY CLASSIFICATION (PIXEL LEVEL) USING BOTH COLOR AND HYBRID DESCRIPTORS (COLOR HISTOGRAMS WITH LBPS). THE RESULTS ARE OBTAINED WITH TREE (C.45).

Image	Recall (%)	Precision (%)	F1-measure (%)	Accuracy (%)	MCC
Color descriptor					
Orthophoto1	68.7	90.8	78.2	96.0	0.77
Orthophoto2	81.6	90.5	85.8	92.8	0.81
Orthophoto3	67.0	91.7	77.4	90.9	0.73
Orthophoto4	86.7	92.0	89.3	96.4	0.87
Orthophoto5	68.6	84.2	75.6	88.2	0.68
Orthophoto6	83.2	89.5	86.2	95.7	0.84
Average	76.0	89.8	82.1	93.3	0.78
Hybrid descriptor					
Orthophoto1	88.3	88.3	88.3	97.6	0.87
Orthophoto2	86.4	94.5	90.3	95.0	0.87
Orthophoto3	75.8	88.8	81.8	92.1	0.77
Orthophoto4	86.3	91.4	88.8	96.3	0.87
Orthophoto5	72.3	86.0	78.6	89.5	0.72
Orthophoto6	83.2	90.8	86.8	95.9	0.85
Average	82.0	90.0	85.8	94.4	0.82

 http://dx.doi.org/10.17562/PB-51-2

ISSN 2395-8618

TABLE VII
RECALL, PRECISION, F1, AND MATTHEWS CORRELATION COEFFICIENT (MCC) CORRESPONDING TO A BINARY CLASSIFICATION (PIXEL LEVEL) USING BOTH COLOR AND HYBRID DESCRIPTORS (COLOR HISTOGRAMS WITH LBPS). THE RESULTS ARE OBTAINED WITH NON LINEAR PARTIAL LEAST SQUARE (PLS).

Image	Recall (%)	Precision (%)	F1-measure (%)	Accuracy (%)	MCC
Color descriptor					
Orthophoto1	71.9	90.1	80.0	96.3	0.79
Orthophoto2	86.3	95.8	90.8	95.3	0.88
Orthophoto3	78.5	94.3	85.7	93.9	0.82
Orthophoto4	91.1	92.4	91.8	97.2	0.90
Orthophoto5	72.9	90.6	80.8	90.8	0.76
Orthophoto6	83.6	89.3	86.4	95.7	0.84
Average	80.7	92.2	85.9	94.9	0.83
Hybrid descriptor					
Orthophoto1	82.9	91.1	86.8	97.4	0.85
Orthophoto2	93.3	95.7	94.5	97.1	0.93
Orthophoto3	86.4	93.1	89.6	95.3	0.87
Orthophoto4	92.8	93.7	93.3	97.7	0.92
Orthophoto5	84.4	88.7	86.5	93.0	0.82
Orthophoto6	93.7	88.0	90.8	96.9	0.89
Average	88.9	91.7	90.3	96.2	0.88

from images. Unlike methods that rely on the interactive image segmentation, our approach does not require any user interaction or any setting of initial algorithm parameters (a threshold of similarity for example). The proposed method involves a supervised scheme in which offline manual delineation and automatic segmentation are carried out to build descriptors and classifiers. At running time, after an over-segmentation of the image, one can classify the segmented regions as object parts or background image using region classification.

In order to show its performance, the proposed method was applied to extract building roofs from orthophotos. This problem is very challenging given the complexity of objects in the orthophotos. While orthophotos construction used Digital Surface Maps, our adopted building detection used image information only. Future work may investigate the use of covariance matrix descriptors as hybrid descriptors. Furthermore, we may investigate whether the use of superpixels algorithms (e.g., [20], [21]) could improve the initial segmentation. The choice of more adapted color space could be also an interesting way to improve the results.

REFERENCES

[1] O. Tournaire, M. Brédif, and M. Boldo, D.and Durupt, "An efficient stochastic approach for buildings footprint extraction from digital elevation models," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, pp. 317–327, 2010.

[2] O. Wang, S. K. Lodha, and D. P. Helmbold, "A Bayesian approach to building footprint extraction from aerial LIDAR data," in *International Symposium on 3D Data Processing, Visualization, and Transmission*, 2006.

[3] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut—interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics (SIGGRAPH'04), New York, NY, USA*, 2004.

[4] K. McGuinness and N. E. O'Connor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, pp. 434–444, 2010.

[5] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," in *IEEE Intl. Conf. on Comput. Vision*, 2001.

[6] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 16, pp. 641–647, 1994.

[7] G. Friedland, K. Jantz, and R. Rojas, "SIOX: simple interactive object extraction in still images," in *IEEE Intl. Symposium on Multimedia*, 2005.

[8] T. Ojala, M. Pietikäinen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, 2002.

[9] V. Takala, T. Ahonen, and M. Pietikäinen, "Block-based methods for image retrieval using local binary patterns," in *Image Analysis, SCIA*, vol. LNCS, 3540, 2005.

[10] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[11] M. Bereta, P. Karczmarek, W. Pedrycz, and M. Reformat, "Local descriptors in application to the aging problem in face recognition," *Pattern Recognition*, vol. 46, pp. 2634–2646, 2013.

[12] D. Huang, C. Shan, M. Ardabilian, and Y. Wang, "Adaptive particle sampling and adaptive appearance for multiple video object tracking," *IEEE Trans. on Systems, Man, and Cybernetics-Part C: Applications and reviews*, vol. 41, no. 6, pp. 765–781, November 2011.

[13] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Faces in Real-Life Images Workshop in ECCV*, 2008.

[14] D. Pollard, *A user's guide to measure theoretic probability*. Cambridge University Press, 2002.

[15] D. Aha, D. Kibler, and M. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[16] D. Mena-Torres, J. Aguilar-Ruiz, and Y. Rodriguez, "An instance based learning model for classification in data streams with concept change," in *11th Mexican International Conference on Artificial Intelligence (MICAI)*, 2012, pp. 58–62.

[17] J. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[18] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, pp. 169–186, 2003.

[19] R. Rosipal and N. Kramer, *Subspace, Latent Structure and Feature Selection Techniques*. Springer, 2006, ch. Overview and recent advances in partial least squares, pp. 34–51.

[20] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.

[21] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

ISSN 2395-8618

Integración de fuentes heterogéneas de datos textuales

Benina Velázquez Ordoñez, Jesús Manuel Olivares Ceja, Miguel Patiño Ortíz,
Julián Patiño Ortíz, Adolfo Guzmán Arenas

***Resumen*—Se ha detectado que en algunas aplicaciones de integración de información de fuentes de datos, en algunos casos pueden ocurrir inconsistencias y en otros, se carece de una entidad para almacenar los datos. Algunas inconsistencias se deben a que los datos se expresan en diferente idioma al utilizado en el repositorio o por el uso de diferentes unidades de medida. En este artículo, la propuesta utiliza reglas en la integración de datos tratando de preservar la consistencia y en otros casos implican modificaciones al esquema. Se seleccionó el modelo orientado a objetos por sus características que facilitan la reutilización de clases. La base de datos de ejemplo utiliza datos obtenidos de fuentes heterogéneas de la Web pertenecientes al dominio de equipos de computación. En la integración, intervienen entidades, atributos, valores y unidades de medida. Esta propuesta se enfoca en el contenido que es una alternativa a la integración de esquemas de datos.**

***Palabras clave*—Integración de datos, información compartida, intercambio de información, bases de datos orientadas a objetos.**

Integration of Heterogeneous Textual Data Sources

***Abstract*—This paper proposes an alternative to data integration from heterogeneous sources or databases. In some cases, inconsistencies may occur, and in others, the schema lacks of any attribute or entity to store the data. Some inconsistencies are consequence of using a language different with the one employed in the schema definition; others are due to the use of distinct units of measure. The object-oriented model provides characteristics that facilitate the class reuse and extension. The samples are obtained from heterogeneous Web sources belonging to the domain of computer equipment. Integration involves entities, attributes, values, and units of measurement.**

Manuscrito recibido el 19 de junio de 2014, aceptado para su publicación el 10 de julio de 2014, publicado el 15 de junio 2015.

Benina Velázquez Ordoñez (autor correspondiente) estudia en el Instituto Politécnico Nacional (IPN), en la Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME), DF, México (correo: bvelazquez@ipn.mx).

Jesús Manuel Olivares Ceja y Adolfo Guzmán Arenas trabajan en el IPN, en el Centro de Investigación en Computación (CIC), México, DF (correo: jesus@cic.ipn.mx, a.guzman@ieee.org)

Miguel Patiño Ortíz y Julián Patiño Ortíz trabajan en el IPN–ESIME, DF, México (correo: {mpatino2002, jpatinoo}@ipn.mx).

***Index Terms*—Data integration, information sharing, information exchange, object oriented databases**.

I. Introducción

Actualmente, como consecuencia de la globalización varias organizaciones requieren integrar datos de fuentes heterogéneas en forma eficiente. El problema de la integración se considera en dos vertientes: integración del esquema y del contenido [9, 22]. La mayoría de los trabajos publicados se enfocan en la integración de esquemas [1, 2, 7, 8, 12, 14, 16, 18, 19], sin embargo [9, 22, 25], también tratan la integración de contenido. El principal problema que se menciona en estos trabajos es la preservación de la *consistencia* de datos y de las relaciones entre ellos al hacer la integración.

Este artículo se enfoca en la integración de documentos heterogéneos obtenidos de la Web en forma de texto plano hacia un modelo de base de datos orientado a objetos; este modelo se seleccionó porque actualmente representa un intermediario entre el modelo semántico más cercano a las abstracciones manejadas por el usuario y aquellas propias del modelo relacional que utiliza (tablas) con la ventaja de ser más eficiente al ejecutarse y validar las restricciones de integridad.

La integración se hace en forma incremental conforme se leen los datos, algunos casos de integración implican modificar el esquema.

En este documento se trata el dominio de equipos de computación obtenidos de diferentes sitios de la Web.

Históricamente en el modelo entidad-relación extendido [20], además de las entidades y relaciones, se incorpora la generalización y especialización; estableciendo las bases para el modelo orientado a objetos utilizado recientemente y cada vez con más frecuencia. Los objetos se asemejan a los marcos de Minsky [26] que se propusieron para representar entidades de conocimiento estereotipadas desde diferentes puntos de vista, con elementos que se dividen en descriptivos y conductuales; sirven para facilitar la inferencia en forma más amplia que los términos y expresiones ofrecidos por la lógica.

El propósito fundamental de los objetos es el manejo de la complejidad, de acuerdo con [23] los principios para el manejo de la complejidad son: la abstracción, la encapsulación, la herencia, la asociación, la comunicación de

ISSN 2395-8618

mensajes mediante métodos, la escala y la clasificación. Por otra parte [24] indica que los objetos están compuestos por cuatro principales elementos: la abstracción, encapsulación, modularidad y jerarquía.

Actualmente, en las aplicaciones, generalmente, se utilizan objetos complejos formados por conjuntos de atributos. Cada atributo, a su vez puede ser simple o complejo. Un atributo simple es un número entero, cadena o booleano; mientras que un atributo complejo es una combinación de atributos simples y complejos. Con base en los atributos puede generarse una jerarquía de objetos complejos.

El objetivo de esta propuesta es la transformación de textos, redactados con diferente vocabulario e idioma, hacia nombres de clases, propiedades, valores y unidad de medida; mismos que empleando conjuntos de reglas permiten hacer la integración hacia un modelo orientado a objetos, preservando la integridad en los datos. Es posible que en algunos casos en que la ambigüedad o los errores léxicos no permitan asegurar la integridad de los datos, se presenten las partes de texto con problemas para que sean revisados por parte del usuario.

El resto del artículo se organiza como sigue: La sección II comenta algunos de los trabajos previos que se han desarrollado hasta ahora. La sección III describe las características de las fuentes textuales heterogéneas y la base de datos que se utiliza para la integración de datos. La sección IV los casos de integración considerados. La sección V describe el método de integración propuesto y la descripción de las fuentes de datos de ejemplo y finalmente en la sección VI se indican las conclusiones y trabajo futuro.

II. Trabajos previos

En [1] se introducen metadatos para resolver semánticamente la heterogeneidad en bases de datos federadas. En [2, 16] se hace la integración de un esquema XML hacia otro esquema basado en el modelo relacional. En [9] se hace el mapeo de dos esquemas de base de datos orientadas a objetos mediante la alineación de un esquema local hacia otro esquema global. En [22] se hace la integración entre dos modelos entidad-relación (ER) resolviendo diferentes relaciones semánticas. En [12] se describe la integración de esquemas orientados a objetos generando un esquema común utilizando tesauros. En [7] se utiliza una técnica basada en clasificadores y redes neuronales para hacer la integración de fuentes de XML a XML. En [14] también se hace un mapeo de XML a un esquema minimal también en XML. En [15] se describe un integrador de fuentes de datos en formato XML del dominio de datos bibliográficos. En [8] se describe la integración de esquemas de base de datos intermediado con una ontología.

En [25] se describe la arquitectura Tisimmis que integra datos de diferentes fuentes utilizadas para hacer consultas con un lenguaje común. En [18] se emplea una arquitectura similar a la de [25] complementada con el uso de una ontología para hacer la integración de fuentes XML. En [19] se describe la unificación de esquemas orientados a objetos realizando una conversión al formato XML y luego utilizando una ontología para producir un esquema común. En [22] se desarrolla la similitud semántica entre entidades del modelo relacional usando diferentes relaciones de similitud semántica.

III. Fuentes y base de datos de ejemplo

En este documento se presenta una alternativa a la integración de datos de fuentes textuales hacia una base de datos orientada a objetos para el dominio de equipos de computación. La base de datos integrada tiene el propósito de ser útil para emitir sugerencias de equipo de cómputo a diferentes usuarios con base en el equipo que mejor satisfaga sus requerimientos. En este caso es muy importante que la base de datos contenga los datos de equipos más actualizados sin importar el proveedor, esto dificulta que puedan establecerse acuerdos para solicitar los datos que se incorporan en la base de datos; como consecuencia, se considera más apropiado integrar los datos de las fuentes textuales hacia la base de datos marcando el proveedor, que en este trabajo se indica como un dato anónimo por cuestiones de derechos de autor de sus marcas registradas.

La propuesta de este trabajo es un módulo integrador de datos que toma las fuentes textuales heterogéneas que deben cumplir con una estructura de encabezado, donde se identifica el nombre de la clase principal y un cuerpo de documento donde se esperan las propiedades, valores y unidades de medida. Algunas de las propiedades pueden ser objetos que componen a la clase principal. Se realiza un análisis léxico de los elementos de las fuentes textuales para identificar los elementos que coinciden con la base de datos (figura 1). El integrador utiliza un diccionario y un conjunto de reglas para encontrar los mapeos entre las palabras de un documento de texto con los elementos de la base de datos.

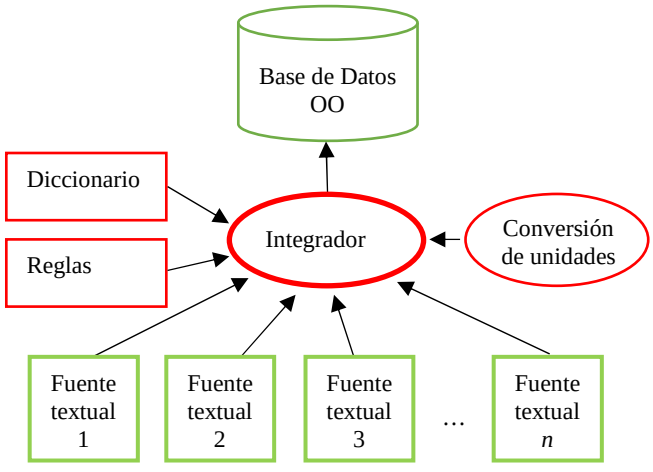


Fig. 1. Integración de fuentes textuales hacia una base de datos

 http://dx.doi.org/10.17562/PB-51-3

ISSN 2395-8618

El modelo semántico de datos [5, 6] tiene la ventaja de utilizar abstracciones cercanas al usuario facilitando la identificación de las principales entidades (clases) con un triángulo y a partir de las que se derivan las subclases identificadas con un círculo. Las clases formadas con la agrupación de propiedades (variables) se indica con un círculo con una cruz, los atributos se indican con un ovalo y cuando son atributos multivaluados se preceden de un circulo con doble cruz. En la figura 2 se utiliza un modelo semántico para mostrar algunas de las entidades principales de la base de datos de ejemplo. A partir del modelo semántico se obtiene el modelo orientado a objetos que como diferencia del anterior las clases y superclases utilizan el mismo símbolo con lo cual se destacan las relaciones de herencia, composición, agregación y asociación. En la figura 3 se indican las clases que destacan las relaciones de la clase computadora con sus componentes y sus relaciones de herencia.

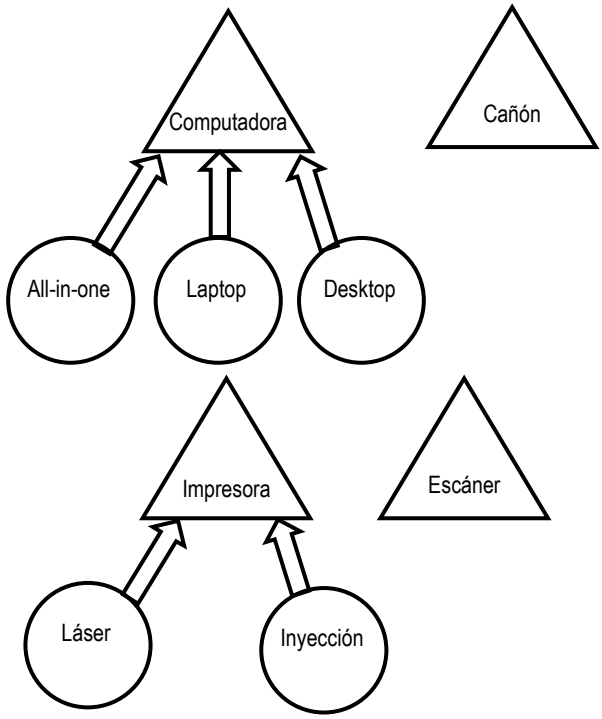


Fig. 2. Modelo semántico de la base de datos de ejemplo

La integración de datos hacia el esquema orientado a objetos utiliza fuentes textuales de equipo de cómputo (figura 4). Se utiliza la convención que cada fuente textual tiene un encabezado que se mapea hacia el nombre de una clase. El resto del documento está formado por una lista de datos que se procesan con un analizador léxico para separar las propiedades, valores y unidades de medida de cada uno. Algunas propiedades pueden ser multivaluadas u objetos que componen al objeto principal.

En los textos se aplican algunas adecuaciones de los símbolos previo al análisis léxico, entre estas está el cambio de los signos ® y ™ a la notación (R) y (TM) respectivamente. Los exponentes se cambian al signo ^ y el valor, de esta forma m2 se cambia a m^2, algunas unidades como los dpi de impresión se escriben como 9.600 dpi por lo que deben cambiarse a 9600 dpi, en general a los números se les eliminan las comas separadoras o los puntos cuando se trata de enteros. Algunas veces las unidades como 2GB se escriben sin espacio por lo que se separan 2 GB.

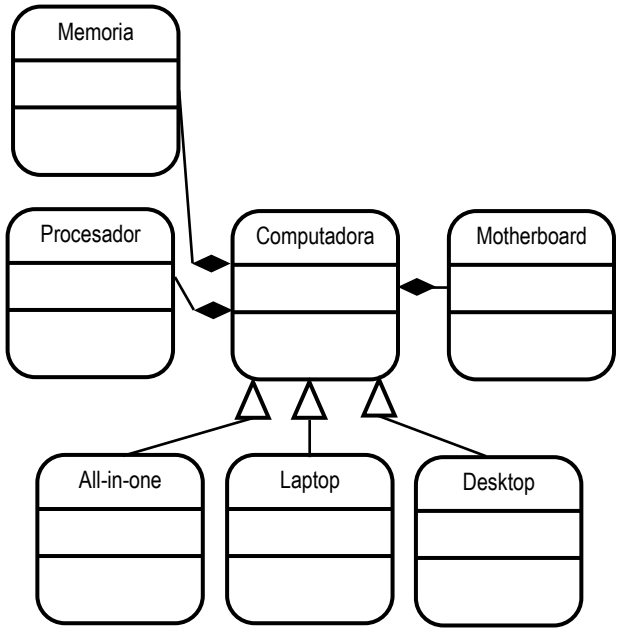


Fig. 3. Algunas clases del modelo orientado a objetos

Las palabras se utilizan respetando las mayúsculas y minúsculas. Después de las adecuaciones de valores y unidades se procesan las unidades léxicas [10] para corregir los errores de ortografía.

El mapeo de palabras a los elementos de la base de datos utiliza un diccionario dividido en secciones para cada uno de los elementos de la base de datos considerados: nombres de clases, identificadores de clases, nombres de propiedad, valores y unidades de medida.

En la figura 5 se muestran las estructuras que componen al diccionario. Las estructuras se llenan en tres etapas, en la etapa inicial se llena la columna de Clase, Identificador, Propiedad, Valor, Unidad con el contenido del esquema de la base de datos orientada a objetos. En la segunda etapa se utilizan recursos léxicos disponibles en la Web para asignar palabras y sinónimos que representan a los elementos de las tablas, registrándolos en la columna Palabras del elemento al que corresponde. La tercera etapa es durante el proceso de integración, cuando se encuentran palabras o símbolos que no se encuentran en el diccionario y que, a criterio del administrador, se añaden. Las palabras e identificadores que no se encuentren en el diccionario se marcarán como desconocidos y no se utilizarán en el mapeo.

ISSN 2395-8618

a) Descripción de una computadora en idioma Inglés

XPS 8700 Special Edition (Computer)
Processor<TAB>4th Generation Intel® Core™ i7
SO<TAB>Windows 8.1
Memory<TAB>16GB
Hard Drive <TAB>2TB
Monitor<TAB>Includes a 23" Dell monitor to maximize your media experience (a $219 value).

b) Descripción de una computadora en idioma Español

XPS 8700 (Computadora)
Procesador<TAB> Cuarta generación del Intel® Core™ i7-4770 (8MB Caché, hasta 3.90 GHz)
SO<TAB>Windows 8.1 Single Language (64Bit) Spanish
Memoria<TAB>16 GB[1] Dos canales SDRAM DDR3 a 1600 MHz
Disco Duro<TAB> SATA de 2TB 7200 RPM (6.0 Gb/s) + Disco Duro de Estado Solido (SSD) de 32GB con Tecnología de Respuesta Inteligente (SRT)
Tarjeta de video<TAB>AMD Radeon™ HD R9 270 2GB GDDR5
garantía Estándar<TAB> 1 Año de, con Servicio en el sitio al siguiente día laborable
Unidad combo<TAB>DVD-RW o Blu-ray Disc
Chipset<TAB>Chipset Intel® Z87 Express

Fig. 4. Ejemplo de fuentes textuales obtenidas de la Web

IV. Integración de fuentes heterogéneas de datos

La integración de las fuentes heterogéneas en este documento se divide en integración hacia el esquema y hacia el contenido.

A. Casos de mapeo del esquema y contenido

La identificación de las clases o subclases se hace utilizando los atributos, la clase se asigna con base en la mayor similitud entre propiedades.

Si se detecta un documento que tiene un nombre de clase y atributos que no mapean con alguna clase existente, se presenta al usuario con sus propiedades para que se considere su integración como una clase o subclase nueva en el modelo de datos.

La integración de datos se hace mediante el mapeo de palabras hacia los elementos del esquema de la base de datos. El encabezado de los documentos textuales se utiliza para encontrar el nombre de la clase y opcionalmente el tipo de equipo. El resto de los elementos se utilizan para encontrar los nombres de las propiedades, los valores y cuando se encuentran las unidades de medida. La integración se hace primero hacia el esquema y luego hacia el contenido. El mapeo del esquema se desarrolla en cuatro casos caracterizados mediante la notación siguiente: a) cuando se puede encontrar un mapeo hacia una clase usando las palabras del texto se indica "$\exists$ clase", en caso contrario se indica "$\neg\exists$ clase"; b) Cuando se encuentra un mapeo hacia las propiedades de alguna clase, no necesariamente de la clase de a), se indica "$\exists$ propiedades", en caso contrario se indica "$\neg\exists$ ropiedades", esto también se aplica cuando al menos una propiedad es diferente.

En los casos se tiene un antecedente que se debe cumplir para que se considere aplicable y un consecuente que indica las acciones que deben aplicarse como resultado de cada caso. Los casos aplicables a la integración del esquema se indican en la tabla I. En forma similar como se aplica en el área de sistemas expertos, se deja al experto humano la decisión de modificar el esquema y el contenido de la base de datos por lo que las reglas se utilizan para proponer cambios en el esquema.

El caso A se presenta cuando no se encuentra una clase en el esquema indicada por las palabras del encabezado de la fuente y tampoco existe una clase que tenga las propiedades encontradas con los mapeos de las palabras del contenido del texto. En este caso se hace el mapeo de valores y unidades de medida para completar una clase con su contenido y se le propone al usuario para darla de alta en el esquema y en la base de datos.

TABLA I
CASOS DE INTEGRACIÓN DEL ESQUEMA

CASO	ANTECEDENTE	CONSECUENTE
A	Si $\neg\exists$ clase $\wedge$ $\neg\exists$ propiedades	Crear una clase nueva con las propiedades detectadas
B	Si $\exists$ clase $\wedge$ $\neg\exists$ propiedades	Crear una clase heredando de una superclase común
C	Si $\neg\exists$ clase $\wedge$ $\neg\exists$ propiedades	Verificar si la clase es sinónimo de una existente o crear otra
D	Si $\exists$ clase $\wedge$ $\exists$ propiedades	Realizar el mapeo de valores y unidades de medida a la clase

En el caso B se encuentra coincidencia en el nombre de la clase pero no de las propiedades de una clase, por lo que se procede a mapear los valores y unidades de medida y se le indica al usuario que revise la clase y en dado caso se tendrá que crear una superclase que agrupe a la clase o clases que tienen propiedades similares con la que se está mapeando.

El caso C se presenta cuando existe coincidencia completa de las propiedades de una clase pero el nombre es diferente. Esta situación se puede presentar cuando algunos dispositivos similares como el caso de las laptop con algunas tabletas pero deben registrarse en clases diferentes. En otros casos similares puede tratarse de un sinónimo como desktop y PC-escritorio.

El caso D se presenta cuando existe coincidencia en el nombre de la clase y sus propiedades, por lo que se requiere proceder a verificar si los valores y unidades de medida existen como algún objeto entre los existentes. Una de las

ISSN 2395-8618

contribuciones en este documento es utilizar las unidades de medida para utilizar funciones de equivalencia y que permiten detectar que dos objetos son iguales después de aplicar las funciones de equivalencia, por ejemplo si dos equipos con propiedades-valor similares tiene un costo de USD$1,000 y otro con costo de MXP$ 13,000 al aplicar una conversión de unidades se encuentra que son iguales. Este caso se divide en cuatro sub-casos como se indica en la tabla II, estos se aplican a una clase con sus propiedades y los objetos de la misma representados por sus valores y unidades de medida.

TABLA II
SUB-CASOS DE INTEGRACIÓN DEL CONTENIDO

SUBCASO	ANTECEDENTE	CONSECUENTE
D.1	Si valores = ∧ unidades =	Objeto existente, se omite
D.2	Si valores = ∧ unidades ≠	Se igualan las unidades y se añade como un objeto nuevo
D.3	Si valores ≠ ∧ unidades =	Se añade como un objeto nuevo
D.4	Si valores ≠ ∧ unidades ≠	Se igualan las unidades y se prueba si es D.1 o D.3

El subcaso D.1 ocurre cuando los valores mapeados de una fuente textual coinciden con los de un objeto de la clase detectada en todas sus propiedades y lo mismo ocurre con las unidades de medida, por lo tanto se le notifica al usuario que se tiene un objeto duplicado y que debe omitirse.

El subcaso D.2 se presenta cuando los valores mapeados coinciden con un objeto existente pero hay diferencias en sus unidades de medida, esto hace necesario que se apliquen funciones de equivalencia que generalmente producen cambios en los valores por lo tanto se vuelve a verificar si es un objeto diferente y se añade a los existentes, en caso que con la aplicación de las funciones de equivalencia se obtenga un objeto existente se procede como en D.1 reportando que es un objeto duplicado.

El subcaso D.3 añade un objeto como consecuencia que se detectaron valores diferentes con las mismas unidades de medida y ningún objeto es igual.

El subcaso D.4 en que los valores y unidades de medida son diferentes se deben aplicar primero las funciones de equivalencia para igualar las unidades de medida transformando los datos al sub-caso D.1 o el D.3.

B. Mapeo de palabras a elementos de base de datos

En el proceso de integración, primero se hacen las adecuaciones a las unidades léxicas y luego se resuelve la identificación de la clase y su identificador junto con las propiedades para determinar el caso que se tiene; si se encuentra el caso D entonces se determina el subcaso de acuerdo con los datos de la fuente textual.

El contenido del diccionario se utiliza para encontrar los elementos del esquema con los que se hace el mapeo del contenido de cada fuente textual.

C. Nombres de clase, propiedades y valores omitidos

En la integración de textos en ocasiones se omiten algunos elementos por lo que se utiliza el contenido del diccionario para inferir la clase o las propiedades que permiten hacer la integración en caso que se omita algún valor.

Clase

nombreClase	palabrasClase
Laptop	Computadora portátil, laptop
Impresora	Printer, printer, impresora

Identificadores

nombreIdent	palabrasIdent
XPS8700	XPS 8700
L-800	L-800

Propiedad

nombreProp	palabrasProp
Procesador	Procesador, CPU, Processor
Memoria	RAM, Memory, Memoria

Valor

nombreVal	palabrasVal
2	2
600x600	600 x 600, 600x600

Unidad

nombreUnid	palabrasUnid
Gb	Gb, GB, gigas, Gigabytes
Tb	Terabytes, teras, TB, Tb

Fig. 5. Estructuras del diccionario para encontrar mapeos

V. PRUEBAS Y RESULTADOS

En esta sección se muestran algunas pruebas y resultados del método de integración propuesto, mostrando la fuente textual, a continuación los elementos mapeados obtenidos y el caso que se aplica.

Sea el texto original:

XPS 8700 Special Edition (Computer)
Processor<TAB>4th Generation Intel® Core™ i7 SO<TAB>Windows 8.1 Memory<TAB>16GB Hard Drive <TAB>2TB Monitor<TAB>Includes a 23" Dell monitor to maximize your media experience (a $219 value).

ISSN 2395-8618

Al efectuar la adecuación se obtiene:

XPS 8700 Special Edition (Computer)
Processor<TAB>4th Generation Intel(R) Core(TM) i7
SO<TAB>Windows 8.1
Memory<TAB>16 GB
Hard Drive <TAB>2 TB
Monitor<TAB>Includes a 23 inches Dell monitor to maximize your media experience (a USD$ 219 value).

Después del mapeo de unidades léxicas se obtienen varias listas de datos que se utilizan para integrarse en la base de datos. En este ejemplo se observa que hace falta el valor de la propiedad velocidad para el caso del procesador y también faltan las propiedades de la clase Motherboard que es una clase que compone a desktop. Se detecta por lo tanto el caso B. Se requiere que previo a su integración, el usuario debe completar los datos faltantes. Las unidades son opcionales porque solamente algunas están presentes con los valores de sus propiedades (figura 6).

Clase	**Instancia**
Desktop	XPS8700

Propiedad	**Valor**	**Unidad**
discoDuro	2	Tb
monitor	tamaño	23 pulgadas

Clase compuesta	**Instancia**
Procesador	core i7

Propiedad	**Valor**	**Unidad**
marca	Intel	
tipo	Core i7	
generacion	4	
velocidad	(INDEFINIDO)	Ghz

Clase compuesta	**Instancia**
Memoria	RAM

Propiedad	**Valor**	**Unidad**
tamaño	16	Gb

Clase compuesta	**Instancia**
Motherboard	(INDEFINIDO)

Propiedad	**Valor**	**Unidad**
fabricante	(INDEFINIDO)	
modelo	(INDEFINIDO)	

Fig. 6. Elementos del ejemplo que se integran en la base de datos

VI. Conclusiones

Se ha presentado una alternativa de integración de datos textuales de fuentes heterogéneas mediante la extracción de los nombres de clases, identificadores, propiedades, valores y unidades de medida. El método se divide en integración del esquema y de contenido. En los casos del esquema una opción es proponer la creación de clases nuevas o adicionar propiedades en alguna clase existente. En la integración de contenido con el apoyo de las funciones de conversión es posible detectar objetos repetidos.

Este método agiliza el proceso de integración cuando se carece de un esquema de datos en los datos fuente, por ejemplo, cuando se aprovechan datos existentes en la Web en forma pública y que se requieren almacenar en un repositorio de datos estructurados.

En la propuesta aún se requiere de la intervención del usuario en la decisión final de creación o actualización del esquema.

El trabajo futuro es aprovechar las estructuras propuestas para dar sugerencias a usuarios que proporcionan información incompleta o con errores.

Agradecimientos

Benina Velázquez Ordoñez y Jesús Manuel Olivares Ceja agradecen el apoyo y comentarios de la Dra. Blanca Lidia Miranda Valencia.

References

[1] G. Aslan and D. McLeod, "Semantic heterogeneity resolution in federated databases by metadata implantation and stepwise evolution," *The VLDB Journal*, vol. 8, no. 2, pp. 120–132, Oct. 1999.

[2] M. Atay, et al., "Efficient schema-based XML-to-Relational data mapping," *Information Systems*, vol. 32, no. 3, pp. 458–476, May 2007.

[3] G. Davies and L. Ekenberg, "Model correspondence as a basis for schema domination," *Knowledge-Based Systems*, vol. 23, no. 7, pp. 693–703, Oct. 2010.

[4] R. C. Goldstein and V. C. Store, "Data abstractions: Why and how?," *Data & Knowledge Engineering*, vol. 29, no. 3, pp. 293–311, Mar. 1999.

[5] R. Hull and R. King, "Semantic database modeling: survey, applications, and research issues," *ACM Computing Surveys*, vol. 19, no. 3, pp. 201–260, Sept. 1987.

[6] R. Hull, "Managing Semantic Heterogeneity in Databases: A Theoretical Perspective," *Proc. ACM Symposium on Principles of Database Systems (PODS'97)*, pp. 51–61, 1997.

[7] B. Jeong, D. Lee, H. Cho and J. Lee, "A novel method for measuring semantic similarity for XML schema matching," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1651–1658, Apr. 2008.

[8] J. Kohler, et al., "Logical and Semantic Database Integration," *Proc. 1st IEEE International Symposium on Bioinformatics and Biomedical Engineering (BIBE '00)*, pp. 77–80, 2000.

[9] E.-P. Lim and R. H. L. Chiang, "Accommodating instance heterogeneities in database integration," *Decision Support Systems*, vol. 38, no. 2, pp. 213–231, Nov. 2004.

[10] C. D. Manning, P. Raghavan and H. Schütze, *An Introduction to Information Retrieval*, Cambridge, MA: Cambridge University Press, 2009.

[11] S. Madria, K. Passi and S. Bhowmick, "An XML Schema integration and query mechanism system," *Data & Knowledge Engineering*, vol. 65, no. 2, pp. 266–303, May 2008.

[12] I. Mirbel, "Semantic integration of conceptual schemas," *Data & Knowledge Engineering*, vol. 21, no. 2, pp. 183–195, Jan. 1997.

ISSN 2395-8618

[13] M. L. Nguyen and A. Shimazu, "A semi supervised learning model for mapping sentences to logical forms with ambiguous supervision," *Data & Knowledge Engineering*, vol. 90, no. 1, pp. 1–12, Mar. 2014.

[14] H.-Q. Nguyen, et al., "Double-layered schema integration of heterogeneous XML sources," *The Journal of Systems and Software*, vol. 84, no. 1, pp. 63–76, Jan. 2011.

[15] H. Nottelmann and U. Straccia, "Information retrieval and machine learning for probabilistic schema matching," *Information Processing and Management*, vol. 43, no. 3, pp. 552–576, May 2007.

[16] G. Della Penna, et al., "Interoperability mapping from XML schemas to ER diagrams," *Data & Knowledge Engineering*, vol. 59, no. 1, pp. 166–188, Oct. 2006.

[17] G. Pirró, "A semantic similarity metric combining features and intrinsic information content," *Data & Knowledge Engineering.* Vol. 68, no. 11, pp. 1289–1308, Nov. 2009.

[18] J.-L. Seng and I.L. Kong, "A schema and ontology-aided intelligent information integration," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10538–10550, Sept. 2009.

[19] R. dos Santos Mello, S. Castano and C. A. Heuser, "A method for the unification of XML schemata," *Information and Software Technology*, vol. 44, no. 4, pp. 241–249, Mar. 2002.

[20] J. M. Smith and D. C. P. Smith, "Database Abstractions: Aggregation and Generalization," *ACM Transactions on Database Systems*, vol. 2, no. 2, pp. 105–133, June 1977.

[21] Victor Vianu. "A Web Odyssey: from Codd to XML," *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems (PODS '01)*, 1–15, 2001.

[22] William Wei Song, Paul Johannesson, Janis A. Bubenko Jr. "Semantic similarity relations and computation in schema integration," *Data & Knowledge Engineering*, Vol. 19, no. 1, pp. 65–97, May 1996.

[23] P. Coad and E. Yourdon, *Object-Oriented Design*, Yourdon Press, New Jersey, 1991.

[24] G. Booch. *Object Oriented Design with Applications*, New York: Benjamin/Cummings, 1994.

[25] H. Garcia-Molina, et al. "The TSIMMIS project: integration of heterogeneous information sources," *Journal of Intelligent Information Systems*, Vol. 8 no. 2, pp. 117–132, 1997.

[26] M. Minsky, "A Framework for Representing Knowledge," MIT-AI Laboratory Memo 306, June, 1974

ISSN 2395-8618

Mobile ACORoute—Route Recommendation Based on Communication by Pheromones

Carla S. G. Pires, Marilton S. de Aguiar, and Paulo R. Ferreira

***Abstract*—Urban mobility problems affects the vast majority of cities nowadays. Thus, systems that provide real time information to assist in planning routes and choosing the most appropriate paths are essential to make transport more effective. As an alternative solution to problems related to mobility in cities, there are the so-called Intelligent Transportation Systems (ITS) which include the Route Recommendation Systems (RRS) and methodologies for congestion prediction that combine Information and Communication Technology (ICT) with Artificial Intelligence (AI) technology to improve the quality of transport systems. In this context, this work proposes the use of pheromone-based communication for building an ITS that offers information about real time traffic flow, taking into account the mobility of vehicles and passengers and the traffic dynamics. The general goal is to provide an Android solution able to suggest users routes calculated by the hybrid algorithm between A* and pheromone mechanism. The idea is to avoid areas of heavy traffic congestion.**

***Index Terms*—Route recommendation systems, intelligent transportation systems, pheromone-based communication.**

I. Introduction

In the past decades, the traffic in medium and large cities, as well as the incovenience caused directly or indirectly by it, cause increasing mobility problems. The Intelligent Transportation Systems (ITS) are shown as an alternative to improve mobility within cities through the application of Information and Communication Technology (ICT) to support the existing traffic infrastructure and improve the quality of transport systems [1].

A wide variety of ITS tools have played important roles in the effectiveness of transport. These systems provide information related to traffic, influencing in various aspects of transport in relation to urban mobility. Most of these ITS tools uses static information aided by the traffic infrastructure integrated technologies [2]. This article proposes to use information from mobile devices to dynamically determine the best path for the driver seeking to avoid traffic congestion.

To this end, this paper proposes an approach for the calculation of the trajectory based on the use of pheromone dynamics. Other approaches to congestion prediction using Swarm Intelligence have been propose [3], [4], [5], [6]. Previous works differ in how pheromones influence the calculation process and do not consider some aspects of their practical application.

The paper is arranged as follows: in Section II the theoretical background are briefly introduced; Section III discusses the conceptual and technological foundations of the work; Section IV discusses about the development of ACORoute and shows how the proposed approach was implemented; V presents the results obtained from the application of the proposed approach in a simulation environment and, finally, Section VI presents the conclusions of the work.

Manuscript received on December 19, 2014, accepted for publication on April 7, 2015, published on June 15, 2015.

The authors are with Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas (UFPEL), Pelotas, RS, Brazil (e-mail: carlasm-pires@gmail.com, marilton@inf.ufpel.edu.br, paulo.ferreira.jr@gmail.com).

II. Related Work

A. Intelligent Transportation Systems (ITS)

The term ITS emerged in USA in the late 1980s with a movement aiming to make transport safer, more effective and reliable. The ideia of the ITS is to support the existing traffic infrastructure without the need to change it [7]. Thus, a number of tools have been developed in order to assist drivers to stay informed about the traffic conditions and the relevant aspects of the dynamics of its flow. Areas of Computing as Artificial Intelligence (AI), Computer Vision, Pattern Recognition, Machine Learning, Data Mining, and Intelligent Control have been intensively explored in this process [8].

The tools developed in ITS enable users to get several kinds of information related to traffic. This information range from current location, alternative routes, road conditions, even weather forecasts along the route, providing greater power of decision on the actions and choices of individuals.

In a simplified way one can say that the ITS aim at the optimization of existing transport systems by making use of a wide range of tools that combine technologies and improvements in information systems, communications, sensors and advanced mathematical methods. The objective is to obtain roads, vehicles and more "intelligent" users, with an attempt to facilitate the flow of traffic, and solve a variety of transportation problems of our days, such as congestion, safety and environmental problems [9].

B. Swarm Intelligence

An interesting approach is the use of techniques of swarm intelligence, as the organization of bee colonies and allocation of tasks among insect societies. The dynamics of insect society

ISSN 2395-8618

are the result of different actions between the population and the environment. This interaction between agents creates a system of communication that contributes to the formation of "Collective Intelligence" assigned to the insect society. This technique arose from observations and studies of the behavior of living beings usually searching for food [10].

Swarm intelligence or collective intelligence has been successfully applied in dynamic optimization problems in various fields, such as the traveling salesman problem, quadratic assignment problems, load balancing and vehicle routing problems. These are just some examples where swarm intelligence is applied [11].

Examples of applications of this nature can be seen in Ando et al. (2003), which introduced systems of bees as a new approach in the area of intelligent swarms applied to problems of transportation engineering. They developed a new heuristic for the traveling salesman problem by defining an artificial environment of bees; swarm intelligence technique inspired by the behavior of ant colonies are also used to optimize the timing of traffic lights [12]; and, the use of the swarm intelligence inspired by the behavior of bees society for task allocation using clustering, grouping agents by skills, considering societies of bees and how they collect the best nectar from the available sources through simple rules of behavior [13].

The use of intelligent insect swarms has already been applied in problems to find shorter routes between cities (TSP), such as Lucic et al. (2003), who developed a new heuristic for the traveling salesman problem by setting an artificial environment of bees, where each bee is an agent performing activities defined by the model and the communication occurs in the interaction between them [14].

This work has focused on ant colony optimization (ACO) that is inspired by the observation of ant communities and their organization to find food sources [15].

Experiments were conducted to understand this behavior, and showed that ants have the ability to discover the shortest path very quickly. It was observed that only the first ants randomly choose the path, thus those who choose the shorter path arrive faster to the nest. Thus, the probability of choosing the shortest way increases continuously and rapidly all ants start using the shortest path [11].

The ACO is based on this behavior, considering that each ant walking on a trail deposits certain amount of pheromone in it, then the next ants follow the path with a proportional probability to the amount of pheromone present in this path and thus reinforcing the current pheromone [15].

ACO was the first swarm intelligence algorithm to be developed, it is used in the development of this work. It has been effectively applied to solve the traveling salesman problem, as well as several other problems related to Transportation and Traffic Engineering. Kurihara (2013) and Narzt et al. (2010) used the swarm intelligence technique inspired in the behavior of ants society (ACO). They employed this methodology for predicting traffic congestion in a simulated environment, considering agents (sensors) installed at the intersections of routes that manage information of pheromones. Ando et al. (2006) evaluated the application of pheromone communication in real traffic conditions applied to congestion prediction and observed that the method is effective for short-term predictions.

III. Mechanism for Congestion Identification based on communication by pheromones

Technologies for congestion identification are a key element to support Intelligent Transportation Systems. Currently, many methods have been proposed, many of them using collective intelligence, based on the idea that societies of insects perform complex tasks using decentralized communication based on pheromones. In this context, pheromones are considered as a means to provide information [16], [3], [5], [6], [17] among others.

In this paper we propose congestion identification using the technique of communication based on pheromones (ACORoute), which was also proposed by [16], [3] and [5] in their works. Different from the adopted in our work, they use the infrastructure provided by local transport system and/or devices installed in vehicles to collect, process and store data related to traffic. In addition, when calculating the route, ACORoute uses the pheromone information in heuristic of the best route, while the papers presented use a common navigator to calculate, merely use pheromones for congestion prediction. A comparison between relevant works can be seen in Table I.

When dealing with congestion prediction, the application WAZE should be mentioned, which recommends routes avoiding places of heavy flow. However, it only offers the collaboration of other users who interact with the application to report occurrences. Thus, apart from Waze (2013), which also makes congestion prediction, other works have results only in simulated environment to date to validate their technical environments. There are no real applications.

IV. ACORoute

A. Pheromone-based Model

The use of pheromone dynamics for congestion identification was discussed in the development of ACORoute, and it was proposed a model that predicts congestion based in the pheromones communication mechanism. The model used considers vehicles as insects that deposit pheromones.

The operating principle of this strategy occurs with vehicles that mark their path by dropping digital pheromones which are perceived by all the vehicles that travel in the environment. This mechanism is used to allow the calculation of a route that avoids heavy traffic.

The vehicle sends information about geopositioning to the server at every 30 seconds and with this information is created of the pheromone map, that is built as follows: when the georeferencing information is sent by the application it is

 http://dx.doi.org/10.17562/PB-51-4

ISSN 2395-8618

TABLE I
CHARACTERISTICS OF THE METHOD FOR CONGESTION PREDICTION

	Self-organising Narzt (2010)	Pheromone Ando (2006)	Traffic-Congestion Kurihara (2013)	Hybrid ACO Ochiai (2014)	WAZE WAZE (2013)	ACORoute
Type of Transport	Individual	Individual	Individual	Individual	Individual	Individual
Sources of Information	No	Yes	Yes	Yes	Yes	Yes
Traffic Infrastructure	Yes	Yes	Yes	No	No	No
Context Informations	No	No	No	No	No	Yes
Mobile Devices	No	No	No	No	Yes	Yes
User Preferences	No	No	No	Yes	No	Yes
Historical Information	No	No	No	No	Yes	Yes
Real Time	No	No	Yes	No	No	Yes
Route Re-planning	Yes	No	No	No	No	Yes
Algorithm		Pheromone		Dijikstra	A*	A*/Pheromone

received by the WebService, the value of 0.8 [3] is incremented at the referenced node (Latitude and Longitude).

When a given node remains without receiving information from pheromones longer than 30 seconds, the evaporation process starts until the amount of pheromones in the node reaches 0. The value of the decrement to pheromone is 0.3. These values were calibrated through simulation, which confirmed to be the most effective.

B. The Best Route Algorithm

To determine a route between a source node (latitude and longitude) and a destination node (latitude and longitude), the solution presented here uses a variation of the routing heuristic in graphs A*, considering besides the distance, the amount of pheromones in this route. The proposed algorithim prefers routes with lower level of pheromones

The map stored in the database is represented by a directed graph with weights representing distances. We can say that the graph is a logic representation of the map, containing the streets of the city of Pelotas, each section of the street is considered an edge and the points are nodes. The search algorithm A*, that considers the pheromone information to calculate the route, is used to find the best route between two points. The database used is derived from the open database of the Open Street Maps only considering the points inside the area of Pelotas.

C. Android Application

For the development of the application, the Eclipse IDE (Integrated Development Environment) combined with Android SDK (Software Develpoment Kit), that provides an API (Application Programming Interface) required to build and to test applications for Android was used.

Two languages were used for the implementation: XML and Java. XML is used to build the graphical part of the application, where the interaction with the user happens, made through the components of the graphic application, the widgets, such as buttons and textboxes. In the control, the Android-Java language, a standard Java subset, which implements specific API for Android application development was used.

In the Android API Activity and Service concepts that are related to the graphical interface of the application are defined which are responsible for the different screens, operations performed in the background, among them communication with the server.

The implementation project of the application was divided into two stages: i) receipt and transmission of georeferenced data, used to establish the historical base and the construction of the pheromone map, these information are obtained through GPS (Global Positioning System) from the device itself and ii) development of the interface with the user where information such as location and path of the calculated routes will be presented.

A service responsible for connecting the application to the GPS, receiving location information periodically every 30 seconds, which can be set remotely from the WebService was created in the first stage of the work. After receiving the information, Service sends it to the WebService, keeping the historical database and pheromones. The application prototype interface is shown in Figure 1.

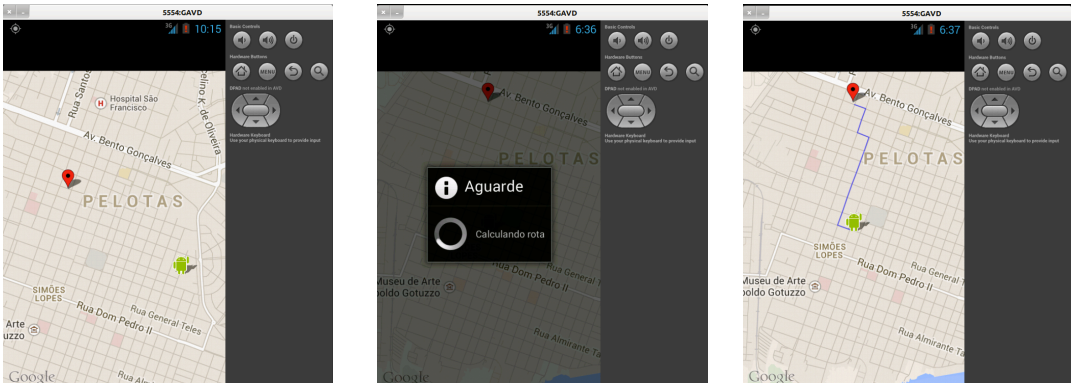
The time to update the user's location was defined in order to optimize the battery consumption of the device without sacrificing the accuracy of the application. When using smaller and more frequent values, it creates several accesses to the GPS which is not considered a good strategy for the substancial increase of battery consumption of the device. In contrast, using higher values can lead to a possible loss of information because it becomes complex to deduce the route taken by the user.

In the second stage of development, two activities were created, the first one, responsible for controlling the transmission of location information and for opening the map; and the second, responsible for displaying the map and all the relevant information, such as location of the user, menus and routes. The map chosen for the application was Google Maps, because it is easy to use and complete, in addition to having its own API for Android programming, which makes the implementation simpler.

V. NUMERICAL EXPERIMENTS AND DISCUSSIONS

After the congestion identification method be defined, it was necessary to evaluate it in terms of precision and,

ISSN 2395-8618



(a) Origin and destination points (b) Calculating the route (c) Route view

Fig. 1. Prototype Interface

consequently, the viability of the method. To that end, simulations comparing the performance of the proposed method with the algorithm A*, standard technique for the route calculation, were made. In a simulation environment developed in Netlogo, 2 (two) cars were inserted, one of them using the methodology that uses pheromones when calculating the route and the other one using pure A* algorithm. In addition, experiments were performed for different amounts of cars of random behaviour for the evaluation in distinct congestion situations (as shown in Table II). The parameterization was made according to Masutani et al. [4] that also used ACO for congestion prediction, where values of 0.7 and 0.8 for evaporation and deposit respectively were pointed as good.

TABLE II
EFFICIENCY OF THE PHEROMONE MODEL

	Number of cars	With pheromones	Without pheromones
Average time (*ticks*)	200	**368.37**	426.91
	400	**488.48**	542.52
	600	**491.14**	572.81

After 6000 simulations of the model, it was possible to observe that the average time (in *ticks* of simulation) of completion of route, when considering the weight of the pheromone in the cost of the Heuristic A* is lower if compared to the route without considering the pheromone for all congestion situations.

It can be observed that the number of cars used in the simulations influence the average times obtained for calculating the best route. As the number of cars increases, the average time for the calculation also increases, thus, the longer times were obtained when 600 cars were used. The average times presented in Table II shows that in all simulations the time of the route using pheromones were lower in relation to the simulations not using pheromones. It could be observed in the simulations that situations where the congestion rate is high, to seek for alternative routes may lead to other congestion situations, which can be worse than waiting for the normalization of the flow. Thus, it was verified that the proposed method is effective.

Subsequent experiments for calibration of the parameters were made, such as increment and decrement of pheromones, weight of the pheromones, weight of the distance to cover all the way, evaporation of pheromone, pheromones limit. Also, experiments were performed to verify the minimum percentage of the number of cars depositing pheromones that makes the model more effective.

The methodology of the tests was systematized in order to compose the best set of parameters for the model. The systematization and planning of experiments was necessary to make possible the evaluation of the behavior of the evaluated parameters. In this sense, an analysis of variance was performed using for this, the full factorial design.

In each experiment were analyzed 2 factors at 3 levels, so we will have a plan where all parameters are studied in three levels thus have a factorial design 2^3 with a planning matrix with 9 experiments.

For planning matrices presented in figure 2 the following response surfaces were obtained.

Considering the data of the correlation between weight of pheromone and weight of distance, was obtained on analysis of variance in Table 4 , where we can observe that the values of p are smaller than 0.05, this means that factors are considered significant, ie, both factors are relevant in the media presented.

Observing the results presented in Table III, it is possible to identify that the best results were obtained with the parameterization Weight of the Pheromone 0.7, Weight of Distance 0.5 and Evaporation 0.3. Also, it can be observed that the highest average times were obtained when the decrements used for Evaporation were higher. This behavior can be attributed to the fact that the algorithm considers without congestion a stretch that is still crowded, and calculate a route that leads the vehicle to get congested rather than calculate a route that avoids a particular stretch. Thus, this parameterization was used in the other simulations.

 http://dx.doi.org/10.17562/PB-51-4

ISSN 2395-8618

2 factors	-1	0	+1
Weight pheromone	0,3	0,5	0,7
Weight distance	0,3	0,5	0,7

Experiments		WP	WD	Time
1	-1, -1	0,3	0,3	445,32
2	-1, 0	0,3	0,5	372,59
3	-1, +1	0,3	0,7	405,22
4	0,-1	0,5	0,3	427,59
5	0,+1	0,5	0,7	419,54
6	+1, -1	0,7	0,3	404,65
7	+1, 0	0,7	0,5	334,45
8	+1, +1	0,7	0,7	423,95
9	0,0	0,5	0,5	398,35

(a)

2 factors	-1	0	+1
Weight pheromone	0,3	0,5	0,7
Evaporation	0,1	0,3	0,5

Experiments			WP	E	Time
1	-1	-1	0,3	0,1	473,2
2	-1	0	0,3	0,3	421,95
3	-1	+1	0,3	0,5	433,72
4	0 -	1	0,5	0,1	425,52
5	0	+1	0,5	0,5	394,72
6	+1	-1	0,7	0,1	399,91
7	+1	0	0,7	0,3	375,1
8	+1	+1	0,7	0,5	404,5
9	0	0	0,5	0,3	455,4

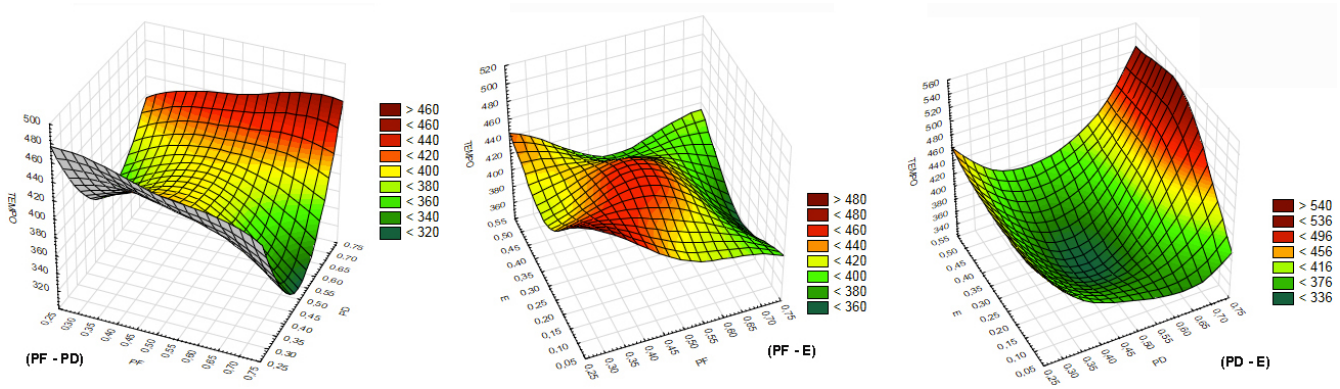
(b)

2 factors	-1	0	+1
Evaporation	0,1	0,3	0,5
Weight distance	0,3	0,5	0,7

Experiments		WD	E	Time
1	-1, -1	0,3	0,1	417,55
2	-1, 0	0,3	0,3	404,65
3	-1, +1	0,3	0,5	420,2
4	0,-1	0,5	0,1	366,00
5	0,+1	0,5	0,5	385,86
6	+1, -1	0,7	0,1	391,59
7	+1, 0	0,7	0,3	482,27
8	+1, +1	0,7	0,5	478,85
9	0,0	0,5	0,3	334,45

(c)

Fig. 2. Planning matrix: (a) Weight pheromone x Weight distance; (b) Weight pheromone x Evaporation; (c) Weight distance x Evaporation



(a) (b) (c)

Fig. 3. Response surface graph: (a) Weight pheromone × Weight distance; (b) Weight pheromone × Evaporation; (c) Weight distance × Evaporation

T-test for Dependent Samples (pfpd)
Marked differences are significant at p < ,05000

Variable	Mean	Std.Dv.	N	Diff.	Std.Dv. Diff.	t	df	p	Confidence -95,000%	Confidence +95,000%
PF	0,5000	0,17321								
TEMPO	403,5178	33,08862	9	-403,018	33,13444	-36,4893	8	0,000000	377,5484	428,4872
PD	0,5000	0,17321								
TEMPO	403,5178	33,08862	9	-403,018	33,11087	-36,5153	8	0,000000	377,5665	428,4690
TEMPO	403,5178	33,08862								
PF	0,5000	0,17321	9	403,018	33,13444	36,4893	8	0,000000	377,5484	428,4872
TEMPO	403,5178	33,08862								
PD	0,5000	0,17321	9	403,018	33,11087	36,5153	8	0,000000	377,5665	428,4690

Fig. 4. Student t table

Experiments with the inclusion of route recalculation were made once the parameterization was defined. The recalculation occurs in situations where the vehicle remains motionless for a certain period. These tests aimed to identify the effectiveness of including recalculation and also calibrate the parameter called Recalculation Limit, which determines the time limit for the route recalculation. The results are displayed in Table IV.

Observing Table IV can be verified that the time limit $1,0$ Ticks to recalculate the route presented better results.

VI. Conclusions and Further Works

This work has as main contribution the development of a methodology to predict congestion using pheromone-based communication. The results obtained in validation tests of the methodology through simulation and implementation of mobile device (ACORoute) show its viability at gaining time with its use.

The results obtained in the simulations confirm the applicability and effectiveness of the proposed method in predicting congestion, obtaining better results in time when compared to the non-use of the technique.

In general, it can be concluded that:

– The average times using Pheromones are better in all cases;
– With the increase of the number of cars, the averages increase, yet the results were better using pheromones;

ISSN 2395-8618

TABLE III
ANALYSIS OF PARAMETERIZATION WITH 600 CARS ON THE ENVIRONMENT, 100% WITH PHEROMONES

Parameterization for 6000 performances			Average time (Ticks)	
Weight pheromone	Weight distance	Evapo-ration	With pheromone	Without pheromone
0,5	0,5	0,3	**455,4**±14,41	398,35±4,64
0,5	0,5	0,1	**425,52**±8,31	418,09±4,14
0,5	0,5	0,5	**394,72**±4,65	408,86±6,23
0,5	0,3	0,3	**427,59**±3,54	480,36±5,45
0,5	0,7	0,3	**419,54**±4,67	460,09±4,41
0,5	0,5	0,3	**398,35**±9,24	455,4±7,86
0,7	0,5	0,1	**366,00**±2,44	399,91±4,34
0,7	0,5	0,5	**385,86**±3,18	404,05±6,23
0,7	0,7	0,1	**391,59**±7,45	399,27±5,87
0,7	**0,5**	**0,3**	**334,45±2,14**	**375,10±2,87**
0,7	0,3	0,3	**404,65**±4,67	440,35±6,14
0,7	0,3	0,1	**417,55**±2,98	430,3±4,32
0,3	0,3	0,3	**445,32**±3,23	479,18±5,90
0,3	0,5	0,3	**372,59**±2,05	421,95±2,97
0,3	0,7	0,3	**405,22**±1,65	459,82±4,03
0,3	0,5	0,1	**473,2**±4,26	445,25±8,90
0,3	0,5	0,3	**421,95**±3,89	372,59±7,98
0,3	0,5	0,5	**433,72**±6,75	392,72±5,73

TABLE IV
ANALYSIS OF DECISION PARAMETERS FOR THE ROUTE RECALCULATION: 600 CARS IN THE ENVIRONMENT

	Average time (*Ticks*)	
Recalculation limit (*Ticks*)	With pheromones	Without pheromones
1.5	**369.87**±3.52	393.15±2.63
1.0	**362.15±4.58**	**399.54±2.41**
0.5	**379.28**±5.96	381.90±2.06

– With the introduction of the recalculation, the average was better;
– The application is in operation and it calculates the route avoiding locations with congestions;
– The application still needs to be optimized in some aspects.

Among the aspects raised to continue the work, the need for some improvement in the application stands out:

– Minimize the battery consumption;
– Testing in a real environment;
– Determine the pattern of behaviour of routes based on historical data, for situations where there is no information of pheromones;
– Implement the use of historical information in route calculation;
– Optimization of route calculation, improving the response time;
– Implement ACORoute for other platforms;
– Make download of the application available to the academic community.

REFERENCES

[1] J. Wahle, O. Annen, C. Schuster, L. Neubert, and M. Schreckenberg, "A dynamic route guidance system based on real traffic data," *European Journal of Operational Research*, vol. 131, no. 2, pp. 302–308, 2001, artificial Intelligence on Transportation Systems and Science. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221700001302

[2] B. Ferris, K. Watkins, and A. Borning, "Location-aware tools for improving public transit usability," *Pervasive Computing, IEEE*, vol. 9, no. 1, pp. 13–19, January–March 2010.

[3] S. Kurihara, "Traffic-congestion forecasting algorithm based on pheromone communication model," *Ant Colony Optimization – Techniques and Applications*, vol. 104, pp. 167–175, 2013. [Online]. Available: http://http://www.academia.edu/2613986/ANT_COLONY_OPTIMIZATION_-_TECHNIQUES_AND_APPLICATIONS

[4] O. Masutani, Y. Ando, H. Sasaki, H. Iwasaki, Y. Fukazawa, and S. Honiden, "Pheromone model: Application to traffic congestion prediction," in *Engineering Self-Organising Systems*, ser. Lecture Notes in Computer Science, S. Brueckner, G. Marzo Serugendo, D. Hales, and F. Zambonelli, Eds., vol. 3910. Springer Berlin Heidelberg, 2006, pp. 182–196. [Online]. Available: http://dx.doi.org/10.1007/11734697_14

[5] W. Narzt, U. Wilflingseder, G. Pomberger, D. Kolb, and H. Hörtner, "Self-organising congestion evasion strategies using ant-based pheromones," *Iet Intelligent Transport Systems*, vol. 4, 2010.

[6] J. Ochiai and H. Kanoh, "Hybrid ant colony optimization for real-world delivery problems based on real time and predicted traffic in wide area road network," *Fourth International conference on Computer Science and Information Technology – CCSIT 2014*, vol. 4, no. 2, February 2014. [Online]. Available: http://airccse.org/V4N19.html

[7] J. L. Adler and V. J. Blue, "Toward the design of intelligent traveler information systems," *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 3, pp. 157–172, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X98000126

[8] I. Steinmacher, V. Vieira, A. C. Salgado, P. Tedesco, V. Times, C. Ferraz, E. Huzita, and A. P. Chaves, "The UbiBus project: Using context and ubiquitous computing to build advanced public transportation systems to support bus passengers," *VIII Simpósio Brasileiro de Sistemas de Informação*, 2012. [Online]. Available: http://www.cin.ufpe.br/~ubibus/publications.html

[9] A. Tito, F. Borgiani, R. dos Santos, P. Tedesco, and A. Salgado, "Contextual information in user information systems in public transportation: A systematic review," in *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, sept. 2012, pp. 361–366.

[10] C. Blum and D. Merkle, *Swarm Intelligence: Introduction and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[11] D. Teodorovic, "Swarm intelligence systems for transportation engineering: Principles and applications," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 6, pp. 651–667, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X08000272

[12] R. Hoar, J. Penner, and C. Jacob, "Evolutionary swarm traffic: if ant roads had traffic lights," in *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02*, vol. 2, 2002, pp. 1910–1915.

[13] D. S. dos Santos and A. L. Bazzan, "Distributed clustering for group formation and task allocation in multiagent systems: A swarm intelligence approach," *Applied Soft Computing*, vol. 12, no. 8, pp. 2123–2131, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494612001044

[14] P. Lucic and D. Teodorovic, "Transportation modeling: An artificial life approach," in *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, ser. ICTAI'02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 216–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850952.853815

[15] H. J. Barbosa, Ed., *Ant Colony Optimization – Techniques and Applications*. Croatia: InTech Chapters published, 2013.

[16] Y. Ando, Y. Fukazawa, O. Masutani, H. Iwasaki, and S. Honiden, "Performance of pheromone model for predicting traffic congestion," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS'06. New York, NY, USA: ACM, 2006, pp. 73–80. [Online]. Available: http://doi.acm.org/10.1145/1160633.1160642

[17] "Waze mobile," https://www.waze.com/wiki/How_Waze_calculates_routes, Waze Ltd., 2013, access date: 03 jan. 2014.

ISSN 2395-8618

Traffic Accidents Forecasting using Singular Value Decomposition and an Autoregressive Neural Network Based on PSO

Lida Barba and Nibaldo Rodríguez

***Abstract*—In this paper, we propose a strategy to improve the forecasting of traffic accidents in Concepción, Chile. The forecasting strategy consists of four stages: embedding, decomposition, estimation and recomposition. At the first stage, the Hankel matrix is used to embed the original time series. At the second stage, the Singular Value Decomposition (SVD) technique is applied. SVD extracts the singular values and the singular vectors, which are used to obtain the components of low and high frequency. At the third stage, the estimation is implemented with an Autoregressive Neural Network (ANN) based on Particle Swarm Optimization (PSO). The final stage is recomposition, where the forecasted value is obtained. The results are compared with the values given by the conventional forecasting process. Our strategy shows high accuracy and is superior to the conventional process.**

***Index Terms*—Autoregressive neural network, particle swarm optimization, singular value decomposition.**

I. Introduction

FORECASTING of time series with neural networks has been widely implemented due to its capability of approximation and universal generalization [1], [2] in diverse areas of knowledge [3], [4]. Conventionally, the neural networks show difference and improvement through the adequate selection of transfer and activation functions [5], [6], the variation in the input dimension and the time delay [7], also changing the number of hidden nodes [8], others researchers propose modifications in the learning algorithms [9], is common also the use of explanatory variables [10], there are works that implement hybrid solutions reaching good performance [11], [12], whereas the decomposition, disaggregation or aggregation of the time series before the forecasting have demonstrated to be an effective strategy [13], [14]. The combination ANN-PSO has improved the forecasting over some classical algorithms [15], [16], [17]

Based on these arguments, in this work we propose a strategy of improving traffic accidents forecasting based on the decomposition of a time series in components of low and high frequency from the singular values of the Hankel matrix. The strategy is applied in four stages, embedding with Hankel matrix, decomposing with SVD, estimation with ANN-PSO, and recomposing with simple addition. The time series are the traffic accidents of Concepción - Chile, sinister number and injured, from year 2000 to 2012, with weekly sampling.

The paper is structured as follows. Section II describes the time series forecasting strategy. Section III presents the forecasting accuracy metrics. Section IV presents the results and discussion. Section V gives conclusions.

Manuscript received on December 24, 2014, accepted for publication on April 20, 2015, published on June 15, 2015.

Lida Barba is with the Pontificia Universidad Católica de Valparaíso, Chile and Universidad Nacional de Chimborazo, Ecuador (e-mail: lbarba@unach.edu.ec).

Nibaldo Rodríguez is with Pontificia Universidad Católica de Valparaíso, Chile (e-mail: nibaldo.rodriguez@ucv.cl).

II. Time series forecasting strategy

Our forecasting strategy is presented in Figure 1. It consists of four stages: embedding, decomposition, estimation, and recomposition. Embedding means to map the time series in a Hankel matrix, decomposition is developed with SVD, the singular values are used to extract the components of low and high frequency, the estimation of the found components is based on an ANN based on PSO, and the recomposition is developed with the simple addition of the ANNs outputs.

The original time series is represented with x, H is the Hankel matrix, S, V, and U are the matrix elements obtained with SVD, C_L is the component of low frequency, C_H is the component of high frequency, $\hat{C_L}$, and $\hat{C_H}$ are the estimated components, $\hat{x}$ is the forecasted time series, and er is the error computed between x and $\hat{x}$.

A. Embedding the time series

The time series is embedded in the Hankel matrix, the process is illustrated as follows:

$$H_{M\times L} = \begin{bmatrix} x_1 & x_2 & \dots & x_L \\ x_2 & x_3 & \dots & x_{L+1} \\ \vdots & \vdots & \vdots & \vdots \\ x_M & x_{M+1} & \dots & x_N \end{bmatrix} \quad (1)$$

where H is a matrix of order $M \times L$, $x_1 \dots x_N$, are the original values of the time series, of length N. The value of L is computed as

$$L = N - M + 1. \quad (2)$$

ISSN 2395-8618

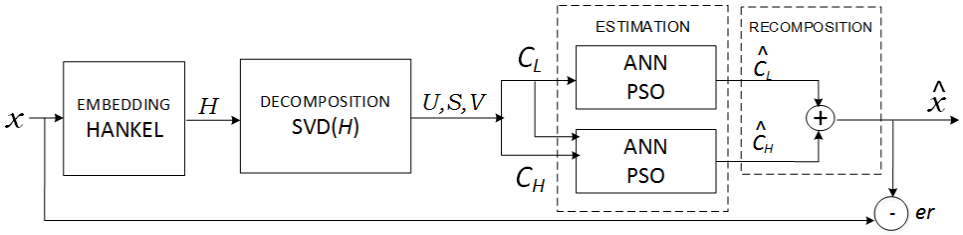


Fig. 1. Time series forecasting strategy

B. Singular Value Decomposition

Let H be an $M \times n$ real matrix, then there exist an $M \times M$ orthogonal matrix U, an $n \times n$ orthogonal matrix V, and and $M \times n$ diagonal matrix S with diagonal entries $s_1 \geq s_2 \geq \ldots \geq s_p$, with $p = min(M,n)$, such that $U^T HV = S$. Moreover, the numbers $s_1, s_2, \ldots, s_p$ are uniquely determined by H [18].

$$H = U \times S \times V^T \tag{3}$$

The extraction of the components is done by means of the singular values s_i, the orthogonal matrix U, and the orthogonal matrix V, for each singular value is obtained one matrix A_i, with $i = 1 \ldots m$:

$$A_i = s(i) \times U(:,i) \times V(:,i)^T \tag{4}$$

Therefore, the matrix A_i contains the $i-th$ component, the extraction process is:

$$C_i = \left[\begin{array}{cc} A_i(1,:) & A_i(2,n:m)^T \end{array} \right] \tag{5}$$

where C_i is the $i-th$ component, the elements of C_i are located in the first row and last column of A_i.

The optimal number of components (M) is given by the maximum peak of differential energy ΔE of each pair of sequential components, and it computation is

$$\Delta E_i = E_i - E_{i+1} \tag{6}$$

where E_i is the energy of the $i-th$ component, and $i = 1, \ldots, M-1$. The energy of each singular is computed with

$$E_i = s_i^2 / (\sum_{i=1}^{M} s_i^2) \tag{7}$$

where s_i is the $i-th$ singular value of the Hankel matrix obtained before. The first component extracted is the component C_L and the second is the component C_H (if $M = 2$). When the optimal number of components $M > 2$, the component C_H is computed with the summation of the components from 2 to $M-th$, as follows

$$C_H = \sum_{i=2}^{M} C_i \tag{8}$$

C. Estimation of components with an Autoregressive Neural Network based on PSO

The ANN is based on the algorithm PSO, it performs the estimation to obtain $\hat{C_L}$, and $\hat{C_H}$. The ANN inputs are the lagged terms of C_L and C_H. The ANN has a common structure of three layers [19], at the hidden layer the sigmoid transfer function is applied, and at the output layer the estimated value is obtained. The ANN output is

$$\hat{x} = \phi(net) \times b \tag{9}$$

where $\hat{x}$ is the estimated value, net is the output of the hidden layer, b is the vector that contains the weights on the connections from the hidden layer to the output layer, net is computed with

$$net = x \times w \tag{10}$$

where x is the data input matrix, with order $N \times P$, N is the sample length, and P is the number of input variables (lags terms), w is the weight matrix of order $P \times N_h$, with N_h hidden units. The sigmoid transfer function is applied at hidden layer with

$$\phi(net) = 1/(1+e^{-net}) \tag{11}$$

The weights of the ANN connections, w and b are adjusted with PSO learning algorithm. In the swarm the N_p particles has a position vector $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$, and a velocity vector $V_i = (V_{i1}, V_{i2}, \ldots, V_{iD})$, each particle is considered a potential solution in a D-dimensional search space. During each iteration the particles are accelerated toward the previous best position denoted by p_{id} and toward the global best position denoted by p_{gd}. The swarm has N_pxD values and is initialized randomly, D is computed with $P \times N_h + N_h$; the process finish when the lowest error is obtained based on the fitness function evaluation, or when the maximum number of iterations is reached [20], [21].

$$V_{id}^{l+1} = I^l \times V_{id}^l + c_1 \times rd_1(p_{id}^l + X_{id}^l) + c_2 \times rd_2(p_{gd}^l + X_{id}^l) \tag{12}$$

$$X_{id}^{l+1} = X_{id}^l + V_{id}^{l+1} \tag{13}$$

$$I^l = I_{max}^l - \frac{I_{max}^l - I_{min}^l}{iter_{max}} \times l, \tag{14}$$

where $i = 1, \ldots, N_p$, $d = 1, \ldots, D$; I denotes the inertia weight, c_1 and c_2 are learning factors, rd_1 and rd_2 are positive random numbers in the range $[0,1]$ under normal distribution,

 http://dx.doi.org/10.17562/PB-51-5

ISSN 2395-8618

l is the lth iteration. Inertia weight has linear decreasing, in equation 14, I_{max} is the maximum value of inertia, I_{min} is the lowest, and $iter_{max}$ is total of iterations.

The particle X_{id} represents the optimal solution of the set of weights in the neural network, therefore X_id contains the ANN connections weights w and b.

D. Recomposing the time series

The recomposition of the time series is done with the addition of the estimated components, then the forecasted time series is obtained using

$$\hat{x} = \hat{C}_L + \hat{C}_H. \tag{15}$$

III. Forecasting accuracy metrics

The number of lags for the ANN is determined with the metric: Generalized Cross Validation (GCV), this determines the best number based on the accuracy of the forecasting probing a determined range of values. The evaluation of the forecasting is computed with the metrics: Mean Absolute Percentage Error (MAPE), Coefficient of determination R^2, Root Mean Squared Error (RMSE), and Relative Error (RE).

$$RMSE = \sqrt{\frac{1}{N_v}\sum_{i=1}^{N_v}(x_i - \hat{x}_i)^2} \tag{16}$$

$$GCV = \frac{RMSE}{(1 - K/N_v)^2} \tag{17}$$

$$MAPE = \left[\frac{1}{N_v}\sum_{i=1}^{N_v}|(x_i - \hat{x}_i)/x_i|\right] \times 100 \tag{18}$$

$$R^2 = 1 - \frac{\sigma^2(er)}{\sigma^2(x)} \tag{19}$$

$$RE = \sum_{i=1}^{N_v}(x_i - \hat{x}_i)/x_i \tag{20}$$

where N_v is the validation (testing) sample size, x_i is the i-th observed value, $\hat{x}_i$ is the i-th estimated value, and K is the number of lagged values.

IV. Results and discussion

The applied data are available from the CONASET web site [22], and they represent the number of accidents and the injured of Concepción-Chile, from year 2000 to 2012 with weekly sampling. The training data set contains the 70% of the sample, consequently the testing data set contains the 30% In the next subsections are evaluated the time series forecasting strategy presented in Fig. 1.

A. Embedding and Decomposition

The time series is embed in a Hankel matrix, the optimal number of components M was determined using and initial number $M = N/2$ components. Once obtained the number the components, the differential energy ΔE, of each component was computed, this is shown in the Fig. 2, the maximum peak represents the optimal M. The embedding and the decomposing is executed again with the optimal M, for time series number of accidents the optimal was $M = 6$, and for the time series injured people the optimal found was $M = 4$. The component of low frequency extracted and estimated for the time series number of accidents is shown in the Fig. 4a, while the component of high frequency the same time series is shown in the Fig. 4b. The component of low frequency extracted and estimated for the time series injured people is shown in the Fig. 5a, while the component of high frequency the same time series is shown in the Fig. 5b.

B. Estimation and Recomposition

The calibration of the number of lags of the ANN was determined with the GCV metric, for the two time series was found an optimal $ANN(K, N_h, 1)$, with $K = 7$ inputs for the time series number of accidents and $K = 5$ for the time series injured people as shown the Fig. 3, and the number of hidden nodes was assigned in $N_h = 6$ for the two time series, this value was computed with the natural logarithm of the training data set length (normally used in our experiments).

The PSO learning algorithm was applied to determine the weights of the ANN, after trial and error they were configured with a swarm of $N_p \times D$ dimension, $N_p = 40$ particles, and $D = Np \times N_h + N_h$, inertia weight parameter I has linear decreasing with a maximum value of 1 and a minimum value of 0.2, the acceleration factors c_1 and c_2 were fixed in 1.05 and 2.95 respectively, the $iter_max$ is 2500.

The evaluation performed at the testing stage for the time series number of accidents is presented in the Fig. 6 and Table I. The observed values vs. the estimated values are illustrated in the Fig. 6a, reaching a good accuracy, while the relative error is presented in the Fig. 6b, which shows that the 98.5% of the points present an error lower than the $\pm 10\%$.

The evaluation performed at the testing stage for the time series number of injured people is presented in the Fig. 7 and Table II. The observed values vs. the estimated values are illustrated in the Fig. 7a, reaching a good accuracy, while the relative error is presented in the Fig. 7b, which shows that the 95.54% of the points present an error lower than the $\pm 10\%$.

TABLE I
Number of Accidents Forecasting

	SVD-ANN-PSO	ANN-PSO
Components	6	—
RMSE	0.0211	0.087
MAPE	3.17%	14.48%
R^2	98.29%	70.71%
$RE \pm 10\%$	98.5%	48.76%

The results presented in Table I show that the major accuracy of the forecasting of the time series number of accidents is achieved with the model SVD-ANN-PSO(7,6,1), with a *RMSE* of 0.0211, and a *MAPE* of 3.17%, the 98.5% of the points have an relative error lower than the $\pm 10\%$.

ISSN 2395-8618

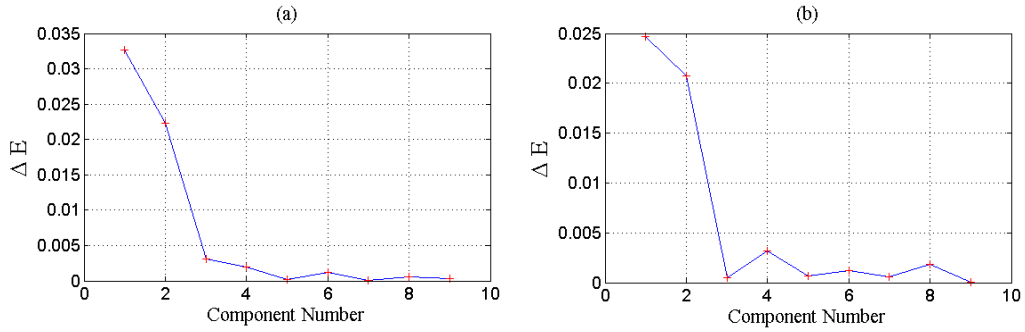


Fig. 2. ΔE : (a) Number of Accidents, (b) Injured people

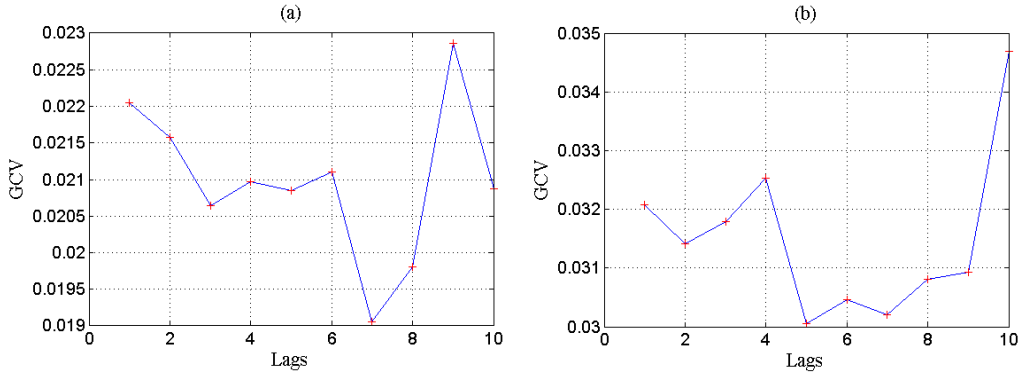


Fig. 3. Lags calibration (a) Number of accidents (b) Injured people

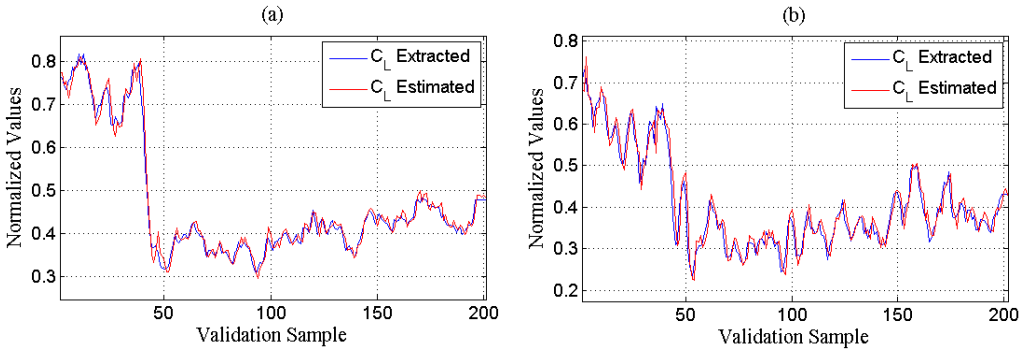


Fig. 4. Number of Accidents components (a) C_L, (b) C_H

TABLE II
INJURED PEOPLE FORECASTING

	SVD-ANN-PSO	ANN-PSO
Components	4	—
RMSE	0.0172	0.101
MAPE	3.58%	21.42%
R^2	98.5%	46.08%
$RE \pm 10\%$	95.54%	40.59%

The results presented in Table II show that the major accuracy of the forecasting of the time series injured people is achieved with the model SVD-ANN-PSO(5,6,1), with a *RMSE* of 0.0172, and a *MAPE* of 3.58%, the 95.54% of the points have an relative error lower than the $\pm 10\%$.

V. CONCLUSIONS

The proposed forecasting strategy is based on the time series decomposition using the singular values of the Hankel matrix. The strategy consists of four stages: embedding, decomposition, estimation, and recomposition. The embedding consists in mapping the time series in a Hankel matrix. The decomposition is based on SVD technique, SVD extracts the components of low and high frequency of the time series, the estimation is executed with an ANN based on PSO, while the recomposition is made with the single addition of the estimated components.

For evaluation of this strategy, we implemented a conventional ANN based on PSO. The best result was obtained

 http://dx.doi.org/10.17562/PB-51-5

ISSN 2395-8618

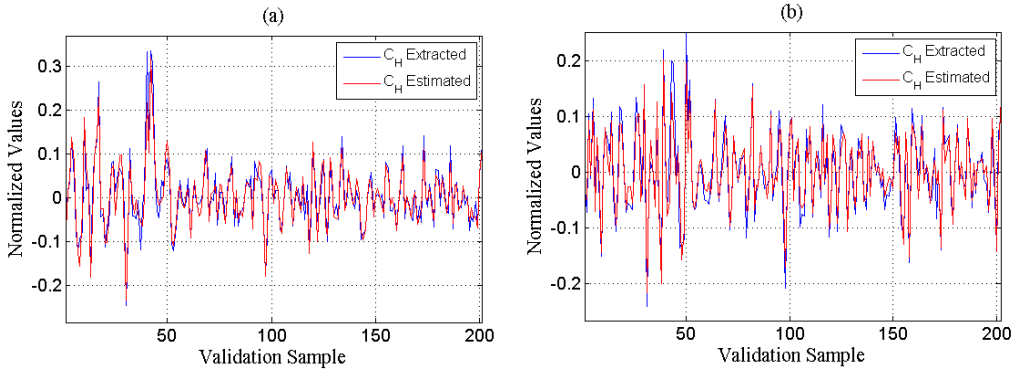


Fig. 5. Injured people components(a) C_L, (b) C_H

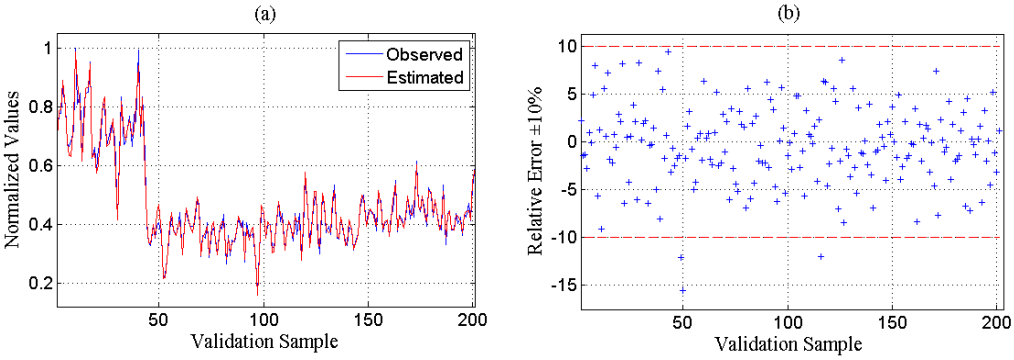


Fig. 6. SVD-ANN-PSO(7,6,1) (a) Observed vs. Estimated (b) Relative Error

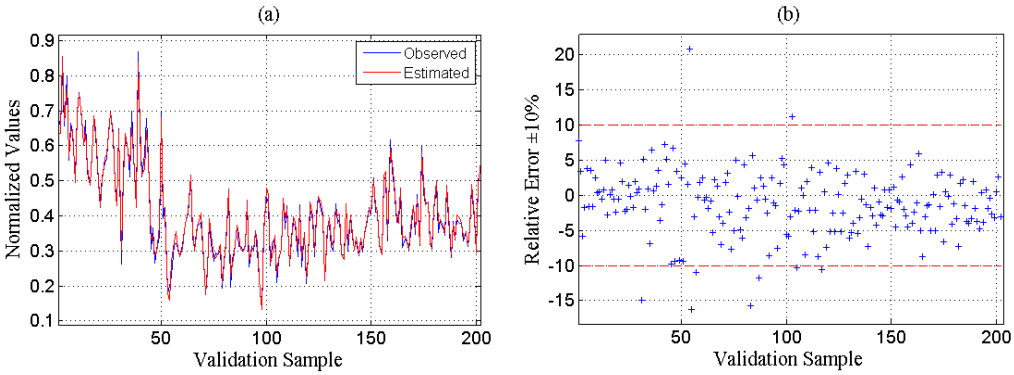


Fig. 7. SVD-ANN-PSO(5,6,1) (a) Observed vs. Estimated (b) Relative Error

with the proposed strategy for the two time series analyzed, SVD-ANN-PSO shows superiority regards the conventional implementation. For the time series number of accidents, SVD-ANN-PSO reaches an *RMSE* of 0.0211, and a *MAPE* of 3.17%, in front of the conventional ANN-PSO that reaches an *RMSE* of 0.087, and a *MAPE* of 14.48%. For traffic accidents reaches an *RMSE* of 0.0172, and a *MAPE* of 3.58%, in front of the conventional ANN-PSO that reaches an *RMSE* of 0.101, and a *MAPE* of 21.42%.

In the future, this strategy will be evaluated with data of traffic accidents of other regions of Chile, other countries, and with time series of other engineering fields.

ACKNOWLEDGEMENTS

This research was partially supported by the Chilean National Science Fund through the project Fondecyt-Regular 1131105 and by the VRIEA of the Pontificia Universidad Católica de Valparaíso.

REFERENCES

[1] Hornik K., Stinchcombe X., White H.: Multilayer feedforward networks are universal approximators. Neural Networks. 2(5), 359–366 (1989)

[2] Svozil D., Kvasnicka V., Pospichal J.: Introduction to multi-layer feed-forward neural networks. Chemometrics and Intelligent Laboratory Systems. 39(1), 43–62 (1997)

ISSN 2395-8618

[3] Chattopadhyay G., Chattopadhyay S.:Autoregressive forecast of monthly total ozone concentration: A neurocomputing approach. Computers & Geosciences. 35(9), 1925–1932 (2009)
[4] Maali Y., Al-Jumaily A.: Multi Neural Networks Investigation based Sleep Apnea Prediction. Procedia Computer Science. 24, 97–102 (2013)
[5] Rojas I., Pomares H., Bernier J.L., Ortega J., Pino B., Pelayo F.J., Prieto A.: Time series analysis using normalized PG-RBF network with regression weights. Neurocomputing. 42(1–4), 267–285 (2002)
[6] Roh S.B., Oh S.K., Pedrycz W.: Design of fuzzy radial basis function-based polynomial neural networks. Fuzzy Sets and Systems. 185(1), 15–37 (2011)
[7] Liu F., Ng G.S., Quek C.: RLDDE: A novel reinforcement learning-based dimension and delay estimator for neural networks in time series prediction. Neurocomputing. 70(7–9), 1331–1341 (2007)
[8] Scarselli F., Chung A.: Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results. Neural Networks. 11(1), 15–37 (1998)
[9] Gheyas I.A., Smith L.S.: A novel neural network ensemble architecture for time series forecasting. Neurocomputing. 74(18), 3855–3864 (2011)
[10] Gao D., Kinouchi Y., Ito K., Zhao X.: Neural networks for event extraction from time series: a back propagation algorithm approach. Future Generation Computer Systems. 21(7), 1096–1105 (2005)
[11] Khashei M., Bijari M., Ali G.: Hybridization of autoregressive integrated moving average (ARIMA) with probabilistic neural networks (PNNs). Computers & Industrial Engineering. 63(1), 37–45 (2012)
[12] Jeong K., Koo C., Hong T.: An estimation model for determining the annual energy cost budget in educational facilities using SARIMA (seasonal autoregressive integrated moving average) and ANN (artificial neural network). Energy. 71, 71–79 (2014)
[13] Wei Y., Chen M.C.: Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. Transportation Research Part C: Emerging Technologies. 21(1), 148–162 (2012)
[14] Shoaib M., Shamseldin A.Y., Melville B.W.: Comparative study of different wavelet based neural network models for rainfall–runoff modeling. Journal of Hydrology. 515, 47–58 (2014)
[15] Zhou J., Duan Z., Li Y., Deng J., Yu D.: PSO-based neural network optimization and its utilization in a boring machine. Journal of Materials Processing Technology. 178(13), 19–23 (2006)
[16] Mohandes M.A.: Modeling global solar radiation using particle swarm optimization PSO. Solar Energy. 86(11), 3137–3145 (2012)
[17] de Mingo López L.F., Blas N.G., Arteta A. The optimal combination: Grammatical swarm, particle swarm optimization and neural networks. Journal of Computational Science. 3(12), 46–55, (2012)
[18] Shores, T.S.: Applied Linear Algebra and Matrix Analysis. Springer, 291–293, (2007)
[19] Freeman J.A., Skapura D.M.: Neural Networks, Algorithms, Applications, and Programming Techniques. Addison-Wesley, California (1991)
[20] Eberhart R.C., Shi Y., Kennedy J.: Swarm Intelligence. Morgan Kaufmann, San Francisco CA (2001)
[21] Yang X.S.: Chapter 7. Particle Swarm Optimization: Nature-Inspired Optimization Algorithms. Elsevier. 99–110 (2014)
[22] National Commission of Transit Security, http://www.conaset.cl

ISSN 2395-8618

Influence of the Binomial Crossover in the DE Variants Based on the Robot Design with Optimum Mechanical Energy

Miguel G. Villarreal-Cervantes, Daniel De-la-Cruz-Muciño, Carlos Ricaño-Rea, Jesus Said Pantoja-García

***Abstract*—Differential evolution (DE) is a powerful algorithm to find an optimal solution in real world problems. Nevertheless, the binomial crossover parameter is an important issue for the success of the algorithm. The proper selection of the binomial crossover parameter depends on the problem at hand. In this work, the effect of the binomial crossover in the DE/Rand/1/bin, DE/Best/1/Bin and DE/Current to rand/1/Bin is empirically studied and analyzed in the optimum design of the kinematic and the dynamic parameters of links for a parallel robot. The optimum design minimizes mechanical energy and consequently reduces the energy provided by the actuator. Based on the experimental results, the range of crossover parameter values that properly explores the search space is obtained. The importance of finding a proper crossover parameter is highlighted. In addition, the optimal design shows a decrease in the parallel robot mechanical energy compared with non-optimal design.**

***Index Terms*—Differential evolution, binomial crossover, optimum design, mechatronic design.**

I. Introduction

DIFFERENTIAL Evolution (DE) has been proved to be a powerful evolutionary algorithm in many real-world problems due to it being highly flexible to adapt to diverse problems (nonlinear, discontinuos, etc.). It presents a superior performance in the majority of applications and it is easy to program. In [1], DE is used and modified to parameterise an equivalent circuit model of lithium-ion batteries. A boundary evolution strategy (BES) is developed and incorporated into the DE to update the parameter boundaries during the parameterizations. The method can parameterize the model without extensive data preparation.The efficiency of the approach is verified through two battery packs, one is an 8-cell battery module and the other from an electrical vehicle. In [2], DE is used to search a global optimum solution for ball bearings link system assembly weight with constraints and mixed design variables. The implementation of the DE algorithm into the particular mechanical design shows a robust performance and obtains an efficient solution to the problem. Beside, the comparisons with other algorithms confirm the effectiveness and the superiority of the DE in terms of the quality of the obtained solution. In [3], DE solves the dimensional synthesis of four and six-bar mechanisms for path generation. In [4], the simulation-optimization approach to determine the optimum location of groundwater production wells is stated as an optimization problem. The DE algorithm and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) technique is used to solve it. A significant conclusion is that the simulation-optimization model consistently finds well locations in less vulnerable areas of the model domain. Nevertheless, in the previous works [1]–[4], the selection of the parameter for the DE algorithm is a crucial factor to find better solutions in such problems. An important open issue is that the performance of the DE algorithm is highly dependent on a mutation and crossover parameter [5]. The binomial (uniform) crossover operator allows the generation of a new individual, called trial vector, from the target and mutant vectors, according to a uniform probability given by the crossover constant $CR \in [0, 1]$. Thus, the crossover constant controls which and how many elements from the current population are mutated. The right selection of the mutation and crossover parameters is a very important factor to determine the quality of the obtained solution and the efficiency of the search [6]. The selection of the suitable parameters depends on the specific problem and the previous experience of the user [7]. Unfortunately, there is no methodology to determine the mutation and crossover parameter. In this paper, an empirical study of the binomial crossover parameters on three different differential evolution variants based on the parallel robot design with optimum mechanical energy is presented.

On the other hand, the demand on high performance mechatronic systems has been a crucial factor to study the mechatronic design approach [8]–[11]. The general philosophy from the mechatronic design approach is to create an integrated design environment which promotes simultaneous design among mechanical engineering, electrical engineering, control engineering and computer engineering. Nevertheless, lately only the mechanical structure design and the control system design have been simultaneously integrated to achieve an optimal system performance due to the complexity for integrating all areas. Therefore, mechatronic system performance not only relies on its controller, but also on its mechanical structure design. Some mechatronic

Manuscript received on July 02, 2014, accepted for publication on January 20, 2015, published on June 15, 2015.

The authors are with the Instituto Politécnico Nacional, CIDETEC, Mechatronic Section, Postgraduate Department, Juan de Dios Bátiz s/n, 07700, DF, Mexico (e-mail: {mvillarrealc, ddelacruzm, cricanor, jpantjag}@ipn.mx).

ISSN 2395-8618

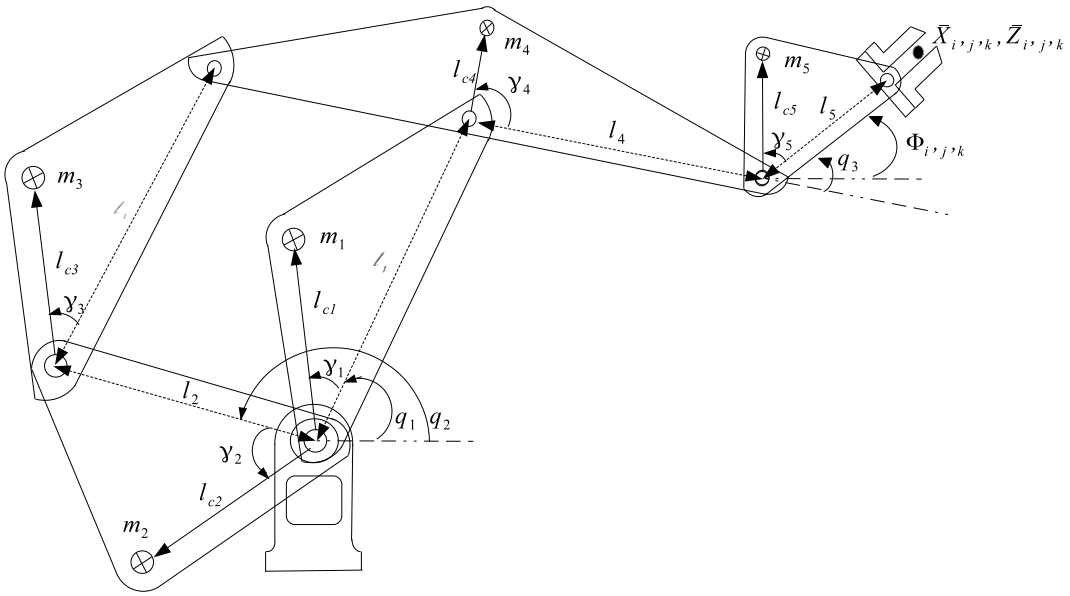


Fig. 1. Schematic diagram of the parallel robot

design works establish optimization problems to integrate both designs due to the non-linear dynamic/static nature. Meta-heuristic algorithms have been used to solve such problems due to the complex relationship in the mechatronic design. Nevertheless, few works are related to the performance of the meta-heuristic algorithm, which is an important issue to be analyzed in order to improve the obtained solutions in the mechatronic design framework.

In [12], a differential evolution algorithm with a constraint handling mechanism is proposed to simultaneously solve the design of the mechanical structure parameters of a parallel robot and the design of the proportional-integral-derivative control system required to perform a task in the Cartesian space. In [13], an approach based on a differential evolution algorithm to promote parametric reconfiguration characteristics on a continuously variable transmission $C.V.T.$ and on a parallel robot optimal design is presented. In [14], a hybrid evolutive-gradient optimization technique is proposed with the purpose of finding the optimal solutions in the search space of the synergetic design of a planar parallel robot and its control system.

Considering the mechatronic design approach, in this paper the mechanical structure of a parallel robot is designed such that the control system is improved from the energy consumption point of view. The parallel robot design is stated as an optimization problem and is solved by using three different variants of the DE algorithm. The selection of the binomial crossover parameter is analyzed to show the importance of adequately selecting such parameter in the mechatronic design framework.

The paper is organized as follows: In Section 2 the design variables, objective function and constraints of the optimization problem are described. The differential evolution algorithm is explained in Section 3. The results obtained by using the DE algorithm are described and discussed in Section 4 as well as the optimum design performance. Finally, in Section 5 the conclusions are drawn.

II. Robot design approach with optimum mechanical energy

The present work states, based on an optimization problem, the optimal dynamic and kinematic design parameters of a parallel robot which reduce the mechanical energy in a defined workspace and guarantee a dextrous workspace. The defined workspace must be described by its vertices $(\bar{x}_{d_{i,j,k}}, \bar{z}_{d_{i,j,k}}) \forall\, i,j = 1,2$. In order to ensure a dextrous workspace, the end-effector of the parallel robot must reach three different desired orientations (described by $\bar{\phi}_{i,j,k}$ $\forall$ $k = 1,2,3$) for each vertex. The parallel robot has three degree of freedom and the end-effector can move in the X–Z plane, as is shown in Fig. 1, where q_i, $\dot{q}_i$, $\ddot{q}_i$, are the joint angular position, joint velocity and joint acceleration. The Cartesian coordinate and the angular position of the manipulator end-effector are represented by $(\bar{x}_{i,j,k}, \bar{z}_{i,j,k})$ and $\bar{\phi}_{i,j,k}$, respectively. The dynamic parameters of the i-th link are the mass m_i, mass center length l_{c_i} and inertia I_i. The kinematic parameter is the i-th link length l_i.

The next sections describe the design variables, objective function, and constraints involved in the optimization problem.

A. Design variable vector

The kinematic and dynamic parameters of the links are considered as the design variables

$$p_m = [l_1, l_2, l_4, l_5, m_1, ..., m_5, l_{c_1}, ..., l_{c_5}, \gamma_1, ..., \gamma_5]^T \in R^{19},$$

ISSN 2395-8618

BEGIN
  $G = 0$;
  Create a random population $\vec{x}_{i,G}\ \forall i = 1, ..., NP$
  Evaluate $J(\vec{x}_{i,G})$, $g(\vec{x}_{i,G})$, $\forall i = 1, ..., NP$
  Do
    For $i = 1$ to NP **Do**
    Select randomly $\{r_1 \neq r_2 \neq r_3\} \in \vec{x}_G$.
    j_{rand} =randint$(1, D)$
      For $j = 1$ to D **Do**
       Mutation and crossover
      End For
      Evaluate $J(\vec{u}_{i,G+1})$, $g(\vec{u}_{i,G+1})$
      If $\vec{u}_{i,G+1}$ is better than $\vec{x}_{i,G}$ (Based on CHM) **Then**
       $\vec{x}_{i,G+1} = \vec{u}_{i,G+1}$
      Else
       $\vec{x}_{i,G+1} = x_{i,G}$
      End
       $G = G + 1$
  While $(G \leq G_{Max})$
END

Fig. 2. The DE algorithm with the constraint handling mechanism

because these modify the mechanical structure of the parallel robot. It is assumed that the length of the third link is $l_3 = l_2$, such that it is not included in the design variable vector. In addition, the joint angle configurations of the robot are chosen as other design variables

$$p_q = [q_{1_{i,j,k}}, q_{2_{i,j,k}}, q_{3_{i,j,k}}]^T \in R^{36} \subseteq W$$

$\forall\, i, j = 1, 2,\ k = 1, 2, 3$, where the space W is defined as $W = \{q | q \in p_q\}$. Hence the design variable vector is described in (1):

$$p = [p_m, p_q]^T \in R^{55}. \tag{1}$$

B. Objective Function

One way to optimize the mechanical energy of the parallel robot is to minimize the robot dynamic load. Then, the sum of the Frobenious norm of the inertia matrix $\|M\|_F$ and the potential energy V of the parallel robot is proposed as the objective function to be minimized. The objective function is shown in (2). The potential energy and the inertia matrix can be obtained in [15]:

$$J(p) = \int_W (\|M(p_m, q)\|_F + V(p_m, q))dW. \tag{2}$$

C. Design constraints

The dextrous workspace of a robot is defined as the set of all reachable points in the Cartesian space by its end-effector with different orientations [15]. Thus, the desired dextrous workspace is bounded by the vertices $(\bar{x}_{d_{i,j,k}}, \bar{z}_{d_{i,j,k}}) \forall i, j = 1, 2$ and is assumed that if the end-effector reaches the four vertices with three different orientations $\bar{\phi}_{d_{i,j,k}} \forall k = 1, 2, 3$, then any point inside the workspace is reachable with at least three different orientations. Therefore a dextrous workspace is promoted. According to the previous comments, the inequality constraint described in (3) is chosen to guarantee a desired dextrous workspace. The vertices of the desired dextrous workspace (see Fig. 1) are chosen as $(\bar{x}_{d_{1,1,k}}, \bar{z}_{d_{1,1,k}}) = (0.2m, -0.23m)$, $(\bar{x}_{d_{1,2,k}}, \bar{z}_{d_{1,2,k}}) = (0.2m, 0.26m)$, $(\bar{x}_{d_{2,1,k}}, \bar{z}_{d_{2,1,k}}) = (0.5m, 0.26m)$ y $(\bar{x}_{d_{2,2,k}}, \bar{z}_{d_{2,2,k}}) = (0.5m, -0.23m)$. An additional point $(\bar{x}_{d_{3,1,k}}, \bar{z}_{d_{3,1,k}}) = (0.2m, 0m)$ is chosen in order to fulfill the three different orientations in the workspace. The subindex k indicates the three different desired orientation for each vertex. The orientations are defined as $\bar{\phi}_{d_{i,j,1}} = -\frac{\pi}{2}\, rad$, $\bar{\phi}_{d_{i,j,2}} = 0\, rad$ and $\bar{\phi}_{d_{i,j,3}} = \frac{\pi}{2}\, rad\ \forall\ i, j = 1, 2$:

$$\begin{aligned} g_1 : \int_w \begin{pmatrix} \bar{x}_{d_{i,j,k}} - (l_1 \cos q_{1_{i,j,k}} - l_4 \cos q_{2_{i,j,k}} \\ -l_5 \cos(q_{2_{i,j,k}} + q_{3_{i,j,k}})) \end{pmatrix}^2 dW + \\ \int_w \begin{pmatrix} \bar{z}_{d_{i,j,k}} - (l_1 \sin q_{1_{i,j,k}} - l_4 \sin q_{2_{i,j,k}} \\ -l_5 \sin(q_{2_{i,j,k}} + q_{3_{i,j,k}})) \end{pmatrix}^2 dW + \\ \tfrac{1.8}{\pi} \int_w \left(\bar{\phi}_{d_{i,j,k}} - (q_{2_{i,j,k}} + q_{3_{i,j,k}} - \pi)\right)^2 dW - \\ 1 \times 10^{-6} \leq 0 \end{aligned} \tag{3}$$

Another important constraint in the parallel robot design is to avoid the collision of links in the parallel structure. Hence, the angular motion must be bounded. The inequality constraints described in (4)-(5) are included to avoid collisions between links, where $Tol_{Max2} = \frac{5\pi}{36}\ rad$ is the minimum security angle between two links:

$$g_{2\to13} : Tol_{Max2} - q_{2_{i,j,k}} + q_{1_{i,j,k}} \leq 0 \tag{4}$$

$$g_{14\to25} : q_{2_{i,j,k}} - q_{1_{i,j,k}} - \pi + Tol_{Max2} \leq 0 \tag{5}$$

The last constraints involve the bound in the design variables vector p. Those are stated in (6)-(9), where the maximum and minimum values are shown in Table I:

$$g_{26\to37} : 0 < q_{1_{i,j,k}} < \pi - Tol_{Max2} \tag{6}$$

$$g_{38\to49} : Tol_{Max2} \leq q_{2_{i,j,k}} \leq \frac{3}{2}\pi - Tol_{Max2} \tag{7}$$

$$g_{50\to61} : -\pi + Tol_{Max1} \leq q_{3_{i,j,k}} \leq \pi - Tol_{Max1} \tag{8}$$

$$g_{62\to80} : p_{m_{Min}} \leq p_m \leq p_{m_{Max}} \tag{9}$$

TABLE I
DESIGN VARIABLE VECTOR BOUNDS

Design variable	Min	Max
$q_1\ [rad]$	0	$\frac{31\pi}{36}$
$q_2\ [rad]$	$\frac{5\pi}{36}$	$\frac{49\pi}{36}$
$q_3\ [rad]$	$-\frac{35\pi}{36}$	$\frac{35\pi}{36}$
$l_i\ [m]\ i \in \{1, 2, 4, 5\}$	0.01	0.5
$m_i\ i \in \{1, 2, 3\}$	0.1	0.35
m_4	0.3	0.35
m_5	0.2	0.35
$l_{c_i}\ i \in \{1, 2, ..., 5\}$	0	0.4
$\gamma_i\ i \in \{1, 2, ..., 5\}$	$-\pi$	π

ISSN 2395-8618

Nomenclature	Variant
rand/1/bin	$u_j^i = \begin{cases} x_j^{r_3} + F(x_j^{r_1} - x_j^{r_2}) & \text{if rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x_{i,\,j} & \text{otherwise} \end{cases}$
best/1/bin	$u_j^i = \begin{cases} x_j^{best} + F(x_j^{r_1} - x_j^{r_2}) & \text{if rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases}$
current-to-rand/1/bin	$u_j^i = \begin{cases} x_j^i + K(x_j^{r_3} - x_j^i) + F(x_j^{r_1} - x_j^{r_2}) & \text{if rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases}$

Fig. 3. DE variants with binomial crossover

D. Optimization problem statement

The optimization problem for the parallel robot design consists in finding the optimal design parameter vector p^* which minimizes the mechanical energy of the robot (2) subject to inequality constraints related to the design such as to have a desired dextrous workspace (3), to avoid the collision between links (4)-(5) and to limit the design variable vector (6)-(9). Then, the optimization problem can be formally stated as in (10)–(11):

$$\underset{p \in R^{55}}{Min}\ J \tag{10}$$

subject to:

$$g(p) \leq 0 \in R^{65}. \tag{11}$$

III. Differential evolution algorithm

The differential evolution (DE) algorithm is a stochastic, population-based algorithm developed by Storn and Price [5], designed for optimization problems in continuous search space. DE is a real-valued number encoded evolutionary strategy for global optimization. It has been shown to be an efficient, effective and robust optimization algorithm. The main advantages of the DE algorithm are: i) The DE is a population based algorithm, ii) No additional computation is needed to define the search direction, such as, gradient vector, Hessian matrix, iii) The DE can be used for different kinds of optimization problems, such as, continuos, discontinuous, etc. Nevertheless, the original DE algorithm lacks a constraint handling mechanism. In this paper a constraint handling mechanism is included into the DE algorithm [16]. The key parameters are: NP - the population size that is the set of individuals, CR - the crossover constant that controls the influence of the parent in the generation of the offspring (higher values mean less influence of the parent), F - the weight applied to the influence of two of the three individuals selected at random in order to generate the offspring (scaling factor). The DE algorithm with the constraint handling mechanism works as follows: the initial population vector called parent is randomly generated. It is mutated and recombined in order to produce another population vector called mutant vector. The offspring vector will inherit features from the mutant vector or from its parent which depends on the uniform crossover. Finally, the new population for the next generation is selected between the parent and the offspring vector taking into account the constraint handling mechanism [16]:

- Any feasible solution is preferred to any infeasible solution.
- Among two feasible solutions, the one having better objective function value is preferred.
- Among two infeasible solutions, the one having smaller constraint violation is preferred.

Once the new population is created, all process (mutation, recombination and selection) are repeated until a pre-specified termination criterion is satisfied. The DE algorithm with the constraint handling mechanism (CHM) is described in Fig. 2.

Different DE variants with binomial crossover are used in this paper. The differences among those variants are in the recombination operator and in the way of selecting the elements in the individual. A summarize of the DE variants used in this paper is shown in Fig. 3.

IV. Results and discussion

The experiments are programmed in Matlab on a windows platform on a PC with 2.8 GHz core $i-7$ with 16GB of RAM. The population size NP of the DE algorithm consists of 36 individuals, the maximum generation is $G_{Max} = 50000$. Five independent runs are carried out with ten different values of crossover parameter $CR = [0, 0.2, ..., 0.9, 1]$.

In Tables II, III and IV, the empirical results of the DE variants with different crossover factor for the optimum parallel robot design are shown. The term J_{mean}, $\sigma(J)_{mean}$ is the mean and the standard deviation of the best objective function in the runs, J_{Best} is the best objective function found in all runs, $Time$ is the mean of the convergence time in the runs and $\#gUF$ is the percentage from the five runs which does not find feasible solution.

In Table II the empirical behavior of the Rand 1 Bin is displayed. It it observed that the best performance function values is $J^* = 0.0742$ and it will be considered as the optimum one. There are runs when $CR \in \{[0, 0.1, 0.2], 1\}$ that neither find feasible solution nor converge to the optimum solution J^*. Moreover, the individuals of those results are dispersed in the search space as is observed in the standard deviation. On the other hand, some runs with $CR \in \{[0.3, 0.5], 0.9\}$ find feasible solution and only with $CR = 0.9$ the converge to the optimal solution is given (see J_{mean} and $\sigma(J)_{mean}$). The results with $CR \in \{[0.3, 0.5]\}$ indicate that

ISSN 2395-8618

TABLE II
EMPIRICAL BEHAVIOR OF THE RAND 1 BIN ALGORITHM

$Algorithm$	CR	J_{mean}	$\sigma(J)_{mean}$	J_{Best}	$Time$ [hr]	$\#gUF$
Rand 1 Bin	0.00	3.0417	1.2605	1.0550	0.32	100%
Rand 1 Bin	0.10	3.1331	0.4526	2.7439	0.34	100%
Rand 1 Bin	0.20	3.6704	0.9251	2.7961	0.33	100%
Rand 1 Bin	0.30	2.6865	1.7699	0.1040	0.33	80%
Rand 1 Bin	0.40	0.6523	0.9613	0.1200	0.33	20%
Rand 1 Bin	0.50	1.2072	2.4889	0.0808	0.33	20%
Rand 1 Bin	**0.60**	**0.0744**	**0.0001**	**0.0742**	**0.33**	**0%**
Rand 1 Bin	**0.70**	**0.0742**	**0.0000**	**0.0742**	**0.33**	**0%**
Rand 1 Bin	**0.80**	**0.0745**	**0.0003**	**0.0742**	**0.34**	**0%**
Rand 1 Bin	0.90	2.8655	1.8061	0.0742	0.33	80%
Rand 1 Bin	1.00	22.2395	17.6979	1.8977	0.33	100%

TABLE III
EMPIRICAL BEHAVIOR OF THE BEST 1 BIN ALGORITHM

$Algorithm$	CR	J_{mean}	$\sigma(J)_{mean}$	J_{Best}	$Time$ [hr]	$\#gUF$
Best 1 Bin	0.00	2.6063	1.1047	1.4722	0.33	100%
Best 1 Bin	0.10	1.8437	0.8270	1.2125	0.32	100%
Best 1 Bin	0.20	0.8046	0.6280	0.3204	0.32	100%
Best 1 Bin	0.30	1.1269	0.5443	0.4283	0.33	100%
Best 1 Bin	0.40	0.4054	0.0803	0.3312	0.32	100%
Best 1 Bin	0.50	0.8405	0.3533	0.4492	0.33	100%
Best 1 Bin	0.60	0.4034	0.3700	0.1189	0.32	100%
Best 1 Bin	0.70	0.1026	0.0924	0.0103	0.32	100%
Best 1 Bin	0.80	0.1035	0.1647	0.0069	0.32	100%
Best 1 Bin	0.90	0.3434	0.2422	0.0310	0.32	100%
Best 1 Bin	1.00	7.5324	5.8768	1.5052	0.32	100%

TABLE IV
EMPIRICAL BEHAVIOR OF THE CURRENT TO RAND 1 BIN ALGORITHM

$Algorithm$	CR	J_{mean}	$\sigma(J)_{mean}$	J_{Best}	$Time$ [hr]	$\#gUF$
Current to rand 1 Bin	0.00	2.7856	1.2458	1.0478	0.32	100%
Current to rand 1 Bin	**0.10**	**0.9398**	**1.1931**	**0.0758**	**0.32**	**40%**
Current to rand 1 Bin	**0.20**	**3.6631**	**2.4611**	**0.0978**	**0.32**	**80%**
Current to rand 1 Bin	**0.30**	**3.6965**	**2.3941**	**0.0753**	**0.33**	**80%**
Current to rand 1 Bin	0.40	4.6247	1.1783	2.9582	0.33	100%
Current to rand 1 Bin	0.50	4.0629	1.5545	2.6370	0.34	100%
Current to rand 1 Bin	0.60	3.6810	0.6680	2.7071	0.34	100%
Current to rand 1 Bin	0.70	4.2893	0.3206	3.9128	0.34	100%
Current to rand 1 Bin	0.80	6.1524	1.0623	4.7779	0.34	100%
Current to rand 1 Bin	0.90	4.5003	2.3716	0.9021	0.34	100%
Current to rand 1 Bin	1.00	27.2530	20.8946	7.8482	0.34	100%

in spite of producing feasible solutions the converge to the optimum one is not reached, which means that suboptimal solutions are found. The best results are given with $CR \in [0.6, 0.8]$ because they find feasible solution in all runs and the convergence to the optimal solution is always reached in all runs (see J_{mean} and $\sigma(J)_{mean}$).

Tables III and IV show the Best 1 Bin and the Current to rand 1 Bin behaviors, respectively. It is observed that the Best 1 Bin algorithm performs poorly. The convergence is towards unfeasible solutions. This indicates that the use of the best individuals in the mutation process accelerate the convergence to unfeasible solutions and there is a lack of diversity in the solution. On the other hand, Current to rand 1 Bin finds local solutions near the optimum one with $CR \in \{[0.1, 0.3]\}$ and those solutions do not converge to a similar performance function value (see the standard deviation).

In all DE variants, the convergence time of the results is competitive among different crossover values. Clearly, selection of the crossover parameter CR is a very important factor in the parallel robot design, because different values of the crossover parameter are required to find feasible solutions among the DE variants. The optimal empirical results indicate that the best DE variant among those analyzed is the DE Rand 1 Bin with the best crossover probability 0.6%–0.8%. Hence the influence of the mutant vector in the generation of the child vector (offspring) must be larger than the parent (target) vector influence. A tradeoff in the selection between mutant and parent vectors must be considered in order to obtain the best solution, and it depends on the problem at hand. A suitable selection of the crossover parameter promotes a better exploration of the search space and the success of the DE variant to find the optimal design parameters of the parallel robot.

ISSN 2395-8618

TABLE V
OPTIMUM PARAMETERS OF THE PARALLEL ROBOT

Mass [Kg]	$m_1 = 0.3499$	$m_2 = 0.3499$	$m_3 = 0.3499$	$m_4 = 0.3$	$m_5 = 0.2$
Length [m]	$l_1 = 0.2263$	$l_2 = 0.0765$	$l_4 = 0.3514$	$l_5 = 0.03$	
Mass center length [m]	$l_{c_1} = 0.1566$	$l_{c_2} = 0.0546$	$l_{c_3} = 0.1520$	$l_{c_4} = 0.0783$	$l_{c_5} = 0.0095$
Mass center angle [rad]	$\gamma_1 = -3.1415$	$\gamma_2 = 0.0842$	$\gamma_3 = 3.1215$	$\gamma_4 = -3.1284$	$\gamma_5 = -1.7150$

TABLE VI
NON-OPTIMUM PARAMETERS OF THE PARALLEL ROBOT

Mass [Kg]	$m_1 = 0.3$	$m_2 = 0.25$	$m_3 = 0.16$	$m_4 = 0.35$	$m_5 = 0.13$
Length [m]	$l_1 = .2$	$l_2 = 0.05$	$l_4 = .25$	$l_5 = 0.072$	
Mass center length [m]	$l_{c_1} = 0.0524$	$l_{c_2} = 0.0114$	$l_{c_3} = 0.1$	$l_{c_4} = 0.0643$	$l_{c_5} = .0185$
Mass center angle [rad]	$\gamma_1 = 0$	$\gamma_2 = 0$	$\gamma_3 = 0$	$\gamma_4 = \pi$	$\gamma_5 = 0$

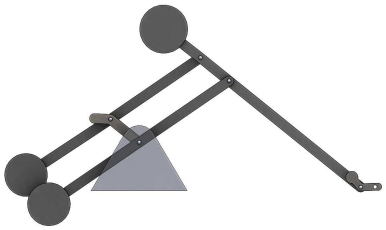


Fig. 4. Parallel robot with optimum links

On the other hand, the optimum design parameter vector is shown in Table V. In Fig. 4 the shape of the links of the parallel robot with the optimum design parameter are displayed. The shapes of the links are obtained by considering the optimum design parameter vector and making empirical Computer Aided Designs (CAD) in Solidworks until the design fulfills the optimum design parameter vector.

In order to verify the mechanical energy of the optimum parallel robot design, simulation results were used. In this case, a circle in the X–Z plane and a sinusoidal signal are chosen as the desired position and orientation to be followed by the end-effector of the parallel robot, respectively. The desired trajectory is shown in (12)-(14). A proportional-integral-derivative (PID) control is selected for this goal:

$$X_d = 0.35 + 0.1\cos(0.6283t) \tag{12}$$

$$Z_d = 0.1\sin(0.6283t) \tag{13}$$

$$\phi_d = 0.0872\sin(2.0943t) \tag{14}$$

The PID gains are selected by a trial and error procedure. Those gains are: $k_{p_1} = 20$, $k_{i_1} = 5$, $k_{d_1} = 3$, $k_{p_2} = 15.8$, $k_{i_2} = 5.4$, $k_{d_2} = 1.1$, $k_{p_3} = 0.8$, $k_{i_3} = 0.8$, $k_{d_3} = 0.005$. In Fig. 5 the trajectory tracking of the end-effector is given. It is observed that the end-effector trajectory is in the desired workspace (bounded by a squared continuous line) and the PID control system stabilizes the end-effector in the trajectory. The control signal (applied torque) to follow the desired trajectory is shown in Fig. 6. It is observed, after the first second, the control torque is low, such that, the mechanical energy of the parallel robot is low too.

TABLE VII
COMPARISON OF THE CONTROL SIGNAL NORM WITH BOTH APPROACHES

Design approach	$\|u_1\|$	$\|u_2\|$	$\|u_3\|$
Optimum	1.1999	0.3245	0.1293
Non optimum	33.0886	0.7917	1.0584

In order to compare the proposed optimum design of the parallel robot, comparative results with a non optimum design are carried out. The non optimum design parameters are chosen with the consideration that the total mass of the parallel robot is smaller than the total mass of the optimum design of the parallel robot. Simulation results are performed with both designs and the comparative results are carried out by analyzing the norm of the control signal of the tracking trajectory. The non optimum design are chosen accordingly to Table VI and its PID gains are proposed as: $k_{p_1} = 130$, $k_{i_1} = 35$, $k_{d_1} = 3$, $k_{p_2} = 55.8$, $k_{i_2} = 5.4$, $k_{d_2} = 1.1$, $k_{p_3} = 0.8$, $k_{i_3} = 0.8$, $k_{d_3} = 0.005$.

In Table VII the norm of the control signals are shown. It is observed that the norm of the control signal in the optimum design is smaller than the non optimum design, in spite of having more total mass. Hence, the optimum mechanical structure of the parallel robot minimizes the mechanical energy, resulting that the torque provided by the control system is reduced. Then, the proposed design approach promotes the mechatronic design approach because the optimum mechanical structure improves the energy efficiency of the control system.

V. CONCLUSIONS

In this work, an optimum design approach for a parallel robot is stated as an optimization problem. This approach finds the dynamic and kinematic parameters of links that fulfill with a structure with less mechanical energy. Hence, as a consequence, the mechanical structure improves the control system behavior w.r.t. the energy consumption in the trajectory tracking.

The main highlights in the selection of the crossover parameter in DE variants are:

– The DE best 1 bin presents a high premature convergence to unfeasible solution in spite of the crossover parameter selection.

 http://dx.doi.org/10.17562/PB-51-6

ISSN 2395-8618

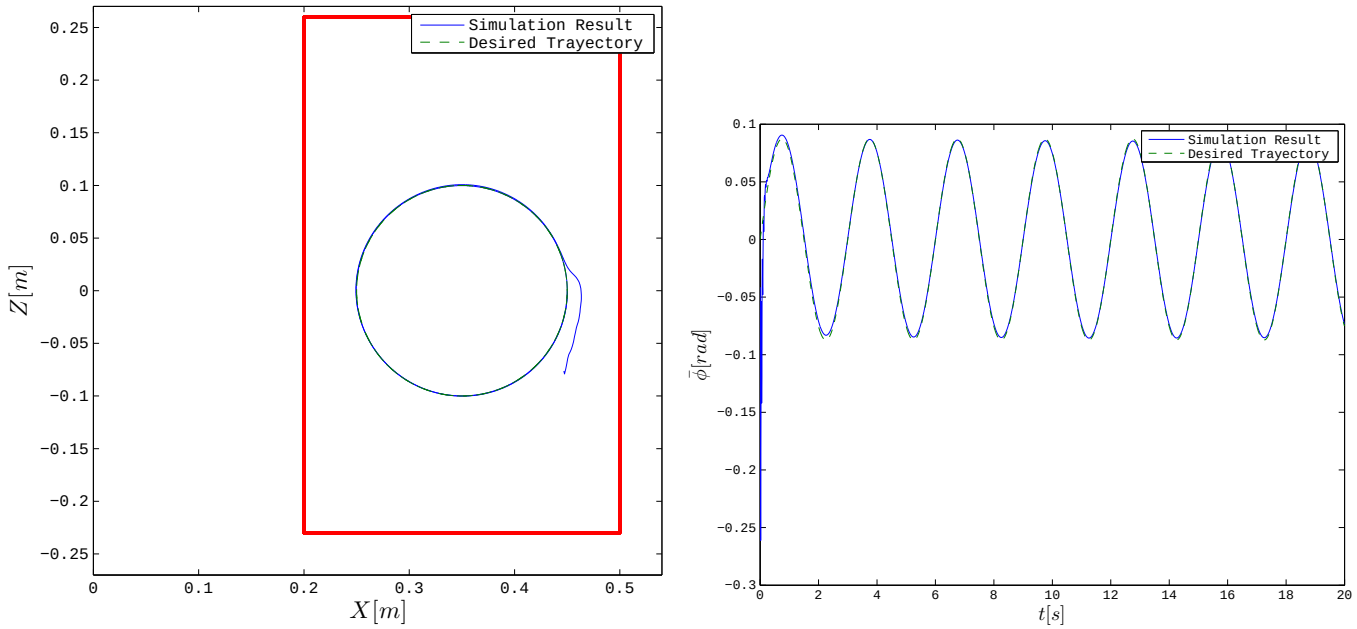


Fig. 5. Trajectory tracking of the optimum parallel robot with the PID control system

– The DE current to Rand 1 Bin converge to a suboptimal solution near the optimum one and it highly depends on the crossover parameter selection.
– The DE Rand 1 Bin presents a good convergence to the optimal solution with $CR = [0.6, 0.8]$. It promotes a better exploration of the search space without converging to local minima and without exhaustive exploration.
– The success of the DE variant to solve the optimum design problem effectively depends on the selection of the crossover parameter and is related to the optimization problem.

Acknowledgements

This work is supported by the Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional (SIP-IPN) under project number SIP-20151212 and the CONACYT under project number 182298. The second to the fourth authors acknowledge support from CONACYT through a scholarship to pursue graduate studies at Instituto Politécnico Nacional.

References

[1] G. Yang, "Battery parameterisation based on differential evolution via a boundary evolution strategy," *Journal of Power Sources*, vol. 245, pp. 583–593, 2014.

[2] H. Saruhan, "Differential evolution and simulated annealing algorithms for mechanical systems design," *Engineering Science and Technology, an International Journal*, 2014.

[3] A. Ortiz, J. Cabrera, F. Nadal, and A. Bonilla, "Dimensional synthesis of mechanisms using differential evolution with auto-adaptive control parameters," *Mechanism and Machine Theory*, vol. 64, pp. 210–229, 2013.

[4] A. Elci and M. T. Ayvaz, "Differential-evolution algorithm based optimization for the site selection of groundwater production wells with the consideration of the vulnerability concept," *Journal of Hydrology*, vol. 511, pp. 736–749, 2014.

[5] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, ser. Natural Computing Series. Springer-Verlag New York, Inc., 2005.

[6] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transaction on Evolutionay Computation*, vol. 3, no. 2, pp. 124–141, 1999.

[7] D. Zaharie, "Influence of crossover on the behavior of differential evolution algorithms," *Applied Soft Computing*, vol. 9, no. 3, pp. 1126–1138, 2009.

[8] M. Villarreal-Cervantes, C. Cruz-Villar, J. Alvarez-Gallegos, and E. Portilla-Flores, "Robust structure-control design approach for mechatronic systems," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 5, pp. 1592–1601, Oct 2013.

[9] C. A. Cruz-Villar, J. Alvarez-Gallegos, and M. G. Villarreal-Cervantes, "Concurrent redesign of an underactuated robot manipulator," *Mechatronics*, vol. 19, no. 2, pp. 178–183, 2009.

[10] Q. Li, W. Zhang, and L. Chen, "Design for control-a concurrent engineering approach for mechatronic systems design," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 2, pp. 161–169, Jun 2001.

[11] S. Alyaqout, P. Papalambros, and A. Ulsoy, "Combined robust design and robust control of an electric DC motor," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 3, pp. 574–582, June 2011.

[12] M. G. Villarreal-Cervantes, C. A. Cruz-Villar, J. Alvarez-Gallegos, and E. A. Portilla-Flores, "Differential evolution techniques for the structure-control design of a five-bar parallel robot," *Engineering Optimization*, vol. 42, no. 6, pp. 535–565, 2010.

[13] E. A. Portilla-Flores, E. Mezura-Montes, J. Alvarez-Gallegos, C. A. Coello-Coello, C. A. Cruz-Villar, and M. G. Villarreal-Cervantes, "Parametric reconfiguration improvement in non-iterative concurrent mechatronic design using an evolutionary-based approach," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 5, pp. 757–771, 2011.

[14] M. G. Villarreal-Cervantes, C. A. Cruz-Villar, and J. Alvarez-Gallegos, "Synergetic structure-control design via a hybrid gradient-evolutionary algorithm," *Optimization & Engineering*, 2014.

[15] M. Spong and S. Hutchinson, *Robot Modeling and Control*. Wiley, 2005.

[16] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer methods in applied mechanics and engineering*, vol. 186, no. 2/4, pp. 311–338, 2000.

ISSN 2395-8618

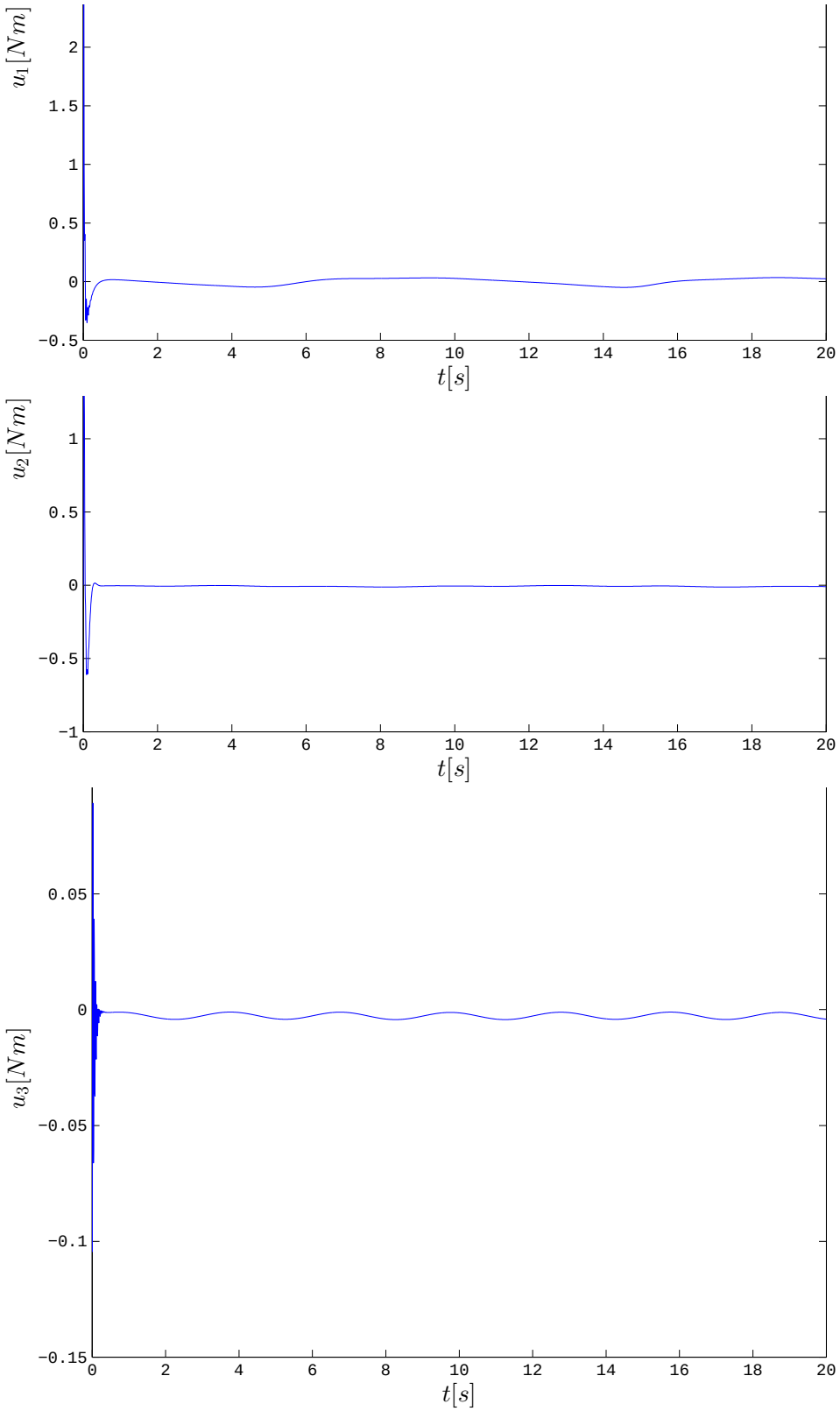


Fig. 6. PID control signal behavior

 http://dx.doi.org/10.17562/PB-51-6

ISSN 2395-8618

The Multiple Knapsack Problem Approached by a Binary Differential Evolution Algorithm with Adaptive Parameters

Leanderson André and Rafael Stubs Parpinelli

***Abstract*—In this paper the well-known 0-1 Multiple Knapsack Problem (MKP) is approached by an adaptive Binary Differential Evolution (aBDE) algorithm. The MKP is a NP-hard optimization problem and the aim is to maximize the total profit subjected to the total weight in each knapsack that must be less than or equal to a given limit. The aBDE self adjusts two parameters, perturbation and mutation rates, using a linear adaptation procedure that changes their probabilities at each generation. Results were obtained using 11 instances of the problem with different degrees of complexity. The results were compared using aBDE, BDE, a standard Genetic Algorithm (GA) and its adaptive version (aGA), and an island-inspired Genetic Algorithm (IGA) and its adaptive version (aIGA). The results show that aBDE obtained better results than the other algorithms. This indicates that the proposed approach is an interesting and a promising strategy to control the parameters and for optimization of complex problems.**

***Index Terms*—Adaptive parameter control, binary differential evolution, multiple knapsack problem, evolutionary computation.**

I. Introduction

THE 0-1 Multiple Knapsack Problem (MKP) is a binary NP-hard combinatorial optimization problem that consists in given a set of items and a set of knapsacks, each item with a mass and a value, determine which item to include in which knapsack. The aim is to maximize the total profit subjected to the total weight in each knapsack that must be less than or equal to a given limit.

Different variants of the MKP can be easily adapted to real problems, such as, capital budgeting, cargo loading and others [1]. Hence, the optimization of resource allocation is one major concern in several areas of logistics, transportation and production [2]. In this way, the search for efficient methods to achieve such optimization aims to increase profits and reduce the use of raw materials.

According to the size of an instance (number of items and number of knapsacks) of the MKP, the search space can become too large to apply exact methods. Hence, a large number of heuristics and metaheuristics have been applied to the MKP. Some examples are the modified binary particle swarm optimization [3], the binary artificial fish swarm algorithm [4], and the binary fruit fly optimization algorithm [5]. In this work, it is investigated the performance of an adaptive Differential Evolution algorithm designed for binary problems.

The Differential Evolution (DE) algorithm is an Evolutionary Algorithm which is inspired by the laws of Darwin where stronger and adapted individuals have greater chances to survive and evolve [6]. Evolutionary Algorithms simulate the evolution of individuals through the selection, reproduction, crossover and mutation methods, stochastically producing better solutions at each generation [7]. In this analogy, the individuals are candidate solutions to optimize a given problem and the environment is the search space.

It is well documented in the literature that DE has a huge ability to perform well in continuous-valued search spaces [8]. However, for discrete or binary search spaces some adaptations are required [9]. Hence, this paper applies a Binary Differential Evolution (BDE) algorithm that is able to handle binary problems, in particular the 0-1 MKP. The BDE algorithm was first applied in [10] for the 0-1 MKP and the results obtained were promising. BDE consists in applying simple operators (crossover and bit-flip mutation) in candidate solutions represented as binary strings. In this work several different instances are approached.

It is known that the optimum values of the control parameters of an algorithm can change over the optimization process [11], directly influencing the efficiency of the method. As most metaheuristic algorithms, DE also has some control parameters to be adjusted. The parameters of an algorithm can be adjusted using one of two approaches: on-line or off-line. The off-line control, or parameter tuning, is performed prior to the execution of the algorithm. In this approach several tests are performed with different parameter settings in order to find good configurations for the parameters. In the on-line control, or parameter control, the values for the parameters change throughout the execution of the algorithm. The control of parameters during the optimization process has been consistently used by several optimization algorithms and applied in different problem domains [12], [13], [14], [15], [16]. In this way, a method to adapt the control parameters (crossover and mutation rates) of DE is applied. The aim is to explore how effective the on-line control strategy is in solving

Manuscript received on January 20, 2015, accepted for publication on March 8, 2015, published on June 15, 2015.

The authors are with the Graduate Program in Applied Computing, Departament of Computer Science, State University of Santa Catarina, Joinville, Brazil (e-mail: leanderson.andre@gmail.com, rafael.parpinelli@udesc.br).

ISSN 2395-8618

Algorithm 1 Binary Differential Evolution (BDE)

1: Parameters : $DIM, POP, ITER, PR, MUT$
2: Generate initial population randomly: $\overrightarrow{x}_i \in \{0,1\}^{DIM}$
3: Evaluate initial population with the fitness function $f(\overrightarrow{x}_i)$
4: **while** termination criteria not met **do**
5: **for** i = 1 **to** POP **do**
6: Select a random individual:
 $k \leftarrow random(1, POP)$, with $k \neq i$
7: Select a random dimension:
 $j_{rand} \leftarrow random(1, DIM)$
8: $\overrightarrow{y} \leftarrow \overrightarrow{x}_i$
9: **for** j = 1 **to** DIM **do**
10: **if** $(random(0,100) < PR)$ **or** $(j == j_{rand})$ **then**
11: **if** $(random(0,100) < MUT\ )$ **then**
12: BitFlip(y_j) {Mutation}
13: **else**
14: $y_j \leftarrow x_{kj}$ {Crossover}
15: **end if**
16: **end if**
17: **end for**
18: Evaluate $f(\overrightarrow{y})$
19: **if** $(f(\overrightarrow{y}) > f(\overrightarrow{x}_i))$ **then** {Greedy Selection}
20: $\overrightarrow{x}_i \leftarrow \overrightarrow{y}$
21: **end if**
22: **end for**
23: Find current best solution $\overrightarrow{x}^*$
24: **end while**
25: Report results

the MKP.

In the following, Section II provides an overview of the Multiple Knapsack Problem. The Binary Differential Evolution algorithm is presented in Section III and the adaptive control parameter mechanism is presented in Section III-A. Section IV gives a brief description of Genetic Algorithms and island-inspired Genetic Algorithm both used in the experiments. The experiments and results are presented in Sections V and VI, respectively. Section VII concludes the paper with final remarks and future research.

II. Multiple Knapsack Problem

The 0-1 Multiple Knapsack Problem (MKP) is a well-known NP-hard combinatorial optimisation problem and its goal is to maximize the profit of items chosen to fulfil a set of knapsacks, subjected to constraints of capacity [2]. The MKP consists of m knapsacks of capacities $C_1, C_2, ...C_m$, and a set of n items $I = \{I_1, I_2, ...I_n\}$. The binary variables $X_i (i = 1, ..., n)$ represent selected items to be carried in m knapsacks. The X_i assumes 1 if item i is in the knapsack and 0 otherwise. Each item I_i has an associated profit $P_i \geq 0$ and weight $W_{ij} \geq 0$ for each knapsack j. The goal is to find the best combination of n items by maximizing the sum of profits P_i multiplied by the binary variable X_i, mathematically represented by Equation 1. Their constraints are the capacity $C_j \geq 0$ of each knapsack. Therefore, the sum of the values of X_i multiplied by W_{ij} must be less than or equal to C_j, represented mathematically by Equation 2.

$$\max \left(\sum_{i=1}^{n} (P_i \times X_i) \right) \tag{1}$$

$$\sum_{j=1}^{m} (W_{ij} \times X_i) \leq C_j \tag{2}$$

A binary exponential function with exponent n assembles all possibilities for n items respecting the capacity of each knapsack m. Hence, the MKP search space depends directly on the values of n and m. Therefore, to find the optimal solution should be tested all 2^n possibilities for each knapsack m, i.e., $m \times 2^n$ possibilities. Depending on the instance, the search space can become intractable by exact methods. In such cases, metaheuristic algorithms are indicated. Hence, the Binary Differential Evolution is an interesting algorithm to be applied to solve the MKP. The algorithm was designed for binary optimization and is shown in next section.

III. Binary Differential Evolution

The Binary Differential Evolution (BDE) [10] is a population-based metaheuristic inspired by the canonical Differential Evolution (DE) [6] and is adapted to handle binary problems. Specifically, the BDE approach is a modification of the DE/rand/1/bin variant.

In BDE, a population of binary encoded candidate solutions with size POP interact with each other. Each binary vector $\overrightarrow{x}_i = [x_{i1}, x_{i2}...x_{iDIM},]$ of dimension DIM is a candidate solution of the problem and is evaluated by an objective function $f(\overrightarrow{x}_i)$ with $i = [1, ..., POP]$. As well as the canonical DE, BDE combines each solution of the current population with a randomly chosen solution through the crossover operator. However, the main modification to the canonical DE, besides the binary representation, is the insertion of a bit-flip mutation operator. This modification adds to the algorithm the capacity to improve its global search ability, enabling diversity.

The pseudo-code of BDE is presented in Algorithm 1. The control parameters are the number of dimensions (DIM), the population size (POP), the maximum number of generations or iterations ($ITER$), the perturbation rate (PR) and the mutation rate (MUT)(line 1). The algorithm begins creating a random initial population (line 2) where each individual represents a point in the search space and is a possible solution to the problem. The individuals are binary vectors that are evaluated by a fitness function (line 3). An evolutive loop is performed until a termination criteria is met (line 4). The termination criteria can be to reach the maximum number of iterations $ITER$. The evolutive loop consists in creating new individuals through the processes of perturbation (mutation

ISSN 2395-8618

Algorithm 2 Genetic Algorithm (GA)

1: Parameters : $DIM, POP, ITER, CR, MUT, ELI$
2: Generate initial population randomly: $\overrightarrow{x}_i \in \{0,1\}^{DIM}$
3: Evaluate initial population with the fitness function $f(\overrightarrow{x}_i)$
4: **while** termination criteria not met **do**
5: Find current best solution $\overrightarrow{x}^*$
6: **if** (ELI) **then**
7: Copy the best individual $\overrightarrow{x}^*$ to next generation
8: **end if**
9: **for** i = 1 **to** (POP / 2) **do**
10: Select two individuals k and y with tournament selection and $k \neq y$
11: **if** ($random(0,100) < CR$) {Uniform Crossover} **then**
12: **for** j = 1 **to** DIM **do**
13: **if** ($random(0,100) < 50$) **then**
14: $offspring\_a_j \leftarrow x_{yj}$
15: $offspring\_b_j \leftarrow x_{kj}$
16: **else**
17: $offspring\_a_j \leftarrow x_{kj}$
18: $offspring\_b_j \leftarrow x_{yj}$
19: **end if**
20: **end for**
21: **end if**
22: **for** j = 1 **to** DIM **do**
23: **if** ($random(0,100) < MUT$) **then**
24: BitFlip($offspring\_a_j$) {Mutation}
25: **end if**
26: **if** ($random(0,100) < MUT$) **then**
27: BitFlip($offspring\_b_j$) {Mutation}
28: **end if**
29: Add new individuals to next generation
30: **end for**
31: **end for**
32: Evaluate new population with the fitness function $f(\overrightarrow{x}_i)$
33: **end while**
34: Find current best solution $\overrightarrow{x}^*$
35: Report results

Algorithm 3 Island Inspired Genetic Algorithm (IGA)

1: Parameters : $DIM, POP, ITER, CR, MUT$
2: Generate initial population randomly: $\overrightarrow{x}_i \in \{0,1\}^{DIM}$
3: Evaluate initial population with the fitness function $f(\overrightarrow{x}_i)$
4: **while** termination criteria not met **do**
5: **for** i = 1 **to** POP **do**
6: Select an individual k, with tournament selection
7: $\overrightarrow{y} \leftarrow \overrightarrow{x}_k$
8: **if** ($random(0,100) < CR$) {Uniform Crossover} **then**
9: **for** j = 1 **to** DIM **do**
10: **if** ($random(0,100) < 50$) **then**
11: $y_j \leftarrow x_{ij}$
12: **end if**
13: **end for**
14: **end if**
15: **for** j = 1 **to** DIM **do**
16: **if** ($random(0,100) < MUT$) **then**
17: BitFlip(y_j) {Mutation}
18: **end if**
19: **end for**
20: Evaluate $f(\overrightarrow{y})$
21: **if** ($f(\overrightarrow{y}) > f(\overrightarrow{x}_k)$) **then** {Greedy Selection}
22: $\overrightarrow{x}_k \leftarrow \overrightarrow{y}$
23: **end if**
24: **end for**
25: Find current best solution $\overrightarrow{x}^*$
26: **end while**
27: Report results

and crossover) (lines 6-17), evaluation of the objective function (line 18), and a greedy selection (lines 19-21).

Inside the evolutive loop, two random indexes k and j_{rand} are selected at each generation. k represents the index of an individual in the population and must be different from the current index of individual i (line 6). j_{rand} represents the index of any dimension of the problem (line 7).

In line 8, the individual $\overrightarrow{x}_i$ is copied to a trial individual $\overrightarrow{y}$. Each dimension of the trial individual is perturbed (or modified) accordingly to the perturbation rate or if the index j is equal to index j_{rand} (line 10). The equality ensures that at least one dimension will be perturbed. The perturbation is carried out by the bit-flip mutation using its probability (line 11-12) or by the crossover operator (line 14).

From the new population of individuals the best solution $\overrightarrow{x}^*$ is found (line 23) and a new generation starts. Algorithm 1 terminates reporting the best solution obtained $\overrightarrow{x}^*$ (line 25).

A. Adaptive Binary Differential Evolution

The Adaptive Binary Differential Evolution (aBDE) algorithm aims to control two parameters: perturbation (PR) and mutation (MUT) rates. To achieve that, a set of discrete values is introduced for each of parameter. Once defined a set of values for each parameter, a single value is chosen at each generation through a roulette wheel selection strategy. The probability of choosing a value is initially defined equally which is subsequently adapted based on a criteria of success. If a selected value for a parameter yielded at least one individual in generation $t+1$ better than the best fitted individual from generation t, then the parameter value has a mark of success. Hence, if at the end of generation $t+1$ the parameter value was successful, its probability is increased with an α value, otherwise, it remains the same. The α is calculated by a linear increase as shows Equation 3:

$$\alpha = min + \left(\frac{max - min}{ITER} \times i \right), \tag{3}$$

ISSN 2395-8618

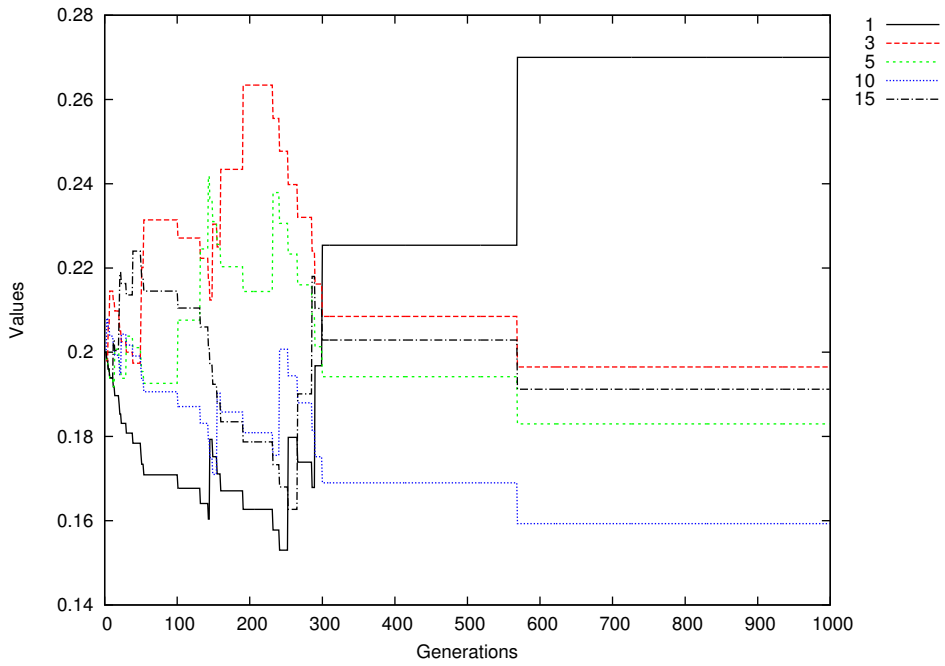


Fig. 1. Adaptive probabilities for mutation rate

where $ITER$ is the number of iterations, i is the current iteration, max is the maximum value of α and min is the minimum value of α. After adjusting the probabilities, the values are normalized between 0 and 1. To ensure a minimum of chance for each value of parameters, a β value is established.

IV. Description of GA and IGA

This section gives a brief description of two algorithms employed in the experiments.

A. Genetic Algorithm

The Genetic Algorithms (GA) are one of the best known and most used algorithms from the Evolutionary Computation field and was proposed by John Holland in 1975 [7]. The inspiration behind GA is based on Darwin's theory of evolution of species. In nature, individuals from different populations compete to survive. According to natural selection, stronger individuals and better adapted to the environment have a greater chance to survive and will continue their species. Thus, GA use the concepts of evolution as an intelligent process for optimization in finding good solutions.

The pseudo-code is presented in Algorithm 2. The control parameters are the number of dimensions (DIM), the population size (POP), the maximum number of generations or iterations ($ITER$), the crossover rate (CR), the mutation rate (MUT) and elitism (ELI) (line 1). The algorithm begins creating a random initial population (line 2) where each individual represents a point in the search space and is a possible solution to the problem. The individuals are binary vectors that are evaluated by a fitness function (line 3). An evolutive loop is performed until a termination criteria is met (line 4). The termination criteria can be to reach the maximum number of iterations $ITER$. From the population of individuals the best solution $\vec{x}^*$ is found (line 5). If elitism is applied, the best individual is placed in the next generation without any change (line 7). The evolutive loop consists in creating new individuals through the operators of crossover and mutation (lines 10-30), the generation of the new population (line 32) and the evaluation of the objective function (line 33).

Inside the evolutive loop, two individuals k and y are selected at each generation by tournament selection (line 10). The individuals are recombined accordingly to the crossover rate (line 11-21). Finally, it is applied the bit-flip mutation using its probability (line 22-30).

The new population is generated from the temporary population t (line 32), is evaluated by a fitness function (line 33) and a new generation starts. Algorithm 2 terminates reporting the best solution obtained $\vec{x}^*$ (line 35-36).

B. Island-inspired Genetic Algorithm

The Island-inspired Genetic Algorithm [17] is a meta-heuristic that uses one population of individuals as islands (island-model GA). This approach uses only one population where each individual is considered to be an island itself.

The pseudo-code of IGA is presented in Algorithm 3. The control parameters are the number of dimensions (DIM), the population size (POP), the maximum number of generations or iterations ($ITER$), the crossover rate (CR) and the mutation rate (MUT)(line 1). The algorithm begins creating

 http://dx.doi.org/10.17562/PB-51-7

ISSN 2395-8618

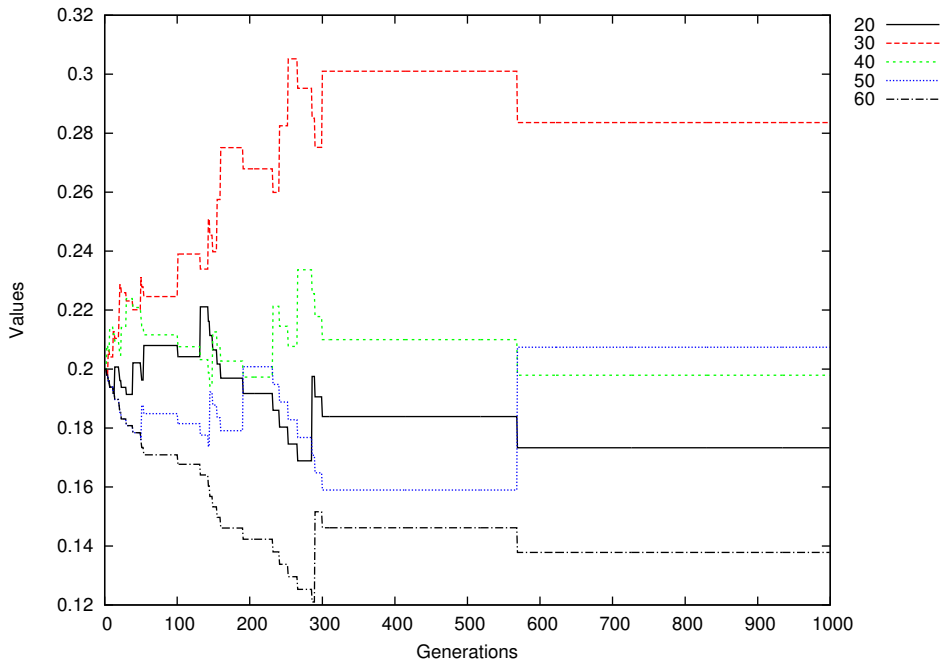


Fig. 2. Adaptive probabilities for perturbation rate

a random initial population (line 2) where each individual represents a point in the search space and is a possible solution to the problem. The individuals are binary vectors that are evaluated by a fitness function (line 3). An evolutive loop is performed until a termination criteria is met (line 4). The termination criteria can be to reach the maximum number of iterations $ITER$. The evolutive loop consists in creating new individuals through the operators of crossover and mutation (lines 6-19), the evaluation of the objective function (line 20), and a greedy selection (lines 21-23).

Inside the evolutive loop, an individual k is selected by tournament selection to where individual i must migrate (line 6). The migration process indicates that individual i will be able to exchange information with individual k. The interaction is made using an uniform crossover that produces one offspring. In line 8, the individual $\overrightarrow{x}_k$ is copied to a trial vector $\overrightarrow{y}$. The trial vector is recombined with individual i accordingly to the crossover rate (line 8-13). Finally, it is applied the bit-flip mutation using its probability (line 15-18).

From the new population of individuals the best solution $\overrightarrow{x}^*$ is found (line 25) and a new generation starts. Algorithm 3 terminates reporting the best solution obtained $\overrightarrow{x}^*$ (line 27).

V. Computational Experiments

For the experiments, 11 instances for the MKP were used[1]. Table I shows the instance reference, the optimum value, the number of knapsacks, and the number of items (or dimensions), respectively. For each instance, 100 independent runs were performed with randomly initialized populations.

[1]Available at: www.cs.nott.ac.uk/˜jqd/mkp/index.html

The algorithms were developed using ANSI C language and the experiments were run on an AMD Phenom II X4 (2.80GHz) with 4GB RAM, under Linux operating system.

TABLE I
Benchmark Instances for the MKP

Instance	Optimum Value	Knapsacks	Items
PB1	3090	4	27
PB2	3186	4	34
PB4	95168	2	29
PB5	2139	10	20
PB6	776	30	40
PB7	1035	30	37
PET7	16537	5	50
SENTO1	7772	30	60
SENTO2	8722	30	60
WEING8	624319	2	105
WEISHI30	11191	5	90

The parameters used for the BDE algorithm are: population size ($POP = 100$), number of iterations ($ITER = 1,000$), perturbation rate ($PR = 50\%$), mutation rate ($MUT = 5\%$).

The Genetic Algorithm (GA) and the Island-inspired Genetic Algorithm (IGA) use tournament selection, uniform crossover and elitism of one individual. For both algorithms the parameters are: population size ($POP = 100$), number of iterations ($ITER = 1,000$), tournament size ($T = 3$), crossover rate ($CR = 80\%$), mutation rate ($MUT = 5\%$), and elitism of one individual.

The strategy to adapt parameters is applied in all algorithms, BDE, GA and IGA, leading to its adaptive versions aBDE, aGA, and aIGA, respectively. The parameters adjusted are PR and MUT for aBDE, and CR and MUT for aGA and aIGA. Thus, the set of values for PR was defined as

ISSN 2395-8618

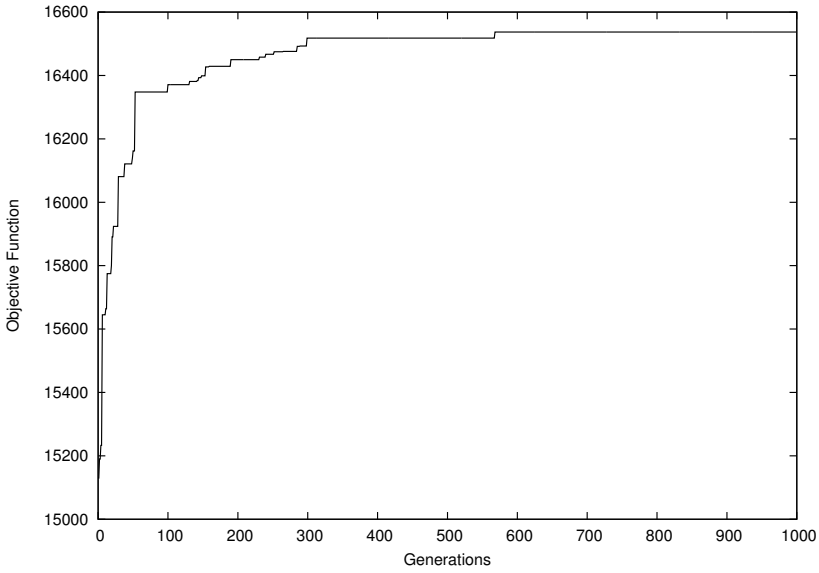


Fig. 3. Convergence graph for instance $PET7$

$\{20, 30, 40, 50, 60\}$ to aBDE, and the set of values for CR was defined as $\{50, 60, 70, 80, 90\}$ to aGA and aIGA. MUT was defined as $\{1, 3, 5, 10, 15\}$ for all algorithms.

A range between $[0.01, 0.1]$ was chosen for α and the β parameter was set to 0.01. The number of function evaluations is the same for all algorithms, resulting in a maximum of 100,000 function evaluations. All choices for the values of parameters were made empirically.

In all approaches, infeasible individuals in the population are fixed by dropping random items from the knapsack until feasibility is obtained. Feasibility of individual is verified inside the objective function as proposed in [18].

VI. Results and Analysis

Table II presents the average and the standard deviation of the best result (*Avg*±*Std*) obtained in all runs for each algorithm, the average number of objective function evaluations (*Eval*) required to achieve the optimum value, the success rate (*Success*) calculated as the percentage that the algorithm reached the optimum value, and the dominance information (P) indicating which algorithms are better than the others concerning both the average best result and the average number of function evaluations. If more than one algorithm is marked in the same benchmark means that they are non-dominated (neither of them are better than the other in both criteria). Also, for each algorithm, the last line (*Average*) shows the average of function evaluations and the average of success rate for all benchmarks. Best results are highlighted in bold.

Analyzing the results obtained by GA and IGA it is possible to notice that IGA achieve better results (success rate) in 8 instances ($PB1$, $PB2$, $PB4$, $PB6$, $PB7$, $PET7$, $SENTO1$, and $SENTO2$), except for $PB5$. This gain can be explained by the model used for exchange information that slows down the premature convergence of the algorithm allowing it to better explore the space of solutions.

Analyzing the results obtained by IGA and BDE we can notice that BDE achieved better results (success rate) in 3 instances ($SENTO2$, $WEING8$, and $WEISH30$), and equivalent results in other 3 instances ($PB4$, $PB6$, and $SENTO1$). In fact, the BDE obtained best results in instances with higher complexity. Also, the average success rate of BDE is better than the average success rate of IGA. This can be explained by the diversification power that BDE employs in its operators.

Comparing the results obtained by BDE and its adaptive version, aBDE, we can notice that the results (success rate) were even better when using the adaptive parameter control strategy for almost all instances except for $SENTO1$ and $WEISHI30$ and equal for $PB4$. Also, the average number of function evaluations decreased when using the parameter control strategy. This improvement can be explained by the adaptive choices for the values of parameters during the optimization process.

Analyzing the effectiveness of the adaptive parameter control strategy, it is possible to notice that aBDE, aIGA, and aGA obtained better success rates for the majority of the instances when compared to its non-adaptive versions. The improvement is boosted in aBDE which has a differentiated diversification mechanism.

Using the dominance information (P) from Table II, it is possible to notice that the Differential Evolution algorithm with adaptive parameter control, aBDE, is present in the non-dominated set in 8 out of 11 instances. This indicates that aBDE is robust concerning both criteria. The aBDE algorithm is dominated in instances $PB2$, $PB5$ and $SENTO1$.

In order to illustrate the behavior of the adaptive control strategy, Figures 1 and 2 show the adaptation of values

 http://dx.doi.org/10.17562/PB-51-7

ISSN 2395-8618

TABLE II
RESULTS OBTAINED BY ALL ALGORITHMS FOR EACH INSTANCE.

Benchmark	**GA**				**aGA**			
	Avg±Std	**Eval**	**Success**	**P**	**Avg±Std**	**Eval**	**Success**	**P**
PB1	3085.26±10.78	34995.18	82.00%		3086.98±8.17	45491.35	86.00%	
PB2	3131.08±40.44	89051.75	17.00%		3142.10±32.96	91786.79	15.00%	
PB4	95071.01±551.51	9251.30	97.00%		94956.92±769.63	21115.21	91.00%	
PB5	2138.15±3.71	29852.48	**95.00%**	x	2136.62±5.90	33728.52	86.00%	
PB6	769.57±10.49	51759.22	68.00%		770.64±10.04	46877.06	72.00%	
PB7	1026.34±6.92	92079.76	17.00%		1024.34±7.98	92400.93	12.00%	
PET7	16428.88±47.93	100100.00	0.00%		16451.34±50.91	98634.27	6.00%	
SENTO1	7640.90±50.75	100100.00	0.00%		7678.39±80.06	95481.81	14.00%	
SENTO2	8620.05±37.74	100100.00	0.00%		8649.13±50.80	99942.68	1.00%	
WEING8	566282.95±12678.93	100100.00	0.00%		583830.05±20597.21	100100.00	0.00%	
WEISHI30	10824.70±92.10	100100.00	0.00%		10962.33±189.93	99851.97	3.00%	
Average		73408.15	34.18%			75037.32	35.09%	

Benchmark	**IGA**				**aIGA**			
	Avg±Std	**Eval**	**Sucess**	**Pareto**	**Avg±Std**	**Eval**	**Sucess**	**Pareto**
PB1	3090.00±0.00	13912.02	**100.00%**	x	3090.00±0.00	17559.92	**100.00%**	x
PB2	3173.19±17.20	74237.64	51.00%		3173.47±18.83	72674.13	**54.00%**	x
PB4	95168.00±0.00	7231.01	**100.00%**	x	95168.00±0.00	8102.60	**100.00%**	x
PB5	2137.13±5.32	30514.89	89.00%		2136.79±5.72	34976.06	87.00%	
PB6	775.86±1.39	12657.32	99.00%		775.89±1.09	12355.48	99.00%	
PB7	1034.32±2.22	42747.69	**83.00%**	x	1034.12±2.62	43877.91	78.00%	
PET7	16529.44±10.97	80221.17	60.00%		16530.22±10.11	76512.13	64.00%	
SENTO1	7771.64±2.06	47496.00	**97.00%**	x	7770.61±3.98	39808.61	89.00%	
SENTO2	8717.77±5.71	87966.44	49.00%		8718.85±4.54	71824.83	55.00%	
WEING8	612963.36±2750.51	100100.00	0.00%		623388.14±1432.22	72758.08	65.00%	
WEISH30	11159.03±13.71	100100.00	0.00%		11190.72±1.02	51125.45	93.00%	
Average		54289.47	66.18%			45597.74	80.36%	

Benchmark	**BDE**				**aBDE**			
	Avg±Std	**Eval**	**Sucess**	**P**	**Avg±Std**	**Eval**	**Sucess**	**P**
PB1	3089.07±4.96	14104.50	96.00%		3089.54±3.52	13074.74	**98.00%**	x
PB2	3144.55±28.43	91164.94	14.00%		3165.17±24.20	78323.80	40.00%	
PB4	95168.00±0.00	4672.21	**100.00%**	x	95168.00±0.00	5584.56	**100.00%**	x
PB5	2135.60±6.80	32052.98	80.00%		2136.79±5.72	26676.96	87.00%	
PB6	775.86±1.39	7200.84	99.00%		776.00±0.00	6865.16	**100.00%**	x
PB7	1034.12±2.57	35502.17	77.00%		1034.47±1.89	33620.13	**82.00%**	x
PET7	16524.58±19.07	65795.81	56.00%		16529.52±15.30	64335.20	**71.00%**	x
SENTO1	7771.44±3.53	17091.08	**97.00%**	x	7770.66±4.61	25110.83	91.00%	
SENTO2	8720.37±3.49	50493.94	67.00%		8721.17±2.37	42285.83	**78.00%**	x
WEING8	624062.37±770.56	55705.08	86.00%		624241.30±457.11	34517.30	**95.00%**	x
WEISHI30	11191.00±0.00	33645.37	**100.00%**	x	11190.84±0.78	26192.99	**96.00%**	x
Average		37038.99	79.27%			**32417.04**	**85.27%**	

for the mutation and perturbation rates, respectively. Also, a convergence plot is show in Figure 3. All three figures were acquired during a successful run of aBDE algorithm using instance $PET7$. For other instances, the behavior observed was similar.

In the first generation of the algorithm, all possibilities for the values of parameters have the same probabilities to be chosen. Through generations, these probabilities can change according to their success of creating better solutions, as explained in Section III-A. From Figures 1 and 2 one can note that in earlier generations, the probabilities of the values for each parameter change more often than in latter generations.

This can be explained by the diversity loss that occurs during the optimization process, as can be seen in the convergence plot (Figure 3). The adaptive method is able to better explore the values of parameters at the beginning of the optimization process, favoring the best values until its end.

VII. CONCLUSION

In this work, a Binary Differential Evolution algorithm with adaptive parameters was applied to the well-known 0-1 MKP. The Adaptive Binary Differential Evolution (aBDE) algorithm aims to control two parameters: perturbation (PR) and mutation (MUT) rates. To achieve that, a set of discrete values is introduced for each of parameter and it is updated based on a criteria of success. If a selected value for a parameter yielded at least one individual in generation $t+1$ better than the best fitted individual from generation t, then the parameter value has a mark of success. Hence, if at the end of generation $t+1$ the parameter value was successful, its probability is increased, otherwise, it remains the same.

Results obtained using 11 instances of the problem strongly suggest that the adaptive selection strategy has advantages when compared with fixed values. This advantages can be seen

ISSN 2395-8618

in the results (average success rate and average number of function evaluations) when comparing aBDE with the other algorithms. This indicates that the proposed approach is an interesting and promising strategy for optimization of complex problems.

As future work, we intend to apply the adaptive method in other metaheuristics. Also, it is planed to investigate the performance of the aBDE in other real-world problems.

ACKNOWLEDGMENT

Authors would like to thank Fundação de Amparo a Pesquisa e Inovação do Estado de Santa Catarina (FAPESC) by the financial support, as well as to State University of Santa Catarina (UDESC).

REFERENCES

[1] M. Vasquez, J.-K. Hao *et al.*, "A hybrid approach for the 0-1 multidimensional knapsack problem," in *IJCAI*, 2001, pp. 328–333.

[2] A. Freville, "The multidimensional 0–1 knapsack problem: An overview," *European Journal of Operational Research*, vol. 155, no. 1, pp. 1–21, 2004.

[3] J. C. Bansal and K. Deep, "A modified binary particle swarm optimization for knapsack problems," *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11 042–11 061, 2012.

[4] M. A. K. Azad, A. M. A. Rocha, and E. M. Fernandes, "Improved binary artificial fish swarm algorithm for the 0–1 multidimensional knapsack problems," *Swarm and Evolutionary Computation*, vol. 14, pp. 66–75, 2014.

[5] L. Wang, X. long Zheng, and S. yao Wang, "A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem," *Knowledge-Based Systems*, vol. 48, no. 0, pp. 17–23, 2013.

[6] R. Storn and K. Price, "Differential evolution : A simple and efficient heuristic for global optimization over continuous spaces," *J. of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[7] K. De Jong, *Evolutionary Computation: A Unified Approach*, ser. Bradford Book. Mit Press, 2006.

[8] X.-S. Yang, "Chapter 6—differential evolution," in *Nature-Inspired Optimization Algorithms*. Oxford: Elsevier, 2014, pp. 89–97.

[9] J. Krause, J. Cordeiro, R. S. Parpinelli, and H. S. Lopes, "A survey of swarm algorithms applied to discrete optimization problems," *Swarm Intelligence and Bio-inspired Computation: Theory and Applications. Elsevier Science & Technology Books*, pp. 169–191, 2013.

[10] J. Krause, R. S. Parpinelli, and H. S. Lopes, "Proposta de um algoritmo inspirado em evolução diferencial aplicado ao problema multidimensional da mochila," *Anais do IX Encontro Nacional de Inteligência Artificial–ENIA. Curitiba, PR: SBC*, 2012.

[11] A. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, Jul 1999.

[12] L. André and R. S. Parpinelli, "Controle de parâmetros em inteligência de enxame e computação evolutiva," *Revista de Informática Teórica e Aplicada*, vol. 21, no. 2, pp. 83–128, 2014.

[13] D. Thierens, "An adaptive pursuit strategy for allocating operator probabilities," in *Proceedings of the 2005 conference on Genetic and evolutionary computation*. ACM, 2005, pp. 1539–1546.

[14] Á. Fialho, L. Da Costa, M. Schoenauer, and M. Sebag, "Extreme value based adaptive operator selection," in *Parallel Problem Solving from Nature–PPSN X*. Springer, 2008, pp. 175–184.

[15] A. Aleti and I. Moser, "Studying feedback mechanisms for adaptive parameter control in evolutionary algorithms," in *IEEE Congress on Evolutionary Computation (CEC)*, June 2013, pp. 3117–3124.

[16] O. Kramer, "Evolutionary self-adaptation: a survey of operators and strategy parameters," *Evolutionary Intelligence*, vol. 3, no. 2, pp. 51–65, 2010.

[17] L. André and R. S. Parpinelli, "An island-inspired genetic algorithm with adaptive parameters applied to the multiple knapsack problem," in *Proceedings of the 5th International Conference on Metaheuristics and Nature Inspired Computing*, October 2014, pp. 1–2.

[18] A. Hoff, A. Løkketangen, and I. Mittet, "Genetic algorithms for 0/1 multidimensional knapsack problems," in *Proceedings Norsk Informatikk Konferanse*. Citeseer, 1996, pp. 291–301.

ISSN 2395-8618

Classification of Group Potency Levels of Software Development Student Teams

Alberto Castro-Hernández, Kathleen Swigger, Fatma Cemile Serçe, and Victor Lopez

***Abstract*—This paper describes the use of an automatic classifier to model group potency levels within software development projects. A set of machine learning experiments that looked at different group characteristics and various collaboration measures extracted from a team's communication activities were used to predict overall group potency levels. These textual communication exchanges were collected from three software development projects involving students living in the US, Turkey and Panama. Based on the group potency literature, group-level measures such as skill diversity, cohesion, and collaboration were developed and then collected for each team. A regression analysis was originally performed on the continuous group potency values to test the relationships between the group-level measures and group potency levels. This method, however, proved to be ineffective. As a result, the group potency values were converted into binary labels and the relationships between the group-level measures and group potency were re-analyzed using machine learning classifiers. Results of this new analysis indicated an improvement in the accuracy of the model. Thus, we were able to successfully characterize teams as having either low or high potency levels. Such information can prove useful to both managers and leaders of teams in any setting.**

***Index Terms*—Software development, group potency, machine learning.**

I. Introduction

BECAUSE of the rapid rise of globalization within industry, the use of virtual teams has dramatically increased in recent years [1]. Despite its known advantages, such as reducing costs and obtaining access to workers with different skills [2], managing remote teams remains challenging, largely because of the difficulty in using electronic media to communicate with members located at remote sites [3]. Nevertheless, this same electronic media now allows managers to keep track of the actions and interactions that occur within a work team. Moreover, the data obtained from these electronic media can be converted into useful information for not only the managers of global teams [4], [5] but also researchers who are looking at various elements of group dynamics. However, determining which information is most useful and how it can be used to predict different team characteristics remains a serious challenge for group-focused researchers.

The search for various team measures, as well as the techniques for modeling the interactions of these measures, has been met with various degrees of success over the past few years. For example, [6] characterized learning groups as graphs, with vertices representing students and edges representing the number of messages interchanged bidirectionally. Using this model, [6] identified *Milson's communication patterns* and count specific *graph theory elements*. The authors then used this new dataset to create decision trees that predicted five levels of performance. The authors' model was able to predict performance with 78.9% accuracy.

The work by [7] is another example of group-related modeling study. The authors of this research developed a tool, called TeCFlow, to analyze the interaction among employees within a company. Interaction rates were calculated by counting the number of messages exchanged between pairs of workers. This information was then displayed in a graphical format. The software was also able to detect collaboration among subgroups by looking at *communication density*. Once a subgroup was detected, the *Group betweenness centrality* measure allowed the user to find interesting events that might have occurred during a specific period. Based on an analysis of data from email exchanges that occurred within a company, the authors argue that they were able to predict groups' productivity as well as suggest ways to improve a group's performance.

A study by [8] proposed a very different type of model that was based on the premise that groups perform better if they use similar words. The authors of this study tried to predict a group cohesion measure (obtained from an Interaction Rating Questionnaire) by calculating percentages for the number of times a team used nine function words (i.e., auxiliary verbs, articles, common adverbs, personal pronouns, indefinite pronouns, prepositions, negations, conjunctions and quantifiers) in their communications. These percentages were then averaged and labeled as the group's Linguistic Style Matching (LSM) index. The authors also calculated percentages for the number of times "We," Future-oriented, and Achievement-oriented words were used by each team. The authors used these four variables to construct regression models to predict cohesion and group performance. Using

Manuscript received on December 29, 2014, accepted for publication on April 27, 2015, published on June 15, 2015.

Alberto Castro-Hernández and Kathleen Swigger are with Computer Science Department, University of North Texas, USA (e-mail: albertocastro-hernandez@my.unt.edu, kathy@cse.unt.edu).

Fatma Cemile Serçe is with Department of Information Systems Engineering, Atilim University, Turkey (e-mail: cemileserce@gmail.com).

Victor Lopez is with Facultad de Ingeniería de Sistemas Computacionales, Universidad Tecnológica de Panamá, Panama (e-mail: victor.lopez@utp.ac.pa).

ISSN 2395-8618

data collected from group chat communications on a small, collaborative task, the authors found that LSM was able to predict cohesiveness and, to some degree, performance.

Despite the intense research activity in group-oriented research, there remain many questions about which measures to model and which modeling techniques are most accurate. One purpose of this paper is to determine the accuracy of our proposed variables in predicting group potency values. A second purpose of the paper is to compare the prediction accuracy of several commonly used modeling techniques (i.e., regression and machine classifiers). A third purpose of the paper is to identify specific feature values that can promote high levels of the group cohesion construct within a distributed software teams.

II. Group potency

The estimation of group potency within teams has been the focus of much research over the past several years. The group potency construct is usually defined as "a collective belief regarding the team's ability to be successful" (as cited by [9]). The importance of the construct was, and continues to be, its strong relationship to group performance [10], [11], [12]. This strong relationship between group potency and performance has been found at both the individual and group levels, although the strength between the two variables has been shown to be higher at the group level [13].

Group potency levels for teams are generally obtained by aggregating individual scores gathered through questionnaires, or by asking the team, as a whole, to agree upon a single score. The latter procedure usually produces higher group potency scores but lower correlations with group performance, because other team members often persuade team members to inflate their group potency scores. Aggregated questionnaires also seem to be the preferred method for obtaining group potency levels for virtual teams [11].

Not surprisingly, a number of theoretical models have been proposed to predict or explain group potency. For example, [14] proposed a model that used the variables of group composition, charismatic leadership, and group size to cause group potency. In a similar study, charismatic leadership was found to be related to group potency [15], [16], because, as the author explained, the presence of a leader helps guide other members toward the team's goals. Other researchers have found that a team's skill level, knowledge, and performance can have a positive effect on group potency [12], [10], largely because such factors tend to increase a group's confidence levels and, thus, affect the member's perception of the group's abilities. In a similar manner, researchers have determined that there is a positive relationship between group potency and communication and cooperation [10], [16]. Both of these factors allow team members to learn about each other's skills and capabilities, hence increasing the group's overall collective confidence. Finally, group size was found to be related to group potency [12], because, as [14] argues, groups with insufficient (or too many) members feel less confident about the team's ability to complete the task.

The group potency construct has also been compared to another construct called group efficacy [17]. Group potency focuses on the team's ability to perform generally, while group efficacy focuses more on the team's ability to complete a specific task [11]. Because research results indicate a strong relationship between group efficacy and team cohesiveness [18], it seens plausible to believe that group potency may also have a positive relation with team cohesion.

What is apparent from this cursory review of the research is that there are many variables that seem related to group potency. What is not so obvious is knowing which variables can be used to model group potency, and which modeling technique performs best with a particular dataset. These two issues were investigated using a database consisting of electronic communications from three global software development projects. The goal of this research was to find an effective model that can successfully predict group potency levels in virtual software development teams.

III. Research methodology

A. Teams

This research involved three sets of teams that participated in three different virtual collaborative projects during 2012-2013. The first set of teams consisted of students from the University of North Texas (UNT in the US) and students from the Atilim University (AU in Turkey). Participants from the US were enrolled in a Human-Computer Interface course, while participants from Turkey were enrolled in a Software Development course. A total of 53 students participated in this collaborative project; 23 US students and 30 Turkish students. Ten teams were created; each team consisting of 4-6 students, with members from both universities.

The second set of teams was made up of students from different courses within UNT. About half of the participants were enrolled in a Human-Computer Interfaces (HCI) course, and the other half were enrolled in an Artificial Intelligence (AI) course. A total of 50 students participated in this project; 28 students from the HCI course and 22 from the AI course. Ten teams were created for this project; each team consisting of between 4-6 members from both courses.

The third set of teams were formed from students enrolled at UNT and the Universidad Tecnológica de Panamá (UTA at Panama). Participants from the US were enrolled in a Human-Computer Interfaces course, and participants from Panama were enrolled in two different database courses. A total of 64 students participated in the third project; 28 students from the US and 36 from Panama. Thirteen teams were formed for this project, each team containing between 3-5 members from both universities.

In summary, the characteristics and behaviors of 33 teams were analyzed in an effort to predict group potency levels.

 http://dx.doi.org/10.17562/PB-51-8

ISSN 2395-8618

B. Software development projects

Separate, but similar, projects were created for each of the three sets of teams that participated in this study. The first project, involving US and Turkish students, occurred in November - December 2012 and lasted for 37 days. Each team was asked to complete a mobile application that could run on an Android phone. Sub teams in the US were responsible for developing the interface, while the Turkish teams implemented the mobile application.

The second collaborative project occurred in April–May 2013 and extended over a 35 day period. Student teams in the US were asked to develop an application that would use a reinforcement learning algorithm to decide where cars should park. The application was also suppose to include a display that would allow users to change the parameters to the algorithm. Sub teams in each course (AI and HCI) were asked to develop and test the application.

The third collaborative task occurred in November–December 2013 and lasted 37 days. Each US-Panama team was asked to re-design an existing website (i.e., the home page, the events page, and the contribution page) and implement a database that could support the various operations that were needed to maintain the pages. US sub team was in charge of developing the website, whereas Panamanian teams were responsible for designing and populating the database for the site.

C. Communication tools

A project-management web application based on the Redmine platform was used to collect the communication activities for each team. This application supports several collaborative tools including chat, forums, wikis, document sharing, etc. Additional programs were added to the Redmine tool that enabled the software to record and timestamp all interaction among team members and store them in a centralized database.

Students who participated in each project were asked to communicate with one another using only the Redmine project management tools. In addition, subjects were asked to use English to communicate with one another. Thus, both the Turkish and Panamanian participants were obviously using a second language to collaborate with the US students.

D. Measures

In order to determine which variables predict group potency levels for virtual teams, we developed three different predictor measures: *Team characteristics*, *Collaboration features*, and *Linguistic features*. Team characteristics represent group variables which are defined before the project's start, and they cannot be changed. Collaboration features describe variables which depend on team members' behavior during the project. Linguistic features are a detailed look into the messages' content exchanged.

The criterion variable of *Group potency level* was obtained by averaging a participant's responses to a group potency survey that was completed at the beginning of each project. This particular survey was developed by [14] and consists of eight questions in a five-point Likert scale that are designed to measure a subject's perceptions of their group's capabilities. The individual scores were then combined into a single *Group potency* score for each group.

1) Team characteristics: The *Team characteristics* variable was defined as *Team size*, *GPA average* (average of individual Grade Point Average) and *Team diversity*. The *Team size* measure was obtained by simply counting the number of members in a team. Both a team's *GPA average* and *Team diversity* scores were obtained by examining surveys completed by all subjects at the beginning of each project. A team's *GPA average* was computed by averaging the members' GPA's. The *Team diversity* score was operationalized as the inequality of GPAs among team members. Inequality was calculated by the Gini coefficent [19] of GPA values within a group. Gini coefficient has values from 0 (members' GPA are the same, or total GPA is distributed equally among team members) to 1 (total GPA comes from only one team member).

2) Collaboration feature characteristics: The *Collaboration feature* variable consisted of seven different measures: *Message average*, *Word average*, *Reply average*, *Message similarity*, *Word similarity*, *Reply Similarity*, and *Cohesion*. The *Message average* variable was computed by simply averaging the number of messages sent by group members. Similarly, the *Word average* for the group was computed by summing all the words in the teams' communications and then dividing the total by the number of members in the group. The *Reply Average* measure was defined as a reply to a message from a member who was different than the sender. The idea behind this measure is to try and capture the level of interaction among different members of a team.

Having collected the raw counts for a group's messages, words, and replies, we then calculated a similarity index for each of these measures. Thus, *Message similarity*, *Word similarity*, and *Reply similarity* were calculated as follows:

$$similarity_{ij} = 1 - \frac{abs(r_{ij} - r_{ji})}{r_{ij} + r_{ji}} \quad (1)$$

Where r_{ij} are the messages (words, replies) sent from member i to member j. A *Member's similarity* was then obtained by averaging all the paired similarity values, as shown in equation 2.

$$similarity_i = \frac{\sum_{j \in M, j \neq i} similarity_{ij}}{|M| - 1} \quad (2)$$

Where j are the teammates of i in team M. For a group-level measure, all team member's *Member's similarity* values were averaged (see equation 3).

$$group\_similarity = \frac{\sum_{i \in M} similarity_i}{|M|} \quad (3)$$

ISSN 2395-8618

These measures were based on the similarity measure proposed by [8]. The scores on each of these measures ranged between 0 and 1, with a 1 representing perfect similarity.

Previous research has found that cohesion is related to performance [20]. Researchers have also found a relationship between Group Potency and performance. Therefore, it seemed reasonable to assume that cohesion would be related to group potency. In order to test this relationship, a *Cohesion* measure was calculated by the LSM equation as proposed in [8]. It is important to mention that researchers who have used this measure have not found a relationship between LSM-based cohesion and performance in tasks that required virtual teams to communicate using emails [5]. Nevertheless, cohesion based on LSM has been used to show a positive relationship between cohesion and performance in chat communication settings [21], [22]. Since synchronous communication (e.g. chat) generates more messages among team members than asynchronous communication [23] (e.g. email), it is possible that group chats induce more language similarity among participants, causing an increase in group cohesiveness and, in turn, effecting group performance. Thus, we believe that in the chat setting described in this paper, the LSM cohesion measure was an appropriate measure to use to determine the relationships among cohesion, group potency and performance.

3) Linguistic features: The Linguistic Inquiry and Word Count (LIWC) tool [24], was used to identify linguistic clues that could help us understand the relationship between a team's language usage and group potency. LIWC is software that analyzes text on a word-by-word basis and calculates a percentage of words falling into one of 88 different categories. It can also be used to detect whether there are specific processes that high group potency teams use more or less than low group potency teams.

All communications from the three projects were analyzed using the LIWC software with the the English dictionary. In addition, the third project was analyzed using the Spanish dictionary, since there were some messages that contained Spanish words. The Spanish counts were then matched to the corresponding English category and included in the final percentages.

IV. Experiments

A total of 167 students participated in the three virtual software development projects. These students were, in turn, organized into 33 different teams. From this dataset, we extracted 1588 communication activities: 1193 chat messages, 388 forum posts, and 7 Wiki pages.

The pre-project survey data yielded profile information (i.e., age, GPA, etc.) for 99.4% of our participants. The missing profile information was estimated using Multiple Imputation method for missing data [25].

Group potency questionnaires were obtained from 74.85% of the participants. Using incomplete data to aggregate to the group-level will cause an overestimation of the group potency and agreement values [26]. Thus, to remove this bias, we corrected group potency values by using the Systematic Nonresponse Parameters (SNP) [27] approach for missing data. Only one team reported insufficient data to estimate a group potency level, so this team was removed from the final dataset. The Group potency average for all groups was 3.63 (SD=0.7146). The agreement within-group members was calculated by the Interrater Agreement (IRA) measure [28]. The IRA average was 77.00%.

TABLE I
Regression models on Group potency

Features	Model	Correlation	RAE
Team+Collaboration	Linear	-0.2918	112.78%
	SMO	0.1221	**97.85%**
Linguistic	Linear	0.1262	177.08%
	SMO	0.1136	178.22%

A. Regression models and Results

In order to test the strength of our model, we designed two feature sets to predict group potency:

- Team characteristics + Collaboration measures
- Linguistic features

Because group potency scores are continuous values, these feature sets were tested using two regression models: 1) Linear regression, and 2) Support Vector Machine for regression (SMO) [29].

Results from our analysis results show a low correlation between our two regression models and group potency (see Table I). Neither Team-Collaboration or Linguistic features were able to predict group potency using either Linear or SVM regression. Table I also reports the Relative Absolute Error (RAE) for each feature and model. The RAE percentage is a measure of extent to which the scheme is an improvement over using the average to predict the outcome variable; a scheme is considered better than average when the RAE percentage is lower than 100%. In Table I, the RAE percentages are above 100 for all the measures, except for the SMO regression model with Team + Collaboration features, which is only slightly better than the average.

A closer inspection of the group potency values for each team revealed that two of the teams appeared to have abnormal averages for their group potency levels (see the two points on the left in Figure 1). We confirmed that these two teams were indeed outliers according to the Chauvenet's criterion [30]. Thus, we removed these two teams from our dataset and did a second analysis. The group potency mean for the 31 remaining teams was 3.74 (SD=0.5682).

The results from the second analysis are presented in Table II. As shown in the table, RAE percentages improved for all models, but the overall correlations between the predictor variables and group cohesion were still low. The SMO technique again produced the best predictive model, but the improvement over using averages was only 8.69%.

ISSN 2395-8618

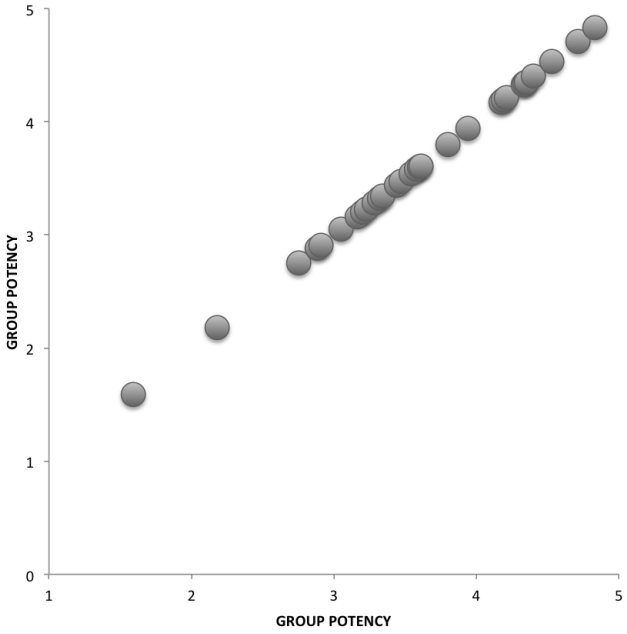


Fig. 1. Group potency values

TABLE II
REGRESSION MODELS (WITHOUT OUTLIERS) ON GROUP POTENCY

Features	Model	Correlation	RAE
Team+Collaboration	Linear	0.1108	98.20%
	SMO	0.3063	**91.31%**
Linguistic	Linear	−0.0486	153.09%
	SMO	0.081	130.34%

B. Binary classification

In order to improve upon our techniques for creating a model to predict group competency levels, several machine learning classifiers were used to test the predictive power of our variables. Since one of our objectives is to provide useful information about a group's internal state or status to managers or project leaders, we decided to convert the team data to a binary problem. Therefore, the group potency data was transformed to achieve better results for binary classification. All thirty-one groups were first ordered according to their group potency level scores. The top fifteen teams were labelled as the *High potency* group, and the bottom fifteen teams were labeled as the *Low potency* group. One team was removed from the analysis in order to maintain a balanced dataset. This new dataset was then analyzed using two common classifiers and two ensemble methods: 1) Support Vector Machine (SMO), 2) Naive Bayes (NB), 3) Bagging-REPTree (Bag), and 4) AdaBoost-DecisionStump (Ada).

Table III shows a comparison of the RAE percentages for the four different classifiers. As is normal, each classifier's accuracy rates are also reported. As anticipated, the RAE percentages for all the classifiers are much lower than in the previous experiments, indicating that our features were much more accurate at predicting group potency when using the binary classification methods as opposed to regression techniques. We also observed that the SMO classifier was a better model for predicting group potency using the combined Team-Collaboration features, whereas the Ada classifier was a better predictor when using only the Linguistic features. These same results are reflected in the accuracy rates reported in Table III.

TABLE III
CLASSIFICATION OF GROUP POTENCY LEVELS

Features	Classifier	Accuracy	RAE
Team+Collaboration	SMO	**70.00%**	**59.31%**
	NB	56.66%	88.10%
	Bag	43.33%	98.56%
	Ada	56.66%	94.96%
Linguistic	SMO	40.00%	118.63%
	NB	50.00%	102.66%
	Bag	53.33%	95.15%
	Ada	**70.00%**	**73.36%**

TABLE IV
ENSEMBLE METHODS ON GROUP POTENCY'S CLASSIFICATION

Features	Classifier	Accuracy	RAE
Team+Collaboration	Bag-SMO	**73.33%**	**67.22%**
	Ada-SMO	63.33%	80.88%

Since the SMO classifier appeared to outperform the other classifying techniques, we then tried to improve the predictive capabilities of this classifier by adding some additional "boosting" power in the form of the AdaBoost and Bagging methods (with Team + Collaboration features). Results of these analyses are presented in Table IV. As reported in Table IV, only the Bag-SMO classifier showed an improvement in the accuracy rate of the classifier. However, the Bag-SMO classifier had a higher RAE percentage. A closer look at the outputs of the two classifiers showed that the SMO classifier was much better at predicating whether an instance was going to be *low potency* versus *high potency*. On the other hand, the Bag-SMO classifier was much better at identifying the exact potency level of an instance, with probabilities ranging from 0.6 to 1. Therefore, the Bag-SMO classifier tended to have higher RAE percentages.

Finally, we tested the three best machine learning classifiers (i.e., SMO, SMO-Bag, Ada-SMO) on a combined dataset of all three predictor measures. Results (see Table V) showed that the predictive powers of these classifiers were not as high as the previous experiment. Perhaps the performance of the classifiers was affected by the normalization of the data. That is, the classifiers may have had difficulty recognizing a feature that could satisfy a "Team+Collaboration or Linguistic" condition, since all of its features were collapsed into a single feature representation.

V. FEATURES RELATED TO HIGH-LEVEL GROUP POTENCY

The high accuracy levels of our binary classifiers led to a more detailed investigation of the specific features that might

ISSN 2395-8618

TABLE V
GROUP POTENCY'S CLASSIFICATION WITH ALL FEATURES

Features	Classifier	Accuracy	RAE
Team+Collaboration+Linguistic	SMO	53.33%	92.27%
	Bag-SMO	56.66%	91.61%
	Ada-SMO	**60.00%**	**80.09%**

have facilitated (or impeded) group potency levels in teams. Therefore, we looked at the output from the best classifiers (i.e., SMO with Team + Collaboration features, and Ada with Linguistic features) and examined the feature values that were used to predict the teams with high group potency levels. Table VI presents the features related to high group potency within teams. It should be noted that the features listed within parenthesis had a negative relationship with high group potency teams.

According to the results from the SMO classifier, negative *Message similarity* and negative *Word similarity* were related to high group potency. Since these two features were correlated with one another, as well as negatively related to high group potency, these results seem to suggest that a single person within the team may have been responsible for most of the communications. This is not an uncommon occurrence in virtual student team projects where it is often the case that a single leader emerges to help manage the task. As seen in other literature [31], individuals that emerge as a leader are often the people who produce the most communications in the teams. This is generally seen as a good thing, because a leader often causes the group to work more closely with one another. The negative relationship between *Reply similarity* and group potency seems to support our emergent leadership theory and shows that such a condition can help strengthen group potency within teams.

There were three variables that had a positive relationship with high group potency levels: *Team size*, *Word average*, and *GPA*. The positive relationship between *Size* and high group potency indicates that students believe that they are more likely to complete the task with more, rather than fewer, team members. The positive relationship between *Word average* and potency levels show that more participation provokes a higher perception of group potency within the team. Finally, the positive relationship of *GPA average* indicates the importance of the skill level of the participants to the potency construct.

The AdaBoost classifier, using the Linguistic features, produced 10 Decision Stump trees. The best performing features are listed in Table VI. The results of this analysis indicate that high group potency teams use fewer "I" words than low group potency teams. According to [32], the use of pronouns tends to show a person's focus. In this context, it appears that low group potency teams pay more attention to themselves (i.e., use of I) as opposed to high potency groups who focus on other group members (i.e., use of "You"). At the same time, low potency teams tend to communicate more about personal matters, such as health (i.e., the use of "Biological Process" words), than high potency groups. The "Verb" category was also negatively related to high potency levels. In a more detailed analysis of the corresponding subcategories within the Verb category, we found that low group potency teams used a much higher percentage of verbs related to the past and present subcategories than high group potency teams. On the other hand, high group potency teams used a higher percentage of verbs related to the future category than low group potency teams. It has been reported that future-oriented words can be be linked to performance indicators [5]. Thus, it is possible that high potency teams tend to use more future verbs because they are more focused on the project's tasks.

The literature on LIWC [32] also argues that the use of the other pronouns, such as "you," indicates that a speaker is more socially oriented. Thus, it appears to high group potency teams are more social than low group potency teams because of their more frequent use of "You" words. The positive relation between high group potency and prepositions suggests these high potency teams exchanged more complex information about a topic [33] than teams with low potency levels.

VI. CONCLUSIONS

In this study, we examined models for predicting group potency levels using aggregated variables that captured team characteristics, collaborative behaviors, and language use. At the same time, we explored a number of modeling techniques to determine which method would yield more accurate results when using typical global software development data to predict group potency. A series of virtual software development projects were developed to collect collaboration data through a distributed collaborative software system. These projects involved students from the US, Turkey, and Panama who worked together in distributed teams. Data obtained from groups' communication activities and surveys were used to predict the group potency construct.

Initial results involving the regression approach showed only a slight improvement over using the mean group potency score. Therefore, the group potency prediction task was converted to a binary classification problem, and several machine learning algorithms were tested and compared. The Bag-SMO classifier yielded the highest accuracy rates (i.e., 73.33%) using the Team + Collaboration feature dataset, while the SMO classifier had the lowest RAE percentage (i.e., 59.31%) on this same dataset. The AdaBoost (Decision Stump) classifier showed the highest accuracy rate (70%) using the Linguistic feature dataset. One explanation for the differences between the RAE percentages with continuous data versus the binary classifiers is that the transformation of the data into two groups allowed the differences among the different predictor variables to emerge. In a similar manner, the reason that the accuracy levels among the machine learning algorithms differed when using the two different features sets (i.e., Team + Collaboration versus Linguistic features) is that the Team + Collaboration features were much more correlated with one another than the Linguistic features. Thus, the results from the different

ISSN 2395-8618

TABLE VI
TOP FEATURES FROM BEST BINARY CLASSIFIERS FOR HIGH GROUP POTENCY TEAMS

Algorithm	Features
SMO with Team + Collaboration	(Message similarity), (Word similarity), Size, Word average, (Reply similarity) and GPA average
Ada with Linguistic	(I), (Biological processes), (Verbs), You, Prepositions

classifiers seemed to be affected by high (or low) variability in the two feature datasets.

The results from the machine learning models were then used to identify the particular values of the linguistic features that were used to predict group potency among team members. These results showed that high group potency teams sent fewer messages and seemed to be more diverse in their language use and message replies than low group potency teams. One explanation for these differences is that high potency teams may have had a leader (which we dubbed as "emergent") who, while dominating the conversation, was able to engender confidence among group members. Not surprising, the Collaborative measures of Size and GPA were also related to high potency group levels.

Results from our linguistic analysis indicated that high potency teams tended to be more focused on their team members (hence the use of "You" words) and communicated more about the task and future events than low group potency teams. In contrast, low potency teams talked more about themselves (hence the use of "I" words) and personal matters and tended to focus on the present and the past.

Although our initial attempt to predict group potency was not successful, we were able to obtain reasonable results by converting the task into a binary classification problem. Despite problem's conversion to binary classification reduce its outcome values, we believe that being able to predict low or high group-potency levels among global teams and provide this information to their corresponding leaders may result in proper interventions to reach a higher distributed team performance.

ACKNOWLEDGMENTS

The first author thanks Veronica Perez-Rosas for her help in the classifiers' selection for some experiments. Also, he gratefully acknowledges financial support from a CONACYT scholarship and from the Support for Graduate Studies Program of SEP. This material was also based upon work supported by the National Science Foundation under Grant No. 0705638.

REFERENCES

[1] J. D. Herbsleb and D. Moitra, "Global software development," *Software, IEEE*, vol. 18, no. 2, pp. 16–20, 2001.

[2] P. J. Agerfalk, B. Fitzgerald, H. H. Olsson, and E. Ó. Conchúir, "Benefits of global software development: the known and unknown," in *Making Globally Distributed Software Development a Success Story*. Springer, 2008, pp. 1–9.

[3] A. M. Townsend, S. M. DeMarie, and A. R. Hendrickson, "Virtual teams: Technology and the workplace of the future," *The Academy of Management Executive*, vol. 12, no. 3, pp. 17–29, 1998.

[4] R. P. Biuk-Aghai and S. J. Simoff, "An integrative framework for knowledge extraction in collaborative virtual environments," in *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, ser. GROUP'01. New York, NY, USA: ACM, 2001, pp. 61–70. [Online]. Available: http://doi.acm.org/10.1145/500286.500298

[5] S. A. Munson, K. Kervin, and L. P. Robert Jr, "Monitoring email to indicate project team performance and mutual attraction," in *Proceeding of CSCW'14, 17th ACM conference on Computer supported cooperative work & social computing*, 2014, pp. 542–549.

[6] G. Chen, C. Wang, and K. Ou, "Using group communication to monitor web-based group learning," *Journal of Computer Assisted Learning*, vol. 19, no. 4, pp. 401–415, 2003.

[7] P. A. Gloor and Y. Zhao, "Tecflow-a temporal communication flow visualizer for social networks analysis," in *CSCW'04 Workshop on Social Networks*, 2004.

[8] A. L. Gonzales, J. T. Hancock, and J. W. Pennebaker, "Language style matching as a predictor of social dynamics in small groups," *Communication Research*, vol. 37, no. 1, pp. 3–19, Feb. 2010.

[9] J. Mathieu, M. T. Maynard, T. Rapp, and L. Gilson, "Team effectiveness 1997-2007: A review of recent advancements and a glimpse into the future," *Journal of Management*, vol. 34, no. 3, pp. 410–476, Jun. 2008.

[10] A. De Jong, K. De Ruyter, and M. Wetzels, "Antecedents and consequences of group potency: A study of self-managing service teams," *Management Science*, vol. 51, no. 11, pp. 1610–1625, 2005.

[11] A. M. Hardin, M. A. Fuller, and J. S. Valacich, "Measuring group efficacy in virtual teams new questions in an old debate," *Small Group Research*, vol. 37, no. 1, pp. 65–85, Feb. 2006. [Online]. Available: http://sgr.sagepub.com/content/37/1/65

[12] A. E. Akgün, H. Keskin, J. Byrne, and S. Z. Imamoglu, "Antecedents and consequences of team potency in software development projects," *Information & Management*, vol. 44, no. 7, pp. 646–656, 2007.

[13] S. M. Gully, K. A. Incalcaterra, A. Joshi, and J. M. Beaubien, "A meta-analysis of team-efficacy, potency, and performance: interdependence and level of analysis as moderators of observed relationships," *Journal of Applied Psychology*, vol. 87, no. 5, p. 819, 2002.

[14] R. A. Guzzo, P. R. Yost, R. J. Campbell, and G. P. Shea, "Potency in groups: Articulating a construct," *British Journal of Social Psychology*, vol. 32, no. 1, pp. 87–106, 1993.

[15] N. Sivasubramaniam, W. D. Murry, B. J. Avolio, and D. I. Jung, "A longitudinal model of the effects of team leadership and group potency on group performance," *Group & Organization Management*, vol. 27, no. 1, pp. 66–96, Mar. 2002. [Online]. Available: http://gom.sagepub.com/content/27/1/66

[16] S. W. Lester, B. M. Meglino, and M. A. Korsgaard, "The antecedents and consequences of group potency: A longitudinal investigation of newly formed work groups," *Academy of Management Journal*, pp. 352–368, 2002. [Online]. Available: http://www.jstor.org/stable/10.2307/3069351

[17] A. D. Stajkovic, D. Lee, and A. J. Nyberg, "Collective efficacy, group potency, and group performance: meta-analyses of their relationships, and test of a mediation model," *Journal of Applied Psychology*, vol. 94, no. 3, p. 814, 2009.

[18] D. M. Paskevich, L. R. Brawley, K. D. Dorsch, and W. Neil, "Relationship between collective efficacy and team cohesion: Conceptual and measurement issues," *Group Dynamics: Theory, Research, and Practice*, vol. 3, no. 3, pp. 210–222, 1999.

[19] T. Ogwang, "A convenient method of computing the gini index and its standard error," *Oxford Bulletin of Economics and Statistics*, vol. 62, no. 1, pp. 123–129, 2000.

[20] S. M. Gully, D. J. Devine, and D. J. Whitney, "A meta-analysis of cohesion and performance effects of level of analysis and task interdependence," *Small Group Research*, vol. 26, no. 4, pp. 497–520, 1995.

ISSN 2395-8618

[21] V. L. Schwanda, K. Barron, J. Lien, G. Schroeder, A. Vernon, and J. T. Hancock, "Temporal patterns of cohesiveness in virtual groups," in *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, ser. CSCW'11. New York, NY, USA: ACM, 2011, pp. 709–712.

[22] Y. R. Tausczik and J. W. Pennebaker, "Improving teamwork using real-time language feedback," in *Proceedings of Human Factors in Computing Systems (CHI)*, 2013, pp. 459–468.

[23] F. C. Serçe, K. Swigger, F. N. Alpaslan, R. Brazile, G. Dafoulas, and V. Lopez, "Online collaboration: Collaborative behavior patterns and factors affecting globally distributed team performance," *Computers in Human Behavior*, vol. 27, no. 1, pp. 490–503, Jan. 2011.

[24] J. W. Pennebaker, C. K. Chung, M. Ireland, A. Gonzales, and R. J. Booth, "The development and psychometric properties of LIWC2007," *Austin, TX, LIWC. Net*, 2007.

[25] R. R. Hirschfeld, M. S. Cole, J. B. Bernerth, and T. E. Rizzuto, "Voluntary survey completion among team members: Implications of noncompliance and missing data for multilevel research," *Journal of Applied Psychology*, vol. 98, no. 3, pp. 454–468, 2013.

[26] D. A. Newman and H.-P. Sin, "How do missing data bias estimates of within-group agreement? Sensitivity of SD WG, CVWG, rWG(J), rWG(J)* , and ICC to systematic nonresponse," *Organizational Research Methods*, vol. 12, no. 1, pp. 113–147, Jan. 2009.

[27] D. A. Newman, C. Lance, and R. Vandenberg, "Missing data techniques and low response rates," *Statistical and methodological myths and urban legends*, pp. 7–36, 2009.

[28] M. K. Lindell, C. J. Brandt, and D. J. Whitney, "A revised index of interrater agreement for multi-item ratings of a single target," *Applied Psychological Measurement*, vol. 23, no. 2, pp. 127–135, Jun. 1999. [Online]. Available: http://apm.sagepub.com/content/23/2/127

[29] A. J. Smola and B. Schoelkopf, "A tutorial on support vector regression," 1998, NeuroCOLT2 Technical Report NC2-TR-1998-030.

[30] H. D. Young, *Statistical treatment of experimental data*. McGraw-Hill, 1962.

[31] I. Brooks and K. Swigger, "Using sentiment analysis to measure the effects of leaders in global software development," in *International Conference on Collaboration Technologies and Systems (CTS)*, 2012, pp. 517–524.

[32] J. Pennebaker, *The Secret Life of Pronouns: What Our Words Say About Us*. Bloomsbury USA, 2013.

[33] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: LIWC and computerized text analysis methods," *Journal of language and social psychology*, vol. 29, no. 1, pp. 24–54, 2010.

ISSN 2395-8618

Soft Cardinality in Semantic Text Processing: Experience of the SemEval International Competitions

Sergio Jimenez, Fabio A. Gonzalez, and Alexander Gelbukh

***Abstract*—Soft cardinality is a generalization of the classic set cardinality (i.e., the number of elements in a set), which exploits similarities between elements to provide a "soft" counting of the number of elements in a collection. This model is so general that can be used interchangeability as cardinality function in resemblance coefficients such as Jaccard's, Dice's, cosine and others. Beyond that, cardinality-based features can be extracted from pairs of objects being compared to learn adaptive similarity functions from training data. This approach can be used for comparing any object that can be represented as a set or bag. We and other international teams used soft cardinality to address a series of natural language processing (NLP) tasks in the recent SemEval (semantic evaluation) competitions from 2012 to 2014. The systems based on soft cardinality have always been among the best systems in all the tasks in which they participated. This paper describes our experience in that journey by presenting the generalities of the model and some practical techniques for using soft cardinality for NLP problems.**

***Index Terms*—Similarity measure, soft computing, set cardinality, semantics, natural language processing.**

I. Introduction

THE SemEval[1] (Semantic Evaluation) competition is a series of academic workshops which aims to bring together the scientific community in the field of natural language processing (NLP) around tasks involving automatic analysis of texts. Each year, a set of challenges is proposed dealing with different aspects of the area of computational semantics attracting the attention of research groups of institutions worldwide. Each challenge follows a peer reviewing screening process ensuring the relevance, correctness, quality, and fairness of each competition. Task organizers pose an interesting challenge by providing a new dataset and a methodology for evaluating systems that address that challenge. For instance, organizers of the semantic textual similarity task (STS) provide several training datasets containing pairs of short texts labeled with a gold standard built using human annotators. Next, participating teams build systems that predict annotations in unseen test data, and organizers evaluate the performance of each system. Finally, organizers and participants describe their experiences and used approaches in peer-reviewed articles, which become de facto state of the art methods.

The authors, researchers from the Universidad Nacional de Colombia and the Centro de Investigación en Computación of the IPN in Mexico, collaborated to participate in several SemEval tasks since 2012. The core component of our participating systems is soft cardinality [1], a recently proposed approach to make the classic cardinality of set theory sensitive to the similarities and differences between the elements in a collection. This approach is particularly appropriate for addressing NLP problems because it allows finding commonalities between texts that do not share words but have words that are similar in some degree. Somehow surprisingly, systems build with this simple approach obtained impressive results in several SemEval challenges defeating considerably more complex and costly approaches. In addition, our team was the one with the highest number of participations from 2012 to 2014 also using the same core approach for addressing all tasks and always obtaining very satisfactory results.

This paper describes our experience in that journey by reviewing our participations in SemEval. Section II presents a brief description of soft cardinality, some parameterized resemblance coefficients, and the method for extracting cardinality-based feature representations. Section III presents some of the techniques and resources used for addressing NLP tasks using soft cardinality. Section IV reviews the systems and particular tasks addressed in SemEval, and a summary of the obtained results is presented. Finally in Section V we provide some concluding remarks.

II. Soft cardinality approach

The cardinality of a collection of elements is the counting of non-repeated elements within. This definition is intrinsically associated with the notion of set, which is a collection of non-repeating elements. The notation of the cardinality of a collection or set A is $|A|$. Jimenez et al. [1] proposed

Manuscript received on February 17, 2015, accepted for publication on May 27, 2015, published on June 15, 2015.

Sergio Jimenez and Fabio A. Gonzalez are with the Departamento de Ingeniería de Sistemas e Industrial of the Universidad Nacional de Colombia, Bogota, Colombia (e-mail: fagonzalezo@unal.edu.co, sergio.jimenez.vargas@gmail.com).

Alexander Gelbukh is with the Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico City, Mexico (e-mail: gelbukh@gelbukh.com).

[1]http://en.wikipedia.org/wiki/SemEval

ISSN 2395-8618

soft cardinality, which uses a notion of similarity between elements for grouping not only identical elements but similar too. That notion of similarity between elements is provided by a similarity function that compares two elements a_i and a_j returning a score in [0,1] interval having $sim(x, x) = 1$. Although, it is not necessary that sim fulfills another mathematical property aside identity, symmetry is also desirable. Thus, the soft cardinality of a collection A, whose elements $a_1, a_2, \ldots, a_{|A|}$ are comparable with a similarity function $sim(a_i, a_j)$, is denoted as $|A|_{sim}$. This soft cardinality is given by the following expression:

$$|A|_{sim} = \sum_{i=1}^{|A|} \frac{w_{a_i}}{\sum_{j=1}^{|A|} sim(a_i, a_j)^p} \quad (1)$$

It is trivial to see that $|A| = |A|_{sim}$ if either $p \to \infty$ or when the function sim is a crisp comparator, i.e., one that returns 1 for identical elements and 0 otherwise. This property shows that soft cardinality generalizes classic cardinality and that the parameter p controls its degree of "softness", the default value is $p = 1$. The values w_{a_i} are optional "importance" weights associated with each element a_i, by default those weights can be assigned to 1.

A. Inferring intersection cardinality

The soft cardinality of the intersection of two collections cannot be calculated directly from $A \cap B$ because the intersection operator is inherently crisp. This means that, if there are no common elements between A and B, their intersection is empty, and so its soft cardinality is 0. The following definition allows inferring the soft cardinality of the intersection through soft cardinalities of each collection and their union.

Let A and B be two collections, the soft cardinality of their intersection is $|A \cap B|_{sim} = |A|_{sim} + |B|_{sim} - |A \cup B|_{sim}$. In this case, the operator $\cup$ means *bag union*, which takes the maximum number of occurrences of the elements in each bag. Example: $\{1, 1, 2, 3\} \cup \{1, 2, 2\} = \{1, 1, 2, 2, 3\}$ [2].

This infers non-empty intersections for pairs of collections that have not common elements, but have similar elements. Once $|A \cup B|_{sim}$, $|A \cap B|_{sim}$, $|A|_{sim}$ and $|B|_{sim}$ are known, it is possible to obtain all other areas in the Venn's diagram of two sets, i.e., $|A \triangle B|_{sim} = |A \cup B|_{sim} - |A \cap B|_{sim}$, $|A \setminus B|_{sim} = |A|_{sim} - |A \cap B|_{sim}$ and $|B \setminus A|_{sim} = |B|_{sim} - |A \cap B|_{sim}$. These are the building blocks of almost any cardinality-based resemblance coefficient.

B. Cardinality-based resemblance coefficients

Since more than a century when Jaccard [3] proposed his well-known index, the classic set cardinality has been used to build similarity functions for set comparison. Basically, any cardinality-based similarity function is an algebraic combination of $|A|$, $|B|$ and either $|A \cap B|$ or $|A \cup B|$ (e.g. Jaccard, Dice [4], Tversky [5], overlap and cosine [6] coefficients). Table I shows some of the most used resemblance coefficients.

TABLE I
NAMED RESEMBLANCE COEFFICIENTS

Resemblance coefficient	$SIM(A, B) =$
Jaccard [3]	$\frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$
Dice or Sørensen [4]	$\frac{\lvert A \cap B \rvert}{0.5(\lvert A \rvert + \lvert B \rvert)}$
Overlap	$\frac{\lvert A \cap B \rvert}{\min(\lvert A \rvert, \lvert B \rvert)}$
Cosine or Ochiai [6]	$\frac{\lvert A \cap B \rvert}{\sqrt{\lvert A \rvert \cdot \lvert B \rvert}}$
Hamming	$\frac{1}{1 + \lvert A \triangle B \rvert}$

The simplest way to build similarity functions with soft cardinality is to replace the classic cardinality $|*|$ by soft cardinality $|*|_{sim}$. These coefficients have mathematical properties (e.g. transitivity, metric properties) that make of them a good option for many applications. When cosine coefficient is used in combination with soft cardinality, the resulting approach is conceptually similar to the soft cosine measure proposed by Sidorov et al. [7].

C. Parameterized resemblance coefficients

Some resemblance coefficients contain in its formulation parameters that allow adaptation to particular tasks. One of them is the Tversky's index [5], which was proposed as a cognitive model of similarity:

$$\begin{aligned} SIM(A, B) &= \frac{|A \cap B|}{\alpha|A \setminus B| + \beta|B \setminus A| + |A \cap B|}; \\ \alpha, \beta &\geq 0 \end{aligned}$$

There, parameters α and β control the balance of the differences between A and B. In Tversky's model, one of the sets being compared is the *referent* and the other is the *variant*, making this similarity measure asymmetric when $\alpha \neq \beta$. This asymmetry makes of Tversky's model an inclusion measure rather than a similarity measure. Nevertheless, in its original form it is still useful in text applications where the texts being compared have an ordinal relation, e.g. question-answer in question answering, query-document in information retrieval, text-hypothesis in textual entailment, text-summary in summarization, and others. In applications such as textual similarity or paraphrase detection, symmetry plays an important role. Jimenez et al. [8] proposed a symmetrization of Tversky's index in the following way:

$$\begin{aligned} SIM(A, B) &= \frac{c}{\beta(\alpha a + (1 - \alpha)b) + c} \quad (2) \\ |c| &= |A \cap B| + bias, \\ a &= \min[|A \setminus B|, |B \setminus A|], \\ b &= \max[|A \setminus B|, |B \setminus A|]. \end{aligned}$$

This formulation also re-arranges parameters α and β in a way that α controls the balance between the differences of

 http://dx.doi.org/10.17562/PB-51-9

ISSN 2395-8618

TABLE II
THE BASIC AND DERIVED FEATURE SETS FOR THE COMPARISON TWO COLLECTIONS OF WORDS.

Basic	Derived set 1	Derived set 2
$\|A\|$	$\|A \cap B\| = \|A\| + \|B\| - \|A \cup B\|$	$\max(\|A\|, \|B\|)$
$\|B\|$	$\|A \triangle B\| = \|A \cup B\| - \|A \cap B\|$	$\min(\|A\|, \|B\|)$
$\|A \cup B\|$	$\|A \setminus B\| = \|A\| - \|A \cap B\|$	$\max(\|A \setminus B\|, \|B \setminus A\|)$
	$\|B \setminus A\| = \|B\| - \|A \cap B\|$	$\min(\|A \setminus B\|, \|B \setminus A\|)$

A and B, and β controls the importance in the denominator between differences and commonalities between A and B. The additional parameter $bias$ allows removing an implicit degree of similarity between *A* and *B*, so usually $bias \leq 0$. This parameter can also be associated with the average or minimum intersections in a dataset. This coefficient generalizes Jaccard ($\alpha = \beta = 1; bias = 0$), Dice ($\alpha = \beta = 0.5; bias = 0$), overlap ($\alpha = 1; \beta = 0; bias = 0$) and Hamming ($\alpha = 1; \beta = 1; bias = 1 - |A \cap B|$).

Another generalization can be made by the observation that Dice and cosine coefficients are the ratio of $|A \cap B|$ and the arithmetic and geometric means, respectively. Therefore, the denominator can be replaced the expression of the generalized mean between $|A|$ and $|B|$:

$$SIM_p(A, B) = \frac{|A \cap B|}{0.5(|A|^p + |B|^p)^{1/p}} \quad (3)$$

Different values of the parameter p produce different known coefficients, i.e., Dice ($p = 1$), cosine ($p \to 0$) and overlap ($p \to \infty$). Other interesting values of p correspond to known means: $p = -1$ is the harmonic mean, $p = 2$ is the quadratic mean and $p \to -\infty$ is the minimum.

De-Baets and De-Meyer [9] proposed another hexaparametric generalized resemblance coefficient (a and b as in Eq. 2:

$$SIM(A, B) = \frac{\alpha a + \beta b + \delta |A \cap B|}{\alpha' a + \beta' b + \delta' |A \cap B|}$$

The values selected for parameters in resemblance coefficients are usually obtained by optimizing some criterion using training data. For example, in a dataset that consist of triples (A, B, g_{AB}) where g_{AB} is a gold standard of similarity (e.g. agreement of human judgments), the optimal set of parameters can be obtained by maximizing the correlation (Pearson or Spearman) between $SIM(A, B)$ and g_{AB} or by minimizing the mean-absolute error (MAE) or root-mean-squared error (RMSE).

D. Cardinality-based features for machine learning models

The parameterized resemblance coefficients allow the exploration and adaptation of a relatively large set of similarity functions to a particular problem. However, the space of possible formulations of similarity functions is huge. Which is the most appropriate similarity function for a particular problem is a question that can be addressed by adjusting parameters in these coefficients, but this strategy is nothing more than an arbitrary bias in the search. In this case, "a problem" means a dataset that needs to be modeling or explained by the similarity function. An exhaustive exploration of candidate similarity function is out of question given the large number of possible formulations. Genetic programming [10] can be used for this, but still the considered functions might be unable to model local non-linearities in some datasets. Using machine learning methods may be an appropriate option to address these issues.

Most machine learning algorithms builds models using a fixed features set (i.e., a vector) to represent each sample (e.g. linear regression, support vector machines, naïve Bayes, decision trees, *K*-means, etc.) Training data is a set of samples wherein each sample is associated with a target variable, a similarity score in our scenario. These labeled samples are used to construct a black box model, which is able, to some extent, to predict the target variable, and it is also able of producing predictions for unlabeled data. There is a variety of methods for obtaining these black box models including approaches whether geometric, probabilistic, algorithmic, information theoretical, among many others. This approach allows learning a similarity function adapted to the problem at hand efficiently and generally with a good level of generalization.

The proposed approach consists in extracting a fixed set of features from each pair of sample objects A and B, building a training dataset using these features, and labeling each sample with a gold-standard of similarity. Next, this training dataset is used to learn a machine learning model for the target variable. Finally, the learned model is used to provide similarity scores for other pairs of objects by extracting from them the same features set.

The proposed features for each pair of objects are based on cardinality, using either classical or soft cardinality. Thus, for a pair of objects (A, B) represented as sets (or bags), the basic set of cardinality-based features consist of $|A|$, $|B|$ and $|A \cup B|$. All other possible cardinality-based features are mathematical combinations thereof these three features. The following obvious features are the other areas in the Venn's diagram of two sets, i.e., $|A \cap B|$, $|A \triangle B|$, $|A \setminus B|$ and $|B \setminus A|$, Table II shows the basic and derived set of features described. An additional set of features aimed to enable machine learning algorithms to identify symmetrical patterns in the objects being compared is built using $\min()$ and $\max()$ functions, see "Derived set 2" in Table II.

Although, many machine learning methods requires or includes previous pre-processing steps of normalization or standardization of the features. Therefore, it makes sense to produce some features whose values are limited in a range.

ISSN 2395-8618

TABLE III
SET OF TEN EXTENDED RATIONAL FEATURES.

	Feature expression		Feature expression
#1	$\frac{\|A\|}{\|A\cup B\|}$	#6	$\frac{\|B\|-\|A\cap B\|}{\|B\|}$
#2	$\frac{\|A\|-\|A\cap B\|}{\|A\|}$	#7	$\frac{\|B\|-\|A\cap B\|}{\|A\cup B\|}$
#3	$\frac{\|A\|-\|A\cap B\|}{\|A\cup B\|}$	#8	$\frac{\|A\cap B\|}{\|B\|}$
#4	$\frac{\|A\cap B\|}{\|A\|}$	#9	$\frac{\|A\cap B\|}{\|A\cup B\|}$
#5	$\frac{\|B\|}{\|A\cup B\|}$	#10	$\frac{\|A\cup B\|-\|A\cap B\|}{\|A\cup B\|}$

Table III shows an extended set of features limited to [0,1] interval. These features are aimed to allow machine learning algorithms for learning patterns from the relative proportions of cardinality magnitudes. In the context of text applications, these rational features allow identifying patterns that are independent of the length of texts.

E. Exploring larger sets of features

Feature sets presented in the previous section have shown effective to address many natural language processing challenges at SemEval competitions. Despite their effectiveness, they seem to be arbitrary. For example, features shown in Table III are rational combinations of some of the features in Table II. Why only select these ten combinations? In fact, if the Table II contains 11 features and number 1 is added to this set, then the number of possible combinations of rational features is $12 \times 11 = 131$. With this new set of 131 features, the ten features in Table III seems to be arbitrary indeed. The reason for including number 1 in the basic feature set is thereby allowing the basic features and their inverses be included in the combined feature set, e.g. $|A \triangle B|$ and $\frac{1}{|A \triangle B|}$. Note that Jaccard index (i.e., $\frac{|A\cap B|}{|A\cup B|}$) is also included in this combined set. Let us call the basic set of features F, formally:

$$\begin{aligned} F(A,B) = & \\ & \{1, |A|, |B|, |A \cup B|, |A \cap B|, |A \triangle B|, \\ & |A \setminus B|, |B \setminus A|, \min(|A|,|B|), \max(|A|,|B|), \\ & \min(|A \setminus B|, ||B \setminus A|), \max(|A \setminus B|, ||B \setminus A|)\} \end{aligned}$$

Before providing a formal definition of the combined set of features, an additional set of basic features from different means (averages) must be considered as well. These additional features allow include Dice, cosine, and other coefficients as features too. For that, the expression of the generalized mean (see denominator at Eq. 3) can be used considering only a representative subset of the possible values for parameter p:

$$\begin{aligned} P = & \{-50, -20, -10, -4, -3, -2, -1, \\ & 0.0001, 1, 2, 3, 4, 10, 20, 50\} \end{aligned}$$

Now, the basic feature set F can be extended to F' by including all the generalized means restricted by P, between $|A|$ and $|B|$, and between $|A \setminus B|$ and $|B \setminus A|$, formally:

$$\begin{aligned} F'(A,B) = & F(A,B) \cup \\ & \{0.5(|A|^p + |B|^p)^{1/p} \,|\forall p \in P\} \cup \\ & \{0.5(|A \setminus B|^p + |B \setminus A|^p)^{1/p} \,|\forall p \in P\} \end{aligned}$$

Now, the number of features in $F'(A,B)$ is $|F(A,B)| + 2|P| = 42$ features. The combined set of features can be defined as:

$$C(A,B) = \left\{ \frac{f_1}{f_2} \,|(f_1, f_2) \in F'(A,B) \times F'(A,B) \wedge f_1 \neq f_2 \right\}$$

This combination produce $42 \times 41 = 1,722$ features in $C(A,B)$. This is a very large number of features for comparing only two set. Clearly, only subsets of this set of features are useful for particular applications. Even different datasets for a same task could require different representations. The idea is to make a selection of features (see [11] for an introductory tutorial) before using any machine learning regressor or classifier for a particular task. This allows to learn an adequate representation for the task prior to learn and adequate black-box (or even an interpretable) model for addressing the task. The optimal feature set for a particular task is very difficult to find because it would require considering $2^{1,772}$ possible subsets. Generally, using known methods only a near-optimal subset can be found, whose size is usually not too small nor too large. Jimenez et al. [12] observed that as a general rule the number of near-optimal features is between 10% and 20% of the number of available training samples. However, the larger the number of possible features explored, the higher the chances of finding smaller feature subsets. For example, Dueñas et al. [13] considering a similar feature set but also including logarithmic functions, found that the most correlated feature to the difficulty of a short-answer question was $\frac{|A \setminus B|}{\log(0.5\sqrt{|A|^2+|B|^2})}$, where A corresponds to the text of the reference answer and B to the question.

Although, we did not use these cardinality-based feature representation learned from training data in SemEval competitions, in subsequent studies showed this approach effective for lexical similarity task and in the analysis of questions for student evaluation. Therefore, we believe this approach may also be useful for other applications of NLP.

III. USING SOFT CARDINALITY FOR NLP

A. Textual similarity

The way to build a text similarity function is *i)* to select a linguistic unit to be compared (e.g. sentences), *ii)* to use a representation of the texts based in bags (e.g. bags of words, *n*-grams, dependencies, etc.), *iii)* to choose a cardinality based similarity coefficient (e.g. Jaccard's, Tversky's, De Beat's coefficients), and *iv)* to provide a pairwise similarity function SIM_{word} for comparing the elements produced by the used text representation (e.g. normalized Levenshtein similarity, nPMI [14], normalized path length in WordNet [15], etc.). The simplest example of such similarity function for sentence pairs is:

$$SIM_{sentence}(A,B) = \frac{|A \cap B|_{SIM_{word}}}{|A \cup B|_{SIM_{word}}}. \quad (4)$$

ISSN 2395-8618

The only parameter to be adjusted in Eq. 4 is p, the softness controller parameter. Jimenez et al. [16] showed that the default $p = 1$ works well for short sentences in English. However, a suitable value for p depends primarily on the range and distribution of the values returned by SIM_{word}, on the length of the texts, and on the task at hand. Clearly, any resemblance coefficient presented in Section II-B and Section II-C can be used.

It is important to note that Eq. 4 is recursive, similar to the popular Monge-Elkan measure [17], [18]. That is, the similarity function $SIM_{sentence}$ is obtained from another similarity function, SIM_{word}. This idea can be recursively used to build a similarity function $SIM_{paragraph}$ based on $SIM_{sentence}$, and so on. Thus, it is possible to build similarity functions exploiting the hierarchical structure of the text and natural language.

B. Term weights

Term weighting is a common practice in NLP to promote informative words and ignore non-informative words. For instance, the so-called stopwords are function words that can be removed of texts preserving their meaning to some extent, examples of these stopwords are *the*, *of*, *for*, etc. Removing stopwords may be interpreted as a binary weighting for the words in a text, i.e., assigning 1 for non-stopwords and 0 otherwise. However, a graded notion of informativeness has proven more effective than the binary approach. Probably the most used term-weighting schemes are tf.idf [19] and BM25 [20].

The soft cardinality allows the use of term weights at w_{a_i} in Eq. 1. It is important to note, that elements with zero weights (or close to 0) should be removed from texts because, although their contribution is 0, their similarities still interacts with the other elements affecting soft cardinality. This issue reveals the fact that most of the properties of the soft cardinality get overwritten because of term weighting. However, that weighted approach still preserves the original motivations of soft cardinality and extends its modeling capability [21].

C. Features for text comparison

In Section II-D we presented a method for extracting basic sets of cardinality-based features from a pair of texts represented as sets or bags of words. When the soft cardinality is being used in short texts, its word-to-word similarity function SIM_{word} plays a central role in the meaning of the extracted features. For instance, if the SIM_{word} compares words morphologically, then features extracted using $|*|_{SIM_{word}}$ reflect morphological features in texts. Additionally, other types of features can be extracted by modifying the set representation of text. For instance, a text A can be enriched with words taken from the dictionary definitions of the words already in A. These and others methods for feature extraction are presented in the following sections.

1) Morphological features: For extracting morphological features of texts it is only necessary to provide a $SIM_{word}(w_1, w_2)$ function based on the characters of the words. Some options are edit distance [22] (converted to a similarity function) or Jaro-Winkler similarity [23] (see [24] for a survey). Our choice was to use the Tversky symmetrized index (Eq. 2) by representing each word by 3-grams of characters, e.g. *house* is represented as {*hou*, *ous*, *use*}. The values of the parameters of the Tversky symmetrized index were obtained by building a simple text similarity function $SIM_{sentence}(A, B)$ using Dice's coefficient and soft cardinality using that function as auxiliary similarity function, i.e., $|*|_{SIM_{word}}$ by Eq. 1. Then the space of parameters were explored by hill-climbing optimizing the Pearson's correlation between the similarity score obtained $SIM_{sentence}$ and the gold standard of the SICK dataset [25]. The optimal values of the parameters were $\alpha = 1.9$, $\beta = 2.36$, $bias = -0.97$. In fact, the size of *n*-grams, $n = 3$, was also optimal for that function. The softness-control parameter of soft cardinality was optimized too, obtaining $p = 0.39$, but it is irrelevant for SIM_{word}. Thus, the proposed similarity function for comparing words is:

$$\begin{aligned} SIM_{word}(w_1, w_2) = & \\ & \frac{|w_1 \cap w_2| - 0.97}{2.36(1.9a - 0.9b) + |w_1 \cap w_2| - 0.97} \\ a &= \min[|w_1 \setminus w_2|, |w_2 \setminus w_1|] \\ b &= \max[|w_1 \setminus w_2|, |w_2 \setminus w_1|] \end{aligned} \quad (5)$$

Finally, having soft cardinality $|*|_{SIM_{word}}$ for each pair of texts A and B the features described in Section II-D or Section II-E can be obtained straightforwardly.

2) Semantic features: The proposed $SIM_{word}(w_1, w_2)$ function in previous section only exploits the superficial information of the words, therefore the extracted features using soft cardinality $|*|_{SIM_{word}}$ convey the same kind of information but at textual level. The obvious next step is to use a function of similarity of words that exploits semantic relationships between the words instead of comparing letters. In that way, the soft cardinality-based features would convey semantic information. There are several choices for that. First, knowledge-based lexical measures based on WordNet can do the job (see background section in [26].) Alternatively, distributional representations that make use of frequencies of the words taken from large corpora (see [27] for some examples) can be used for semantic lexical similarity. Recently, neural word embedding [28], [29] has become the state-of-the-art for semantic lexical similarity. The approach consists in building a predictive model for each word in the vocabulary of a large corpus based in local contexts. For this, each vocabulary word is represented as a fixed dimensional vector (usually from 100 to 300 dimensions). These representations are those that maximize the probability of generating the entire corpus. Although, the process of

ISSN 2395-8618

obtaining these representations is computationally expensive, pre-trained vectors are freely-available for use.[2] To obtain similarity scores with this approach, the cosine similarity between their vectorial representations is used.

3) ESA Features: For this set of features, we used the idea proposed by Gabrilovich and Markovitch [30] of extending the representation of a text by representing each word by its textual definition in a knowledge base, i.e., explicit semantic analysis (ESA). For that, we used as knowledge base the synset's textual definitions provided by WordNet. First, in order to determine the textual definition associated to each word, the texts were tagged using the maximum entropy POS tagger included in the NLTK.[3] Next, the Adapted Lesk's algorithm [31] for word sense disambiguation was applied in the texts disambiguating one word at the time. The software package used for this disambiguation process was *pywsd*.[4] The argument parameters needed for the disambiguation of each word are the POS tag of the target word and the entire sentence as context. Once all the words are disambiguated with their corresponding WordNet synsets, each word is replaced by all the words in their textual definition jointly with the same word and its lemma. The final result of this stage is that each text in the dataset is replaced by a longer text including the original text and some related words. The motivation of this procedure is that the extended versions of each pair of texts have more chance of sharing common words that the original texts.

Once the extended versions of the texts were available, the same features described in Section III-C1 or Section III-C2 can be obtained.

4) Features for each part-of-speech category: This set of features is motivated by the idea proposed by Corley and Mihalcea [32] of grouping words by their POS category before being compared for semantic textual similarity. Our approach provides a version of each text pair in the dataset for each POS category including only the words belonging to that category. For instance, the pair of texts {"*A beautiful girl is playing tennis*", "*A nice and handsome boy is playing football*"} produces new pairs such as: {"*beautiful*", "*nice handsome*"} for the ADJ tag, {"*girl tennis*", "*boy football*"} for NOUN and {"*is playing*", "*is playing*"} for VERB.

Again, the POS tags were provided by the NLTK's maximum entropy tagger. The 28 POS categories were simplified to nine categories in order to avoid an excessive number of features and hence sparseness; used mapping is shown in Table IV. Next, for each one of the nine new POS categories a set of features is extracted reusing again the method proposed in section II-D. The only difference consideration is the stopwords should not be removed and stemming should not be performed. The motivation for generating this feature sets grouped by POS category is that the machine learning algorithms could weight differently each category. The intuition behind this is that it is reasonable that categories such as VERB and NOUN could play a more important role for the task at hand than others such as ADV or PREP. Using these categorized features, such discrimination among POS categories can be discovered from the training data.

TABLE IV
MAPPING REDUCTION OF THE POS TAG SET

Reduced tag set	NLTK's POS tag set
ADJ	JJ,JJR,JJS
NOUN	NN,NNP,NNPS,NNS
ADV	RB,RBR,RBS,WRB
VERB	VB,VBD,VBG,VBN,VBP,VBZ
PRO	WP,WP$,PRP,PRP$
PREP	RP,IN
DET	PDT,DT,WDT
EX	EX
CC	CC

5) Features from dependencies: Syntactic soft cardinality [33], [34] extends the soft cardinality approach by representing texts as bags of dependencies instead of bags of words. Each dependency is a 3-tuple composed of two syntactically related words and the type of their relationship. For instance, the sentence "*The boy plays football*" is be represented with 3 dependencies: [**det**,"*boy*","*The*"], [**subj**,"*plays*","*boy*"] and [**obj**,"*plays*","*football*"]. Clearly, this representation distinguishes pairs of texts such as {"*The dog bites a boy*","*The boy bites a* dog"}, which are indistinguishable when they are represented as bags of words. This representation can be obtained automatically using the Stanford Parser [35], which, in addition, provides a dependency identifying the root word in a sentence.

Once the texts are represented as bags of dependencies, it is necessary to provide a similarity function between two dependency tuples in order to use soft cardinality, and hence to obtain the cardinality-based features in Table II. Such function can be obtained using the SIM_{word} function (Eq. 5) for comparing the first and second words between the dependencies and even the labels of the dependency types. Let's consider two dependencies tuples $d = [d_{dep}, d_{w_1}, d_{w_2}]$ and $p = [p_{dep}, p_{w_1}, p_{w_2}]$ where d_{dep} and p_{dep} are the labels of the dependency type; d_{w_1} and p_{w_1} are the first words on each dependency tuple; and d_{w_2} and p_{w_2} are the second words. The similarity function for comparing two dependency tuples can be a linear combination of the sim scores between the corresponding elements of the dependency tuples by the following expression:

$$sim_{dep}(d,p) = \gamma\, sim(d_{dep}, p_{dep}) + \delta\, sim(d_{w_1}, p_{w_2}) + \lambda\, sim(d_{w_2}, p_{w_2}).$$

Although, it is unusual to compare the dependencies' type labels d_{dep} and p_{dep} with a similarity function designed for words, we observed experimentally that this approach yield better overall performance in the textual relatedness task in comparison with a simple exact comparison. The optimal

[2] http://code.google.com/p/word2vec/; http://nlp.stanford.edu/projects/glove/
[3] http://www.nltk.org/
[4] https://github.com/alvations/pywsd

ISSN 2395-8618

values for the parameters $\gamma = -3$, $\delta = 10$ and $\lambda = 3$ were determined with the same methodology used in Section II-C for determining α, β and $bias$. Clearly, the fact that $\delta > \lambda$ means that the first words in the dependency tuples plays a more important role than the second ones. However, the fact that $\gamma < 0$ is counter intuitive because it means that the lower the similarity between the dependency type labels is, the larger the similarity between the two dependencies. Up to date, we have been unable to find a plausible explanation for this phenomenon.

IV. Soft cardinality at SemEval

The soft cardinality approach has been used by several teams for participating in several tasks in the recent SemEval campaigns (2012 to 2014). In SemEval, the task organizers propose a NLP task, provide datasets, and an evaluation setup that is carried out by them. This methodology ensures a fair comparison of the performance of the methods used by competitors. The participating systems that incorporated soft cardinality among their used methods have obtained very satisfactory results, obtaining in most of the cases rankings among the top systems. In this section, a brief overview of these participations is presented.

A. Semantic textual similarity

The task of automatically comparing the similarity or relatedness between pairs of texts is fundamental in NLP, which attracted the attention of many researchers in the last decade [36], [32]. This task consists in building a system able to compare pairs of texts, using (or not) training data and return graded predictions of similarity or relatedness. The system performance is evaluated by correlating its predictions against a gold standard built using human judgments in a graded scale. Table V contains a summary of the results obtained by the systems that used soft cardinality.

In 2012, soft cardinality was used for the first time [16] in the pilot of the Semantic Textual Similarity (STS) task [37]. The approach consisted in building a cardinality-based similarity function $SIM_{sentence}$ combining soft cardinality with a coefficient similar to Tversky's (see Subsection III-A.) The function SIM_{word} used for comparing pairs of words was based on *n*-grams of characters combined with the same rational coefficient used at sentence level (see Eq. 5.) The parameters p, n and those of both coefficients were obtained by looking for an optimal combination in the provided training data. Finally, tf-idf weights were associated with the words (weights w_{a_i} in Eq. 1.) This simple approach obtained an unexpected third place in the official ranking among 89 participating systems. Besides, as Table V shows, this system was pretty close to the top system, which used considerably more resources [37]. Besides, comparing the rankings obtained for individual datasets and the overall ranking (3^{rd}), it can be seen that the soft cardinality system was more consistent across different data sets than most of the other systems.

In 2013, the STS task was proposed again but with increased difficulty because no additional data was provided for training. Our 2012 approach was extended by building an additional similarity function for sentences using nPMI [14] as the comparator of words. Moreover, the predictions were obtained training a regression SVM with the features described in Subsection II-D. This system ranked 19^{th} among 89 systems. However, in addition to the official results, we discovered that the same 2012 function averaged with the new nPMI function correlated much better (4^{th}) [8].

In addition, in 2013, a pilot for the Typed Similarity task was proposed. It consisted in comparing pairs of text records associated with objects from the Europeana[5] database. Croce et al. [34] built a system based on the previously proposed *syntactic soft cardinality* [33]. This consists in representing texts as sets of triples (*word1*, *word2*, *relation*) extracted from dependency graphs, and combine them using soft cardinality with a similarity function for those triplets. This system ranked first among 15 participants.

In 2014, the task 10 at SemEval was the third STS version [38], which included additional datasets in Spanish. Lynum et al. [39] proposed a system for the data sets in English using features (among others) extracted with soft cardinality ranking first in 4 out of 6 data sets among 37 participating systems. Similarly, Jimenez et al. [40] proposed a system based on the soft cardinality for the Spanish data sets, ranking first in one of the data sets and third overall among 22 systems. This system also participated in tasks 1 [25] and 3 [41], which addressed text relatedness and similarity between different lexical levels (e.g. paragraph to sentence) respectively. In these tasks, the systems based on the soft cardinality ranked 4^{th} out of 17, and 3^{rd} out of 38 systems. The used features were a combination of the feature sets presented in Sections III-C1, III-C2, III-C3, III-C3, III-C4, and III-C5.

These results show that soft cardinality is a very competitive tool for building text similarity functions with relatively few resources, namely: a similarity function for comparing pairs of words, soft cardinality, and a cardinality-based coefficient or a regression method to learn this coefficient.

B. Textual Entailment

Textual entailment (TE) is the task that consists in determining whether or not a text entails another one. It was proposed under the name *cross-lingual textual entailment* (CLTE) [42], [43] in SemEval with the additional difficulty of having the two texts in different languages. The results obtained by the systems based on the soft cardinality that participated in this task in 2012 and 2013 are shown in Table VI. The approach consisted in providing two versions of the pair of texts, each one in a single language, using machine translations from Google translate.[6] Once in a single language,

[5]http://www.europeana.eu/

[6]https://translate.google.com

ISSN 2395-8618

TABLE V
BEST RESULTS OBTAINED BY SYSTEMS THAT USED SOFT CARDINALITY AT SEMEVAL 2012–2014 FOR TEXTUAL SIMILARITY AND RELATEDNESS (PEARSON CORRELATION)

Year	Task	Dataset	Rank	Soft Card.†	Top Sys.‡	Ref.
2012	STS	MSRpar	7^{th}/89	0.6405	0.7343	[16]
		MSRvid	9^{th}/89	0.8562	0.8803	
		SMT-eur	9^{th}/89	0.5152	0.5666	
		OnWN	3^{rd}/89	0.7109	0.7273	
		SMT-news	11^{th}/89	0.4833	0.6085	
		All (w. mean)	**3^{rd}/89**	**0.6708**	**0.6773**	
2013	STS	Headlines	30^{th}/90	0.6713	0.7838	[8]
		OnWN	7^{th}/90	0.7412	0.8431	
		FNWM	22^{th}/90	0.3838	0.5818	
		SMT	54^{th}/90	0.3035	0.4035	
		All (w. mean)	**19^{th}/90**	**0.5402**	**0.6181**	
		All (unofficial)	**4^{th}/90**	**0.5747**	**0.6181**	
	Typed sim.	**Europeana**	**1^{st}/15**	**0.7620**	**0.7620**	[34]
2014	Task 1-STS	**SICK**	**4^{th}/17**	**0.8043**	**0.8280**	[40]
	Task 3	Para2Sent	1^{st}/38	0.8370	0.837	
		Sent2Phr	6^{th}/38	0.7390	0.7770	
		Phr2Word	3^{rd}/22	0.2740	0.4150	
		Word2Sense	5^{th}/20	0.2560	0.3890	
		All (w. mean)	**3^{rd}/38**	**0.5260**	**0.5810**	
	Task 10 (en)	deft-forum	1^{st}/38	0.5305	0.5305	[39]
		deft-news	2^{nd}/37	0.7813	0.7850	
		headlines	1^{st}/37	0.7837	0.7837	
		images	1^{st}/37	0.8343	0.8343	
		OnWN	4^{th}/37	0.8502	0.8745	
		tweet-news	1^{st}/37	0.7921	0.7921	
		All (w. mean)	**3^{rd}/38**	**0.7549**	**0.7610**	
	Task 10 (es)	Wikipedia	1^{st}/22	0.7804	0.7804	[40]
		news	7^{th}/22	0.8154	0.8454	
		All (w. mean)	**3^{rd}/22**	**0.8013**	**0.8072**	

† Results for the best system using the soft cardinality. ‡ Results for the best system in competition.

the soft cardinality features explained in Subsection II-D were extracted for each text pair using the same word-to-word similarity function SIM_{word} used for STS. Finally, these features were combined by a classifier to determine the type of entailment. In 2013, Jimenez et al. [44] showed that these features are also language independent, making possible to train a single classifier using data in different languages. This approach produced (not included in the official ranking) state-of-the-art results for all CLTE datasets [45].

In 2014, the textual entailment task was proposed for the SICK dataset (Sentences Involving Compositional Knowledge) [25]. Using the same approach as in CLTE, but combining additional features from soft cardinalities obtained with word similarity functions based on WordNet, ESA and dependency graphs, the soft-cardinality system [40] ranked 3^{rd} of 18. Table VII shows the results obtained by the soft cardinality system both in textual entailment and textual relatedness sub-tasks.

C. Automatic students' answer grading

The task consisted in grading the correctness of a student answer (*SA*) to a question (*Q*) given a reference answer (*RA*) [47]. The approach of the system that used soft cardinality [45] consisted in extracting features for pairs (*SA*,*Q*), (*Q*,*RA*), (*SA*,*RA*) (again using the simple SIM_{word} word similarity function) and training with them a J48-graft tree classifier. Table VIII shows the results obtained by the soft cardinality system predicting correctness in 5 categories. In all other numbers of categories and evaluation measures, the soft cardinality system also ranked 1^{st} overall datasets [47]. Recently, Leeman-Munk et al. [48] integrated the soft cardinality approach in an experimental automatic tutoring system.

V. CONCLUSION

We presented our experience participating in SemEval competitions using soft cardinality and cardinality-based feature representations. This article describes the basic methods and particular methods for addressing textual similarity, multilingual textual similarity, typed-textual similarity, textual entailment, cross-lingual textual entailment and automatic students' answer grading. A summary of the official results obtained in SemEval challenges provides the evidence of the effectiveness of the used methods in open competition. It can be come to the conclusion that soft cardinality is a practical and effective tool to address several NLP problems. Furthermore, the soft cardinality model is general enough to be used in other domains and applications.

ISSN 2395-8618

TABLE VI
BEST RESULTS OBTAINED BY THE SYSTEMS THAT USED THE SOFT CARDINALITY AT SEMEVAL 2012–2014 FOR THE TEXTUAL ENTAILMENT TASK (ACCURACY)

Year	Task	Dataset	Rank	Soft Card.	Top Sys.	Reference
2012	CLTE	Spanish-English	5^{th}/29	0.552	0.632	[46]
		Italian-English	1^{st}/21	0.566	0.566	
		French-English	1^{st}/21	0.570	0.570	
		German-English	3^{rd}/21	0.550	0.558	
2013	CLTE	Spanish-English	1^{st}/15	0.434	0.434	[44]
		Italian-English	1^{st}/15	0.454	0.454	
		French-English	6^{th}/15	0.426	0.458	
		German-English	6^{th}/16	0.414	0.452	

TABLE VII
RESULTS FOR SEMEVAL TASK 1 IN 2014

	Entailment		Relatedness			
system	accuracy	official rank	Pearson	Spearman	MSE	official rank
UNAL-NLP_run1 (primary)	83.05%	3rd/18	0.8043	0.7458	0.3593	4th/17
UNAL-NLP_run2	79.81%	–	0.7482	0.7033	0.4487	-
UNAL-NLP_run3	80.15%	–	0.7747	0.7286	0.4081	-
UNAL-NLP_run4	80.21%	–	0.7662	0.7142	0.4210	-
UNAL-NLP_run5	83.24%	–	0.8070	0.7489	0.3550	-
ECNU_run1	83.64%	2nd/18	**0.8280**	**0.7689**	0.3250	1st/17
Stanford_run5	74.49%	12th/18	0.8272	0.7559	**0.3230**	2nd/17
Illinois-LH_run1	**84.58%**	1st/18	0.7993	0.7538	0.3692	5th/17

TABLE VIII
BEST RESULTS OBTAINED BY THE SOFT-CARDINALITY SYSTEM ON THE STUDENT RESPONSE ANALYSIS TASK AT SEMEVAL 2013 (WEIGHTED-AVERAGE F_1 IN 5 CORRECTNESS LEVELS)

Dataset	Testing group	Size	Rank	Soft Cardinality	Top System
Beetle	unseen answers	439	4^{th}/9	0.558	0.705
	unseen questions	819	4^{th}/9	0.450	0.614
SciEntsBank	unseen answers	540	4^{th}/9	0.537	0.625
	unseen questions	733	1^{st}/9	0.492	0.492
	unseen domains	4,562	1^{st}/9	0.471	0.471
F_1 weighted average		**7,093**	$\mathbf{1^{th}}$**/9**	**0.502**	**0.502**

ACKNOWLEDGMENT

The second author acknowledges the support of LACCIR R1212LAC006 under the project "Multimodal image retrieval to support medical case-based scientific literature search." The third author acknowledges the support of the Mexican Government via SNI, CONACYT, and the Instituto Politécnico Nacional, SIP-IPN grants 20152100 and 20152095.

REFERENCES

[1] S. Jimenez, F. Gonzalez, and A. Gelbukh, "Text Comparison Using Soft Cardinality," in *String Processing and Information Retrieval*, ser. LNCS, E. Chavez and S. Lonardi, Eds. Berlin, Heidelberg: Springer, 2010, vol. 6393, pp. 297–302.

[2] S. P. Jena, S. K. Ghosh, and B. K. Tripathy, "On the theory of bags and lists," *Information Sciences*, vol. 132, no. 1-4, pp. 241–254, 2001.

[3] P. Jaccard, "Etude comparative de la distribution florare dans une portion des {A}lpes et des {J}ura," *Bulletin de la Société Vaudoise des Sciences Naturelles*, pp. 547–579, 1901.

[4] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[5] A. Tversky, "Features of similarity," *Psychological Review*, vol. 84, no. 4, pp. 327–352, 1977.

[6] Ochiai, Akira, "Zoogeographical studies on the soleoid fishes found Japan and its neighboring regions," *Jap. Soc. Sci. Fish.*, vol. 22, no. 9, pp. 526–530, 1957.

[7] G. Sidorov, A. Gelbukh, H. Gomez-Adorno, and D. Pinto, "Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model," *Computacion y Sistemas*, vol. 18, no. 3, pp. 491–504, 2014.

[8] S. Jimenez, C. Becerra, and A. Gelbukh, "SOFTCARDINALITY-CORE: Improving Text Overlap with Distributional Measures for Semantic Textual Similarity," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*. Atlanta, Georgia, USA: ACL, Jun. 2013, pp. 194–201.

[9] B. D. Baets, H. D. Meyer, and H. Naessens, "A class of rational cardinality-based similarity measures," *Journal of Computational and Applied Mathematics*, vol. 132, no. 1, pp. 51–69, Jul. 2001.

[10] R. Poli, W. B. Langdon, N. F. McPhee, and J. R. Koza, *A field guide to genetic programming*. Lulu. com, 2008.

[11] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[12] Jimenez, Sergio, Gonzalez, Fabio A., and Gelbukh, Alexander, "Cardinality-based lexical similarity in WordNet: Bridging the gap to neural embedding," *to appear*, 2015.

[13] Dueñas, George, Jimenez, Sergio, and Julia, Baquero, "Automatic prediction of item difficulty for short-answer questions," in *to appear*, 2015.

[14] Bouma, Gerlof, "Normalized (pointwise) mutual information in collocation extraction," in *Proceedings of the Biennial GSCL Conference*, 2009, pp. 31–40.

ISSN 2395-8618

[15] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet::Similarity: measuring the relatedness of concepts," in *Proceedings HLT-NAACL–Demonstration Papers*. Stroudsburg, PA, USA: ACL, 2004.

[16] S. Jimenez, C. Becerra, and A. Gelbukh, "Soft Cardinality+ ML: Learning Adaptive Similarity Functions for Cross-lingual Textual Entailment," in *First Joint Conference on Lexical and Computational Semantics (*SEM)*. Montreal, Canada: ACL, 2012, pp. 684–688.

[17] A. E. Monge and C. Elkan, "The field matching problem: Algorithms and applications," in *Proceeding of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, Portland, OR, 1996, pp. 267–270.

[18] S. Jimenez, C. Becerra, A. Gelbukh, and F. Gonzalez, "Generalized Mongue-Elkan Method for Approximate Text String Comparison," in *Computational Linguistics and Intelligent Text Processing*, ser. Lecture Notes in Computer Science, A. Gelbukh, Ed. Springer, Jan. 2009, no. 5449, pp. 559–570.

[19] G. Salton, *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[20] S. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-3," in *Proceedings of the Third Text REtrieval Conference (TREC 1994)*, Gaithersburg, USA, 1994, pp. 109–126.

[21] Jimenez, Sergio, Gonzalez, Fabio A., and Gelbukh, Alexander, "Mathematical properties of Soft Cardinality: Enhancing Jaccard, Dice and cosine similarity measures with element-wise distance," *to appear*, 2015.

[22] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[23] W. E. Winkler, "The State of Record Linkage and Current Research Problems," *Statistical Research Division, US Census Bureau*, 1999.

[24] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate Record Detection: A Survey," *IEEE Trans. on Knowl. and Data Eng.*, vol. 19, no. 1, pp. 1–16, 2007.

[25] M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli, "Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: ACL, 2014, pp. 1–8.

[26] B. T. McInnes, T. Pedersen, Y. Liu, G. B. Melton, and S. V. Pakhomov, "U-path: An undirected path-based measure of semantic similarity," in *AMIA Annual Symposium Proceedings*, vol. 2014. American Medical Informatics Association, 2014, p. 882.

[27] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa, "A study on similarity and relatedness using distributional and WordNet-based approaches," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. NAACL'09. Stroudsburg, PA, USA: ACL, 2009, pp. 19–27.

[28] Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg, and Dean, Jeff, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 3111–3119.

[29] Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D., "Glove: Global vectors for word representation," in *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, vol. 12, Doha, Qatar, 2014, pp. 1532–1543.

[30] E. Gabrilovich and S. Markovitch, "Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, ser. IJCAI'07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, pp. 1606–1611.

[31] S. Banerjee and T. Pedersen, "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet," in *Computational Linguistics and Intelligent Text Processing*, ser. Lecture Notes in Computer Science, A. Gelbukh, Ed. Springer, 2002, no. 2276, pp. 136–145.

[32] C. Corley and R. Mihalcea, "Measuring the semantic similarity of texts," in *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, ser. EMSEE'05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 13–18.

[33] D. Croce, V. Storch, P. Annesi, and R. Basili, "Distributional Compositional Semantics and Text Similarity," in *Proceedings of the IEEE Sixth International Conference on Semantic Computing (ICSC)*, Sep. 2012, pp. 242–249.

[34] D. Croce, V. Storch, and R. Basili, "UNITOR-CORE TYPED: Combining Text Similarity and Semantic Filters through SV Regression," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: SemanticTextual Similarity*. Atlanta, Georgia, USA: ACL, 2013, pp. 59–65.

[35] M.-C. De Marneffe, B. MacCartney, C. D. Manning, and others, "Generating typed dependency parses from phrase structure parses," in *Proceedings of LREC*, vol. 6, 2006, pp. 449–454.

[36] M. D. Lee, B. Pincombe, and M. Welsh, "An empirical evaluation of models of text document similarity," in *In CogSci2005*. Erlbaum, 2005, pp. 1254–1259.

[37] E. Agirre, D. Cer, M. Diab, and G.-A. Aitor, "SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity," in *First Joint Conference on Lexical and Computational Semantics (*SEM)*. Montreal,Canada: ACL, 2012, pp. 385–393.

[38] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe, "SemEval-2014 Task 10: Multilingual semantic textual similarity," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: ACL, 2014, pp. 81–91.

[39] A. Lynum, P. Pakray, B. Gambäck, and S. Jimenez, "NTNU: Measuring Semantic Similarity with Sublexical Feature Representations and Soft Cardinality," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: ACL, 2014, pp. 448–453.

[40] S. Jimenez, G. Duenas, J. Baquero, and A. Gelbukh, "UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: ACL, 2014, pp. 732–742.

[41] D. Jurgens, M. T. Pilehvar, and R. Navigli, "SemEval-2014 Task 3: Cross-level semantic similarity," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: ACL, 2014, pp. 17–26.

[42] M. Negri, A. Marchetti, Y. Mehdad, L. Bentivogli, and D. Giampiccolo, "2012. Semeval-2012 Task 8: Cross-lingual Textual Entailment for Content Synchronization," in *First Joint Conference on Lexical and Computational Semantics (*SEM)*. Montreal, Canada: ACL, 2012, pp. 399–407.

[43] M. Negri, A. Marchetti, Y. Mehdad, and L. Bentivogli, "Semeval-2013 Task 8: Cross-lingual Textual Entailment for Content Synchronization," in *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: ACL, 2013, pp. 25–33.

[44] S. Jimenez, C. Becerra, and A. Gelbukh, "SOFTCARDINALITY: Hierarchical Text Overlap for Student Response Analysis," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: ACL, 2013, pp. 280–284.

[45] ——, "SOFTCARDINALITY: Learning to Identify Directional Cross-Lingual Entailment from Cardinalities and SMT," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: ACL, Jun. 2013, pp. 34–38.

[46] ——, "Soft Cardinality: A Parameterized Similarity Function for Text Comparison," in *First Joint Conference on Lexical and Computational Semantics (*SEM)*. Montreal, Canada: ACL, 2012, pp. 449–453.

[47] M. O. Dzikovska, R. D. Nielsen, C. Brew, C. Leacock, D. Giampiccolo, L. Bentivogli, P. Clark, I. Dagan, and H. T. Dang, "SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge," in *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantcis (*SEM 2013)*. Atlanta, Georgia, USA: ACL, 2013, pp. 263–274.

[48] S. P. Leeman-Munk, E. N. Wiebe, and J. C. Lester, "Assessing elementary students' science competency with text analytics," in *Proceedins of the Fourth International Conference on Learning Analytics And Knowledge (LAK 14)*. Indianapolis, Indiana, USA: ACM, 2014, pp. 143–147.

ISSN 2395-8618

Applying the Technology Acceptance Model to Evaluation of Recommender Systems

Marcelo G. Armentano, Ingrid Christensen, and Silvia Schiaffino

***Abstract*—In general, the study of recommender systems emphasizes the efficiency of techniques to provide accurate recommendations rather than factors influencing users' acceptance of the system; however, accuracy alone cannot account for users' satisfying experience. Bearing in mind this gap in the research, we apply the technology acceptance model (TAM) to evaluate user acceptance of a recommender system in the movies domain. Within the basic TAM model, we incorporate a new latent variable representing self-assessed user skills to use a recommender system. The experiment included 116 users who answered a satisfaction survey after using a movie recommender system. The results evince that perceived usefulness of the system has more impact than perceived ease of use to motivate acceptance of recommendations. Additionally, users' previous skills strongly influence perceived ease of use, which directly impacts on perceived usefulness of the system. These findings can assist developers of recommender systems in their attempt to maximize users' experience.**

***Index Terms*—Recommender systems, evaluation, user acceptance, technology acceptance model.**

I. Introduction

THE popularization of the Web 2.0 has resulted in plethora of applications suggesting unlimited alternative items for users, which stresses the need for effective recommendations systems. In this context, recommender systems [1] are a popular solution, as they provide suggestions based on data about users' preferences, item attributes and relationships among users and items. Although the main objective has been placed on improving the algorithms to generate recommendations, it is now possible o understand that recommendation accuracy by itself is not enough to provide users with a satisfying experience. Therefore research has started to explore the factors that might have a direct impact on a user's acceptance of a recommendation technology, in order to maximize the popularity of a recommendation system; some of those factors are the user's satisfaction with the recommendations, suggested item attractiveness, accurate understanding of the user's preferences, intention to reuse the system, facility to evaluate candidate items, ease of use, capacity to learn and interact with the system. Swearingen and Sinha [2] were among the firsts to argue that the effectiveness of a recommender system depends on factors that go beyond the quality of the prediction algorithm. Several models attempted to address this problem, explaining and predicting the use of a system; nonetheless the Technology Acceptance Model (TAM) has been the one that has met with approval within the Information Systems community [3].

This paper aims at exploring potential user acceptance issues on a traditional recommender system, using the TAM. Within the basic TAM model, we incorporate a new latent variable representing self-assessed user skills to use a recommender system. We conducted an empirical user study using a movie recommender system as a testbed, as well as a questionnaire applicable to any recommender system in the entertainment domain (books, music, movies, etc.). The results evidence that the two main factors impacting on user acceptance are perceived usefulness and perceived ease of use, which is also affected by supposed skills in the use of recommender systems.

The remainder of this paper is organized as follows. Section 2 presents the Technology Acceptance Model used in our study. Section 3 presents some related work regarding the application of TAM to recommender systems. Section 4 describes the methodology used in our study and Section 5 presents the results obtained. Finally, Section 6 presents our conclusions.

II. Background

Users' acceptance of a recommendation technology involves a set of variables regarding the users' experience in the use of the system that are related to the positive aspects of the interaction and to the fact of being captivated by a web application that leads to using it in a regular basis. User acceptance is a complex concept that goes far beyond having an attractive and easy-to-use user interface. It has been shown that two systems with identical user interface might be perceived differently by users if, for example, the underlying recommendation algorithm is changed [4].

So, what are the factors that influence the acceptance or rejection of an information technology? Davis [5] was among the firsts to study this question. First, he found that people will use an application if they believe it will help them to perform a given task better than when not using the application. Second, he found that even if users believe that a given application is useful, if the application is hard to use, then the perceived benefits of using the application are outweighed by the effort needed to use it. He call the first variable

Manuscript received on May 5, 2015, accepted for publication on June 5, 2015, published on June 15, 2015.

The authors are with the ISISTAN Research Institute (CONICET / UNICEN), Tandil, Argentina (e-mail: {marcelo.armentano, ingrid.christensen, silvia.schiaffino}@isistan.unicen.edu.ar).

ISSN 2395-8618

"perceived usefulness" and the second variable "perceived ease of use". With these findings Davis proposed the Technology Acceptance Model (TAM), which is an adaptation of the Theory of Reasoned Action (TRA) [6] to specifically deal with the prediction of the acceptability of an information system. The purpose of this model is to predict the acceptability of a tool and to identify the modifications that must be brought to the system in order to make it acceptable to users.

As shown in Figure 1, TAM suggests that Perceived Usefulness (PU) and Perceived Ease of Use (PEOU) determine an individual's intention to use a system with intention to use serving as a mediator of actual system use.

The Perceived usefulness is defined by Davis as "the degree to which a person believes that using a particular system would enhance his or her job performance". A system scoring high in perceived usefulness is then one for which a user believes in the existence of a positive user-performance relationship.

Perceived ease of use, in contrast, refers to "the degree to which a person believes that using a particular system would be free from effort". Effort is a limited resource that a person may allocate to the different activities he/she is performing. If we make all other factors invariable, a system perceived to be easier to use than another is more likely to be accepted by users

III. Previous Work

Some works have applied the TAM model to evaluate user acceptance in recommender systems, with different purposes. For example, in [7] a virtual community recommender recommends optimal virtual communities for an active user using behavioral factors suggested in TAM. Authors of this article include a filtering function based on the user's needs type, which makes the recommendation process more effective and efficient. In [8] the TAM model is used to evaluate the adoption of a recommender system in retail industry and banking sector.

In [9] the authors evaluate an existing personality-based recommender system using the technology acceptance model. They also consider that when recommending music other factors such as emotion and mood have to be considered. Then, in [10] a modified version of the technology acceptance model is applied to assess the customer's acceptance of individual personalized recommendations generated in an online shopping experience. Additionally, in [11] the authors present a framework questionnaire based on TAM, named as ResQue (Recommender systems' quality of user experience), which categorizes a set of questions into four dimensions: (1) perceived system qualities, (2) user's belief derived from these qualities, (3) user's subjective attitude, and (4) user's behavioral intentions.

In [12] the technology acceptance model and partial least squares regression are used to investigate learners' acceptance of a learning companion recommendation system (LCRS) in Facebook. They considered the usage of Facebook and the system design characteristics as external variables in TAM. Moreover, in [13] the authors propose a framework to evaluate recommender systems from the user's perspective. This framework describes that user experience depends on the user's subjective perception about some objective aspects of the system, such as the recommendation approach applied or the user interface, together with personal and situational characteristics. Similarly, in [14] a travel information recommender system is evaluated. The study found that most travelers tend to acquire the recommendation from the Internet or word-of-mouth by friends and family. Therefore authors suggest that travel information websites should consider showing friends' travel information as an important issue. An extension of TAM, UTAUT [15], is studied in a recommendation system in the context of e-commerce in [16]. Specifically, the concept of trust on technological artifacts is adapted to the UTAUT model and both hedonic and utilitarian product characteristics were considered attempting to present a comprehensive range of recommender systems.

Finally, in [17] the authors present a detailed review of the state-of-the-art about user experience and user acceptance research in recommender systems.

IV. Methodology

The experiment was conducted with an invitation to students and researchers from two universities in Argentina, in which we introduced the new movie recommender system, shown in Figure 2. In order to have a balanced study, invitations were sent to people in the area of Computer Sciences and people in other areas of study, such as Economics, Law, Business Administration and Finances. We asked participants to register in the recommender system website and to use it until the system recommended at least 20 interesting movies. Finally, we asked participants to answer an online survey, composed of 19 questions in a Likert-5 scale with 1 corresponding to "strongly disagree" and 5 corresponding to "strongly agree". The questions presented to participants, along with the associated TAM variable are detailed below:

- SKILLS_01: I believe I have the ability to use recommender systems to get useful recommendations.
- SKILLS_02: I believe I am able to identify my preferences regarding the products offered by the recommender system to get useful recommendations.
- SKILLS_03: I believe I have the ability to evaluate and use the recommendations of the recommender system to choose good movies to watch.
- PEOU_01: My interaction with the recommender system was clear and easy to understand.
- PEOU_02: I found the recommender system easy to use.
- PEOU_03: It was easy for me to learn how to use the recommender system.
- PU_01: I found the recommended movies attractive.
- PU_02: The recommended movies were adequate for my mood.
- PU_03: The recommended movies were tailored to my taste.

 http://dx.doi.org/10.17562/PB-51-10

ISSN 2395-8618

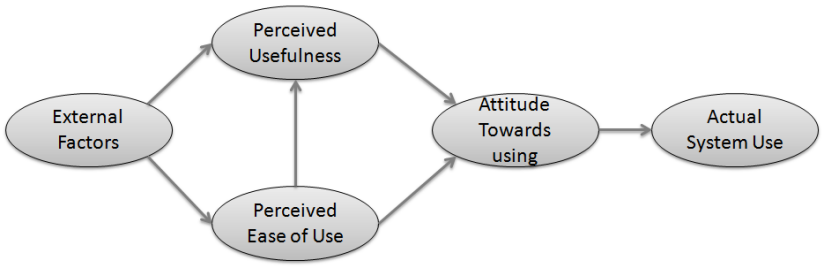


Fig. 1. Technology Acceptance Model

– PU_04: The recommended movies that I have already seen were movies I liked.
– PU_05: In general, I am satisfied with the recommended movies.
– PU_06: The recommended movies were as good as those that a friend would recommend.
– PU_07: The technology used by the recommender system is accurate.
– PU_08: The system understands my preferences regarding movies.
– ACCEPTANCE_01: I like the fact that the system learns about my preferences.
– ACCEPTANCE_02: I would use other recommender system in a different domain (songs, books, etc.).
– ACCEPTANCE_03: I want to own the recommended movies.
– ACCEPTANCE_04: I found the recommender system useful to find movies I liked and therefore I would use it again.
– ACCEPTANCE_05: I found the recommender system useful to find new movies that I would like to see and therefore I would use it again.

The experiment was open during December 2013, when we collected 116 cases. Table I shows some statistics of the participants of the experiment.

TABLE I
PARTICIPANTS STATISTICS

Attribute	Variable	Rate	Amount
Sex	Male	63.8%	74
	Female	36.2%	42
Age range	20-30	76.7%	89
	31-40	14.7%	17
	>40	8.6%	10
Area of Expertise	Business	32.8%	38
	Computer Sciences	34.5%	40
	Economics	17.2%	20
	Other	2.6%	3

V. EXPERIMENTS

The variables of interest of TAM are often unobserved variables (latent variables). The "perceived usefulness", "perceived ease of use", "skills" and "acceptance" are variables that can not be directly observed, but that can be infered from some indicators, which are the answers to the questionary. The latent variables are modeled by specifying a measurement model and a structural model. The measurement model specifies the relationships between the observed indicators and the latent variables while the structural equation model specifies the relationships amongst the latent variables.

We performed an analysis that consisted in examining the reliability and validity of the measurement model (Section V-A) and examining the significance and prediction of path coefficients in the structural model (Section V-B).

A. Measurement Model

The first step was to determine the reliability and validity of the measurement model with item loadings, convergent validity, reliability of measure and discriminat validity.

Exploratory Factor Analysis (EFA) is a statistical approach for determining the correlation among the variables in a dataset. This type of analysis groups variables based on strong correlations, providing a factor structure. In Exploratory Factor Analysis there is no a priori theory about which items belong to which constructs. This means the EFA will be able to spot problematic questions in the experiment that do not fit well the latent variables they try to describe.

Principal Component Analysis (PCA) is a statistical procedure to perform EFA. It uses an orthogonal transformation to convert a set of observations of possibly correlated variables (questions in the questionary) into a set of values of linearly uncorrelated variables called principal components (TAM variables, in our case). Basically, PCA seeks a linear combination of variables such that maximum variance is extracted. This transformation assigns the largest possible variance (that is, accounts for as much of the variability in the data as possible) to the first principal component. Each succeeding component has the highest variance possible under the constraint that it is uncorrelated (orthogonal) with the preceding components. The principal components are orthogonal because they are the eigenvectors of the covariance matrix, which is symmetric.

In order to make the interpretation of the factors that are considered relevant, the first selection step is generally followed by a rotation of the factors that were retained. Two main types of rotation are used: orthogonal when the new

ISSN 2395-8618

Fig. 2. Snapshot of the movie recommender system evaluated

axes are also orthogonal to each other, and oblique when the new axes are not required to be orthogonal to each other. Varimax [18] is the most popular rotation method. After a varimax rotation, each original variable tends to be associated with one or a small number of factors, and each factor represents only a small number of variables. In addition, the factors can often be interpreted from the opposition of few variables with positive loadings to few variables with negative loadings.

We performed Principal Component Analysis with Varimax rotation to extract factors from the questions asked to participants. Several tests were performed to check the suitability of the data for factor extraction:

- The **Kaiser-Meyer-Olkin Measure** (KMO) of sampling adequacy is a statistic that indicates the proportion of variance in the variables that might be caused by underlying factors. Values close to 1.0 generally indicate that a factor analysis may be useful with the data since patterns of correlations are relatively compact and then factor analysis should yield distinct and reliable factors. If the value is less than 0.50, the results of the factor analysis probably will not be very useful. From our data, KMO measured 0.860, which is indeed a very good index.
- **Bartlett's test of sphericity** tests the hypothesis that the correlation matrix is an identity matrix, which would indicate that the variables are unrelated and therefore unsuitable for structure detection. For factor analysis to work we need some relationships between variables and if the R-matrix were an identity matrix then all correlations coefficients would be zero. For our data, Barlett's test is highly significant ($p<0.001$) and therefore factor analysis is appropriate.
- **Extraction communalities** are estimates of the variance in each variable accounted for by the factors in the factor solution. Small values indicate variables that do not fit well with the factor solution, and should possibly be dropped from the analysis. The extraction communalities for our factors are acceptable, with the lowest 0.487 corresponding to ACCEPTANCE_01 (users like the fact that the system learns about their preferences). This means that 48.7% of the variance associated to ACCEPTANCE_01 is common, or shared, variance.

There are many indicators of the number of factors to retain from a EFA. The first approach is to consider the total variance explained by the retained factors. The total variance in the data is defined as the sum of the variances of the individual components. This quantity is simply the

ISSN 2395-8618

trace of the covariance matrix, since the diagonal elements of the latter contain the variances. On the other hand, the K1 method proposed by [19] is perhaps the best known and most utilized in practice. According to this rule, only the factors that have eigenvalues greater than one are retained for interpretation. Another popular approach is based on the Cattell's Scree test [20], which involves the visual exploration of a graphical representation of the eigenvalues. In this method, the eigenvalues are presented in descending order and linked with a line. Afterwards, the graph is examined to determine the point at which the last significant drop or break takes place—in other words, where the line levels off. The logic behind this method is that this point divides the important or major factors from the minor or trivial factors

Four factors in the initial solution resulted in eigenvalues greater than 1. Table II shows the variance explained by each factor. Together, they account for 68.634% of the variability in the original variables. This suggests that, as expected for our research model, four latent influences are representative. This conclusion is supported by the scree plot in Figure 3

TABLE II
VARIANCE EXPLAINED BY EXTRACTED FACTORS

Factor	Variance (%)	Cumulative (%)	Associated construct
1	38.53	38.53	Perceived Usefulness
2	14.26	52.79	Acceptance
3	8.87	61.66	Perceived Ease of Use
4	6.97	68.63	Skills

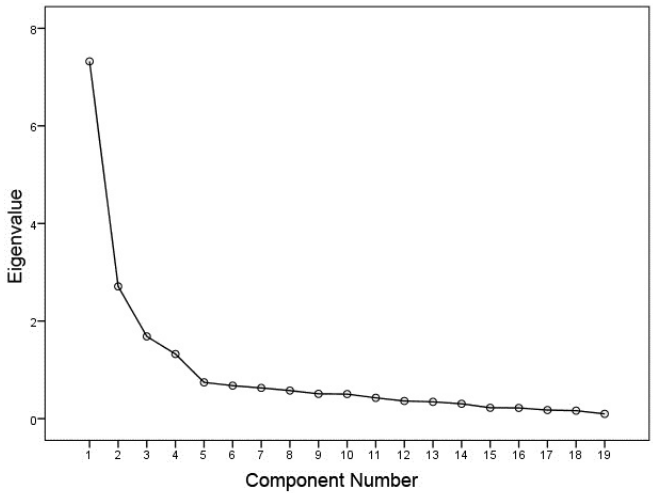


Fig. 3. Scree plot

Reliability refers to the consistency of the item-level errors within a single factor. A "reliable" set of variables will consistently load on the same factor. Cronbach's alpha is considered to be a measure of scale reliability or internal consistency. Cronbach's alpha can be written as a function of the number of test items and the average inter-correlation among the items. This metric measures how closely related a set of items are as a group. Table III shows the Cronbach's alpha coefficient on different factors. We can see that Cronbach-alpha is higher than 0.7 for all factors, indicating that the reliability of data can be considered to be sufficient.

TABLE III
CRONBACH'S ALPHA FOR DIFFERENT FACTORS

Factor	Cronbach's alpha
Skills	0.789
PEOU	0.836
PU	0.924
Acceptance	0.827
TOTAL	0.897

Convergent validity means that the variables within a single factor are highly correlated. This is evident by the factor loadings. The factors extracted from the data demonstrate sufficient convergent validity, as their loadings were all above the recommended minimum threshold of 0.55 for samples size of 100 [21]. On the other hand, **discriminant validity** refers to the extent to which factors are distinct and uncorrelated. By examining the component matrix (Table IV) we can see that variables load significantly only on one factor, demonstrating sufficient discriminant validity.

TABLE IV
ROTATED COMPONENT MATRIX. EXTRACTION METHOD: PRINCIPAL COMPONENTS, ROTATION METHOD: VARIMAX WITH KAISER NORMALIZATION. ROTATION CONVERGED IN 6 ITERATIONS.

	Component			
	PU	Accept.	PEOU	Skills
PU_03	.902			
PU_05	.847			
PU_01	.812			
PU_02	.805			
PU_04	.763			
PU_06	.692			
PU_08	.653			
PU_07	.650			
ACCEPTANCE_04		.781		
ACCEPTANCE_02		.739		
ACCEPTANCE_05		.737		
ACCEPTANCE_03		.643		
ACCEPTANCE_01		.633		
PEOU_02			.909	
PEOU_03			.839	
PEOU_01			.729	
SKILLS_01				.844
SKILLS_02				.810
SKILLS_03				.806

B. Structural Model

The next step after exploratory factor analysis is to confirm the factor structure we extracted. The objective of confirmatory factor analysis (CFA) is to test whether the data fit our research model. Model fit refers to how well our proposed model (in this case, the model of the factor structure) accounts for the correlations between variables in the questionary. If we are accounting for all the major correlations inherent in the answers to the questionary (with regards to the variables in TAM), then we will have good fit. Otherwhise, there is a

ISSN 2395-8618

significant "discrepancy" between the correlations proposed and the correlations observed, and thus we have poor model fit.

Regression Evaluation for the constructs was performed using AMOS. Figure 4 shows the stardardized estimates regarding each question, all estimates are significant at p<0.001 level.

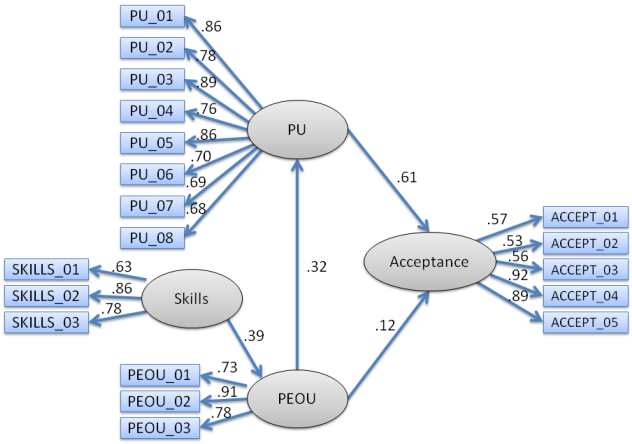


Fig. 4. Standardized estimates for our research model

Two measures that are useful for establishing validity and reliability are the **Composite Reliability** (CR) and the **Average Variance Extracted** (AVE). To test for **convergent validity** we calculated the AVE. For all factors, the AVE was above 0.5 [21] as shown in Table V. To test for **discriminant validity** we compared the square root of the AVE (on the diagonal in the Table V) to all inter-factor correlations. All factors demonstrated adequate discriminant validity since the diagonal values are greater than the correlations.

We also computed the composite reliability for each factor. In all cases the CR was above the minimum threshold of 0.70 [21], indicating we have reliability in our factors.

TABLE V
VALIDITY AND RELIABILITY INDICATORS

	CR	AVE	PEOU	PU	Accep.	Skills
PEOU	.850	.656	**.810**			
PU	.926	.612	.320	**.783**		
Accep.	.832	.512	.314	.647	**.716**	
Skills	.803	.579	.388	.080	.174	**.761**

Modification indices were consulted to determine if there was an opportunity to improve the model, but there was no need to add any covariance relationship. There are specific measures that are usually computed to determine goodness of fit. Some of these metrics are listed in Table VI, along with their acceptable thresholds according to Hair et al. [21].

For our model, we obtained cmin/df=1.527, CFI=0.938, RMSEA=0.068 and PCLOSE=0.058. These values indicate that the goodness of fit for our measurement model is acceptable according to the guideline thresholds.

TABLE VI
MODEL FIT METRICS AND RECOMMENDED THRESHOLDS, ACCORDING TO HAIR ET AL. [21].

Metric	Threshold
Chi-square / degrees of freedom (cmin/df)	< 3.000 good; < 5 sometimes permissible
CFI	> 0.95 great; > 0.90 traditional; > 0.80 sometimes permissible
RMSEA	< 0.05 good; 0.05 to 0.10 moderate; > 0.10 bad
PCLOSE	>0.05

Next, composite variables for factors were created using factor scores in AMOS. After adding the corresponding paths in the model, we performed model fit again. The consulted indicators resulted as follows: cmin/df=1.278, CFI=0.995, RMSEA=0.049 and PCLOSE=0.376. As we can see, the model is within the acceptable range of fitting.

Regression Evaluation of the structure model was performed using AMOS. Table VII shows the estimates resulting for each path. The p-value stands for the degree of significance that the estimate shows the effect on each path, where *** means that the effects on path is significant in terms of p-value is below 0.001. Thus, the regression weight for (1) Skills in the prediction of Effort, (2) Effort in the prediction of Quality and (3) Quality in the prediction of Acceptance are significantly different from zero at the 0,001 level (two-tailed). On the other hand, the regression weight for Effort in the prediction of Acceptance is only significantly different from zero at the 0.10 level.

TABLE VII
REGRESSION WEIGHTS

Path	Estimate	SE	CR	P
PEOU ← Skills	.477	.091	5.264	***
PU ← PEOU	.415	.104	3.984	***
Accep ← PU	.417	.046	9.096	***
Accep ← PEOU	.093	.055	1.702	0.089

VI. CONCLUSIONS

We presented in this work an approach to evaluate the users' acceptance of recommender systems, based on the Technology Acceptance Model. We performed an experiment with a new movie recommender system with real users. Participants answered a post treatment questionary related to a set of variables that influence each latent variable in TAM. Furthermore, we introduced a new latent variable corresponding to believed skills in the use of recommender systems.

A exploratory factor analysis validated the hypothesis that the proposed variables were able to describe adequate, reliable and valid constructs. Then, a confirmatory factor analysis validated the fact that the data fit well in the proposed model.

ISSN 2395-8618

Our experiments confirmed that perceived usefulness plays a predominant role for users to accept a new recommender system, as proposed in TAM. Perceived ease of use, on the other hand, did not show to have as much importance as perceived usefulness in the acceptance of the system. However, it did show to have an important role in determining the perceived usefulness itself. Finally, we observed that the skills that the user believes he/she has to use recommender systems have a positive impact on the effort needed to use the new system. These findings would be useful to recommender systems developers both in the academic and commercial areas.

ACKNOWLEGMENT

This study was partially supported by research projects PIP-0181-CONICET and PICT-2011-0366 awarded by ANPCyT, Argentina.

REFERENCES

[1] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*. Springer, 2011, ch. 1, pp. 1–35.

[2] K. Swearingen and R. Sinha, "Beyond algorithms: An HCI perspective on recommender systems," in *ACM SIGIR Workshop on Recommender Systems*, vol. 13, 2001, pp. 393–408.

[3] M. Chuttur, "Overview of the technology acceptance model: Origins, developments and future directions," *Working Papers on Information Systems*, vol. 9, no. 37, pp. 1–22, 2009.

[4] M. G. Armentano, R. Abalde, S. Schiaffino, and A. Amandi, "User acceptance of recommender systems: Influence of the preference elicitation algorithm," in *9th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, Nov. 2014, pp. 72–76.

[5] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Q.*, vol. 13, no. 3, pp. 319–340, Sep. 1989.

[6] I. Ajzen and M. Fishbein, *Understanding Attitudes and Predicting Social Behavior*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[7] H.-Y. Lee, H. Ahn, and I. Han, "VCR: Virtual community recommender using the technology acceptance model and the user's needs type," *Expert Systems with Applications*, vol. 33, no. 4, pp. 984–995, Nov. 2007.

[8] A. Asosheh, S. Bagherpour, and N. Yahyapour, "Extended acceptance models for recommender system adaption, case of retail and banking service in iran," *WSEAS Trans. on Business and Economics*, vol. 5, no. 5, pp. 189–200, May 2008.

[9] R. Hu and P. Pu, "Acceptance issues of personality based recommender systems," in *Proc. of ACM RecSys'09*. New York, NY, USA: ACM, 2009, pp. 221–224.

[10] D. Baier and E. Stüber, "Acceptance of recommendations to buy in online retailing," *Journal of Retailing and Consumer Services*, vol. 17, no. 3, pp. 173–180, 2010, new Technologies and Retailing: Trends and Directions. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0969698910000202

[11] P. Pu, L. Chen, and R. Hu, "A user-centric evaluation framework for recommender systems," in *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 2011, pp. 157–164.

[12] H.-C. Chen, C.-C. Hsu, C.-H. Chang, and Y.-M. Huang, "Applying the technology acceptance model to evaluate the learning companion recommendation system on Facebook," in *IEEE Fourth International Conference on Technology for Education (T4E)*, 2012, pp. 160–163.

[13] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell, "Explaining the user experience of recommender systems," *User Modeling and User-Adapted Interaction*, vol. 22, no. 4-5, pp. 441–504, 2012.

[14] Y.-H. Hung, P.-C. Hu, and W.-T. Lee, "Improving the design and adoption of travel websites: An user experience study on travel information recommender systems," in *5th IASDR International Conference*, Tokio, Japan, 2013.

[15] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User acceptance of information technology: Toward a unified view," *MIS Quarterly*, vol. 27, no. 3, pp. 425–478, 2003.

[16] Y. yao Wang, "Antecedents of review and recommendation systems acceptance," Ph.D. dissertation, Iowa State University, 2011.

[17] P. Pu, L. Chen, and R. Hu, "Evaluating recommender systems from the user's perspective: survey of the state of the art," *User Modeling and User-Adapted Interaction*, vol. 22, no. 4-5, pp. 317–355, 2012.

[18] H. F. Kaiser, "The varimax criterion for analytic rotation in factor analysis," *Psychometrika*, vol. 23, no. 3, pp. 187–200, 9 1958.

[19] ——, "The application of electronic computers to factor analysis," *Educational and Psychological Measurement*, vol. 20, pp. 141–151, 1960.

[20] R. B. Cattell, "The scree test for the number of factors," *Multivariate Behavioral Research*, vol. 1, pp. 245–276, 1966.

[21] J. F. Hair, W. C. Black, B. J. Babin, and R. E. Anderson, *Multivariate data analysis*. Prentice Hall Higher Education, 2010.

ISSN 2395-8618

Journal Information and Instructions for Authors

I. Journal Information

Polibits is a half-yearly open-access research journal published since 1989 by the *Centro de Innovación y Desarrollo Tecnológico en Cómputo* (CIDETEC: Center of Innovation and Technological Development in Computing) of the *Instituto Politécnico Nacional* (IPN: National Polytechnic Institute), Mexico City, Mexico.

The journal has double-blind review procedure. It publishes papers in English and Spanish (with abstract in English). Publication has no cost for the authors.

A. Main Topics of Interest

The journal publishes research papers in all areas of computer science and computer engineering, with emphasis on applied research. The main topics of interest include, but are not limited to, the following:

- Artificial Intelligence
- Natural Language Processing
- Fuzzy Logic
- Computer Vision
- Multiagent Systems
- Bioinformatics
- Neural Networks
- Evolutionary Algorithms
- Knowledge Representation
- Expert Systems
- Intelligent Interfaces
- Multimedia and Virtual Reality
- Machine Learning
- Pattern Recognition
- Intelligent Tutoring Systems
- Semantic Web
- Robotics
- Geo-processing
- Database Systems
- Data Mining
- Software Engineering
- Web Design
- Compilers
- Formal Languages
- Operating Systems
- Distributed Systems
- Parallelism
- Real Time Systems
- Algorithm Theory
- Scientific Computing
- High-Performance Computing
- Networks and Connectivity
- Cryptography
- Informatics Security
- Digital Systems Design
- Digital Signal Processing
- Control Systems
- Virtual Instrumentation
- Computer Architectures

B. Indexing

The journal is listed in the list of excellence of the CONACYT (Mexican Ministry of Science) and indexed in the following international indices: LatIndex, SciELO, Periódica, e-revistas, and Cabell's Directories.

There are currently only two Mexican computer science journals recognized by the CONACYT in its list of excellence, *Polibits* being one of them.

II. Instructions for Authors

A. Submission

Papers ready for peer review are received through the Web submission system on www.easychair.org/conferences/?conf=polibits1; see also updated information on the web page of the journal, www.cidetec.ipn.mx/polibits.

The papers can be written in English or Spanish. In case of Spanish, author names, abstract, and keywords must be provided in both Spanish and English; in recent issues of the journal you can find examples of how they are formatted.

The papers should be structures in a way traditional for scientific paper. Only full papers are reviewed; abstracts are not considered as submissions. The review procedure is double-blind. Therefore, papers should be submitted without names and affiliations of the authors and without any other data that reveal the authors' identity.

For review, a PDF file is to be submitted. In case of acceptance, the authors will need to upload the source code of the paper, either Microsoft Word or LaTeX with all supplementary files necessary for compilation. Upon acceptance notification, the authors receive further instructions on uploading the camera-ready source files.

Papers can be submitted at any moment; if accepted, the paper will be scheduled for inclusion in one of forthcoming issues, according to availability and the size of backlog.

See more detailed information at the website of the journal.

B. Format

The papers should be submitted in the format of the IEEE Transactions 8x11 2-column format, see http://www.ieee.org/publications_standards/publications/authors/author_templates.html. (while the journal uses this format for submissions, it is in no way affiliated with, or endorsed by, IEEE). The actual publication format differs from the one mentioned above; the papers will be adjusted by the editorial team.

There is no specific page limit: we welcome both short and long papers, provided that the quality and novelty of the paper adequately justifies its length. Usually the papers are between 10 and 20 pages; much shorter papers often do not offer sufficient detail to justify publication.

The editors keep the right to copyedit or modify the format and style of the final version of the paper if necessary.

See more detailed information at the website of the journal.