Editorial

ARTIFICIAL INTELLIGENCE is ubiquitous nowadays. It penetrates in the areas of science and computing that have been traditionally considered very far from heuristic methods—from networking and hardware design to economy, biology, medicine, forensics, etc. It is perhaps difficult to think of an area of modern computing that would not heavily depend on intelligent methods, or of an area of modern science or technology that would not greatly benefit from applications of Artificial Intelligence.

In Mexico there exists a large and rapidly growing community of researchers and students working in this area. The Mexican Society of Artificial Intelligence (SMIA; www.SMIA.org) consists of more than two hundreds computer scientists and interdisciplinary researchers. The SMIA devotes great effort to dissemination of both relevant scientific results and the general knowledge about Artificial Intelligence and its potential to solve practical problems.

The main forums organized by the SMIA for discussion and exchange of scientific ideas are the Mexican International Conference on Artificial Intelligence (MICAI, www.MICAI.org), which receives two to five hundred submissions every year from thirty to forty countries all over the world, and the Mexican Conference on Artificial Intelligence (COMIA), oriented to Mexican students, where the talks are given in Spanish.

The journal Komputer Sapiens (www.komputersapiens.org), another flagship product of the SMIA, is devoted to bringing knowledge about Artificial Intelligence and the practical applications and solutions that it offers, to general public, from students to businesspersons and decision makers, in simple and understandable terms, yet with in-depth exposition and thoughtful discussion.

It is my pleasure to present to the readers this issue of Polibits, with more than half of its papers devoted to various areas of Artificial Intelligence.

The issue offers to the reader eleven papers written by authors from eight countries: Cuba, Czech Republic, India, Italy, Mexico, Spain, Tunisia, and Vietnam.

The first six papers give a representative selection of topics that comprise Artificial Intelligence: optimization algorithms, neural networks, robotics, and natural language processing.

Gonzalo Nápoles, Isel Grau, and Rafael Bello (Cuba) in the paper “Constricted Particle Swarm Optimization based Algorithm for Global Optimization” discuss a modification of the Particle Swarm Optimization (PSO), which is a bio-inspired algorithm oriented on solution of complex problems. The modification presented in this paper helps the algorithm to explore the whole search space in order to find the global optimum, instead of being trapped in a local optimum.

Beatriz A. Garro, Humberto Sossa, and Roberto A. Vazquez (Mexico) in the paper “Automatic Design of Artificial Neural Networks by means of Differential Evolution” suggest an automatic method for designing optimal neural network—which is an activity that so far has been more an art than a science, heavily dependent on the trial-and-error approaches. Specifically, they use the method of Differential Evolution.

Jaroslav Moravec (Czech Republic) in his paper “Map Building of Unknown Environment Using L1-norm, Point-to-Point Metric and Evolutionary Computation” shows yet another application of the method of Differential Evolution, in this case to the task of automatically building maps of previously unknown environment for orientation of a robot, which is currently a tedious, time-consuming, and error-prone task.

Eduardo Cendejas, Grettel Barceló, Gigori Sidorov, Alexander Gelbukh, and Liliana Chanona-Hernández (Mexico) in the paper “Aligned Word Networks as a Resource for Extraction of Lexical Translation Equivalents, and their Application to the Text Alignment Task” study the potential of multilingual semantic dictionaries known as WordNets, or word networks, for tasks useful for machine translation, in particular, for word-level alignment of parallel bilingual texts.

Laroussi Merhben, Anis Zouaghi, and Mounir Zrigui (Tunisia) in the paper “Lexical Disambiguation of Arabic Language: An Experimental Study” continue the discussion of multilingualism and semantic issues in natural language processing. They analyze a number of well-known supervised learning algorithms for work sense disambiguation and test them on Arabic language data.

Anup Kumar Kolya, Asif Ekbal, and Sivaji Bandyopadhyay (India) present a third paper devoted to natural language processing: “A Hybrid Approach for Event Extraction”. Event extraction consists in identifying in natural language texts mentions of event that occurred at some specific time mentioned in the text, and extracting their characteristics, such as time of occurrence. They show that a combination of different machine learning approaches gives the best results on this task.

E. Lebano-Perez, C. A. Gracios-Marin, J. F. Guerrero-Castellanos and G. A. Munoz-Hernandez (Mexico) in their paper “Graphical Description of Soft Fault on Manufacturing Systems Using FDI Strategy: a SCL Approach” show the benefits of a graphical design tool to model failures in manufacturing process involving robotic instruments. A number of screenshots they included in the paper gives a rich feeling of the system they have developed.

Marva Angélica Mora Lumbreras, Álvaro Jair Martínez Varela, Julio Cesar Calva Plata, Rubén Alfredo Mejorada Lira, Brian Manuel González Contreras, and Alberto Portilla (Mexico) with the paper “VirtUATx: A Virtual Reality and Visualization Center” continue the topic of visual tools. They present a virtual reality and visualization center that they have developed for the Autonomous University of Tlaxcala.

Tran Khanh Dang and Tuan Anh Truong (Vietnam and Italy) in the paper “Anonymizing but Deteriorating Location Databases” address an important privacy and security issue: preventing malicious data miners or attackers from extracting sensitive personal information from very large databases that contain personal data of citizens or users. The authors test their approach on real-world databases.

Fernando Rodríguez-Haro, Felix Freitag, and Leandro Navarro (Spain and Mexico) in the paper “A QoS App-SLO Manager for Virtualized Infrastructure” present a formal specification of high-level component for an enhanced hypervisor in cloud computing, which allows implementing a Quality of Service policy in the virtual machine.

Finally, Marco Antonio Acevedo Mosqueda, Emmanuel Martínez Zavala, María Elena Acevedo Mosqueda, and Oleksiy Pogrebnyak (Mexico) conclude the volume with the paper “A Novel Method of Beamforming to Improve the Space Diversity”, in which they show how to improve the space diversity of an antenna array, thus reusing frequency of transmission and improving throughput of radio communication channel.

I am grateful to the Editorial Board of Polibits for the great opportunity for me to serve as a guest editor of the journal, and I would like to congratulate both the authors and the readers with yet another successful issue of this excellent journal.

Dr. Raúl Monroy

President, Mexican Society
of Artificial Intelligence (SMIA);
Professor, Tecnológico de Monterrey

Guest Editor

Constricted Particle Swarm Optimization based Algorithm for Global Optimization

Gonzalo Nápoles, Isel Grau, and Rafael Bello

***Abstract*—Particle Swarm Optimization (PSO) is a bioinspired meta-heuristic for solving complex global optimization problems. In standard PSO, the particle swarm frequently gets attracted by suboptimal solutions, causing premature convergence of the algorithm and swarm stagnation. Once the particles have been attracted to a local optimum, they continue the search process within a minuscule region of the solution space, and escaping from this local optimum may be difficult. This paper presents a modified variant of constricted PSO that uses random samples in variable neighborhoods for dispersing the swarm whenever a premature convergence (or stagnation) state is detected, offering an escaping alternative from local optima. The performance of the proposed algorithm is discussed and experimental results show its ability to approximate to the global minimum in each of the nine well-known studied benchmark functions.**

***Index Terms*—Particle Swarm Optimization, Local optima, Global Optimization, Premature Convergence, Random Samples, Variable Neighborhoods.**

I. Introduction

PARTICLE Swarm Optimization is a bioinspired search technique that simulates the social behavior observed in groups or swarms of biological individuals [1], [2]. It is based on the principle that intelligence does not lie in individuals but in the collective, allowing for the solution of complex optimization problems from a distributed point of view, without centralized control in a specific individual. Each organism (particle) adjusts its position by using a combination of an attraction to the best solution that they individually have found, and an attraction to the best solution that any particle has found [3], imitating those who have a better performance. Thus, the particle swarm overflies the search space detecting promising regions.

Although the PSO meta-heuristic has proved to be efficient for solving real-value optimization problems, the particle swarm is often attracted to stable points that are not necessarily global optima [4], [5]. This behavior causes premature convergence of the algorithm, in which the particles are grouped about suboptimal solutions with little chance of escaping from this situation. Once the particles have converged prematurely, they continue converging within extremely close proximity of one another so that the global best and all personal bests are within one minuscule region of the search space [6], limiting the algorithm exploration.

Several approaches have been proposed in the literature for addressing this undesirable situation. For example, in Attraction-Repulsion based PSO (ATREPSO) [7] the swarm switches between the attraction phase, repulsion phase, and in between phase which consist of a combination of attraction and repulsion to the optimal position. Another approach called Quadratic Interpolation based PSO (QIPSO) [8] uses a quadratic crossover operator and the swarm diversity as a measure to guide the population for finding a better solution in the search space. Using a certain threshold of diversity, Gaussian Mutation PSO (GMPSO) [9] activates a mutation operator with the hope to increase the diversity of the swarm. Lastly, in [10] the authors propose a new hybrid variant called HPSO-SA that combines PSO and Simulating Annealing (SA) to avoid premature convergence using the strong local-search ability of SA.

Although these approaches generally outperform classical PSO, there is a need to incorporate alternative procedures for enhancing the PSO search process. In this work we present a modified constricted PSO called Particle Swarm Optimization with Random Sampling in Variable Neighborhoods (PSO-RSVN) which detects and treats the premature convergence state, achieving promising results compared to several approaches reported in the literature.

The rest of the paper is organized as follows: in next Section II a theoretical background of standard PSO is described. In Section III we introduce the proposed PSO-RSVN algorithm. Section IV gives the experimental settings, the numerical benchmark problems used for comparison and the result discussion. Finally, conclusions and further research aspects are given in Section V.

II. Particle Swarm Optimization

The PSO technique involves a set of agents or particles known as swarm which “flies” through the solution space trying to locate promising regions. The particles are interpreted as possible solutions for the optimization problem and are represented as points in n-dimensional search space. In the case of standard PSO, each particle (X_i) has its own velocity (V_i) bounded by a maximum value (V_{max}), a memory

Manuscript received June 20, 2012. Manuscript accepted for publication July 24, 2012.

Gonzalo Nápoles and Rafael Bello are with the Laboratory of Artificial Intelligence, Universidad Central “Marta Abreu” de Las Villas (UCLV), Santa Clara, Cuba (e-mail: {gnapoles, rbellop}@ uclv.edu.cu).

Isel Grau is with the Laboratory of Bioinformatics, Universidad Central “Marta Abreu” de Las Villas (UCLV), Santa Clara, Cuba (e-mail: igrau@ uclv.edu.cu).

of the best position it has obtained (P_i) and knowledge of the best solution found in its neighborhood (G). In the search process the particles adjust their positions according to the following equations (1) and (2):

$$\mathrm{V}_i^{(k+1)} = \mathrm{V}_i^{(k)} + \mathrm{c}_1\mathrm{r}_1\left(\mathrm{P}_i - \mathrm{X}_i^{(k)}\right) + \mathrm{c}_2\mathrm{r}_2\left(\mathrm{G} - \mathrm{X}_i^{(k)}\right) \quad (1)$$

$$\mathrm{X}_i^{(k+1)} = \mathrm{X}_i^{(k)} + \mathrm{V}_i^{(k+1)} \quad (2)$$

where k indexes the current generation, c_1 and c_2 are positive constants, r_1 and r_2 are random numbers with uniform distribution on the interval $[0, 1]$.

A commonly used parameter that changes the original PSO is the *constriction coefficient* (χ), which was introduced by Clerc *et al.* [11] to guarantee the algorithm convergence, avoiding the explosion of the particle swarm (i.e. the state where the particles velocities and positional coordinates careen toward infinity). It can be expressed in terms of c_1 and c_2 as shown in (3):

$$\chi = \frac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|} \text{ and } \varphi = \mathrm{c}_1 + \mathrm{c}_2\,, \varphi > 4 \quad (3)$$

Another parameter that modifies the standard PSO is the inertia weight (ϖ) added by Shi *et al.* [12]. The incorporation of this parameter guarantees the balance between the capacities of local and global search; a higher weight value ($\varpi > 1$) will facilitate the exploration, while a low weight ($\varpi < 1$) facilitates the exploitation. The wrong choice of this parameter value will affect the algorithm convergence speed, so it is recommended to adjust it dynamically as shown in the following equation (4):

$$\varpi_k = \varpi_{max} - \frac{\varpi_{max} - \varpi_{min}}{\mathrm{F}_{max}}\mathrm{F}_k \quad (4)$$

where, ϖ_{min} and ϖ_{max}match the end points of the interval on which the *k*-th inertia weight is defined, F_k denotes the number of evaluations at the *k*-cycle, whereas F_{max} corresponds to the maximal number of evaluations allowed. So, both factors are applied to the equation (1) as follow:

$$V_i^{(k+1)} = \chi\left(\varpi_{k+1}\mathrm{V}_i^{(k)} + \delta_1\left(\mathrm{P}_i - \mathrm{X}_i^{(k)}\right) + \delta_2\left(\mathrm{G} - \mathrm{X}_i^{(k)}\right)\right) \quad (5)$$

Extensive experiments carried out in [3] showed that constricted PSO returns improved performance over the original PSO; however, it provides no mechanism to detect and treat premature convergence (or stagnation state) of the particle swarm, which could adversely affect the algorithm effectiveness.

III. PSO-RSVN Algorithm

In this section we introduce a modification for constricted PSO algorithm defined by equations (2) and (5). First, several mechanisms to detect the premature convergence state along the progress of the algorithm are discussed. Next, a new procedure of swarm reorganization based on randomly selected particles from the neighborhoods of the global best particle is presented.

A. Detection of the premature convergence state

The first step to enhance the performance of constricted PSO algorithm is to detect the premature convergence state. When PSO falls into a local optimum all individuals are grouped around this solution, which is why diversity is lost among the swarm particles, making more difficult to find better solutions in the algorithm progress. In [13] several ways to detect this state are discussed:

i. **Cluster analysis:** a percentage of the particles are at a certain Euclidean distance of the best global particle.
ii. **Objective function without progress:** the objective function does not suffer significantly improvement in several iterations of the generational cycle. This criterion also may be used for detecting the stagnation state.
iii. **Maximum radius of the swarm:** the particle with more Euclidean distance respect to the global best particle found, have a distance less than a pre-set threshold. This criterion it is formally defined as:

$$\rho(k) = \frac{\max\limits_{1 \le i \le |\Omega|}\left\|\mathrm{X}_i^{(k)} - \mathrm{G}\right\|}{|\sigma_{max} - \sigma_{min}|} \quad (6)$$

where ‖.‖ denotes the Euclidean norm on $\mathbb{R}^n$, while σ_{min} and σ_{max} are the end points on which each dimension of the particle X_i is defined (assuming same domain for all dimensions), whereas **Ω** represents the particle swarm. In this way the threshold is normalized for each generation *k*. Particularly, we use ii) to detect the stagnation state and iii) for identify premature convergence.

B. Treatment of the Premature Convergence State

Once premature convergence signals are detected it is necessary to take some action to allow the algorithm to escape from this state, for example:

i. Moving the position of the globally best particle found.
ii. Reorganizing the particle swarm (applying genetic operators in order to diversify the population, re-initializing the swarm, etc.).

Although both strategies have reported good results in solving global optimization problems, they have some drawbacks. In the first case is not trivial to find a better particle, and even when once found, the population is poorly diversified.

In the second case, diversification increases the chances of escaping the local optimum but if the swarm is not properly reorganized, the particles may converge to the same solution or indiscriminately move away from the promising areas that were found. In order to mitigate these problems in this paper we assume a hybrid approach consisting in diversifying the population while trying to move the position of the best global particle found (G) in the search process.

The Variable Neighborhood Search (VNS) [14] is a simple and effective meta-heuristic for combinatorial problems and global optimization which is based on the systematic change of the neighborhood in the search process. Inspired by this idea, we present a procedure called Random Sampling in Variable Neighborhoods (RSVN) which aims to disperse the swarm when the premature convergence or stagnation state is detected. The main idea of this procedure is to restructure the particle swarm from the selection of random samples uniformly distributed in several neighborhoods generated around the n-dimensional point G. Equations (7), (8) and (9) formalize the way to generate the set of samples in each neighborhood:

$$\lambda_{jd}{}^{-}=\begin{cases}\tau_{jd}, & \tau_{jd} \geq \sigma_{min} \\ \sigma_{min}, & \tau_{jd} < \sigma_{min}\end{cases} \text{ and } \tau_{jd}= G_d - \xi_j |\sigma_{max} - \sigma_{min}| \quad (7)$$

$$\lambda_{jd}{}^{+}=\begin{cases}\varsigma_{jd}, & \varsigma_{jd} \leq \sigma_{max} \\ \sigma_{max}, & \varsigma_{jd} > \sigma_{max}\end{cases} \text{ and } \varsigma_{jd}= G_d + \xi_j |\sigma_{max} - \sigma_{min}| \quad (8)$$

$$X_t \in \Psi_j \,|X_{td} \sim U\left(\lambda_{jd}{}^{-}, \lambda_{jd}{}^{+}\right),\ t = 1, \dots, |\Psi_j|;\ \ j = 1, \dots, M \quad (9)$$

where d indexes the particle dimension, M is a user-specified integer parameter that denotes the number of neighborhoods, whereas $\xi_j \in (0,1]$is a fractional value called neighborhood factor that denotes the j-th neighborhood proportion to the size of the search space; and it is calculated as: $\xi_j = j/M$. Lastly, $\Psi_j$represents the j-th set of uniformly distributed samples in the domain that defines the interval $[\lambda_{jd}{}^{-}, \lambda_{jd}{}^{+}]$.

After collecting the samples, a selection process of the particles takes place. These agents will form the new swarm as shown below:

$$\Omega^k = \Phi_1 \cup \Phi_2 \cup \dots \cup \Phi_m = \bigcup_{j=1}^{m} \Phi_j \ | \ \Phi_j \subseteq \Psi_j, \forall j \qquad (10)$$

where $\Phi_j$is a subset of good enough particles compared to all samples Ψ_j using an elitist criterion. In this procedure each particle X_i is a candidate to replace the best global particle, which complements the swarm dispersion process. Next, a pseudocode summarizes the main ideas and constriction parameters of the PSO-RSVN algorithm.

An important aspect to be discussed is the selection of each subset Φ_j; due to the high computational cost that generally involve the evaluation of the objective function the elitist criterion may be replaced by a heuristic criterion, for example: "select the particles with greater Euclidean distance respect to the global best particle". However, for high-dimensional problems the extra-computational cost needed to compute the Euclidean distance could become significant. For solving this inconvenient, a simple but effective alternative might be: for each set of samples make $|\Phi_j| = |\Psi_j| = |\Omega|/M$, i.e., the selection process is omitted and consequently reduced the extra-computational cost required for computing the swarm reorganization process. In fact, this last criterion is used in all experiments carried out in next Section IV.

Pseudocode of PSO with Random Sampling in Variable Neighborhoods

```
Generate the swarm vector and the velocity vector randomly
Select the best global particle of the swarm (G)
Initialize $\omega_{max}$ = 1.4, $\omega_{min}$ = 0.4, $c_1$ = 2.05, $c_2$ = 2.05
While(the maximal number of evaluations is not met)
  Calculate $\varpi_k$ dynamically according to expression (4)
  ForEach $X_i^{(k)} \varepsilon \Omega$
    Calculate $V_i^{(k+1)}$according to expression(5)
    Adjust the position of $X_i^{(k+1)}$ according to expression (2)
    Evaluate the new particle $X_i^{(k+1)}$
    IF($X_i^{(k+1)}$ is the best record for the i-th particle)
      Update the best record for the i-th particle with $X_i^{(k+1)}$
      IF($X_i^{(k+1)}$ is a better particle than G)
        Update G with $X_i^{(k+1)}$ as the best global particle
      endIF
    endIF
  endForEach
  IF (premature convergence or stagnation state is detected)
    Disperse the swarm $\Omega$ according to the expressions (7-10)
    Reset velocity vector using a random sequence
    Update the vector P and the best particle G
    Reset the inertia weight
  endIF
endWhile
```

IV. Performance Study

In this section two implementations of the PSO-RSVN algorithm are evaluated; the first implementation called PSO-RSVN-α detects the premature convergence state, comparing the maximum radius of the swarm (according to equation (6)) with a pre-set threshold α. The second variant called PSO-RSVN-p is useful for identify a premature convergence state as well as a possible stagnation state, and involve a parameter (p) that represents the allowed maximal number of evaluations without progress. It must be mentioned that PSO-RSVN-p induce less extra-computational cost, since the estimation of the maximum radius of the swarm is not required.

Table I describes nine well-known benchmark functions taken from [15], which are used to compare the performance of the proposal with several approaches reported in the literature. These functions are minimization problems characterized by multiple local optima, especially when the complexity of the function increases, that is, when the dimensionality of the search space increases. The first seven problems are scalable and includes unimodal, multimodal and noisy functions, whereas the last two problems are highly multimodal in nature. Therefore, the use of these functions helps in deciding the credibility of an optimization algorithm.

The experimental results discussed in this section are addressed in two sub-sections which study the performance of the PSO-RSVN algorithm for 20-dimensional and 30-dimensional solution search spaces.

A. PSO-RSVN behavior in 20-dimensional search spaces

This section compares both PSO-RSVN variants against five approaches evaluated and discussed in [16], [10]: PSO,

TABLE I
STANDARD BENCHMARK FUNCTIONS USED IN THIS WORK. LAST COLUMN (F_{MIN}) REFERS TO GLOBAL MINIMUM VALUE EXISTING IN THE DOMAIN DEFINED BY BOUNDARIES.

Mathematical Formulation	Search Range	F_{min}
$f_1(\vec{x}) = \sum_{i=1}^{n} x_i^2$	[-100.0,100.0]	0.0000000
$f_2(\vec{x}) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	[-5.120,5.120]	0.0000000
$f_3(\vec{x}) = \left(\frac{1}{4000}\right) \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600.0,600.0]	0.0000000
$f_4(\vec{x}) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[-2.048,2.048]	0.0000000
$f_5(\vec{x}) = \sum_{i=1}^{n} i x_i^4 + rand[0,1]$	[-1.280,1.280]	0.0000000
$f_6(\vec{x}) = -\sum_{i=1}^{n} (x_i \sin(\sqrt{\lvert x_i \rvert}))$	[-500.0,500.0]	-420.968n
$f_7(\vec{x}) = 20 + e - 20e^{-0.2\left(\left(\frac{1}{n}\right) \sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}}} - e^{(1/n)\sum_{i=1}^{n} \cos(2\pi x_i)}$	[-32.0,32.0]	0.0000000
$f_8(\vec{x}) = \sum_{i=1}^{5} icos((i+1)x_1 + i) \sum_{i=1}^{5} icos((i+1)x_2 + i)$	[-10.0,10.0]	-186.7309
$f_9(\vec{x}) = (x_2 + x_1^2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + x_1$	[-5.00,5.00]	-3.783961

[a] F_8 and F_9 are defined in $\mathbb{R}^2$ space.

TABLE II
AVERAGE ERROR OBTAINED IN THE OPTIMIZATION PROCESS. THE BEST PERFORMING ALGORITHM FOR EACH FUNCTION IS EMPHASIZED IN BOLDFACE.

ID	PSO	QIPSO	ATREPSO	GMPSO	HPSO-SA	RSVN-α	RSVN-p
F_1	1.167E-45	**0.0000000**	4.000E-17	7.263E-17	5.365E-32	**0.0000000**	**0.0000000**
F_2	22.339158	11.946888	19.425979	20.079185	**0.0000000**	**0.0000000**	**0.0000000**
F_3	0.0316460	0.0115800	0.0251580	0.0244620	3.322E-20	**0.0000000**	**0.0000000**
F_4	22.191725	8.9390110	19.490820	14.159547	0.2270481	2.635E-16	**7.312E-25**
F_5	8.6816020	0.4511090	8.0466170	7.1606750	0.0020199	**4.372E-06**	1.386E-05
F_6	2240.8010	2063.7740	2235.6830	2371.6900	**39.700000**	43.651000	1280.4350
F_7	3.483E-18	**2.461E-24**	0.0184930	1.474E-18	7.435E-16	4.440E-16	4.440E-16
F_8	1.420E-05	**0.0000000**	1.420E-05	1.530E-05	8.670E-14	**0.0000000**	**0.0000000**
F_9	0.4524730	**0.0000000**	0.0325030	0.3237280	1.000E-06	1.000E-06	**0.0000000**

QIPSO, ATREPSO, GMPSO, and HPSO-SA. In each simulation we used 30 particles and 300.000 objective function evaluations in a 20-dimensional search space. In addition, five variable neighborhoods (M=5) are used. For the PSO-RSVN-α implementation, a threshold for the maximum radius of the swarm α= 1.0E-5 is adopted, whereas for PSO-RSVN-p the allowed number of evaluations without progress is set to 200. Table II summarizes the average error obtained respect to the global optimum for each algorithm from 30 independent trials. In all tables, PSO-RSVN-αis abbreviated as RSVN-α and PSO-RSVN-p as RSVN-p.

Analyzing the results shown in Table II we observed that for the functions F_1, F_2, F_3 and F_8 both PSO-RSVN variants always finds the global optimum satisfactorily. For Rosenbrock (F_4) function PSO-RSVN-p outperforms all examined algorithms; in this function the stagnation state is frequently presented due to the search space properties. For the noisy function F_5PSO-RSVN-α computes the best performance, whereas for Shwefel (F_6), HPSO-SA locates better solutions. PSO-RSVN-α is slightly the best approach for minimizing the Himmelblau (F_9) function. Finally, QIPSO has the best results reported for Ackley (F_7), followed by both PSO-RSVN algorithms.

In a deeper statistical study of the algorithms performance we used several test for exploring significant differences among them. Depending on the concrete type of data employed, statistical procedures are grouped in two classes: parametric and nonparametric [18]. Parametric tests have been often used in the analysis of experiments in computational intelligence. Unfortunately, they are based on assumptions (independence, normally, homoscedasticity) which are most probably violated when analyzing the performance of stochastic algorithms based on computational intelligence [19], [20]. To overcome this problem, the researchers frequently use nonparametric statistical procedures when these previous assumptions cannot be satisfied.

First, we compute the Friedman test (Friedman two-way analysis of variances by ranks) [21], [22]. This test is a multiple comparisons procedure for detecting significant differences between the behaviors of two or more algorithms; i.e. it can be used for detecting whether at least two of the samples represent populations with different median values or not, in a set of *n* samples ($n \geq 2$). Table III shows the mean rank and the p-value associated with this test. Using a significance level of 0.05, corresponding to the 95% confidence interval, the Friedman test suggest rejecting the null hypothesis ($p\text{-value} < 0.05$), thus, there exist highly significant differences between at least two methods across benchmark. Also can be observed that PSO-RSVN-p and PSO-RSVN-α are the best ranked; however this information cannot be used to conclude that our proposals are involved on this differences.

TABLE III
MEAN RANK ACHIEVED BY THE FRIEDMAN TEST

Evaluated Algorithms	Mean Rank[a]
RSVN-p	2.39
RSVN-α	2.39
HPSO-SA	2.89
QIPSO	3.44
GMPSO	5.39
ATREPSO	5.50
PSO	6.00

[a] Monte Carlo signification (p-value) = 0.00

The main drawback of the Friedman's tests is that they only can detect significant differences over the whole multiple comparisons, being unable to establish proper comparisons between some of the algorithms considered [23]. For this reason we also compute the Wilcoxon signed ranks test [24]; it is used for answering a simple question: do two samples represents two different populations? Thus, Wilcoxon is a pairwise procedure that aims to detect significant differences between two sample means, that is, the behavior of two algorithms.

Table IV shows the p-values associated with each pairwise comparison. Then some important conclusions came out:

i. Using a significance level of 0.05, corresponding to the 95% confidence interval, the Wilcoxon test suggest to reject the null hypothesis ($p\text{-value} < 0.05$) for the following pairwise comparisons: RSVN-p vs. GMPSO, RSVN-p vs. ATREPSO, RSVN-p vs. PSO, RSVN-α vs. PSO, RSVN-α vs. GMPSO and RSVN-α vs. ATREPSO; thus we can conclude that there exist highly significant differences between them.
ii. Using a significance level of 0.1, corresponding to the 90% confidence interval, the Wilcoxon test suggest to reject the null hypothesis ($p\text{-value} < 0.1$) for the following pairwise comparisons: RSVN-p vs. QIPSO and RSVN-α vs. QIPSO; i.e. there exist fairly significant differences between them.
iii. For the pairwise comparisons that involve the following methods: RSVN-α, RSVN-p and HPSO-SA, there not exist perceptible differences among them. These results confirm the improvement of the proposed procedures.

TABLE IV
WILCOXON SIGNED RANKS TEST RESULTS

Pairwise Comparison	p-value[a]
RSVN-p vs. RSVN-α	0.621
RSVN-p vs. HPSO-SA	0.195
RSVN-p vs. GMPSO	0.013
RSVN-p vs. ATREPSO	0.004
RSVN-p vs. QIPSO	0.066
RSVN-p vs. PSO	0.013
RSVN-α vs. HPSO-SA	0.292
RSVN-α vs. GMPSO	0.013
RSVN-α vs. ATREPSO	0.004
RSVN-α vs. QIPSO	0.081
RSVN-α vs. PSO	0.013

[a] Monte Carlo signification

B. PSO-RSVN behavior in 30-dimensional search spaces

One of the most important variations to PSO is the introduction of the local model or local topology (*lbest*). In this model, each particle can only communicate with a subset of particles, limiting the overall exchange of information. In contrast to the global model or the global topology (*gbest*), the local model converges more slowly but is less prone to being trapped in suboptimal solutions. In fact, several authors suggest using the local topology to optimize complex multimodal functions, and the global topology to optimize unimodal functions [3].

Four different approaches have been evaluated in [6] that include considerations about the topology of the particle swarm in 30-dimensional spaces: *lbest* PSO with a ring topology, *gbest* PSO, Regrouping PSO (RegPSO) [6] and Opposition based PSO (OPSO) [17]. These simulations allow studying the stability of the PSO-RSVN algorithm when increasing dimensionality of the solution search space. Table V summarizes the average error obtained for each algorithm from 50 trials, 20 particles as the swarm size and 800.000 objective function evaluations. Moreover, five variable neighborhoods (M=5) are used. For the PSO-RSVN-α implementation, a tolerance for the maximum radius of the swarm α = 1.0E-5 is adopted, whereas for PSO-RSVN-p the allowed number of evaluations without progress is set to 500.

From the numerical results shown in Table V a conclusion came out: both variants of PSO-RSVN outperform other approaches in all cases. PSO-RSVN-α and PSO-RSVN-p always find the global optimum satisfactorily for Sphere (F_1),

TABLE V
AVERAGE ERROR OBTAINED IN THE OPTIMIZATION PROCESS. THE BEST PERFORMING ALGORITHM FOR EACH FUNCTION IS EMPHASIZED IN BOLDFACE.

ID	*gbest* PSO	*lbest* PSO	OPSO	RegPSO	RSVN-α	RSVN-p
F_1	2.470E-323	5.513E-160	9.881E-324	9.2696E-15	**0.00000000**	**0.00000000**
F_2	71.6368600	54.2849000	66.1646300	2.6824E-11	**0.00000000**	**0.00000000**
F_3	0.05500800	0.00939970	0.02574900	0.01386100	**0.00000000**	**0.00000000**
F_4	2.06915000	3.25523000	1.86410000	0.00393510	7.9800E-04	**2.4299E-18**
F_5	0.00394380	0.01325000	0.00101660	0.00064366	**2.1731E-06**	1.4155E-05
F_7	3.91150000	0.07546900	2.67240000	4.6915E-07	**4.4408E-16**	**4.4408E-16**

Rastringin (F_2), Griewank (F_3) and Ackley (F_7) functions. For Rosenbrock (F_4) PSO-RSVN-p has the best performance, whereas for Quartic (F_5) the other proposal achieves the best approximations. These results reveal that both PSO-RSVN are quite consistent across benchmarks when the dimensionality of the search space increases.

Figure 1 illustrates, as an example, the behavior of the swarm diversity in the optimization process of the Rastringin function, for proposed PSO-RSVN-αalgorithm and constricted PSO. In this simulation we use the maximum radius of the swarm for measuring the swarm diversity. So, in the generation number 4900 both methods prematurely converge to a local optimum, i.e. the whole population is grouped in a minuscule region of the search space. This situation degrades the PSO search capabilities. However, the swarm diversity introduced by PSO-RSVN-α ensures the exploration of new areas of the solution space, increasing the possibility of escape from suboptimal solutions.

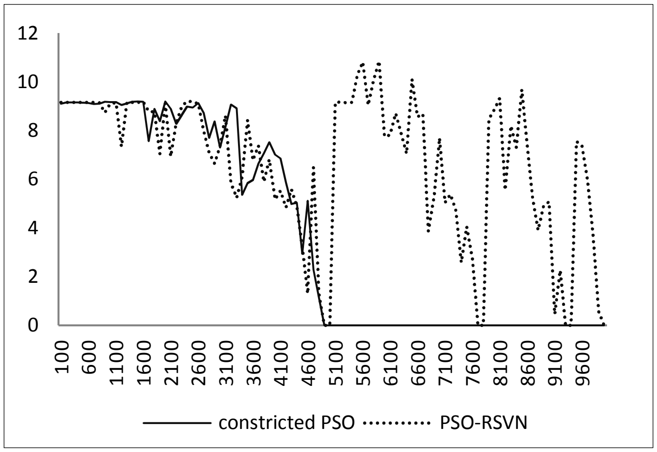


Fig. 1. Behavior of the swarm diversity for PSO-RSVN and constricted PSO, during the optimization process. The horizontal axis denotes the number of objective functions evaluated and the vertical axis refers to the maximum radius of the swarm.

In general, PSO-RSVN introduces a new parameter to be estimated by the user: the number of neighborhoods M; it is an integer value used to organize the sampling process by dividing the search space in several partitions. Recommended values for this parameter could be M = 5 or M =10, although other values are allowed. For the first implementation a normalized threshold α for the maximum radius of the swarm should be specified. The wrong selection of this parameter may be relevant to the algorithm performance: a higher value of α will affect the PSO-RSVN-α exploitation capability, due to a false premature convergence state could be induced, whereas a lower value could never detect an existing premature convergence state. Empirical experiments show that values from 1.0E-2 to 1.0E-8 are a good choice. Finally, for the PSO-RSVN-p implementation, a parameter for controlling the allowed number of evaluations without progress is required. This parameter is easy to set and will depend on the maximal number of the objective function evaluations.

Although PSO-RSVN generally provides superior results regarding examined approaches for well-known benchmark functions described in Table I, future work will study the algorithm performance across other well-known benchmark functions, for example, shifted or badly scaled functions.

V. CONCLUSIONS

In this paper was proposed a modified variant of the constricted PSO called PSO-RSVN, for enhancing the search capability of this algorithm, when solving complex optimization problems. Two PSO-RSVN implementations are presented: the first one is capable to detect the premature convergence state, and the second one is able to detect the premature convergence as well as the stagnation state. Both variants treat these undesirable states by reorganizing the particle swarm, which is, conducting a random sampling in several neighborhoods, emphasizing the neighborhood of the best particle found so far. The swarm diversity introduced by the proposed dispersion mechanism ensures the exploration of new areas of the solution space, increasing the possibility of escape from suboptimal solutions.

It was evaluated the algorithm performance in comparison with other variants of PSO reported in the literature, by using nine well-known benchmark functions for 20-dimensional search spaces. In addition it was verified the stability of the algorithm upon increasing the dimensionality of the search space, by using 30-dimensional spaces. In both experiments the new algorithm (PSO-RSVN) provides superior results in most cases. In fact, due to its simplicity, elitist properties and low computational cost, the RSVN procedure could be adapted and successfully integrated into other evolutionary paradigms. As mentioned, future work will be focused on extending the study of the algorithm performance across other well-known benchmark functions.

ACKNOWLEDGMENT

We would like to thank Prof. Dr. Ricardo Grau Abalo, from the Centre of Studies on Informatics, UCLV, for fruitful discussions and statistical advice.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proc. of the 1995 IEEE International Conference on Neural Networks*, Australia, 1995, pp. 1942–1948.

[2] R. Eberhart and J. Kennedy, "A New Optimizer using Particle Swarm Theory," in *Proc. of the Sixth International Symposium on Micromachine and Human Science*, Japan, 1995, pp. 39–43.

[3] D. Bratton and J. Kennedy, "Defining a Standard for Particle Swarm Optimization," in *Proc. of the 2007 IEEE Swarm Intelligence Symposium*, Honolulu, 2007, pp. 120–127.

[4] V.J. Huang, et al, "Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems," *Int. J. of Intelligent Systems*, vol. 21, pp. 209–226, 2006.

[5] J. Kennedy and C.E. Russell, *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001.

[6] G. Evers and M.B. Ghalia, "Regrouping Particle Swarm Optimization: A new Global Optimization Algorithm with Improved Performance Consistency Across Benchmarks," in *IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, 2009, pp. 3901–3908.

[7] J. Riget and J.S. Vesterstrom, "A diversity-guide particle swarm optimizer – the arPSO", Tech. Rep., 2002.

[8] M. Pant and T. Radha, "A new Particle Swarm Optimization with quadratic crossover," in *International Conference on Advanced Computing and Communications*, India, 2007, pp. 81–86.

[9] M. Pant, T. Radha and V.P. Singh, "A new diversity based Particle Swarm Optimization using Gaussian Mutation," *Int. J. of Mathematical Modeling, Simulation and Applications*, vol. 1, pp. 47–60, 2007.

[10] L. Idoumghar, M. Melkemi, R. Schott and M.I. Aouad, "Hybrid PSO-SA Type Algorithms for Multimodal Function Optimization and Reducing Energy Consumption in Embedded Systems," *Applied Computational Intelligence and Soft Computing*, 12 pages, 2011.

[11] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evolutionary Computation*, vol. 6, pp. 58–73, 2002.

[12] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer," in *Proc. of the IEEE Congress on Evolutionary Computation*, 1998, pp. 69–73.

[13] F. Van den Bergh, "An Analysis of Particle Swarm Optimizers," PhD thesis, Univ. of Pretoria, South Africa, 2002.

[14] P. Hansen and N. Mladenović, "Variable neighborhood search: Principles and applications," *European Journal of Operations Research*, pp. 449–467, 2001.

[15] P.N. Suganthan, et al., "Problem definition and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Tech. Rep. 2005005, 2005.

[16] M. Pant, R. Thangaraj and A. Abraham, "Particle swarm based metaheuristic for function optimization and engineering applications," in *Proc. of the 7th Computer Information Systems and Industrial Management Applications*, Washington, 2008, pp. 84–90.

[17] H. Wang, et al., "Opposition-based Particle Swarm Algorithm with Cauchy mutation," in *Proc. of the IEEE Congress on Evolutionary Computation*, 2007, pp. 4750–4756.

[18] J. Higgins, *Introduction to Modern Nonparametric Statistics*, Duxbury Press, 2003.

[19] S. García, A. Fernández, J. Luengo and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability," *Soft Computing*, vol. 13, no. 10, pp. 959–977, 2009.

[20] S. García, D. Molina, M. Lozano and F. Herrera, "A study on the use of nonparametric tests for analyzing the evolutionary algorithms' behavior: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, pp. 617–644, 2009.

[21] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, pp. 674–701, 1937.

[22] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Annals of Mathematical Statistics*, vol. 11, pp. 86–92, 1940.

[23] J. Derrac, S. Garcia, D. Molina and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, pp. 3–18, 2011.

[24] J.D. Gibbons and S. Chakraborti, "Nonparametric Statistical Inference," 5th ed., Chapman & Hall, 2010.

Diseño Automático de Redes Neuronales Artificiales mediante el uso del Algoritmo de Evolución Diferencial (ED)

Beatriz A. Garro, Humberto Sossa, Roberto A. Vazquez

***Resumen*—En el área de la Inteligencia Artificial, las Redes Neuronales Artificiales (RNA) han sido aplicadas para la solución de múltiples tareas. A pesar de su declive y del resurgimiento de su desarrollo y aplicación, su diseño se ha caracterizado por un mecanismo de prueba y error, el cual puede originar un desempeño bajo. Por otro lado, los algoritmos de aprendizaje que se utilizan como el algoritmo de retropropagación y otros basados en el gradiente descenciente, presentan una desventaja: no pueden resolver problemas no continuos ni problemas multimodales. Por esta razón surge la idea de aplicar algoritmos evolutivos para diseñar de manera automática una RNA. En esta investigación, el algoritmo de Evolución Diferencial (ED) encuentra los mejores elementos principales de una RNA: la arquitectura, los pesos sinápticos y las funciones de transferencia. Por otro lado, dos funciones de aptitud son propuestas: el error cuadrático medio (MSE por sus siglas en inglés) y el error de clasificación (CER) las cuales, involucran la etapa de validación para garantizar un buen desempeño de la RNA. Primero se realizó un estudio de las diferentes configuraciones del algoritmo de ED, y al determinar cuál fue la mejor configuración se realizó una experimentación exhaustiva para medir el desempeño de la metodología propuesta al resolver problemas de clasificación de patrones. También, se presenta una comparativa contra dos algoritmos clásicos de entrenamiento: Gradiente descendiente y Levenberg-Marquardt.**

***Palabras clave*—Evolución diferencial, evolución de redes neuronales artificiales, clasificación de patrones.**

Automatic Design of Artificial Neural Networks by means of Differential Evolution (DE) Algorithm

***Abstract*—Artificial Neural Networks (ANN) have been applied in several tasks in the field of Artificial Intelligence. Despite their decline and then resurgence, the ANN design is currently a trial-and-error process, which can stay trapped in bad solutions. In addition, the learning algorithms used, such as back-propagation and other algorithms based in the gradient descent, present a disadvantage: they cannot be used to solve non-continuous and multimodal problems. For this reason, the application of evolutionary algorithms to automatically designing ANNs is proposed. In this research, the Differential Evolution (DE) algorithm finds the best design for the main elements of ANN: the architecture, the set of synaptic weights, and the set of transfer functions. Also two fitness functions are used (the mean square error—MSE and the classification error—CER) which involve the validation stage to guarantee a good ANN performance. First, a study of the best parameter configuration for DE algorithm is conducted. The experimental results show the performance of the proposed methodology to solve pattern classification problems. Next, a comparison with two classic learning algorithms—gradiant descent and Levenberg-Marquardt—are presented.**

***Index Terms*—Differential evolution, evolutionary neural networks, pattern classification.**

I. Introducción

Las redes neuronales artificiales han sido por muchos años una herramienta indispensable en el área de la Inteligencia Artificial debido a su aplicación satisfactoria en lo concerniente a la clasificación de patrones y la predicción de series de tiempo, entre otras problemáticas. Estos sistemas se basan en el comportamiento que tiene la red neuronal biológica del cerebro. Ramón y Cajal [1], fue el primer científico en demostrar que el sistema nervioso se compone de células individuales llamadas neuronas, las cuales se conectan entre ellas, creando un sistema complejo de comunicación y cuyo procesamiento de información es hasta el día de hoy, un misterio científico.

Una RNA emplea un mecanismo de aprendizaje en la etapa de entrenamiento, donde se optimiza una función que evalúa la salida de la red; con ello se determina la eficiencia del aprendizaje. Después de ser entrenada, la RNA puede ser utilizada para resolver algún problema con información totalmente desconocida pero, que puede dar un veredicto correcto conforme lo aprendido. Esta etapa recibe el nombre de generalización.

Los pesos sinápticos, las funciones de transferencia, el número de neuronas y el tipo de arquitectura o topología (determinado por las conexiones entre neuronas) son escenciales y determinantes en el desempeño de una RNA. Por este motivo, es importante seleccionar los parámetros del diseño de manera adecuada para obtener la mejor

Manuscript received April 22, 2012. Manuscript accepted for publication July 20, 2012.

Beatriz A. Garro y Humberto Sossa pertenecen al Centro de Investigación en Computación del Instituto Politécnico Nacional, CIC-IPN, Av. Juan de Dios Bátiz s/n, esquina con Miguel de Othón de Mendizábal, 07738, Ciudad de México, México (Email: bgarrol@ipn.mx, hsossa@cic.ipn.mx).

Roberto A. Vazquez pertenece al Grupo de Sistemas Inteligentes, Facultad de Ingeniería, Universidad la Salle, Benjamín Franklin 47, Col. Hipódromo Condesa, 06140, Ciudad de México, México (Email: ravem@lasallistas.org.mx).

eficiencia en la red neuronal. Sin embargo, expertos en el área generan arquitecturas en un procedimiento de prueba y error, seleccionando de entre ellas, aquella que otorga el mejor desempeño. Esto puede provocar que no se explore adecuadamente otras formas de diseñar que pueden otorgar mejores resultados. Por otra parte, para entrenar la red neuronal se selecciona un tipo de algoritmo que ajusta los pesos sinápticos hasta otorgar el comportamiento deseado de la red. El más utilizado es el algoritmo de retropropagación (back-propagation BP) [2], el cuál se basa en la técnica del gradiente descendiente. Las técnicas de ajuste que se basan en el cálculo de derivadas como BP, presentan un problema: no pueden ajustar información proveniente de problemas que son no continuos. Por otro lado, sólo pueden trabajar con problemas donde sólo existe un valor óptimo (o en el peor caso, el mejor) esto quiere decir que al presentarse problemas multimodales (donde se presentan varios valores óptimos) el ajuste podría no ser el mejor ya que podría hacercarse a una falsa solución.

Por tal motivo, el deseo de construir un diseño adecuado que resuelva problemas del mundo real giró hacia otras técnicas inspiradas en la naturaleza como los llamados algoritmos evolutivos [3]. Estos algoritrmos heurísticos se utilizan para resolver problemas no lineales que no pueden ser resueltos por técnicas clásicas de optimización, donde el espacio de búsqueda es muy grande, combinatorial y/o multimodal. Estas técnicas se basan en conceptos biológicos como lo es el Neo-Darwinismo, teoría formada por el pensamiento de Charles Darwin sobre evolución de las especies [4], el pensamiento de August Weismann sobre el plasma germinal responsable de la herencia [5] y el pensamiento de Gregor Mendel sobre la teoría de las leyes de la herencia [6].

Los algoritmos evolutivos son procesos que mezclan un concepto de individuos distribuidos en un espacio de búsqueda determinado. La población de individuos no es más que un conjunto de posibles soluciones que, en un determinado tiempo, deben converger a la solución óptima (si se conoce) o simplemente converger a la mejor solución (se espera que sea la más cercana a la óptima). Estos individuos cambian (evolucionan) al realizar operaciones de cruza o mutación para mejorar su desempeño mientras se incrementa el tiempo. Para indicar cuál individuo representa la mejor solución se deben evaluar cada una de las soluciones o individuos en una función de aptitud o función objetivo. La función de aptitud es diseñada de tal manera que se realice una optimización en el desempeño de los individuos y se indique que tan bueno es cada uno para resolver un determinado problema. El proceso termina al guardar el mejor individuo que genera la mejor evaluación en la función de aptitud.

Estas técnicas de optimización han sido empleadas para ajustar los pesos sinápticos de una RNA debido a su eficiencia para resolver problemas complejos. Algunas investigaciones como [7], [8] proponen una modificación de Evolución diferencial para ajustar los pesos sinápticos de una red neuronal multicapa. En [9] tres arquitecturas con diferentes técnicas de entrenamiento, entre ellas ED, son aplicados a la predicción del clima. En [10], los autores utilizan ED para el ajuste de los pesos sinápticos de una RNA, así mismo utilizan la técnica llamada optimización por enjambre de partículas (PSO por sus siglas en inglés), proporcionando como métrica el error cuadrático medio (MSE por sus siglas en inglés). La incursión del algoritmo de Evolución diferencial no ha sido muy explorado para el diseño de RNA. Sin embargo, otros algoritmos evolutivos y algoritmos bioinspirados han sido aplicados en el entrenamiento de las RNA y la selección del número de neuronas en un número de capas especificado por el diseñador [11]. En [12] los autores presentan el diseño de una RNA generada por el algoritmo ED, el cuál evoluciona al mismo tiempo la arquitectura (topología), pesos sinápticos y funciones de transferencia utilizando como métrica el error cuadrático medio.

En esta investigación se decribe la metodología que permite diseñar de manera automática la arquitectura, el conjunto de pesos sinápticos y las funciones de transferencia por cada neurona que componene a una RNA. El diseño será generado al aplicar el algoritmo evolutivo llamado Evolución diferencial, el cuál evaluará las soluciones mediante dos funciones de aptitud. La primera función toma en cuenta el error cuadrático medio (mean square error-MSE) y la etapa de validación la cuál impide generar redes neuronales con el problema de sobreaprendizaje. La segunda utiliza el error de clasificación (CER) también considerando la validación. Estas funciones de aptitud, son muy adecuadas ya que los problemas que se quieren resolver son problemas de clasificación de patrones.

La estructura del escrito está dividida en las siguientes secciones: la sección 2 describe los conceptos básicos de una RNA; en la sección 3 se introduce a la técnica de Evolución diferencial; la descripción del diseño automático se detalla en la sección 4; la sección 5 describe cómo se obtiene la salida general de la RNA; los resultados experimentales se encuentran en la sección 6 y finalmente en la sección 7 se pueden encontrar las conclusiones que cierran esta investigación.

II. Redes neuronales artificiales

La red neuronal artificial (RNA) basada en el sistema neuronal biológico, es un sistema computacional que permite realizar un mapeo de un conjunto de datos o patrones de entrada a un conjunto de salida. En [13], Kohonen describe: “Las redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico”. Las RNA se componen por unidades básicas llamadas neuronas, las cuales estan conectadas entre sí y dependiendo de la capa a la que pertenezcan pueden modificar la información que reciben o simplemente la envían tal y como la recibieron a otras neuronas con las que tienen conexión sináptica.

Las neuronas se encuentran organizadas por capas. Algunas neuronas componen la capa de entrada, la cuál, se encarga de recibir la información del entorno o problema a resolver. Otras neuronas forman la capa de salida las cuales, entregan un patrón asociado al patrón de la entrada. Las neuronas restantes constituyen la llamada capa oculta donde la información es procesada, enviada a otras neuronas y evaluada en funciones de transferencia que entregarán una salida por cada neurona. Cada conexión indica un peso sináptico y está representado por un valor numérico determinado. Dependiendo del tipo de conexión que se tenga, es el tipo de flujo de información. Puede ser hacia adelante, cuando la información fluye desde la capa de entrada hacia la de salida (flujo unidireccional) [14] o puede ser recurrente [15] es decir, cuando la información fluye en ambos sentidos con presencia de posibles retroalimentaciones. La salida de la red neuronal está dada por las neuronas de la capa de salida, las cuales conjuntan toda la información procedente de capas anteriores. Dicha salida permite evaluar el diseño de la red neuronal.

Para que una RNA resuelva un determinado problema, su diseño necesita ser evaluado mediante una etapa de entrenameinto en la cuál se lleva a cabo el aprendizaje. Esta etapa consiste en alimentar la red con patrones que codifican un problema específico. Esta información pasa por cada capa de la red en donde es procesada por los pesos sinápticos y después se transforma por medio de funciones de transferencia. Este hecho se da hasta alcanzar la capa de salida. Si la métrica que se utiliza para medir la salida no es la deseada, los pesos sinápticos cambian con ayuda de una regla de aprendizaje con el fin de volver al paso anterior y así generar una mejor salida que la anterior.

Existen varios tipos de aprendizaje de una RNA [16]. El aprendizaje supervisado será utilizado en esta investigación. El aprendizaje supervisado consite en asociar un conjunto de patrones de entrada con un correspondiente patrón deseado el cuál, es conocido. De tal manera que se puede supervisar si la salida de la RNA es la deseada o no.

Cuando la red neuronal ya ha aprendido, es de sumo cuidado conocer si no aprendió de más, es decir que se haya convertido en un sistema experto en la resolución del problema con el que se entrenó. Al aprender de manera experta cada entrada, el sistema neuronal será incapás de reconocer alguna entrada contaminada con algún error y no podrá reconocer nueva información que determina el mismo problema a resolver. Este problema se resuelve utilizando una etapa de validación [17] la cuál consite en tomar un conjunto de patrones diferentes al conjunto de entrenamiento, y probarlos con la red entrenada. Si el error que se genera es menor al error en el aprendizaje, la RNA continúa ajustando sus pesos sinápticos, pero, si el error que se genera con el conjunto de validación es mayor al error generado por el entrenamiento, la etapa de aprendizaje debe ser suspendida para evitar el sobreaprendizaje.

Después de ser entrenada y probada con el conjunto de validación, la RNA estará lista para recibir patrones de datos diferentes a los utilizados durante el entrenamiento y así realizar una generalización eficiente, la cuál determina qué tan bueno fue el aprendizaje de la red y que tan robusta para resolver el problema, en este caso de clasificación de patrones.

Para entender más a detalle cómo opera una RNA a continuacíon se explica el funcionamiento de sus elementos esenciales [18].

A. *Entradas de la RNA*

El conjunto de entradas $x_j(t)$ de la RNA es un conjunto de patrones los cuales codifican la información de un problema que se quiere resolver.

Las variables de entrada y de salida pueden ser binarias (digitales) o continuas (analógicas), dependiendo del modelo y la aplicación. Por ejemplo un perceptrón multicapa (MLP Multilayer Perceptron, por sus siglas en inglés) puede trabajar con ambos tipos de señales. En el caso de salidas digitales las señales se pueden representar por $0, +1$, en el caso de las salidas analgógicas la señal se da en un cierto intervalo.

B. *Pesos sinápticos*

Los pesos sinápticos w_{ij} de la neurona i son variables relacionadas a la sinapsis o conexión entre neuronas, los cuales representan la intensidad de iteracción entre la neurona presináptica j y la postsináptica i. Dada una entrada positiva (puede ser del conjunto de datos de entrada o de la salida de otra neurona), si el peso es positivo tenderá a exitar a la neurona postsináptica, si el peso es negativo tenderá a inhibirla.

C. *Regla de propagación*

La regla de propagación permite obtener, a partir de las entradas y los pesos sinápticos, el valor del potencial postsináptico h_i de la neurona i en función de sus pesos y entradas.

$$h_i(t) = \sigma_i(w_{ij}, x_j(t)) \tag{1}$$

La función más habitual es de tipo lineal, y se basa en la suma ponderada de las entradas con los pesos sinápticos

$$h_i(t) = \sum_j w_{ij} x_j \tag{2}$$

que también puede interpretarse como el producto escalar de los vectores de entrada y pesos

$$h_i(t) = \sum_j w_{ij} x_j = \mathbf{w}_i^T \mathbf{x} \tag{3}$$

D. Función de transferencia

La función de transferencia f de la neurona i proporciona el estado de activación actual $a_i(t)$ a partir del potencial postsináptico $h_i(t)$ y del propio estado de activación anterior $a_i(t-1)$

$$a_i(t) = f_i(a_i(t-1), h_i(t)) \tag{4}$$

Sin embargo, en muchos modelos de RNA se considera que el estado actual de la neurona no depende de su estado anterior, si no únicamente del actual

$$a_i(t) = f_i(h_i(t)) \tag{5}$$

E. Función de salida

La función de salida proporciona la salida global de la neurona $y_i(t)$ en función de su estado de activación actual $a_i(t)$. Muy frecuentemente la función de salida es simplemente la identidad $F(x) = x$ de tal modo que el estado de activación de la neurona se considera como la propia salida de la red neuronal

$$y_i(t) = F_i(a_i(t)) = a_i(t) \tag{6}$$

III. Evolución diferencial

En 1994 surgió un adaptativo y eficiente esquema: el algoritmo de Evolución diferencial, propuesto por Kenneth Price y Rainer Storn. Este algoritmo se utilizó para la optimización global sobre espacios continuos [19]. Debido a su capacidad de exploración sobre un espacio de búsqueda, dado un problema, el algoritmo de Evolución diferencial (DE por sus siglas en inglés) evita quedar atrapado en mínimos locales. Este algoritmo tiene pocos parámetros y converge más rápido al óptimo en comparación con otras técnicas evolutivas. Todas estas características convierten a este algoritmo en una excelente técnica para la optimización de problemas no diferenciables. La idea detrás de esta técnica de optimización es generar vectores de prueba.

Dado una población de vectores $\vec{x_i} \in \mathbb{R}^D, i = 1, ..., M$ en un espacio multidimensional D, el algoritmo consiste, en eligir de manera aleatoria un vector objetivo $\mathbf{x}_i$ y un vector base $\mathbf{x}_{r_3}$, donde r es un número aleatorio entre $[1, M]$. Por otro lado, se deben elegir aleatoriamente dos miembros de la población $\mathbf{x}_{r_1}$ y $\mathbf{x}_{r_2}$, y se realiza una diferencia entre ellos. A continuación, el resultado es operado por un factor constante, denotado por F, así se obtendrá un vector ponderado. Inmediatamente después, el vector ponderado y el vector base son sumados. El nuevo vector que surge, se le llamará vector mutante $\mathbf{u}_i$.

Finalmente se realiza la operación de cruza, la cuál involucra una comparación (variable por variable) entre el vector mutante y el vector objetivo. De la operación de cruza se genera un nuevo vector llamado vector de prueba. La comparación consiste en una simple regla: si un número aleatorio es menor que el factor de cruza CR entonces, la variable del vector que se elige es la del vector mutante; si no, entonces se elige la variable del vector objetivo, así el vector prueba será una mezcla de variables del vector mutante y el vector objetivo. Finalmente el último paso es la selección del mejor vector (aquél con la mejor aptitud, según sea el tipo de optimización). Esta selección involucra comparar la aptitud del vector objetivo y la del vector prueba.

Existen varias estrategias de ED [20]. La estrategia descrita en esta sección es la estrategia técnicamente llamada "DE/rand/1/bin" cuyo pseudocódigo se muestra en el Algoritmo 1 tomado de [21]. Esta nomenclatura cambia dependiendo del tipo de estrategia que se esté implementando. Las diferentes estrategias varían al cambiar la siguiente nomenclatura, donde DE/*x*/*y*/*z* toma las siguientes variables: *x* se refiere a cómo será elegido el vector objetivo, puede ser aleatorio o el mejor de la población. *y* se refiere a cuántos pares de vectores se tomarán para realizar la diferencia; puede ser un par y sumarle un tercer vector (vector base) o puede ser dos pares cuya respectiva diferencia se sume a un quinto vector (vector base).

El tipo de cruza se representa por *z*. Ésta puede ser del tipo *bin* (cruza binomial) en donde para cada variable dada una probabilidad, se hereda la información de uno u del otro vector o puede ser del tipo *exp* (cruza exponencial) en donde dado un número aleatorio entre $(0, 1)$, si dicho número es mayor a CR entonces se suspende la cruza [22].

IV. Diseño automático de una RNA mediante ED

Para aplicar el algoritmo de Evolución diferencial es necesario codificar el individuo o la solución con la información que se requiere. El algoritmo ED generará una población de dichas soluciones que evolucionarán en un número de generaciones (iteraciones) y se evaluará cada una de ellas en una función de aptitud. Dicha función de aptitud nos indicará que tan buena es la solución, guardando la mejor al finalizar la ejecución del mismo.

La descripción de cada elemento en el algoritmo evolutivo se explica a continuación.

A. Codificación del individuo

Un individuo representa una solución. La codificación de ese individuo consiste en contar con la información necesaria para el diseño de una red neuronal artificial. Como se mencionó en las restricciones de esta investigación, la metodología será aplicada por el momento a problemas de clasificación de patrones.

En general el problema a resolver se puede describir de la siguiente manera.

Dado un conjunto X con p patrones que definen un problema de clasificación definido por $X = \{\mathbf{x}^1, ..., \mathbf{x}^p\}, \mathbf{x} \in \mathbb{R}^n$, y dado un conjunto D con p patrones deseados que definen la clase a la que pertenece cada patrón definido por $D = \{\mathbf{d}^1, ..., \mathbf{d}^p\}, \mathbf{d} \in \mathbb{R}^m$; encontrar una RNA, cuyo diseño está representado por una matriz $\mathbf{W}$ donde $W \in \mathbb{R}^{q \times (q+3)}$ tal

Algorithm 1 Pseodocódigo de Evolución diferencial al aplicar la estrategia “DE/rand/1/bin”. CR es un número entre $(0,1)$, $MAXITER$ es el número máximo de iteraciones, G es una iteración específica, M es el número total de individuos en la población , $randint(1,D)$ es una función que regresa un número entero entre 0 y D. $rand_j[0,1)$ es una función que regresa un número real entre $(0,1)$. Ambas funciones basadas en una probabilidad de distribución uniforme.

```
$G = 0$
Crear una población inicial aleatoria $\vec{\mathbf{x}}_{i,G} \forall i, i = 1, \ldots, M$
Evaluar $f(\vec{\mathbf{x}}_{i,G}) \forall i, i = 1, \ldots, M$
for $G = 1$ hasta $MAXITER$ do
  for $i = 1$ hasta $M$ do
    Seleccionar aleatoriamente $r_1 \neq r_2 \neq r_3$
    $j_{rand} = randint(1, D)$
    for $i = 1$ hasta $M$ do
      if $rand_j[0,1) < CR$ o $j = j_{rand}$ then
        $\mathbf{u}_{i,j,G+1} = \mathbf{x}_{r_3,j,G} + F \cdot (\mathbf{x}_{r_1,j,G} - \mathbf{x}_{r_2,j,G})$
      else
        $\mathbf{u}_{i,j,G+1} = \mathbf{x}_{i,j,G}$
      end if
    end for
    if $f(\vec{\mathbf{u}}_{i,G+1}) \leq f(\vec{\mathbf{x}}_{i,G})$ then
      $\vec{\mathbf{x}}_{i,G+1} = \vec{\mathbf{u}}_{i,G+1}$
    else
      $\vec{\mathbf{x}}_{i,G+1} = \vec{\mathbf{x}}_{i,G}$
    end if
  end for
  $G = G + 1$
end for
```

que una función f sea optimizada $(min) f(D, X, W)$, donde q es el número de neuronas.

B. Individuo

El individuo que representa el diseño de una RNA, está dado por una matriz **W**. Esta matriz está compuesta por tres partes: la topología (*T*), los pesos sinápticos (*SW*) y las funciones de transferencia (*TF*), tal como se muestra en la Figura 1.

El tamaño del individuo depende de un número máximo de neuronas (MNN), el cuál está definido por q. En esta investigación se desarrolló una ecuación para obtener el MNN, la cuál depende del problema a resolver. Ésta se encuentra definida a continuación:

$$q = (m+n) + \left(\frac{m+n}{2}\right) \tag{7}$$

donde n es la dimensión del vector de los patrones de entrada y m es la dimensión del vector de los patrones deseados.

$$\left[\underbrace{\begin{matrix} w_{1,1} \\ \vdots \\ w_{1,1} \end{matrix}}_{T} \; \underbrace{\begin{matrix} w_{1,2} & \cdots & w_{1,MNN+1} \\ \vdots & \ddots & \vdots \\ w_{MNN,2} & \cdots & w_{MNN,MNN+1} \end{matrix}}_{SW} \; \underbrace{\begin{matrix} w_{1,MNN+2} \\ \vdots \\ w_{MNN,MNN+2} \end{matrix}}_{TF}\right]$$

Fig. 1. Representación del individuo que codifica la topología (T), los pesos sinápticos (SW) y las funciones de transferencia TF).

Debido a que la matriz está compuesta por tres diferentes informaciones, se consideró tres rangos diferentes para cada una. En el caso de la topología, el rango se encuentra entre $\left[1, 2^{MNN} - 1\right]$, los pesos sinápticos tienen un rango de $[-4, 4]$ y para las funciones de transferencia el rango es $[1, nF]$, donde nF es el número total de funciones de transferencia a utilizar. Al generar los individuos, todas las matrices **W** están compuestas de valores reales en sus respectivos rangos y al momento de decodificar la información para ser evaluados en la función de aptitud, tanto la arquitectura y la función de transferencia sufren un redondeo del tipo $\lfloor x \rfloor$.

C. Arquitectura y pesos sinápticos

Para poder evaluar los diseños de las RNA, es necesario decodificar la información del individuo. La primera información a decodificar es la topología o arquitectura (*T*), la cual se evalúa con los pesos sinápticos (*SW*) y las funciones de transferencia (*TF*) codificados en la misma matriz.

Para hacer válida una topología de RNA para la metodología propuesta, se deben de seguir ciertas reglas. Las RNA generadas están compuestas de tres capas: la capa de entrada, la capa oculta y la capa de salida y las reglas de conexión entre las neuronas de cada capa siguen las siguientes condiciones.

Donde ILN es el conjunto de I neuronas que componen la capa de entrada, HLN es el conjunto de J neuronas pertenecientes a la capa oculta y OLN es el conjunto de K neuronas, las cuales pertenecen a la capa de salida.

Primera regla (para las neuronas de la capa de entrada-ILN): La neurona ILN_i, i = 1,..., I sólo puede enviar información a las neuronas de la capa oculta HLN_j y a las neuronas de la capa de salida OLN_k.

Segunda regla (para las neuronas de la capa oculta-HLN): La neurona HLN_j, j = 1,..., J sólo puede enviar información a las neuronas de la capa de salida OLN_k y a las neuronas de la capa oculta HLN_j pero, para ésta última con una restricción. Para la neurona HLN_j sólo puede haber conexión con las neuronas del tipo $HLN_{j+1}, ..., HLN_J$.

Tercera regla (para las neuronas de la capa de salida-OLN): La neurona OLN_k, k = 1,..., K sólo puede enviar información a otras neuronas de su misma capa pero, con una restricción. Para la neurona OLN_k sólo puede existir conexión con neuronas del tipo $OLN_{k+1}, ..., OLN_K$.

Para decodificar la arquitectura siguiendo las tres reglas de conexión y recordando que la información en $\mathbf{W}_{ij}$ con $i = 1, ..., MNN$ y $j = 1$ está en base decimal, sufre un redondeo como se explicó anteriormente y se codifica a binario en una matriz **Z**. Esta matriz representará un grafo (arquitectura de la RNA), donde cada componente de la matriz z_{ij} indica las aristas o conexiones entre la neurona (vértices del grafo) i y la neurona j cuando $z_{ij} = 1$. Por ejemplo: supongamos que $\mathbf{W}_{i,1}$ tiene un número entero “57” el cuál será transformado en su correspondiente número binario “0111001”. Este número binario indica las conexiones de la neurona i-ésima con siete neuronas (número de bits en la cadena binaria). En este caso, solo las neuronas en la posición de la cadena binaria de izquierda a derecha donde exista un 1 como la neurona dos, tres, cuatro y siete, tendrán una conexión a la neurona i.

Al extraer de la matriz **W** la arquitectura se evaluará con los correspondientes pesos sinápticos de la componente $\mathbf{W}_{ij}$ con $i = 1, ..., MNN$ y $j = 2, ..., MNN + 1$. Finalmente cada neurona de la arquitectura calculará su salida con su correspondiente función de transferencia indicada en la misma matriz.

D. *Funciones de transferencia*

Las funciones de transferencia (*TF*) se encuentran representadas en la componente $\mathbf{W}_{ij}$ con $i = 1, ..., MNN$ y $j = MNN + 3$. Dependiendo del valor entero en la componente, se eligirá una de las funciones propuestas en esta investigación.

Aunque existen otras funciones de transferencia que pueden ser implementadas en el contexto de las RNA, en esta investigación se utilizarán sólo las más utilizadas en el área. Estas TF con su respectiva nomenclatura son: la función sigmoide (LS), la función hipertangencial (HT), la función seno (SN), la función gausiana (GS), la función lineal (LN) y la función límite duro (HL).

Hasta este momento se ha explicado la codificación del individuo y la forma de decodificar la información cuando se hace la evaluación de la solución en la función de aptitud. En la siguiente sección se explican las diferentes funciones de aptitud desarrolladas en esta investigación.

E. *Funciones de Aptitud*

La función de aptitud permite saber que tan buena es la solución dependiendo del problema de optimización que se quiere resolver. En este caso, el problema de optimización es del tipo $minf(x)|x \in A \subseteq \mathbb{R}^n$ donde $\mathbf{x} = (x_1, ..., x_n)$ y n es la dimensiónalidad.

En esta investigación dos funciones de aptitud fueron aplicadas. La primera consiste en evaluar a las diferentes soluciones que se generen utilizando el error cuadrático medio (MSE) sobre el conjunto de entrenamiento MSE_T y sobre el conjunto de validación MSE_V ver Ec. 8. La segunda función consiste en considerar al mismo tiempo el error de clasificación (CER) sobre el conjunto de entrenamiento CER_T y el de validación CER_V, dándole más peso al error de validación, ver Ec. 9.

$$F_1 = 0.4 \times (MSE_T) + 0.6 \times (MSE_V) \qquad (8)$$

$$F_2 = 0.4 \times (CER_T) + 0.6 \times (CER_V) \qquad (9)$$

El desempeño de estas funciones de aptitud serán presentadas más adelante.

V. Salida de la red neuronal artificial

Una vez decodificada la información del individuo, se calcula la salida de la RNA, de tal forma que, es posible determinar la eficiencia de la red mediante la función de aptitud. Dicha salida se calcula aplicando el Algoritmo V.

Algorithm 2 Pseudocódigo de la salida de la RNA. o_i es la salida de la neurona i, a_j es el patrón de entrada a la RNA, n es la dimensionalidad del patrón de entrada, m es la dimensionalidad del patrón deseado y y_i es la salida de la RNA.

```
for $i = 1$ hasta $n$ do
    Calcular $o_i = a_i$
end for
for $i = n + 1$ hasta $MNN$ do
    Obtener el vector de conexiones z de la neurona $i$ a partir del individuo $W_i$.
    Obtener los pesos sinápticos s de la neurona $i$ a partir del individuo $W_i$.
    Obtener el bias $b$ de la neurona $i$ a partir del individuo $W_i$.
    Obtener el índice $t$ de la función de transferencia de la neurona $i$ a partir del individuo $W_i$.
    Calcular la salida de la de la neurona $i$ como $o_i = f_t\left(\sum_{j=1}^{i} s_j \cdot z_j \cdot a_j + b_j\right)$.
end for
for $i = MNN - m$ hasta $MNN$ do
    Calcular la salida de la RNA con, $y_{i-MNN-m+1} = o_i$.
end for
```

Para el caso de la función de aptitud F_2, la salida de la red será modificada mediante la técnica del *ganador toma todo*, es decir la neurona que genere el valor mas alto en su salida se le asignará el valor binario de uno y a las restantes se les asignara el valor de cero. Esta nueva salida binaria será comparada con el conjunto de patrones deseados asignados para cada problema de clasificación.

VI. Resultados experimentales

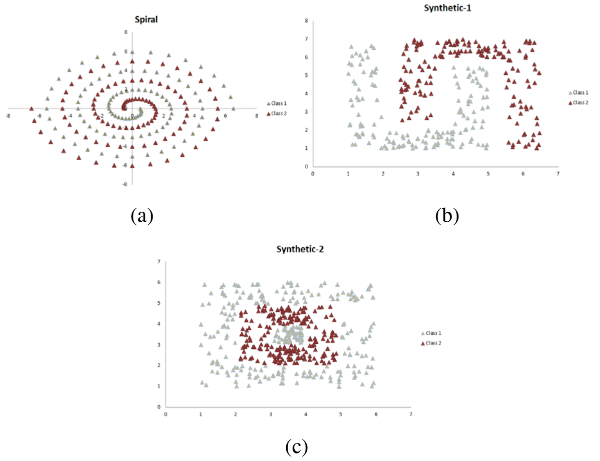
La metodologá propuesta se evaluó al resolver problemas de clasificación de patrones. Como se mencionó anteriormente, se utilizará un aprendizaje supervisado, lo que indica que será utilizado un conjunto de patrones deseados.

Debido que el algoritmo de Evolución diferencial presenta algunos parámetros, la configuración de los mismos puede repercutir en el desempeño de los resultados. Por tal motivo, se realizó un estudio de sensibilidad para encontrar la mejor configuración de los parámetros de ED y así obtener los mejores resultados en la experimentación. Para encontrar dicha configuración se propusieron diferentes valores para cada parámetro y se evaluaron los diseños de RNA con las funciones de aptitud F_1 y F_2.

Al obtener la mejor configuración del algoritmo se procedió a realizar la experimentación que proporcionará datos estadísticamente válidos sobre el reconocimiento, los mejores resultados y la evaluación de los errores que se generaron al aplicar la metodología propuesta.

A. *Descripción de la experimentación*

Para evaluar el desempeño de la metodología, se seleccionaron diez problemas de clasificación de patrones de diferente complejidad. El problema de la planta del Iris, el del vino, el cáncer de mama, el problema de diabetes, el de desórdenes del hígado y el problema del vidrio son problemas que se encuentran en el repositorio de UCI machine learning [23]. El problema de reconocimiento de objetos se obtubo de [24], y los problemas como la espiral y las dos sintéticas se desarrollaron en nuestro laboratorio. La Figura 2 muestra los patrones dispersos de éstos últimos problemas.



(a)

(b)

(c)

Fig. 2. Dispersión de datos para los problemas sintéticos. (a) Datos del problema de espiral. (b) Datos del problema sintético 1. (c) Datos del problema sintético 2.

En la Tabla I se encuentra la descripción de los patrones de cada problema de clasificación.

Para obtener los tres conjuntos de datos para entrenar y validar la RNA, se dividió el número de patrones totales de cada base de datos en tres conjuntos: el de entrenamiento, el de validación y el de generalización. La selección de los patrones que componene cada conjunto se realizó de manera aleatoria con el fin de validar estadísticamente los resultados obtenidos. Esta selección tiene una distribución del 33% de los patrones totales para el entrenamiento, 33% para la validación y 34% para la generalización para cada problema de clasificación.

TABLA I
DESCRIPCIÓN DE LOS PROBLEMAS DE CLASIFICACIÓN DE PATRONES.

Problemas de clasificación	Descripción de los patrones	Patrones totales
Rec. objetos.	7 caractéristicas que describen 5 clases	100
Planta Iris	4 caractéristicas que describen 3 clases	150
Vino	13 caractéristicas que describen 3 clases	178
Cáncer de mama	7 caractéristicas que describen 2 clases	683
Diabetes	8 caractéristicas que describen 2 clases	768
Desórdenes del hígado	6 caractéristicas que describen 2 clases	345
Vidrio	9 caractéristicas que describen 6 clases	214
Espiral	2 caractéristicas que describen 2 clases	194
Sintética 1	2 caractéristicas que describen 2 clases	300
Sintética 2	2 caractéristicas que describen 2 clases	450

B. *Análisis de sensibilidad*

El análisis de sensibilidad consiste en evaluar los resultados obtenidos con diferentes valores asignados a los parámetros del algoritmo de Evolución deferencial. De este modo, se puede determinar cómo las diferentes configuraciones proporcionan desempeños variados, de los cuales podemos seleccionar la mejor configuración con el que el algoritmo se desempeña mejor.

La configuración para detrerminar cuál es el valor para cada parámetro está determinada por la siguiente secuencia: $v-w-x-y-z$ donde cada variable representa un parámetro. Para el caso del número de individuos en la población la variable $v = \{50, 100\}$ donde el elemento 1 corresponde a 50 individuos y el elemento 2 corresponde a 100. En el caso de el tamaño del espacio de búsqueda $w = \{2, 4\}$ el primer elemento indica que el rango se establece en $[-2, 2]$ y el caso del segundo elemento indica que el rango está determinado entre $[-4, 4]$. Para determinar el tipo de función de aptitud, la variable $x = \{3, 4\}$ indica con el primer elemento, que será seleccionada la función F_1 y el segundo elemento define la selección de la función F_2. El algoritmo ED, tiene dos parámetros propios: el factor de cruza CR y una constante F, las cuales se representan por las variables y y z respectivamente. Ambas variables toman los valores $y = z = \{1, 2, 3\}$ donde dichos elementos estan asociados a los siguientes valores $\{0.1, 0.5, 0.9\}$. El número de generaciones o iteraciones del algoritmo se fijó en 2000 y el número de experimentos para cada combinación de los parametros, para cada problema de clasificación y cada función de aptitud se fijó en cinco corridas del algoritmo. De tal manera que la secuencia 2-1-3-2-3 significa que la configuración correspondiente es: $100 - [-2, 2] - F_1 - 0.5 - 0.9$

Se debe considerar que para cada experimento se obtienen dos valores: el error de entrenamiento y el error de generalización. Por ese motivo, se obtuvo una ponderación de estos dos valores y así se determinó quien presenta mejores resultados. Con ayuda de la evaluación ponderada de la suma del entrenameinto y la generalización se decidió asignar mayor peso a la etapa de generalización al ser multiplicada por un factor de 0.6 y en el caso del entrenamiento el factor es 0.4. La ecuación que determina la ponderación es la que se describe en Ec. 10.

$$rp = 0.4 \times (Training) + 0.6 \times (Test) \qquad (10)$$

donde $Training$ representa el porcentaje de reconocimiento durante la etapa de entrenamiento y $Test$ representa el porcentaje de reconocimineto obtenido durante la etapa de generalización.

Para cada configuración de los parámetros y cada problema se realizaron 5 experimentos, de los cuales se obtubo un promedio ponderado rp. La Tabla II muestra el mejor promedio ponderado para las dos funciones de aptitud y cada problema de clasificación de patrones, así como también muestra las configuraciones de los parámetros que generaron dichos valores.

TABLA II
PROMEDIO PONDERADO PARA CADA PROBLEMA DE CLASIFICACIÓN Y CADA FUNCIÓN DE APTITUD.

Problemas de clasificación	Función aptitud F_1		Función aptitud F_2	
	Promedio P.	Config.	Promedio P.	Config.
Espiral	0.3753	2,2,3,2,2	0.3062	2,2,4,1,1
Sintética 1	0.0128	1,2,3,1,2	0.0060	2,1,4,2,1
Sintética 2	0.1400	2,2,3,2,2	0.0997	2,2,4,2,1
Planta de Iris	0.0256	2,2,3,3,3	0.0216	2,2,4,1,3
Cáncer de mama	0.0209	2,1,3,3,3	0.0204	2,1,4,2,2
Diabetes	0.2181	1,1,3,3,3	0.2170	2,1,4,3,1
Desórdenes del hígado	0.2845	1,1,3,2,1	0.2793	2,2,4,3,2
Rec. Objetos	0.0	2,2,3,3,2	0.0	1,1,4,1,1
Vino	0.0183	2,1,3,3,2	0.0237	2,1,4,1,3
Vidrio	0.3346	1,1,3,3,3	0.3425	2,1,4,1,1

En la Tabla II se puede observar que los mejores valores fueron obtenidos con la función CER, mostrando valores mínimos para ocho de diez problemas; la función de aptitud F_1 presentó mejores resultados aislados para el caso de los problemas del Vino y del Vidrio.

Los resultados anteriores muestran para cada problema de clasificación y cada función de aptitud, una configuración específica que genera el mejor desempeño del algoritmo evolutivo para el diseño de RNA. Sin embargo, debido a que se desea crear una metodologá no específica para cada problema a resolver, se buscó aquella configuación que en promedio resolviera mejor todas los bases de datos (una solución general), por lo que se obtubo por cada función de aptitud, el promedio que involucra el promedio ponderado rp de todos los problemas de clasificación. De lo anterior, se obtubo que para el caso de la función de aptitud F_1 el porcentaje de error ponderado fue de 15.37% y en el caso de la función de aptitud F_2 el porcentaje del error ponderado fue de 14.83%.

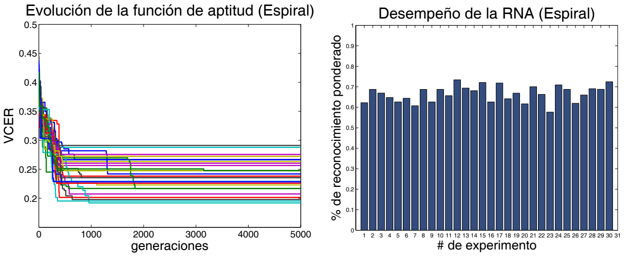
Como se mencionó anteriormente, el análisis de sensibilidad permite conocer cómo los valores de los diferentes parámetros afectan el desempeño del algoritmo ED y por consiguiente el diseño de las RNA. Por este motivo, se requiere de la minimización del error que se genere con las diferentes configuraciones. En este caso, el error mínimo se generó con la función de aptitud F_2 con la configuración 2-2-4-3-2. Esta configuración representa la mejor de entre todas las que se generaron al realizar la experimentación con todas las posibles configuraciones. Esta mejor configuración, será utilizada para realizar el análisis experimental que involucra todos los problemas de clasificación de patrones.

C. Análisis experimental

Para realizar una experimentación estadísticamente válida, se generaron 30 experimentos para cada problema de clasificación utilizando la mejor configuración para el algoritmo de ED, la cuál fue encontrada previamente en el análisis de sensibilidad. Los resultados para cada problema al utilizar la mejor configuración: una población de 100 individuos, en un espacio de búsqueda de $[-4, 4]$ con la función de aptitud F_2 y los valores para $CR = 0.9$ y para $F = 0.5$ se describen a continuación. Estos resultados se evaluaron durante un número de generaciones de 5000. Los resultados presentan la evolución del error, el porcentaje de reconocimiento ponderado que conjunta la etapa de entrenamiento y generalización así como las arquitecturas con las que se generó el mejor y el peor error.

A continuación se describen los resultados obtenidos con la mejor configuración para cada uno de los diez problemas de clasificación.

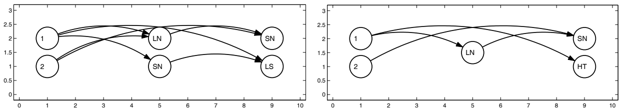
1) Espiral: La evolución del error obtenida de la función de aptitud F_2 se muestra en la Figura 3, en la cual el error de clasificación y la validación se conjuntan para obtener una red neuronal con el mejor desempeño, es decir con el mínimo valor encontrado por la función de aptitud. Como se puede observar, para la mayoría de las experimentaciones el error desciende casi en su totalidad antes de las 1000 generaciones, después el error se mantiene constante.



(a) (b)

Fig. 3. Resultados experimentales en los 30 experimentos para el problema de Espiral. (a) Evolución del error. (b) Porcentaje de reconocimiento ponderado.

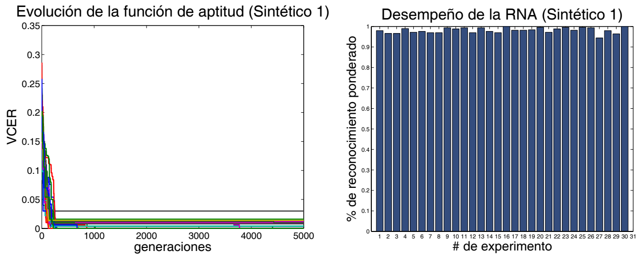
En la Figura 4 se muestran dos de las 30 arquitecturas que se generaron para el problema de la Espiral. La Figura 4(a) es el diseño con el cuál se generó el mejor error, en donde se puede apreciar que, para las neuronas de la capa intermedia se utilizaron las funciones lineal y seno como funciones de transferencia y en la capa de salida las funciones seno y sigmiode. En la Figura 4(b) se muestra la arquitectura que generó el peor error durante la experimentación. A diferencia de la mejor, esta arquitectura presenta menos conexiones. Cabe señalar que las funciones de transferencia son las mismas utilizadas por la mejor configuración.



(a) (b)

Fig. 4. Diseño de arquitecturas de RNA generadas para el problema de Espiral en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.

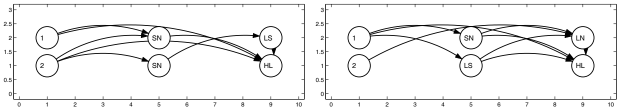
2) Sintética 1: Como se puede ver en la Figura 5(a), la evolución del error converge rápidamente a un error mínimo. La Figura 5(b) muestra el porcentaje de reconocimiento para la base de datos Sintética 1. En ella podemos observar que para los 30 experimentos, el desempeño de las RNA diseñadas presentan un entrenamiento y generalización muy altos, por arriba del 95% y en algunos casos alcanza el reconocimiento máximo (100%) para ambas etapas.



(a) (b)

Fig. 5. Resultados experimentales en los 30 experimentos para el problema Sintético 1. (a) Evolución del error. (b) Porcentaje de reconocimiento ponderado.

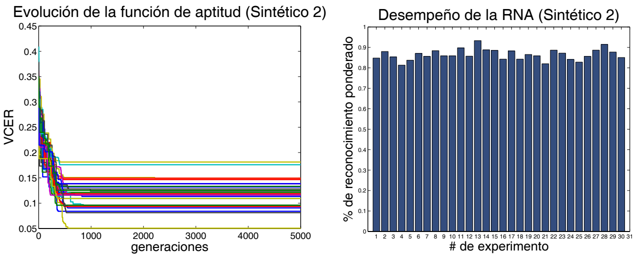
En la Figura 6 se presentan la mejor y la peor arquitectura encontradas para el problema Sintético 1. En ella podemos apreciar que la mejor arquitectura o topología presenta conexiones directas desde la capa de entrada hasta la capa de salida. Esto se debe a las reglas de conexiones propuestas anteriormente. Las funciones de transferencia utilizadas para dicha arquitectura fue la función seno, la sigmoide y la límite duro.



(a) (b)

Fig. 6. Diseño de arquitecturas de RNA generadas para el problema Sintético 1 en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.

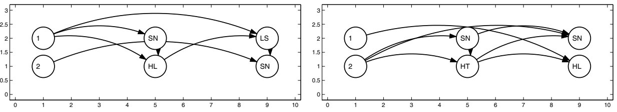
3) Sintética 2: La evolución del error para este problema de clasificación se muestra en la Figura 7(a), en donde se puede observar que el mínimo error generado por la función de aptitud no mejora después de las 1000 iteraciones. Por otro lado, la Figura 7(b) presenta el porcentaje de reconocimiento ponderado, en ella podemos ver que en más de la mitad de los experimentos totales, se obtubo un porcentaje por arriba del 85%.



(a) (b)

Fig. 7. Resultados experimentales en los 30 experimentos para el problema de Sintético 2. (a) Evolución del error. (b) Porcentaje de reconocimiento ponderado.

La Figura 8(a) muestra la arquitectura que alcanza el mínimo error (el mejor desempeño). Su arquitectura describe algunas conexiones directas desde la capa de entrada hasta la de salida. El conjunto de funciones de transferencia que utiliza cada neurona son: las funciones seno, límite duro y sigmoide. Para el caso de la peor arquitectura encontrada durante la experimentación, la Figura 8(b) muestra dicha topología diseñada con las funciones de transferencia seno, hipertangencial y la límite duro.



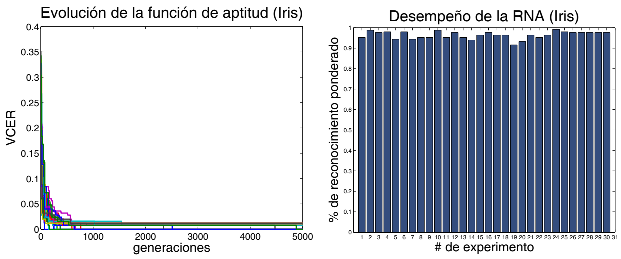
(a) (b)

Fig. 8. Diseño de arquitecturas de RNA generadas para el problema de Sintético 2 en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.

4) Planta de Iris: La Figura 9(a), muestra para las 5000 generaciones la evolución del error encontrado por la función de aptitud F_2. En ella podemos observar que al incrementar el número de generaciones el error disminuye drásticamente en algunos experimentos. Sin embargo, la mayoría de los experimentos alcanza su mejor valor antes de las 1000 generaciones.

En la Figura 9(b) se muestra el porcentaje de reconocimiento ponderado para cada experimento. Como se puede apreciar el desempeño de las RNA generadas alcanza un reconocimiento mayor al 90%.

En la Figura 10(a) se muestra la mejor arquitectura encontrada durante la experimentación. Este interesante y peculiar diseño, tiene una neurona que carece de conexión con otras en la capa de salida. Esta neurona trabaja sólo con



(a) (b)

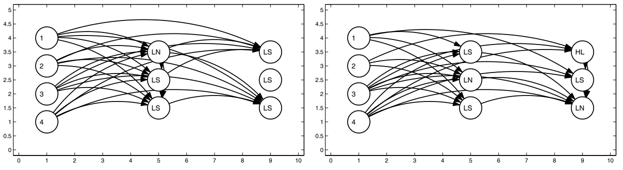
Fig. 9. Resultados experimentales en los 30 experimentos para el problema de la Planta de Iris. (a) Evolución del error. (b) Porcentaje de reconocimiento ponderado.

el bias correspondiente, que al ser evaluado en la respectiva función de transferencia sigmoide, se transforma a un valor que será exitosamente utilizado al momento de evaluar la salida de la RNA con la técnica el *ganador toma todo*. Al generarse las correspondientes salidas para cada patrón de entrada las neuronas restantes en la capa de salida son las que detallan la clase a la que pertenece cada patrón, mientras que la neurona que no tiene conexión funciona como un umbral fijo. Por ejemplo, suponga que la segunda neurona de salida genera en su salida un valor de 0.45, sin importar el patrón de entrada; por otro lado la primera y tercera neurona generan en su salida los valores de 0.65 y 0.55 respectivamente, al ser estimuladas con un patrón de entrada que pertenece a la clase 1. Al ser evaluada la salida de la RNA por la técnica del ganador toma todo, se obtendría la salida 1,0,0 la cuál indica que el patrón de entrada pertenece a la clase 1. Ahora suponga que la salida de la primera y tercera neurona generan en su salida los valores de 0.35 y 0.25 respectivamente, al ser estimuladas con un patrón de entrada que pertenece a la clase 2. En este caso, al ser evaluada la salida de la RNA por la técnica del *ganador toma todo*, se obtendría la salida 0,1,0 la cuál indica que el patrón de entrada pertenece a la clase 2, recuerde que la salida de la segunda neurona no cambia, es decir, permanece en 0.45 porque no está conectada con otras neuronas. Finalmente, en un tercer caso suponga que la salida de la primera y tercera neurona generan en su salida los valores de 0.35 y 0.65 respectivamente, al ser estimuladas con un patrón de entrada que pertenece a la clase 3. En este caso, al ser evaluada la salida de la RNA por la técnica del *ganador toma todo*, se obtendría la salida 0,0,1 la cuál indica que el patrón de entrada pertenece a la clase 3.

Las funciones de transferencia que este diseño necesita son las funciones lineal y sigmoide.

La arquitectura que genera el peor error durante las 30 experimentaciones se muestra en la Figura 10(b), la cual utiliza las funciones sigmide, lineal, sigmoide e hipertangencial como funciones de trasnferencia para cada neurona.

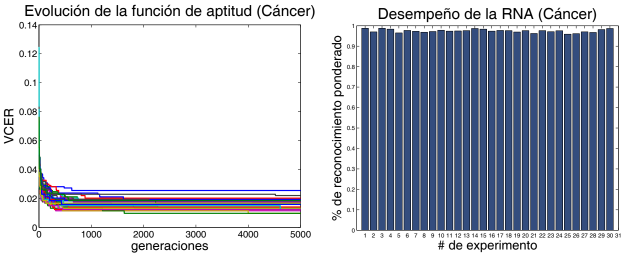
5) Cáncer de mama: Para el caso del problema de cáncer de mama, la evolución del error generado mediante la función de aptitud F_2 es presentado en la Figura 11(a), donde se puede observar que el error se mantiene casi constante para las 30 experimentaciones durante el tiempo límite especificado en 5000 generaciones.



(a) (b)

Fig. 10. Diseño de arquitecturas de RNA generadas para el problema de la Planta de Iris en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.

Por otro lado, el porcentaje de reconocimento ponderado para las etapas de entrenamiento y generalización se presentan en la Figura 11(b). El porcentaje se mantiene exitosamente, para todas las experimentaciones, por arriba del 95%, lo que indica que se encontraron los mejores diseños de las redes para resolver este problema de clasificación.



(a) (b)

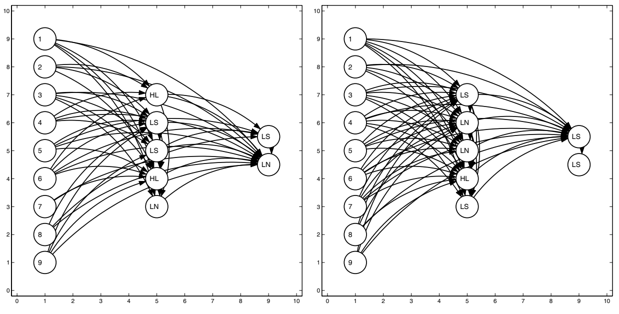
Fig. 11. Resultados experimentales en los 30 experimentos para el problema de Cáncer de mama. (a) Evolución del error. (b) Porcentaje de reconocimiento ponderado.

El mejor y el peor error de entrenamiento fue alcanzado con las arquitecturas mostradas en la Figura 12, donde también se presentan las funciones de transferencia utilizadas para cada neurona. Para el caso del mejor diseño, las funciones de transferencia seleccionadas por la metodología fueron el límite duro, sigmoide y lineal. Para el caso del peor diseño se obtubo el conjunto de funciones compuesto por la función sigmoide, la lineal y el límite duro.

6) Diabetes: Para el problema de la diabetes, la evolución del error a diferencia de las Figuras anteriores, muestra que el proceso de convergencia tarda más generaciones, ver Figura 13(a).

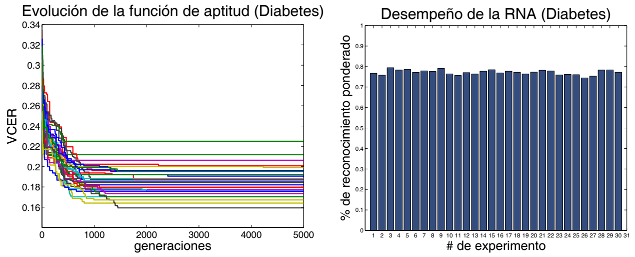
En el caso del reconocimento ponderado, el porcentaje alcanzado para toda la experimentación no fue mayor del 80%, ver Figura 13(b).

La arquitectura que genera el mejor desempeño es la que se muestra en la Figura 14(a), donde hay cuatro neuronas que componenen la capa intermedia con las siguientes funciones de transferencia: sigmoide, gaussiana y lineal. El peor caso se muestra en la Figura 14(b) donde se generó una arquitectura con una sola neurona. Esta arquitectura tiene en la capa de entrada, una neurona que no presenta conexión.



(a) (b)

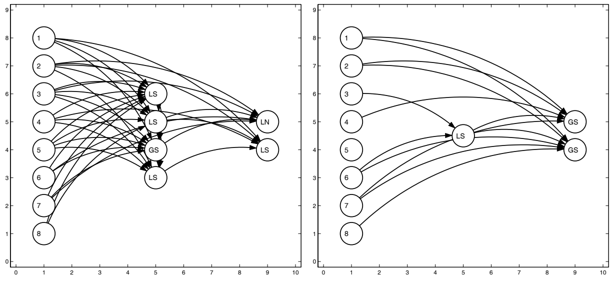
Fig. 12. Diseño de arquitecturas de RNA generadas para el problema de Cáncer de mama en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.



(a) (b)

Fig. 13. Resultados experimentales en los 30 experimentos para el problema de Diabetes. (a) Evolución del error. (b) Porcentaje de reconocimiento ponderado.

Este hecho se puede presentar al diseñar las RNA y no significa que el desempeño se reduzca; al contrario significa que la dimensionalidad del patrón de entrada se puede reducir evitando tener información redundante.

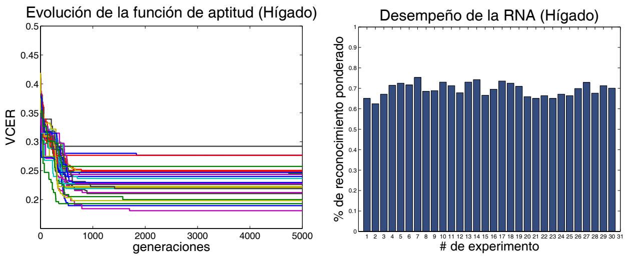


(a) (b)

Fig. 14. Diseño de arquitecturas de RNA generadas para el problema de Diabetes en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.

7) Desórdenes del hígado: La Figura 15 muestra la evolución del error para los 30 experimentos. Antes de las 2000 generaciones el algoritmo de ED encuentra el mejor error para cada experimento, después se mantiene constante.

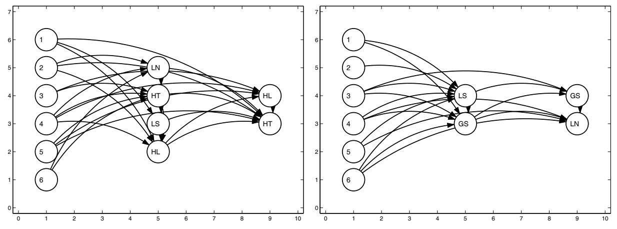
Para el caso del porcentaje de reconocimiento, éste se encuentra arriba del 60% para algunos casos y para otros arriba del 70%.



(a) (b)

Fig. 15. Resultados experimentales en los 30 experimentos para el problema de Desórdenes del hígado. (a) Evolución del error. (b) Porcentaje de reconocimiento ponderado.

La mejor y la peor arquitectura para el problema de desórdenes del hígado se muestran en la Figura 16. El diseño con el mejor desempeño utiliza las funciones de transferencia lineal, hipertangencial y sigmoide. En el caso del peor desempeño, el conjunto de funciones de transferencia está compuesto por: la sigmoide, gausiana y la lineal.

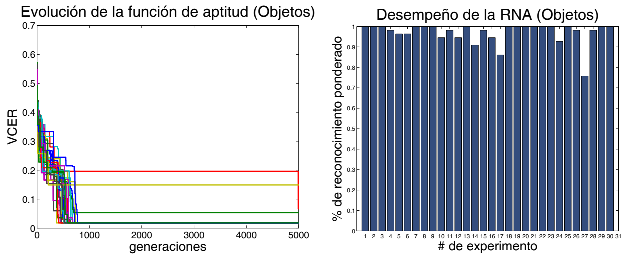


(a) (b)

Fig. 16. Diseño de arquitecturas de RNA generadas para el problema de Desórdenes del hígado en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.

8) Reconocimiento de objetos: La evolución del error para el problema de reconocimiento de objetos se muestra en la Figura 17(a). En ella se observa que la evaluación de la función de aptitud alcanza el error mínimo en menos de 1000 generaciones.

El resultado de esa evolución se ve reflejada en el reconocimiento ponderado para cada experimentación. De las 30 arquitecturas diseñadas, 16 alcanzaron un reconocimiento del 100%, ver Figura 17(b). El desempeño de la mayoría de las arquitecturas restantes se encuentran por arriba del 90%.



(a) (b)

Fig. 17. Resultados experimentales en los 30 experimentos para el problema de Reconocimiento de objetos. (a) Evolución del error. (b) Porcentaje de reconocimiento ponderado.

La Figura 18(a) muestra la arquitectura con el mejor desempeño y en Figura 18(b) aquella con el peor.

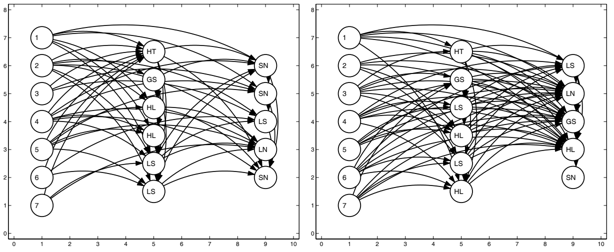
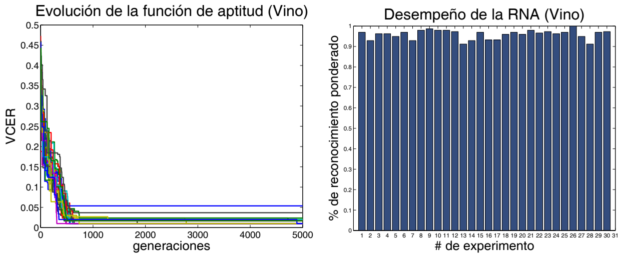

(a) (b)

Fig. 18. Diseño de arquitecturas de RNA generadas para el problema de Reconocimiento de objetos en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.

9) Vino: La Figura 19(a) muestra que la evolución del error tiende a un valor mínimo antes de las 1000 generaciones. En este caso se puede observar que el error de la mayoría de las experimentaciones convergen a valores cercanos.

En el caso del porcentaje de reconocimento ponderado para el problema del vino, el cuál presenta los patrones con mayor número de características, se generó un porcentaje por arriba del 90% y en un experimento se logró generar el diseño con el mejor desempeño, es decir con el 100% de reconocimiento, ver Figura fig19(b).



(a) (b)

Fig. 19. Resultados experimentales en los 30 experimentos para el problema del Vino. (a) Evolución del error. (b) Porcentaje de reconocimiento ponderado.

Dos diseños generados por la metodología propuesta se presentan en la Figura 20, estos diseños son aquellos que generaron el mejor y el peor desempeño para el problema del vino.

10) Vidrio: El problema del vidrio es aquél problema de clasificación que presenta el mayor número de características por cada patrón en la salida. La evolución del error generado continúa descendiendo durante las primeras 3500 generaciones, ver Figura 21(a).

En el caso del porcentaje de reconocimiento para el problema del vidrio, se tiene que la mayoría de la experimentación se encuentra por arriba del 60% con algunos ejemplos por debajo del mismo, ver Figura 21(b).

La Figura 22 muestra las arquitecturas con el mejor y el peor desempeño durante la experimentación.

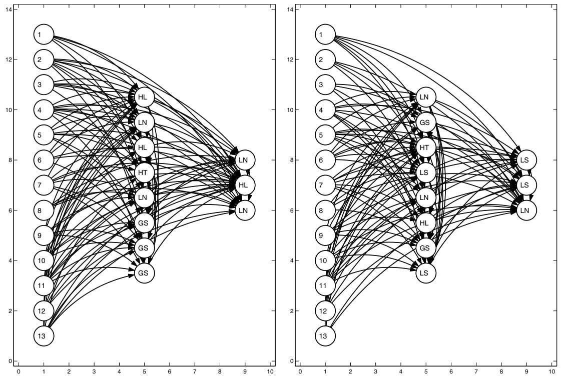
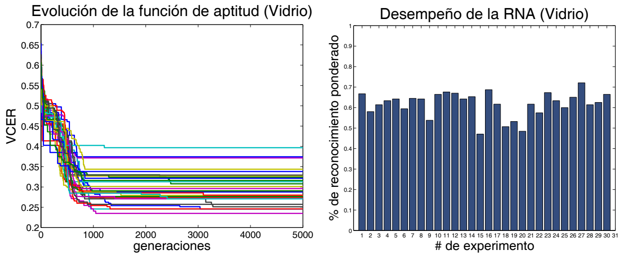

(a) (b)

Fig. 20. Diseño de arquitecturas de RNA generadas para el problema del Vino en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.



(a) (b)

Fig. 21. Resultados experimentales en los 30 experimentos para el problema del Vidrio. (a) Evolución del error. (b) Porcentaje de reconocimiento ponderado.

D. Discusión general

A continuación se muestran los desempeños promedio de los 30 experimentos para cada uno de los problemas de clasificación.

La Figura 23(a) muestra el error promedio para cada problema de clasificación. En dicha Figura se muestra que la función de aptitud con la que se realizaron las experiementaciones presentó en general un error bajo. Sin

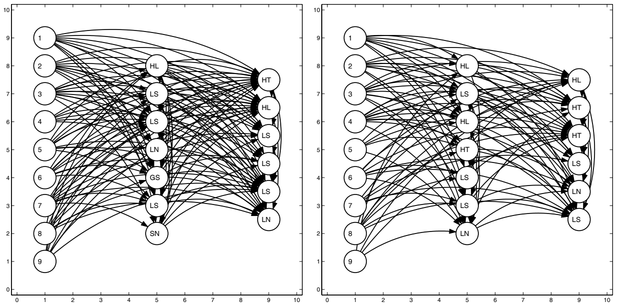
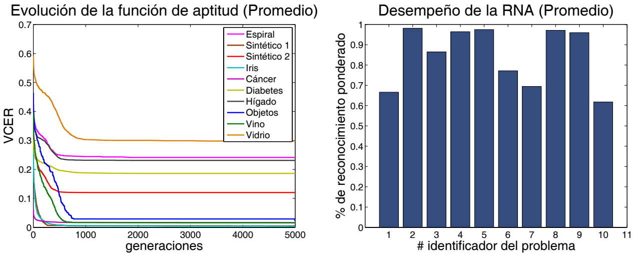

(a) (b)

Fig. 22. Diseño de arquitecturas de RNA generadas para el problema del Vidrio en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.

embargo, las bases de datos que presentan un mínimo en el error durante las 5000 generaciones fueron: reconocimiento de objetos, vino, planta de Iris, cáncer de mama y el problema sintético 1.

En la Figura 23(b) muestra el porcentaje de reconocimiento ponderado promedio. Las bases de datos alcanzaron el siguiente porcentaje: espiral de 66.62%, sintética 1 de 98.12%, sintética 2 de 86.49%, planta de Iris 96.41%, cáncer de mama un 97.47%, diabetes 77.17%, desórdenes del hígado un 69.45%, reconocimieno de objetos 97.09%, para el caso del Vino 95.95% y por último para el caso del problema del vidrio, éste alcanzó un reconocimiento del 61.79%.



(a) (b)

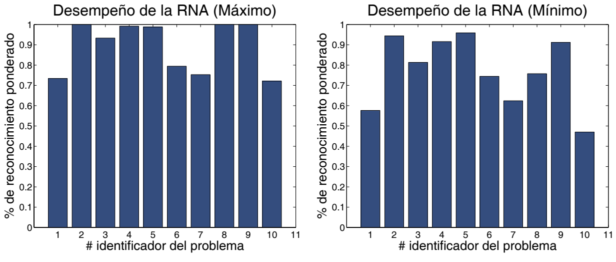
Fig. 23. Diseño de arquitecturas de RNA generadas para el problema del Vidrio en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.

La Figura 24 que a continuación se muestra, presenta los porcentajes de reconocimiento ponderado máximo y mínimo, con el fin de sintetizar cuál fue el mejor desempeño alcanzado para cada problema de clasificación al utilizar la metodología propuesta.

En la Figura 24(a) se muestra que para el caso del problema Sintético 1, del reconocimieto de objetos y el vino, se alcanzó el 100% de reconocimento. Para el caso de los problemas restantes de (izquierda a derecha) el máximo porcentaje obtenido fue: Espiral 73.41%, problema sintético 2 con 93.33%, para la planta de Iris de 99.20%, cáncer de mama %98.77, para el problema de diabetes se alcanzó un 79.45%, para el problema de desórdenes del hígado el máximo porcentaje de reconocimiento fue 75.30% y para el problema del vidrio de 72.11%.

Al contrario, la Figura 24(b) muestra el desempeño promedio de las RNA en términos del porcentaje de reconocimiento ponderado mínimo para cada problema. Para el caso de espiral, el mínimo porcentaje fue 57.66%, para sintético 1 fue de 94.40%, para problema sintético 2 de 81.33%, para la planta de Iris de 91.60%, cáncer de mama 95.87%, para el problema de diabetes se alcanzó un 74.45%, para el problema de desórdenes del hígado el mínimo porcentaje de reconocimiento fue 62.43%, para el problema de reconocimiento de objetos de 75.76%, para el vino de 91.19% y por último para el vidrio fue de 47.04%.

Por otro lado, el número de veces que fueron seleccionadas las diferentes funciones de transferencia para cada problema está descrita en la Tabla III. En ella, se puede apreciar que para el caso del problema de la espiral la función de transferencia que fue seleccionada con mayor frecuencia es la función sigmoide, para el caso del problema sintético 1 fue también la función sigmoide, para el caso del problema sintético 2 fue la función seno, para el problema de la planta de Iris fue la función sigmoide, para el problema del cáncer de mama la función con una mayor selección fue la sigmoide, para el problema de diabetes también la función sigmoide fue seleccionada con mayor frecuencia, para el problema de desórdenes del hígado las funciones sigmoide y lineal tuvieron el mismo número de frecuencia con la que fueron seleccionadas; el problema de reconocimiento de objetos, el vino y el vidrio utilizaron también con mayor frecuencia la función sigmoide.



(a) (b)

Fig. 24. Diseño de arquitecturas de RNA generadas para el problema del Vidrio en los 30 experimentos. (a) Arquitectura con el mejor desempeño. (b) Arquitectura con el peor desempeño.

TABLA III
NÚMERO DE VECES EN QUE CADA FUNCIÓN DE TRANSFERENCIA FUE SELECCIONADA PARA CADA PROBLEMA DE CLASIFICACIÓN. FUNCIONES: LS: SIGMOIDE, HT: HIPERTANG, SN: SENO, GS: GAUSIANA, LN: LINEAL, HL: LÍM. DURO.

Problemas de clasificación	LS	HT	SN	GS	LN	HL
Espiral	14	11	41	8	35	5
Sintética 1	39	3	31	7	11	23
Sintética 2	37	7	40	13	9	6
Planta de Iris	78	12	5	18	39	24
Cáncer de mama	99	19	11	10	26	40
Diabetes	75	19	13	23	32	24
Desórdenes del hígado	37	27	17	23	37	23
Rec. Objetos	75	49	56	60	52	36
Vino	141	44	4	28	54	50
Vidrio	133	50	26	43	71	65
Total	728	241	244	233	366	296

Finalmente, se realizó una comparación con resultados generados por el método del gradiente descendiente (algoritmo de retropropagación) y el algoritmo de Levenberg-Marquardt; algoritmos clásicos de entrenamiento para las RNA. El porcentaje promedio ponderado de cada algoritmo es mostrado en la Tabla IV. Se realizaron dos configuraciones diferentes para las arquitecturas de las RNA entrenadas con cada algoritmo clásico. Estas arquitecturas consiste en generar una capa oculta y otra con dos capas ocultas.

El número máximo de neuronas totales MNN de las RNA entrenadas con los algoritmos clásicos, se generaron mediante la misma ecuación en nuestra metodología propuesta para el caso de una capa oculta, ver Ec. 7 pero, para RNA con dos capas, la distribución se hace a traves de la Ec. 11.

$$DN = 0.6 \times (MNN) + 0.4 \times (MNN) \qquad (11)$$

en donde la primera capa tiene el 60% de las neuronas ocultas y el 40% de las neuronas de la capa oculta está en la segunda capa, es decir una arquitectura piramidal.

Los parámetros para el algoritmo del gradiente descendiente y Levenberg-Marquardt tuvieron dos criterios de paro: al alcanzar las 5000 generaciones o alcanzar un error de 0.000001. Los problemas de clasificación fueron divididos en tres partes: el 40% de los patrones totales fue utilizado para el entrenamiento, el 50% fue utilizado para la generalización y el 10% fue utilizado para la validación. Se utilizó un tasa de aprendizaje del 0.1.

Estas redes neuronales, generadas para la aplicación de los algoritmos clásicos, fueron diseñadas con el fin de obtener los mejores desempeños y así poder compararlas contra las RNA generadas por la metodología propuesta.

En la Tabla IV, podemos observar que el porcentaje promedio ponderado generado por la metodología propuesta es mejor en seis problemas de clasificación: en la espiral, sintético 1, sintético 2, cáncer de mama, diabetes y desórdenes del hígado. En el caso de la planta de Iris y del problema de reconocimiento de objetos, el mejor promedio ponderado se alcanzó con el algoritmo de Lebenberg-Marquardt de una capa. Para los problemas del vino y del vidrio, el algoritmo de Levenberg-Marquardt de dos capas obtubo el mejor porcentaje de reconocimiento ponderado.

A pesar de que el algoritmo clásico Levenberg-Marquardt, obtuvo en algunos casos mejores resultados que la metodología propuesta, el promedio general de todos los problemas de clasificación fue mayor con la metodología propuesta.

VII. Conclusiones

Aunque la metodología propuesta ya fue presentada en [25] y [26], los autores no habían contemplado el incluir el conjunto de validación durante la etapa de entrenamiento, ni se había realizado un estudio de sensibilidad previa a la experimentación, aunado a ésto, se agragaron más problemas de clasificación.

En esta investigación, se presentó la metodología que permite diseñar de manera automática una red neuronal. Este diseño incluye, la arquitectura (cómo se conectan las neuronas y cuántas neuronas son suficientes), el valor del conjunto de pesos sinápticos y el tipo de funciones de transferencia dado un conjunto. El algoritmo evolutivo que se aplicó fue el llamado Evolución diferencial.

En una primera etapa de la experimentación, se realizó un estudio de sensibilidad de los parámetros de dicho algoritmo evolutivo. En esta experimentación se probaron diferentes valores de los parámetros y se seleccionó la mejor configuración. Esta configuración (que generó el mejor desempeño en las RNA generadas para cada problema) se dió al utilizar un número de individuos de 100, en un espacio de búsqueda entre $[-4, 4]$.

De dos funciones de aptitud seleccionadas para el anális de sensibilidad (las funciones de MSE y CER), las cuales incluyeron la etapa de validación, se encontró que, la mejor fue $F_2 = 0.4 \times (CER_T) + 0.6 \times (CER_V)$. Para el caso del factor de cruza CR el mejor valor fue 0.9 y para la constante F un valor de 0.5.

Como es bien sabido, la etapa de validación juega un papel indispensable en la etapa de entrenamiento de una red neuronal artificial, ya que impide el sobreaprendizaje. Por ese motivo, se decidió implementar la etapa de validación en la función de aptitud, con el fin de encontrar una solución con error mínimo de clasificación y al mismo tiempo que no generara un sobreaprendizaje.

Para validar estadísticamente los resultados, dicha configuración se aplicó a 30 corridas del algoritmo en cada problema de clasificación. Se encontró que el desempeño de las RNA diseñadas por la metodología bajo las condiciones dadas en el párrafo anterior, presentan un porcentaje de reconocimiento alto: en el 50% de los problemas el reconocimeinto es mayor al 95% siendo el más bajo de 61.79%.

En el caso específico para cada problema, durante las 30 experimentaciones se alcanzaron porcentajes de reconocimiento del 100% tanto en la etapa de entrenameitno como en la generalización.

Con esto, podemos decir que la etapa de validación y la mejor configuración del algoritmo de Evolución diferenial generaron resultados exitosos. Recordemos que los tres conjuntos, a saber, los conjuntos de entrenamiento, validación y generalización en los que se dividió cada problema se eligieron de manera aleatoria para cada experimento, lo que hace aún más valiosos los resultados obtenidos, pues esto indica que los resultados se validan estadísticamente y experimental.

La metodología propuesta presenta un desempeño general (en todos los problemas de clasificación) mejor que el generado con los algoritmos clásicos de entrenamiento. A pesar que hubo algunos casos donde el mejor promedio de reconocimiento ponderado se alcanzó con Levenberg-Marquardt, la metodología propuesta presenta varias ventajas: la primera es que el diseño se realiza de manera automática. En segundo lugar, la metodología no depende de un algoritmo basado en el cálculo de derivadas, lo que la hace robusto ante problemas más complejos.

Con esto podemos concluir que es posible generar diseños de redes neuronales artificiales con desempeños hasta con el 100% de reconocimineto en la etapa de entrenamiento y generalización, que se puede encontrar varios diseños que resuelven el mismo problema con diferente configuración y mismos resultados y que algunos diseños presentan

TABLA IV
PROMEDIOS DEL PORCENTAJE DE RECONOCIMIENTO PONDERADO PARA LOS ALGORITMOS CLÁSICOS Y LA METODOLOGÍA PROPUESTA.

Problemas de clasificación	Gradiente Descendiente (1 capa)	Gradiente Descendiente (2 capas)	Levenberg Marquardt (1 capa)	Levenberg Marquardt (2 capas)	Metodología propuesta (ED)
Espiral	0.500824742	0.50137457	0.509209622	0.50137457	**0.666185897**
Sintética 1	0.749911111	0.770444444	0.790088889	0.777288889	**0.9812**
Sintética 2	0.544859259	0.51442963	0.69997037	0.562488889	**0.864888889**
Planta de Iris	0.932266667	0.652266667	**0.979111111**	0.756266667	0.964133333
Cáncer de mama	0.967696547	0.944751762	0.969269149	0.957415353	**0.974685447**
Diabetes	0.757864583	0.727604167	0.765260417	0.760902778	**0.771692708**
Desórdenes del hígado	0.604435184	0.576515437	0.675610073	0.662586369	**0.694492754**
Rec. Objetos	0.744533333	0.694133333	**0.982133333**	0.727466667	0.970909091
Vino	0.982921348	0.933782772	0.968614232	**0.979101124**	0.959548023
Vidrio	0.707040498	0.685358255	0.789034268	**0.798380062**	0.617934272
Total	0.749235327	0.700066104	0.812830146	0.748327137	**0.846567041**

características como la reducción de la dimensionalidad de las características de los patrones de entada.

AGRADECIMIENTOS

H. Sossa agradece a la SIP-IPN y al DAAD, por el apoyo económico bajo el número 20111016 y al DAAD-PROALMEX J000.426/2009. B. A. Garro agradece al CONACYT por la beca otorgada durante sus estudios doctorales. H. Sossa también agradece a la Unión Europea y el CONACYT por el apoyo económico FONCICYT 93829. El contenido de este artículo es responsabilidad exclusiva del CIC-IPN y no puede ser considerado como posición de la Unión Europea.

REFERENCES

[1] S. R. y Cajal, Ed., *Elementos de histología normal y de técnica micrográfica para uso de estudiantes*, 3rd ed. Madrid, España: Imprenta y librería de Nicolas Moya, 1901.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning internal representations by error propagation*. Cambridge, MA, USA: MIT Press, 1988, pp. 673–695. [Online]. Available: http://dl.acm.org/citation.cfm?id=65669.104449

[3] T. Back, D. B. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*, 1st ed. Bristol, UK, UK: IOP Publishing Ltd., 1997.

[4] C. Darwin, *The origin of species /.* New York :P.F. Collier,, 1859, http://www.biodiversitylibrary.org/bibliography/24252. [Online]. Available: http://www.biodiversitylibrary.org/item/65514

[5] A. Weismann, *Die Kontinuität des Keimplasmas als Grundlage einer Theorie der Vererbung /.* Jena, Germany: Gustav Fischer, 1885.

[6] G. Mendel and W. Bateson, *Experiments in plant-hybridisation /.* Cambridge, Mass. :Harvard University Press,, 1925, http://www.biodiversitylibrary.org/bibliography/4532. [Online]. Available: http://www.biodiversitylibrary.org/item/23469

[7] J. Ilonen, J.-K. Kamarainen, and J. Lampinen, "Differential evolution training algorithm for feed-forward neural networks," *Neural Process. Lett.*, vol. 17, no. 1, pp. 93–105, Mar. 2003. [Online]. Available: http://dx.doi.org/10.1023/A:1022995128597

[8] N. Guiying and Z. Yongquan, "A modified differential evolution algorithm for opti-mization neural network," in *Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering*, ser. Advances in Intelligent Systems Research, 2007.

[9] H. M. Abdul-Kader, "Neural networks training based on differential evolution al-gorithm compared with other architectures for weather forecasting34," *International Journal of Computer Science and Network Security*, vol. 9, no. 3, pp. 92–99, march 2009.

[10] B. Garro, H. Sossa, and R. Vazquez, "Evolving neural networks: A comparison between differential evolution and particle swarm optimization," in *Advances in Swarm Intelligence*, ser. Lecture Notes in Computer Science, Y. Tan, Y. Shi, Y. Chai, and G. Wang, Eds. Springer Berlin / Heidelberg, 2011, vol. 6728, pp. 447–454.

[11] X. Yao, "Evolving artificial neural networks," 1999.

[12] B. A. Garro, H. Sossa, and R. A. Vázquez, "Design of artificial neural networks using differential evolution algorithm," in *Proceedings of the 17th international conference on Neural information processing: models and applications - Volume Part II*, ser. ICONIP'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 201–208. [Online]. Available: http://dl.acm.org/citation.cfm?id=1939751.1939779

[13] T. Kohonen, "An introduction to neural computing," *Neural Networks*, vol. 1, no. 1, pp. 3–16, 1988.

[14] F. Rosenblatt, *Principles of Neurodynamics*. Spartan Book, 1962.

[15] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982.

[16] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.

[17] P. Isasi Viñuela and I. M. Galván León, *Redes de neuronas artificiales: Un enfoque práctico*. Madrid, España: Pearson Educación, 2004.

[18] B. Martín del Brío and A. Saenz Molina, *Redes Neuronales y Sistemas Borrosos*. Madrid, España: Alfaomega, 2007.

[19] R. Storn and K. Price, "Differential evolution — a simple and efficient adaptive scheme for global optimization over continuous spaces," International Computer Science Institute, Berkeley, Tech. Rep., 1995. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.9696

[20] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[21] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello, "A comparative study of differential evolution variants for global optimization," in *GECCO*, M. Cattolico, Ed. ACM, 2006, pp. 485–492.

[22] R. Storn and K. Price. (2012, april) Official web site this is a test entry of type @ONLINE. [Online]. Available: http://www.icsi.berkeley.edu/storn/code.html,

[23] D. N. A. Asuncion, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html

[24] R. A. Vázquez and H. Sossa, "A new associative model with dynamical synapses," *Neural Processing Letters*, vol. 28, no. 3, pp. 189–207, 2008.

[25] B. A. G. Licon, J. H. S. Azuela, and R. A. Vázquez, "Design of artificial neural networks using a modified particle swarm optimization algorithm," in *IJCNN*. IEEE, 2009, pp. 938–945.

[26] B. A. Garro, H. Sossa, and R. A. Vázquez, "Artificial neural network synthesis by means of artificial bee colony (abc) algorithm," in *IEEE Congress on Evolutionary Computation*. IEEE, 2011, pp. 331–338.

Map Building of Unknown Environment Using L1-norm, Point-to-Point Metric and Evolutionary Computation

Jaroslav Moravec

Abstract—**In the present paper, a method for building a map of an unknown environment (SLAM) derived from the ICP algorithm using point-to-point metric is proposed. The polar-scan matching technology is used for estimation of the robot location change between two scans in sequence estimate the correct position of the robot. Since map building is fairly time-consuming, the algorithm of differential evolution (DE) is used in the calculation. This efficient optimizer provides very good results in different types of small office environment (unstructured and structured). The new type of an algorithm for map building is based purely on simple geometric primitives—vectors and integrates the modern evolutionary algorithm—DE. The presented algorithm falls into the wider group of geometric map builders and is able to build a map of indoor, mostly office, environment without moving objects.**

Index Terms—**SLAM, robot localization, evolutionary robotics, differential evolution, L1-norm.**

I. Introduction

SIGNIFICANT effort of many different research groups in the area of the map building has brought good results in the last several decades. The integration of modern evolutionary algorithms is not taken for granted that much in this field. Disadvantage of nearly all EA (evolutionary algorithms) methods is a necessity to find proper working parameters. Many EA methods suffer from premature convergence to local optimum, which they´re not able to release from any more. Algorithms for the map building are very sensitive about the failure of the estimator which performs the estimation of position and turning transformation. All these exact reasons lead to elect differential evolution optimizer as an appropriate EA tool, as it provides very good results for a given task.

There are many different approaches in an area of the robot localization and map building which can be classified into several main groups. The amount of publications in particular groups is approximately the same. (A) Probabilistic algorithms usually use different versions of an occupancy grid. A map is represented by set of occupancy probability eventually emptiness probability. The map is formed by a set of cells in the shape of usual square area [23, 9, 3, 13]. (B) Map is represented by geometric primitives such as lines, circles, points or curves (b-spline curves) etc. The geometric model of environment is created from these elements. The core of the position estimator might be compiled by using various types of algebraic criteria [18, 19, 20, 21]. (C) The third and these days probably the widest group is the one using combinations of algorithms from two previous groups (so called hybrid algorithms) or it uses very specific patterns that represented the model of the world and different methods for localization and map building. It´s for example the case of events when a map is represented by a cloud of points, alternatively by different kinds of landmarks [17] like RFID, sound sources etc. Relatively new and perspective way in the field of map building and navigation is called cognitive maps [38, 39]. This approach exploits and integrates more information sources. A precise geometric map similar to [13] does not exist here.

Conventional methods for creating a map of unknown environment such as e.g. these publications [23, 1, 21] use gradient optimizers. But this is an approach a few decades old. The advantages of gradient optimizers are their simplicity and implementation speed. They still interest many researchers thanks to these qualities. They may be found e.g. in [40]. However, they have their insignificant limitations. Due to an intensive research in the field of evolutionary computer technology and fairly huge amount of publications analyzing their possibilities on different types of problems, EA methods have come to the foreground in the area of robotics as well. Their application is broad – map building using 2D or 3D laser scanner, global and local localization, semantic classification, the area of machine learning. A relatively big disadvantage is that they may also extend significantly the implementation of a basic navigation algorithm. MoteCarlo algorithm is the most common optimizer that is possible to come across and is used in the connection with probabilistic algorithms [9, 3, 13].

In 1998 an interesting article [42] based on Island Model Genetic Algorithm (IGA) was published. The theme of distributed GA can be found earlier for example [43, 44]. IGA is a derivative of the genetic algorithms that works with several populations which search functional space in parallel. The authors were successful to prove that IGA provides better results especially for linearly separable problems comparing to SGA [48] that is used e.g. in [26]. Using IGA optimizer as a computational accelerator also depends strongly on a type of the basic method(s). These methods were used for the purposes of localization and map creation (SLAM) in [45], [46]. It is also possible to find a very interesting connection of the SLAM algorithm and the fuzzy logic [47]. One of the first papers using untreated SGA was published in [11]. Interesting

Manuscript received June 20, 2012. Manuscript accepted for publication July 24, 2012.

J. Moravec is with the Czech Technical University in Prague, Prague, Czech Republic (e-mail: j.moravec.email@seznam.cz; webpage and source code: robomap.4fan.cz, www.openslam.org).

results like that may be found in the article [55] as well. Practical use of the SLAM algorithm with the complete analysis of the issue can be found e.g. in [25]. The base of the presented method is formed by the probabilistic occupancy grid [23, 13] again. Kwok et al. presented a small study in [16] which compares the performance of three different evolutionary algorithms SGA [48, 49], PSO [53] and AntSystem [52]. The tested EA methods solve three-dimensional problem – here the dimensions are represented by X, Y coordinates in the Cartesian coordinate system and the third dimension is then the angle α that includes the robot´s view direction along with axis X. The disadvantage of this approach is the necessity of using a fairly huge number of individuals in the population and substantially higher computation demands related to that. Similar results are to be found in paper [24] as well.

The differential evolution algorithm (DE) proposed by Kenneth Price a Rainer Storn [36] is used in this paper as an efficient and powerful computational accelerator (optimizer). However, DE is only suitable for certain types of problems – see [14]. SLAM under low noise levels falls to such suitable groups of problems as well. Finding of the correct working parameters of the optimizer is not so easy and usually takes a fairly long time. Valuable results of a research in this area can be found in [2, 12, 37, 14, 30, 22].

II. Pose Estimation in Partially Known Environment

Consider a general evolutionary SLAM problem

$$min_{x\in\boldsymbol{X}} f\big(g(x)\big) \text{ or } max_{x\in\boldsymbol{X}} f\big(g(x)\big) \tag{1}$$

Denote by f_{OPT+}, f_{OPT-} the optimal values of this problem and by f^* and f_* the maximum and minimum value of f over $\boldsymbol{X}$. $x \in \boldsymbol{X}$ represents optimal trajectory in state space (it is for example: x, y, α is a robot pose; x, y in Cartesian coordinates and α is heading with regard to the axis X, and of course all working parameters of the presented methods have to be included as well). $g(\ldots)$ represents sensing model and a pose estimator (in the case of the SLAM problem, presented here, it is $\mathcal{F}_2$ or $\mathcal{F}_3$ strategy – see below), $f(\ldots)$ represents evolutionary pseudo-random process – i.e. DE algorithm for example.

Definition: Given $\varepsilon \in \langle -\infty, +\infty \rangle$, a functional $\tilde{x} \in \boldsymbol{X}$ is said to be an ε-approximate solution of the problem (1) if possible solution exists in the sense

$$\begin{gathered}|f(\tilde{x}) - f_{OPT-}| \le \varepsilon |f_* - f_{OPT-}| \text{ or} \\ |f(\tilde{x}) - f_{OPT+}| \le \varepsilon |f^* - f_{OPT+}|\end{gathered} \tag{2}$$

Unfortunately, $\tilde{x} \in \boldsymbol{X}$ strongly depends on $g(\ldots)$ and "system" represented by robot, environment and all moving objects and is non-separable and non-stationary (so called t-variant). In this task robot always affects itself through other objects moving in the given environment and that´s thanks to the used control systems. The presented work transforms the general optimization problem utilizing evolutionary computation to:

$$\begin{gathered}\mathcal{F}_{2,3}: f\big(g(x)\big) \rightarrow f_{EA}\big(g(x)\big)\,,\, \boldsymbol{D}(f, f_{EA}) \in \mathbb{R}, \\ f_{EA} \sim \mathit{pseudo\text{-}random\ process}\end{gathered} \tag{3}$$

Evolutionary computations are used to accelerate the map building process. The pose estimation process is based on comparison of a set of simulated data from a virtual 2DLS (two dimensional laser scanner) from positions obtained by using the EA process and scan from the real 2DLS. We can denote: $P_{e(i)} = (x, y, \alpha)$, $P_{s(j)} = (x, y, \alpha)$, $i, j \in \mathbb{N}$, $\exists j | P_{opt} \equiv P_{s(j)} \sim P_{e(i)}$ is a real and simulated pose and from the text above $P_{e(i)}, P_{s(j)} \in \boldsymbol{X}$, $D_{e(i)} = \big[v_{e(1)}, v_{e(2)}, \ldots, v_{e(m)}\big]$, $i \in \mathbb{N}$, $\forall v_e | v_e \le L_{max}$, $1 \le m \le \mathbb{C}_{real}$ and $D_{s(j)} = \big[v_{s(1)}, v_{s(2)}, \ldots, v_{s(n)}\big], j \in \mathbb{N}$, $\forall v_s | v_s \le L_{max}$, $1 \le n \le \mathbb{C}_{sim}$, $m < n$, the real and simulated sensing. $\mathbb{C}_{sim} \ge \mathbb{C}_{real}$. α is heading with regard to axis X. For every $P_{e(i)}$ it is necessary to create set of $P_{s(j)}, j \in \mathbb{N}$. $\varphi_{det} = \pi$ is a detection angle of the real 2DLS, $\varphi_{stp} = 0.5°$ is a resolution of the real 2D LS - i.e. 361 beams can be used for example. γ_{det} is a detection angle of the pose estimator $g(x)$ i.e. γ_{det} is the angle which the matching process works in. γ_{det} is set to $^{\pi}/_{3}$ for all experiments. $P_{opt}|_1^{\boldsymbol{n}} \in \tilde{x}$, L_{max} is a beam limitation given by used sensor with $\mathbb{C}_{real}$ beams. $\mathbb{C}_{sim}$ denotes the number of simulated beams. Environment is represented by set of short lines. EA methods use set of individuals $P_{s(j)}$. From the theoretical point of view, only one point $P_{opt}, P_{opt} \in P_s$ defines the correct robot's pose. In the real world and thanks to the existence of noise (estimated by S_{p3} or S_{p2} function), more than one point can provide well acceptable result(s). Every individual of the EA represents one possible solution which is evaluated in the sense:

$$\boldsymbol{C} = |\boldsymbol{A} - \boldsymbol{B}|, \tag{4}$$

where

$$\boldsymbol{A} = \begin{bmatrix} v_{s'(1)} & v_{s'(2)} & v_{s'(3)} & \cdot & \cdot & \cdot & \cdot & v_{s'(m-1)} & v_{s'(m)} \\ v_{s'(2)} & v_{s'(3)} & \cdot & \cdot & \cdot & \cdot & v_{s'(m-1)} & v_{s'(m)} & v_{s'(m+1)} \\ v_{s'(3)} & \cdot & \cdot & \cdot & \cdot & v_{s'(m-1)} & v_{s'(m)} & v_{s'(m+1)} & v_{s'(m+2)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ v_{s'(n-m+1)} & v_{s'(n-m+2)} & \cdot & \cdot & \cdot & \cdot & \cdot & v_{s'(n-1)} & v_{s'(n)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & v_{s'(n-1)} & v_{s'(n)} & v_{s'(n+1)} \\ v_{s'(n-3)} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ v_{s'(n-2)} & v_{s'(n-1)} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & v_{s'(n+m-3)} \\ v_{s'(n-1)} & v_{s'(n)} & v_{s'(n+1)} & \cdot & \cdot & \cdot & \cdot & v_{s'(n+m-3)} & v_{s'(n+m-2)} \\ v_{s'(n)} & v_{s'(n+1)} & v_{s'(n+2)} & \cdot & \cdot & \cdot & v_{s'(n+m-3)} & v_{s'(n+m-2)} & v_{s'(n+m-1)} \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} v_{e(1)} & v_{e(2)} & \cdot & v_{e(m)} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ v_{e(1)} & v_{e(2)} & \cdot & v_{e(m)} \end{bmatrix}, \boldsymbol{A}, \boldsymbol{B} \text{ are } n \times m.$$

$$\begin{gathered} v_{s'(1..n)} = v_{s(1..n)}\ ,\ v_{s'((n+1)..(n+m-1))} = v_{s(1..(m-1))}\ ; \\ a \equiv v_{s'}\ ,\ b \equiv v_e\ ; n, i \text{ is } row\ ,\ m, j \text{ is } column\ ; \\ \boldsymbol{u}^{\boldsymbol{T}} = \big[\textstyle\sum_{j=1}^{m} c_{1,j}\,, \sum_{j=1}^{m} c_{2,j}, \ldots, \sum_{j=1}^{m} c_{n-1,j}, \sum_{j=1}^{m} c_{n,j}\big] \end{gathered}$$

Matrices $\boldsymbol{A}, \boldsymbol{B}$ are expressed generally in (4) i.e. $\gamma_{det} = 2\pi$. If γ_{det} is smaller than 2π, number of rows of the used matrices will be adequately smaller too. Presented map building algorithm is based on two independent strategies. Both strategies $\mathcal{F}_2$ and $\mathcal{F}_3$ enable correct $\tilde{x} \in \boldsymbol{X}$ estimation. $\mathcal{F}_2$ and $\mathcal{F}_3$ are given by equations:

$$\mathcal{F}_2|_{P_{e(i=1)}}^{i=t} \equiv \left\{S_{r2} = argmax_{P_s(x,y)} max_{u_{(i)}} |_{i=1}^{n} \left[\sum_{j=1}^{m} \begin{cases} 1, \; if \left(\left|a_{(i,j)} - b_{(i,j)}\right| \leq S_{p2}\right) \\ 0, \; if \left(\left|a_{(i,j)} - b_{(i,j)}\right| > S_{p2}\right) \end{cases}\right]\right\} \quad (5)$$

$$\forall x, y | x, y \in \mathbb{R}\,, S_{r2} \leftrightarrow P_{opt}; S_{p2} = \frac{1}{n} + \varepsilon_t(b_{i,j} + a_{i,j}) + 1, \; \varepsilon_t = \frac{59}{6000}$$

$$\mathcal{F}_3|_{P_{e(i=1)}}^{i=t} \equiv \left\{S_{r3} = argmin_{P_s(x,y)} min_{u_{(i)}} |_{i=1}^{n} \left(\sum_{j=1}^{m} \begin{cases} \left|a_{(i,j)} - b_{(i,j)}\right|, if \; a_{(i,j)} < L_{max} \\ 0, \quad if \; a_{(i,j)} = L_{max} \end{cases}\right)\right\} \quad (6)$$

$$\forall x, y | x, y \in \mathbb{R}\,, S_{r3} \leftrightarrow P_{opt}\,, u_{(i)}|_{i=1}^{n} \in \{0,1\}$$

$$u_{(i)}|_{i=1}^{n} = \textstyle\sum_{j=1}^{m} \left|a_{(i,j)} - b_{(i,j)}\right| ; S_{p3} = \left[\frac{1}{n} + (\varepsilon_t a_{i,j}) + 1\right]$$

where ε_t is a slope of the accuracy curve of the used 2DLS (for example $\frac{59}{6000}$) – it is classic linear dependency according to the manufacturer recommendations. t is the number of all collected scans from the real 2DLS.

Here, S_{r2} and S_{r3} represents the fitness value of the best founded estimated pose. S_{p3} and S_{p2} represent equations of the linearized model of the 2DLS sensor—simple noise model for one(every) beam. Correct pose and heading estimation according to the selected strategy is given by P_{opt} and heading α of the robot is given by:

$$\boldsymbol{\mathcal{F}_3}\!: \; min \, u_k|_{k=1}^{n} \Rightarrow (index)k \,, \; \boldsymbol{\mathcal{F}_2}\!: \; max \, u_k|_{k=1}^{n} \Rightarrow (index)k \,; \qquad (6)$$

$$\boldsymbol{\mathcal{F}_{2,3}}\!: \alpha = \left(\varphi_{stp}(k-1)\right) + \left(\tfrac{1}{2}\gamma_{det}\right)$$

'$index$' means k-th element of the $u_k$vector, for which the fitness function takes the smallest ($\mathcal{F}_3$) or the biggest ($\mathcal{F}_2$) value.

Sensorial data from 2DLS are used only (no data from odometry). The pose estimator described in here is based on point-to-point metric. It is the core of the proposed SLAM method. $\mathcal{F}_2$ or $\mathcal{F}_3$ strategy is used to dissimilarity measurement—dissimilarity between simulated vector D_s and the real sensing D_e. $\mathcal{F}_3$ strategy has universal features and is suitable for structured or unstructured environment, long or small corridors (hallways) or environment with or without moving objects.

Generally, $\mathcal{F}_3$ provides somewhat worse results at heading estimation—of about 5–7 percent in comparison to $\mathcal{F}_2$. $\mathcal{F}_2$ has identical features to $\mathcal{F}_3$, but it is not suitable for work in long hallways and has quarter noise resistance abilities only – see [54]. $\mathcal{F}_2$ is suitable for small and very structured environment with or without moving objects.

At correct pose estimation and if searching area is 60x60cm for example, equation (4) must be evaluated 60 × 60=3600 times. It takes a long time. DE optimizer is able to estimate correct solution approx. 25–35x faster. Fig. 1 depicts Fitness function projection of the $\mathcal{F}_3$ and $\mathcal{F}_2$ strategy to 3D space—identical environment and identical robot's pose is used.

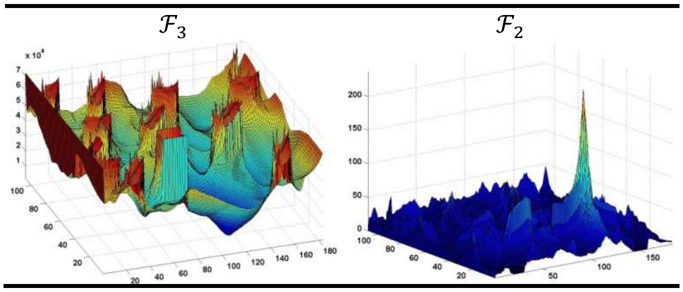


Fig. 1. Projection of the Fitness function of the $\mathcal{F}_3$ and $\mathcal{F}_2$ strategy to 3D space - example. $\mathcal{F}_3$ - robot is in the place with min. Fitness function. $\mathcal{F}_2$ - robot is in the place with max. Fitness function.

TABLE I
PROPOSED GSLAM ALGORITHM

Input data: $P_{actual}(x,y,\alpha)$ – Start (alias actual) robot's pose – corresponding to the $D_{e(1)}$, $D_{e(i)}$ – Set of sensorial data from 2DLS, $\boldsymbol{GM}$ - empty global map.

```
A1  1  Approximate scan $D_{e(1)}$ by set of lines $U_k, k \in \mathbb{N}, k \neq 0$
    2  Compute parameters of the $U_k$, necessary for SLAM.
A2  3  Insert the $U_k, k \in \mathbb{N}, k \neq 0$ into $GM$
    4  for all $D_{e(i)}$; $i \in \mathbb{N}, i \in \{2,..,p\}$
A3  5      Create $TM$ based on the $P_{actual}(x,y,\alpha)$
           _and $GM$.
A4  6      Find correct pose and heading $P_{opt}$ for $D_{e(i)}$,
           _use the $TM$ and selected strategy $\mathcal{F}_3$ or
           _$\mathcal{F}_2$ accelerated by DE.
A1  7      Approx. actual $D_{e(i)}$ vector, by set of
           _lines $U_k, k \in \mathbb{N}, k \neq 0$.
    8      Compute params. of the $U_k$ and insert
           _$U_k$ into the $LM$.
A2  9      Use the $LM$, build a new set of lines $U_{new}$,
           _suitabe to be inserted into $GM$.
           _Clear $LM$. Insert the new set of
           _lines $U_{new}$ into $LM$.
   10      Insert all lines from $LM$ to $GM$ and
           _clear $LM$.
A5 11      Use heuristic rules, merge all possible
           _lines in $GM$, if it is possiblle.
   12  end for
A5 13  Final map 'refinement' – $GM$ map
```

Output data: $\boldsymbol{GM}$ – Lines list – global map of environment, $P_k(x,y,\alpha)$ – Set of positions. Robot's pose and heading corresponding with $D_{e(i)}$ vectors.

Key: $\boldsymbol{TM}$ – Temporary Map of local environment. This map is used at pose estimation utilizing EA computations. $\boldsymbol{LM}$ – Local Map contains all lines found in $D_{e(i)}$. A1.. A5 individual parts of the gSLAM algorithm. A1 consists of a recursive line splitting marked as B1 and line pose improvements by LSQ algorithm marked as B2. A3 use classic ray-tracing. In step A4, EA method is used according to the equation (4). Line 13 can be optionally omitted.

DE optimizer in this case seeks for an optimum (minimum or maximum) according to the u_k (alias $\mathcal{F}_3$: $min \, u_k|_{k=1}^{n}$, $\boldsymbol{\mathcal{F}_2}$: $max \, u_k|_{k=1}^{n}$, see (3) – (6)) value for every individuum of the population. Fig. 1 shows large area 600 × 800cm, for better understanding.

III. GSLAM ALGORITHM

Proposed gSLAM algorithm uses raw 2DLS data to estimate the correct pose in polygonal environment by modified simulated-point-to-point matching technique in temporary map $\boldsymbol{TM}$ which represents the temporary polygonal model-view from the last estimated correct pose P_{opt} in $\boldsymbol{GM}$ – global map. $\boldsymbol{TM}$ is obtained by using 2D ray-tracing. The complete gSLAM algorithm is described in Table I. Once the correct robot's pose is found according to the $\boldsymbol{GM}$, the actual sensorial data $D_{e(i)}$ are approximated by set of lines and the local model $\boldsymbol{LM}$ is created from them. The set of lines is confronted with the global $\boldsymbol{GM}$ map of environment (previously built) and some parts of lines in $\boldsymbol{LM}$ are marked as suitable for inserting into $\boldsymbol{GM}$. This process uses several heuristic rules.

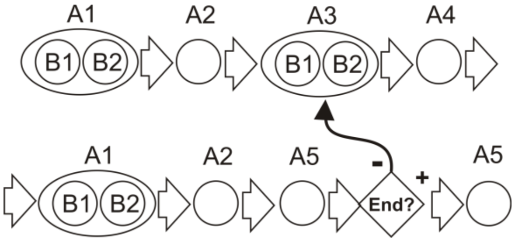
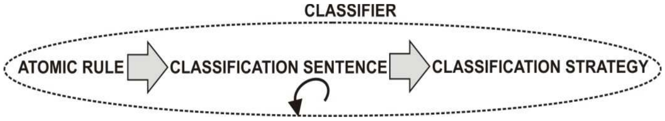


Fig. 2. Flowchart diagram of the gSLAM – sequential ordering of the A1 .. A5 algorithms.

The flowchart diagram of insertion process is depicted in Fig. 3. The recursive line splitting algorithm (LR-LSQ) based on L1-norm is used. 'LocalMap($\boldsymbol{LM}$)' serves like a helper only. Table I contains individual steps of the proposed gSLAM algorithm. gSLAM consists of 5 main parts. Input data are $P_{actual}(x, y, \alpha)$ – start (alias actual) robot's pose – corresponding to the $D_{e(1)}$, $D_{e(i)}$ – set of sensorial data 2DLS. Output data are $\boldsymbol{GM}$ – global map of environment, $P_k(x, y, \alpha)$ – set of estimated positions resp. robot's pose and heading corresponding with $D_{e(i)}$ vectors.



* A small circular arrow means that a classification sentence can be repeated according to the actual state of the $\boldsymbol{LM}$ map, until all abscissae are successfully classified. Deadlock is handled.

Fig. 3. The flowchart diagram of the lines relationship classifier.

A. The Line Fitting Method

The line fitting algorithm (marked as A1 in Fig. 2) uses approximation of a point set by multi-line. The presented line fitting algorithm uses combination of several methods - successive Edge Following – SEF [31] and Iterative End Point Fit (IEPF) [10, 6, 27, 28, 29]:

1) Transform D_e vector from polar to Cartesian coordinates.
2) Eliminate all points b_i where: $\forall b_i(x, y)$, $\nexists b_j(x, y)$ so that $b_i(x, y) \in \{b_j(x, y): \|b_i, b_j\| \leq \delta, j \neq i\}$ in E^2, δ is constant; 10cm for example.
3) Use SEF algorithm to b_i points and exclude all unsuitable points from b_i. New set of points will be b_i^*. Use IEPF algorithm to b_i^*.
4) Use linear regression (LR-LSQ) [32], [15] algorithm to merge two consecutive lines if it is possible + final refinement of every line in the sense of (LR-LSQ) algorithm.

Abscissae of the $\boldsymbol{LM}$ map cannot be inserted directly. Several atomic rules were defined enabling to estimate correctly, which part of the line is to be inserted into the $\boldsymbol{GM}$ – see Table II. These rules are designed to be useful for line-to-line algorithms. There are 17 atomic rules – see Fig. 4. Suitable combinations of atomic rules form four classification sentences. The classification sentences form a classification strategy. The number of applied classification sentences is not known a priori and is variable. The proposed classification sentences (insertion process of the $\boldsymbol{LM}$ into $\boldsymbol{GM}$):

1) If two abscissae AB and CD do not lie on an identical line (consecutively), the distance of C and D points from the line identical to abscissa AB is shorter than the limit, CD abscissa does not overhang the boundary C and D points of AB abscissa and the angle between AB and CD is smaller than the limit, remove CD abscissa. (Atomic rules 1,11, 12, 5, 4)
2) If two abscissae AB a CD do not lie on an identical line (consecutively), C point and D point overhang the end points of AB abscissa, the distance of A and B points to CD abscissa is shorter than the limit and the angle between AB and CD is shorter than the limit, let in $\boldsymbol{LM}$ parts of CD abscissa, which overhang AB abscissa. (Atomic rules 1, 3, 4, 7, 8, 9, 10, 13, 14, 16).
3) If two abscissae AB a CD do not lie on an identical line (consecutively), C point overhangs AB abscissa and the distance of the second point to AB abscissa is less than the limit, let in $\boldsymbol{LM}$ part of CD abscissa, which overhangs AB abscissa only. (Atomic rules 1, 2, 3, 7, 12, 13, 14, 15).
4) If two abscissae AB a CD do not lie on an identical line (consecutively), D point overhangs AB abscissa and the distance of the second point to AB abscissa is less than the limit, let in $\boldsymbol{LM}$ part of CD abscissa which overhangs AB abscissa only. (Atomic rules 1, 2, 4, 6, 12, 13, 16, 17).

The atomic rules are formularized by AND logic operator and rules 16 and 17 are placed inside of the 4. condition clause.

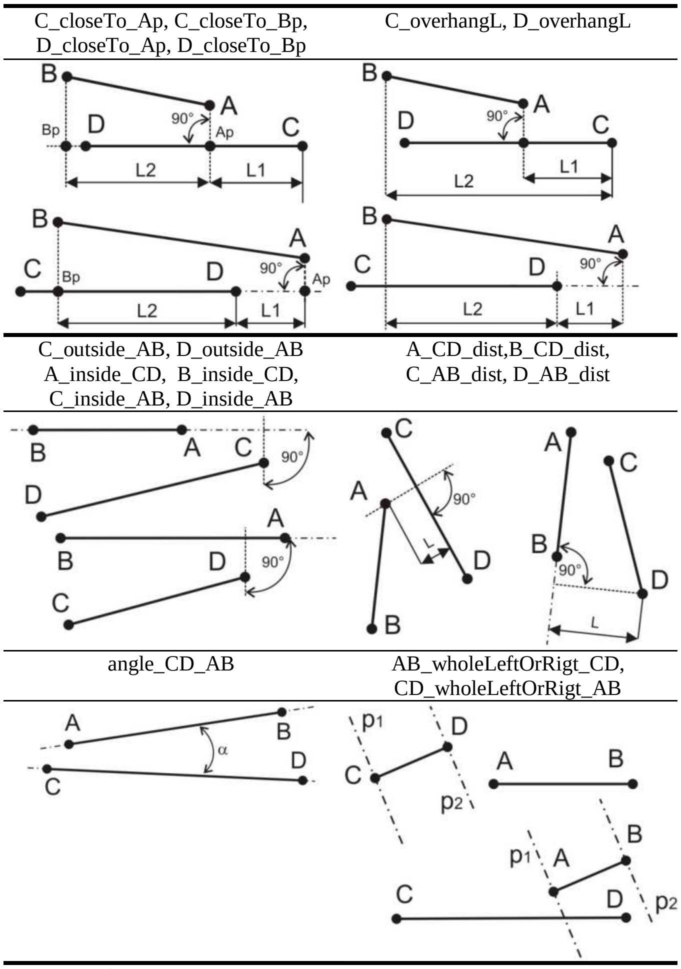
Four classification sentences represent a heuristic schema that only enables to define such parts of the $\boldsymbol{LM}$ map which are suitable for inserting into the $\boldsymbol{GM}$ map. If any "less suitable" abscissae appear in $\boldsymbol{LM}$, classifier inserts such abscissae into $\boldsymbol{GM}$ without any change. Usually perpendicular abscissae are considered – perpendicular to existing walls of the environment model.

TABLE II
ATOMIC RULE LIST – INSERTION PROCESS

Nr.	Atomic rule	Description
1	`AB_wholeLeftOrRigt_CD`	Abscissa AB lies on the left or right side of CD abscissa.
2	`CD_wholeLeftOrRigt_AB`	Abscissa CD lies on the left or right side of AB abscissa.
3	`C_overhangL <= L1`	Point C of CD abscissa does not overhang any of A or B points
4	`D_overhangL <= L1`	Point D of CD abscissa does not overhang any of A or B points
5	`C_overhangL > L1`	Point C of CD abscissa overhangs one of the outer points of AB abscissa AB.
6	`D_overhangL > L1`	Point D of CD abscissa overhangs one of the outer points of AB abscissa.
7	`C_outside_AB = true`	Intersect Cp of the line passing the point C perpendicular to AB abscissa does not lie between AB points.
8	`D_outside_AB = true`	Intersect Dp of the line passing the point D perpendicular to AB abscissa does not lie between AB points.
9	`A_CD_dist < L2`	Distance of the point A from the line which the CD abscissa lies on.
10	`B_CD_dist < L2`	Distance of the point B from the line which the CD abscissa lies on.
11	`C_AB_dist < L2`	Distance of the point C from the line which AB abscissa lies on.
12	`D_AB_dist < L2`	Distance of the point D from the line which the AB abscissa lies on.
13	`angle_CD_AB < L3`	Angle of AB and CD lines
14	`C_closeTo_Ap = true`	Distance of the C point and intersect Ap line passing the point A perpendicular to line on which the points C,D lie is shorter than distance of the point C and intersect Bp line passing the point B perpendicular to CD.
15	`C_closeTo_Bp = true`	Distance of the point C and intersect Bp line passing the point B perpendicular to line on which the points C,D lie is shorter than distance of the point C and intersect Ap line passing the point A perpendicular to CD.
16	`D_closeTo_Ap = true`	Distance of the D point and intersect Ap line passing the point A perpendicular to line on which the points C,D lie is shorter than distance of the point D and intersect Bp line passing the point B perpendicular to CD.
17	`D_closeTo_Bp = true`	Distance of the D point and intersect Bp line passing the point B perpendicular to line on which the points C,D lie is shorter than distance of the point D and intersect Ap line passing the point A perpendicular to CD.

If the rule has an operator (<,>,= et.c) it is only used in this form. If no operator is present, any type of operator can be used in algorithm in the sense of the particular rule. For example ‘`C_overhangL`’ is real a number at computations and ‘`L1`’ is constant.

If any abscissa is transformed using any classification sentence, only suitable parts of it are moved back to $\boldsymbol{LM}$ map. Once classifier finishes its job, all abscissae are moved to $\boldsymbol{GM}$ at once. Fig. 4 shows a graphical representation of the atomic rule list in Table II. used at an insertion process. It is a classic conceptional relation between two abscissae AB and CD.


C_closeTo_Ap, C_closeTo_Bp, D_closeTo_Ap, D_closeTo_Bp
C_overhangL, D_overhangL
B
A
D
C
Bp
Ap
90°
L2
L1
B
A
C
D
Bp
Ap
90°
L2
L1
C_outside_AB, D_outside_AB A_inside_CD, B_inside_CD, C_inside_AB, D_inside_AB
A_CD_dist,B_CD_dist, C_AB_dist, D_AB_dist
B
A
C
D
90°
L
angle_CD_AB
AB_wholeLeftOrRigt_CD, CD_wholeLeftOrRigt_AB
A
B
C
D
α
p1
p2
90°

Examples of the presented rules:
- C_closeTo_Ap = true, C_closeTo_Bp = false,
- D_closeTo_Ap = true, D_closeTo_Bp = false
- C_overhangL = L1, D_overhangL = -1*L1
- C_outside_AB = true, D_outside_AB = false,
- C_inside_AB = false, D_inside_AB = true
- A_CD_dist = L, D_AB_dist = L
- angle_CD_AB = α
- AB_wholeLeftOrRigt_CD = true, CD_wholeLeftOrRigt_AB = false

*L, L1, L2, α are elected constants at SLAM process.

Fig. 4. Atomic rule list, conceptional relation between two abscissae AB and CD; graphic representation.

B. Merging – minimizing the number of abscissae in $\boldsymbol{GM}$

Once the list of abscissae suitable for inserting into $\boldsymbol{GM}$ is completed, it´s inserted into $\boldsymbol{GM}$ immediately. Merging process ensures minimum and acceptable number of abscissae in $\boldsymbol{GM}$. Merging process is not a necessary step in order the final map to be fully consistent. The method presented in here is based on Skrzypczyňski [33], [34] and Crowley [6], [7], [8], but the method is significantly modified. Similarly to the insertion process, the merging process uses the same basic scheme as the insertion process – see Fig. 3. The proposed merging process consists of 3 classification sentences:

1) Abscissae AB and CD lie consecutivey in a line and the distance of end-points is shorter than the limit. AB and CD will be concatenaded. (Atomic rules 13, 11, 12, 1, 18, 19, 20, 21, 22, 23).

2) If abscissae AB and CD intersect and the distance of any point A, point B, point C or point D from intersect AB and CD is less than the limit, cut-off this small protrusion. (Atomic rules 24, 25, 26, 27, 28).

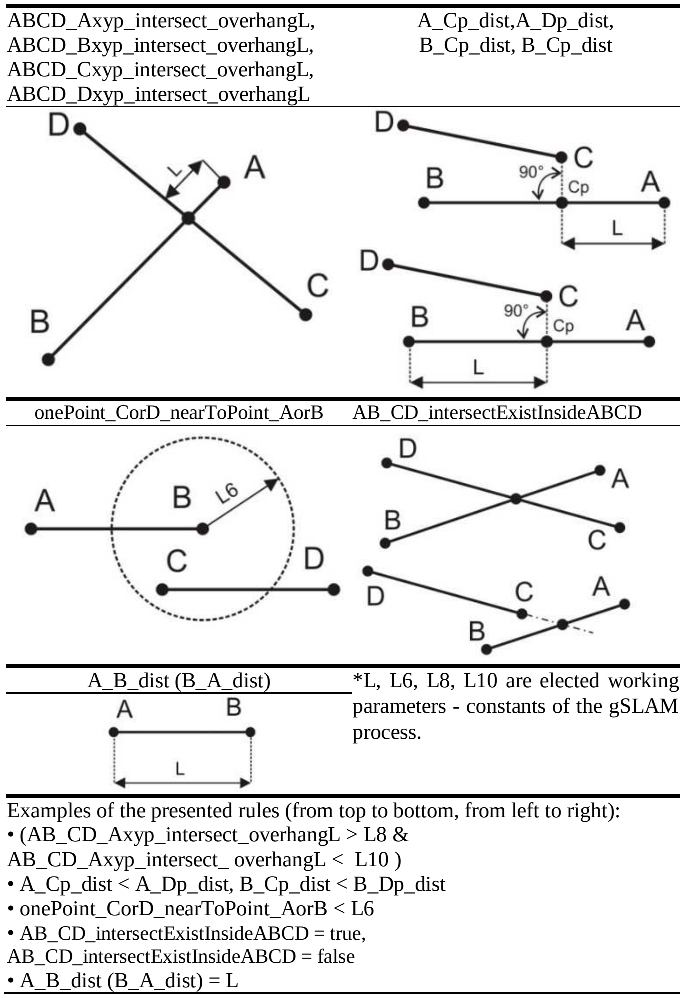
3) The intersect of the AB and CD does not esist and any point $p_1 = \{A, B, C, D\}$ is close to any other point $p_2 = \{A, B, C, D\}$, $p_1 \neq p_2$, and point p_1 or p_2 lies between AB or CD, merges with the nearest points. Only two points can be merged. (Atomic rules 24, 31, 35, 32, 36, 29, 33, 30, 33, 34).

TABLE III
ATOMIC RULE LIST – MERGING PROCESS

Nr.	Atomic rule	Description
18	onePoint_CorD_nearToPoint_AorB	One point C or D lie close to point A or B of AB abscissa. Max. distance is defined by fix. threshold.
19	A_B_dist, B_A_dist	Length of AB abscissa.
20	A_Cp_dist	Distance of the point A and intersect of AB and perpendicular line passing the point C.
21	A_Dp_dist	Distance of the point A and intersect of the AB and perpendicular line passing the point D.
22	B_Cp_dist	Distance of the point B and intersect of the AB and perpendicular line passing the point C.
23	B_Dp_dist	Distance of the point B and intersect of the AB and perpendicular line passing the point D.
24	AB_CD_intersectExistInsideABCD	This rule tells us that abscissae AB and CD have one intersect between AB and CD points. 'True' if yes, 'False' if intersect does not esist.
25	ABCD_Axyp_intersect_overhangL	Distance of the point A and intersect AB and CD abscissae, if rule 24 is true.
26	ABCD_Bxyp_intersect_overhangL	Distance of the point B and intersect AB and CD abscissae, if rule 24 is true.
27	ABCD_Cxyp_intersect_overhangL	Distance of the point C and intersect AB and CD abscissae, if rule 24 is true.
28	ABCD_Dxyp_intersect_overhangL	Distance of the point D and intersect AB and CD abscissae, if rule 24 is true.
29	A_inside_CD	Intersect Ap of the line passing the point A, perpendicular to CD is or is not inside CD abscissa. Rule can be true or false.
30	B_inside_CD	Intersect Bp of the line passing the point B perpendicular to CD is or is not inside CD abscissa. Rule can be true or false.
31	C_inside_AB	Intersect Cp of the line passing the point C perpendicular to AB is or is not inside AB abscissa. Rule can be true or false.
32	D_inside_AB	Intersect Dp of the line passing the point D perpendicular to AB is or is not inside AB abscissa. Rule can be true or false.
33	A_CD_dist	Distance of the point A from line which CD abscissa lies on.
34	B_CD_dist	Distance of the point B from line which CD abscissa lies on.
35	C_AB_dist	Distance of the point C from line which AB abscissa lies on.
36	D_AB_dist	Distance of the point D from line which AB abscissa lies on.

The merging process uses the atomic rule list mentioned in Table III – rules 18-36. Graphical representation of the presented atomic rules is depicted in Fig. 5.

The merging process occurs only in a limited range of possible combinations of AB and CD positions, which is also sufficient for the construction of a high-quality vector map. If unclassifiable schema appears, it is inserted into $\boldsymbol{GM}$ without any change. Such problems appear if random perpendicular abscissae, perpendicular to walls in environment, have to be processed for example.


ABCD_Axyp_intersect_overhangL, ABCD_Bxyp_intersect_overhangL, ABCD_Cxyp_intersect_overhangL, ABCD_Dxyp_intersect_overhangL

A_Cp_dist,A_Dp_dist, B_Cp_dist, B_Cp_dist

onePoint_CorD_nearToPoint_AorB

AB_CD_intersectExistInsideABCD

A_B_dist (B_A_dist)

*L, L6, L8, L10 are elected working parameters - constants of the gSLAM process.

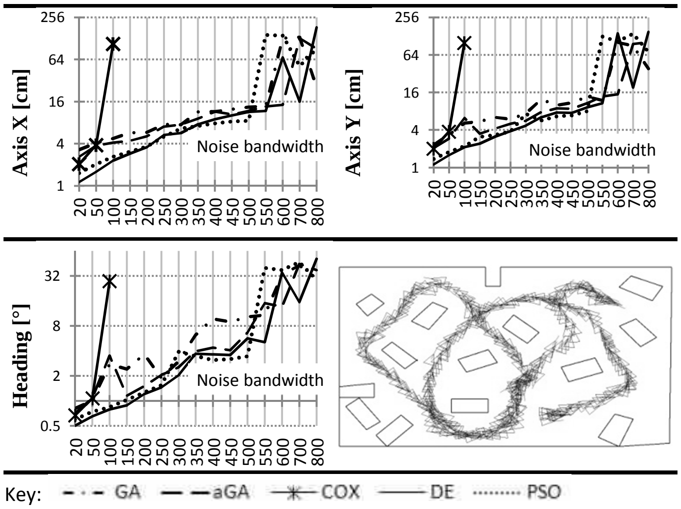
Examples of the presented rules (from top to bottom, from left to right):
- (AB_CD_Axyp_intersect_overhangL > L8 & AB_CD_Axyp_intersect_ overhangL < L10)
- A_Cp_dist < A_Dp_dist, B_Cp_dist < B_Dp_dist
- onePoint_CorD_nearToPoint_AorB < L6
- AB_CD_intersectExistInsideABCD = true, AB_CD_intersectExistInsideABCD = false
- A_B_dist (B_A_dist) = L

Fig. 5. Atomic rule list, conceptional relation between two abscissae AB and CD; graphic representation.

IV. Experimental Results

A. DE efficiency and relevancy – short discussion

DE is a stochastic optimizer. The optimized task is continual, separable, t-variant, unimodal (for small searching area only). Several different evolutionary optimizers (EA) were tested in a continual localization task – see Fig. 6. All tested algorithms: SGA[48], [49], [55], aGA[54], PSO[53] and DE applied to (3) and (4) equations (alias $\mathcal{F}_2$, $\mathcal{F}_3$ strategy) provide well usable results. Beside these optimizers classic Cox's [5] gradient method was tested as well. All methods were tested under heavy-duty operation conditions at a continual localization task in known environment (known geometric map) to get their reliability and capabilities. Unfortunately, the map building algorithm is not highly noise resistant that much. Additive Gaussian noise with different noise bandwidth (independent, fixed noise bandwidth) from zero to 800cm was tested. The sample of noise was obtained according to the equation: $D_{noisy\_e(i)} = D_{e(i)} + rnd(-L, +L)$, where $2L$ is equal to noise bandwidth and rnd represents a random numbers generator – normal distribution, mean zero. DE and PSO algorithms proved the best results without regard to different types of environments – see [54]. In contrast to other used EA, DE and PSO provide stable and almost identical results, especially on lower noise levels. Occasional malfunction (which SGA suffers from so much) was never observed. Permanent malfunction of any EA method caused by additive noise was observed only on highest noise levels from the level of about 550cm.



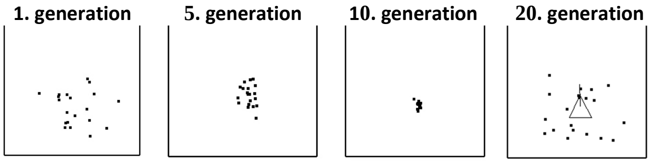
Vertical axis in $\boldsymbol{log_2}$ scale. Bottom right - Localization in environment A, noise bandwidth +/-225cm.

Fig. 6. Graphs of accuracy of estimating the position and heading - SGA, aGA, PSO, DE and Cox's optimizers.

For comparison classic, Cox gradient method is only able to work at noise level no higher than about 50. On Table 1, line 8, block A1, DE algorithm is used. Input of this optimizer is the area of width x height approx. 60 × 60cm (or bigger) around the last known robot's pose, $\boldsymbol{TM}$ (temporary map) obtained from $\boldsymbol{GM}$ based on classic ray-tracing algorithm + line fitting method – block A1; ($\boldsymbol{TM}$ is used for computation acceleration purposes only) and actual D_e vector.

DE optimizer used in presented gSLAM method solves a classic two-dimensional optimization problem. Heading of the robot is calculated separately because of the accuracy and higher speed. Step in heading is 0.5°. If a three dimensional optimizer would be used similar to [16], the number of generations and number of individuals in every population must be minimally three times higher.

In Fig. 7, population convergence is depicted. Individuals in the first generation cover equally the whole searching area (100x100cm is elected in here). After 10 generations, all individuals are almost at the correct pose. Normally 15 generations/10 individuals are efficient to ensure the correct convergence. After 20 generations, DE optimizer found the correct pose. In Fig. 7, robot is depicted by a small triangle; heading is depicted by a short line. 10 classic basic perturbation vectors were tested to get which vector is the most suitable. DE/Rand1/Exp provides the best results. Beside the DE, PSO optimizer provides good results too, but DE is better of approx. 5% if the additive (Gaussian) noise is small or zero. In structured environment PSO is significantly better from noise level approx. 250cm. Presented gSLAM algorithm is designed to build a map of unknown environment without moving objects and under noise stress no more than $\pm(8-10)$cm. Noise level value was obtained based on practical experiments. Ability to work under higher noise levels is significantly reduced thanks to A1 and A5 algorithms - line fitting and merging methods.



Key: 20 generations, 20 individuals, searching area 100x100cm. DE/Rand1/Exp, F=0.6,Pcr=1.0.

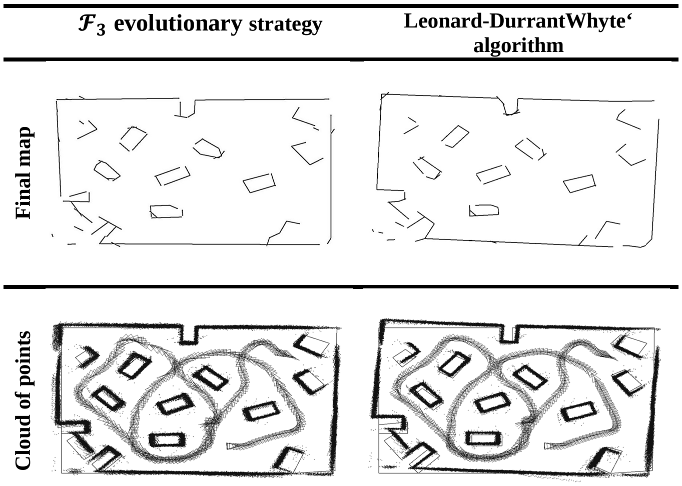
Fig. 7. Population convergence - Differential evolution.

B. *Experimental verification of the gSLAM algorithm*

The experimental verification of the proposed gSLAM algorithm was performed in two structured environments – A and B. The first environment was build-up for test purposes by cardboard boxes in laboratory. It is a common indoor office environment of 10 × 10 meters with several obstacles inside. The obstacles are cardboard boxes (approx. 40 × 60cm). Trajectory length was approx. 3001.11cm. The robot obtained 350 D_e vectors. The environment consists of 60 walls. Differential evolution - working parameters: DE/rand/1/exp, $POP = 25$, $GEN = 25$, searching area 32×32cm, $F = 0.6$, $P_{CR} = 0.9$. Such working parameters were obtained from practical experiments. If F value is smaller, time to correct pose evaluation is significantly longer. There is a linear dependence. Thanks to quantizing noise of used 2DLS Sick-PLS100 theoretical accuracy is ± 5 cm. Practically, it is twice as worse. The second environment B is a large cluster of offices. Dimensions are 2560 × 1880cm. Trajectory length is 23629.60cm. The number of lines is 329; the number of D_e vectors is 1832. Robot passed the trajectory which many times intersected itself. Environment B consists of 10 small offices and one long hallway. The working parameters of DE estimator were identical to the first experiment.

In the first experiment (see Fig. 8) robot passed the trajectory which intersects itself in three points. Presented estimator $\mathcal{F}_3$(or $\mathcal{F}_2$) does not use 'closing loop' mechanism (global localization based pose corrector) capable of improving the correct pose estimation globally. This makes the estimator more sensitive to noise.

The robot was able to pass through the environment without loss of orientation. Leonard-DurrantWhyte's algorithm [18, 19, 20] was tested for comparison of the efficiency of tested methods $\mathcal{F}_3$(or $\mathcal{F}_2$). Fig. 8 shows that both methods were able to build-up the map of unknown environment without any problems.


$\mathcal{F}_3$ evolutionary strategy
Leonard-DurrantWhyte‘ algorithm
Final map
Cloud of points

*Correct shape of the tested environment is plotted as well. $\mathcal{F}_3$ and $\mathcal{F}_2$ strategies provide almost identical results – identical map. Differences are unimportant.

Fig. 8. Environment A – SLAM.

Every method has its own specific characteristics. Leonard-DurrantWhyte‘s algorithm provides slightly turned map of about 3°. The correct shape of the tested environments was bild by hands for comparison purposes only. In several places there are minor but visible inaccuracies in comparison to the map buit-up by $\mathcal{F}_3$ strategy. In both cases the final line map of the built environment can be used for localization process.

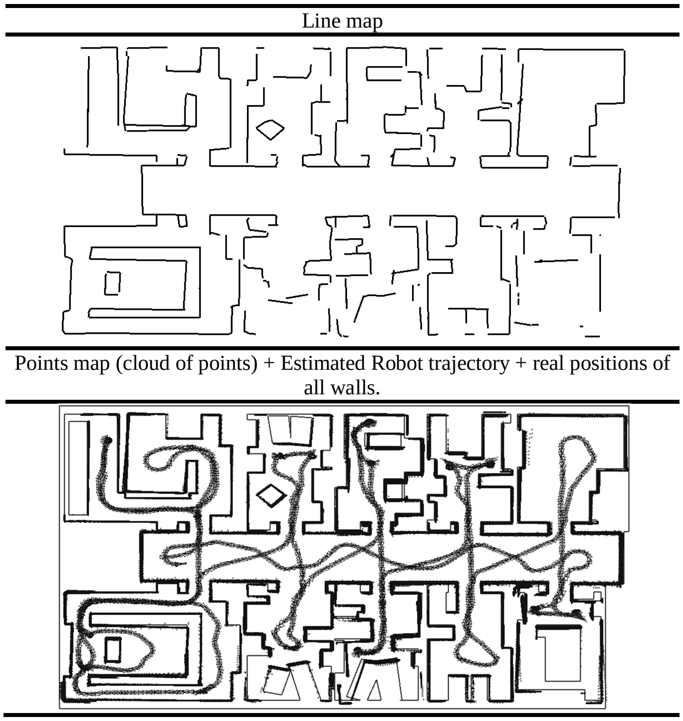

Line map
Points map (cloud of points) + Estimated Robot trajectory + real positions of all walls.

Fig. 9. Environment B - Computer Science Department, University of Bonn

The second experiment was conducted in a large office environment. The trajectory was obtained by computer simulation from the geometrical map of Computer Science Department, University of Bonn. The sensorial data were burdened by additive noise (Gaussian noise, mean zero) with bandwidth 6cm ($\pm$3cm) – it matches the new type of 2DLS Sick LMS-200. Because dimensions of the environment were set to approx. 30x20 meters only, beams were not trimmed by L_{max} function. All beams reflect the walls at any time. It is a big advantage. By this step 2DLS provides more useful information. Fig. 9 shows the result of the experiment.

The robot was able to build the map of environment without any problems. Several places are not well mapped (charted) because of small inaccuracies at heading estimation. Mostly it is small structured space. Thanks to existence of a long corridor in map B, $\mathcal{F}_3$ strategy was only used. Thanks to EA method use, the robot has to keep a minimal distance approx. 50cm from all obstacles – searching area may not overcome the walls of the environment. Sometimes this distance was slightly exceeded. Large and well structured environment is a more suitable area for gSLAM algorithm, especially if there is one central point and all offices are attainable from this. Election of suitable trajectory has a big influence too. The presented gSLAM algorithm only uses one data source – sensorial data from 2DLS. Behavior of such a method is a little different from behavior of a classic probabilistic method. When the robot is moving in a long corridor, the trajectory similar to ship cruising is the best choice. Turning should be made along large circular trajectory, if it is possible. On the other side, gSLAM shows that sensorial data as the only data source can be practically usable for such purposes. Both strategies $\mathcal{F}_3$ and $\mathcal{F}_2$ provide identical results. No malfunction was observed at testing time. Differential evolution provides a stable and very powerful tool.

V. Conclusion

The simultaneous localization and the mapping algorithm were developed and presented in this paper. The core of the presented algorithm is based on geometric primitives and evolutionary computations. Such approach provides an efficient and stable tool. The differential evolution forms a substantial part of this project. Based on practical experiments DE was elected as one of the most suitable algorithms for map building purposes especially on zero or lower noise levels. Results presented in here were obtained from two experiments—in a small indoor office environment and a cluster of small offices and provide us with a wider view on possibilities of the evolutionary robotics and map building process in general.

The proposed algorithm and basic methodology were tested in different types of environments with stable results. Navigation algorithms enabling both, global or local pose estimation and map building (SLAM process) still belong to highly interesting areas of mobile robotics. Constantly increasing computer power provides immense possibilities to create more complicated and more sophisticated algorithms for regular available computers. Thanks to the possibilities of joining the groups of different strategies, great results can be reached regarding to the type of working conditions. The presented map building method only uses one data source and thanks to the natural addition of additive errors at the pose estimation process, the size of mapped areas will be always limited to some extent.

Acknowledgment

This research was supported by Tanja Agency, to which we would like to express our cordial thanks.

References

[1] P.J. Besl, and H.D. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.

[2] J. Brest, S. Greiner, B. Boškovič, M. Mernik, and V. Žumer, “Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems,” *IEEE Trans. on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[3] W. Burgard, D. Fox, D. Hennig, and T. Schmidt, “Estimating the absolute position of a mobile robot using position probability grids,” *in Proc. of the National Conference on Artificial Intelligence*, pp. 896-901, 1996.

[4] A. Censi, “An ICP variant using a point-to-line metric,” *IEEE International Conference on Robotics and Automation ICRA 2008*, pp. 19-25, 2008.

[5] I.J. Cox, “Blanche – An experiment in Guidance and Navigation of an Autonomous Robot Vehicle,” *IEEE Trans. on Robotics and Automation*, vol. 7, no. 2, pp. 193-204, 1991.

[6] J.L. Crowley, “World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging,” *International Conference on Robotics and Automation*, pp. 674-680, 1989.

[7] J.L. Crowley, “Dynamic World Modeling for an Intelligent Mobile Robot Using a Rotating Ultra-Sonic Ranging Device,” tech. report CMU-RI-TR-84-27, Robotics Institute, Carnegie Mellon University, 1984.

[8] J.L. Crowley, “Navigation for an intelligent mobile robot,” *IEEE Journal of robotics and automation*, vol. 1, no. 1, 1985.

[9] F. Dellaert, D. Fox, W. Burgard, S. Thrun, “MonteCarlo localization for mobile robots,” *Journal of Artificial Intelligence*, vol. 128, no. 1-2, pp. 99-141, 1999.

[10] R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience, 1976.

[11] M. Ebner, “Evolving environment model for robot localization,” *Euro GP 1999*, Ebenhard-Karls-Universitat Tubingen, Germany, Springer Verlag, pp. 184-192, 1999

[12] A.E. Eiben, R. Hinterding, and Z. Michalewicz, “Parameter control in evolutionary algorithms,” *IEEE Transaction on evolutionary computation*, vol. 3, no. 2, pp. 124–141, 1999.

[13] D. Fox, W. Burgard, and S. Thrun, “Markov localization for mobile robots in dynamic environments,” *Journal of Artificial Intelligence Research*, pp. 391–427, 1999.

[14] R. Gamperle, S.D. Muller, and P. Koumoutsakos, “A parameter study for differential evolution”, *WSEAS Int. Conference on advances in intelligent systems*, pp. 293–298, 2002.

[15] Q. Ke, and T. Kanade, “Robust L1 Norm Factorization in the Presence of Outliers and Missing Data by Alternative Convex Programming,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 2005.

[16] N.M. Kwok, D.K. Liu, and G. Dissanayake, “Evolutionary computing based mobile robot localization,” *Engineering Applications of Artificial Intelligence*, vol. 19, pp. 857–868, 2006.

[17] J.C. Latombe, and A. Lazanas, “Landmark-Based Robot Navigation,” *Algoritmica*, vol. 13, no. 5, pp. 472-501, 1997.

[18] J.J. Leonard, and H.F. Durrant-Whyte, “Simultaneous map building and localization for an autonomous mobile robot,” *Conference IROS-91*, Osaka, Japan, pp. 1442-1447, 1991.

[19] J.J. Leonard, and H. F. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons,” *IEEE Trans. on Robotics and Automation*, vol. 7, no. 3, pp. 376-382, 1991.

[20] J.J. Leonard, I.J. Cox, and H.F. Durrant-Whyte, “Dynamic map building for an autonomous mobile robot,” *Int. journal on Robotics Research*, vol. 11, no./4, pp. 286-298, 1992.

[21] F. Lu, and E. Milios, “Robot pose estimation in unknown environments by matching 2D range scans,” *Journal of Intelligent Robotics Systems*, vol. 18, no. 3, pp. 249–275, 1997.

[22] E.M. Montes, and A.G.P. Ortiz, “Self-adaptive and Deterministic Parameter Control in Differential Evolution for Constrained Optimization,” *IEEE Congress on Evolutionary Computation*, pp. 1375 - 1382, 2009.

[23] H.P. Moravec, and A. Elfes, “High resolution maps from wide angle sonar,” *in Proc. IEEE Int. Conf. Robotics and Automation*, pp. 116–121, 1985.

[24] M.R. Mohammadi, and S.S. Ghidary, “Integrated PSO and line based representation approach for SLAM,” *in Proceedings of the 2011 ACM Symposium on Applied Computing*, pp.1382-1388, 2011.

[25] L. Moreno, S. Garrido, D. Blanco, and M.L. Muñoz, “Differential evolution solution to the SLAM problem,” *Journal of Robotics and Autonomous Systems*, vol. 57, no. 4, 2009.

[26] L. Moreno, J.M. Armingol, S. Garrido, A. Escalera, and M.A. Salichs, “A Genetic Algorithm for Mobile Robot Localization Using Ultrasonic Sensors,” *Journal of Intelligent and Robotic Systems*, vol. 34, no. 2, 2002.

[27] T. Pavlidis, and S.L. Horowitz, “Segmentation of Plane Curves,” *IEEE Trans. on Computers*, vol. 23, no. 8, pp. 860–870, 1974.

[28] S.T. Pfister, S.I. Roumeliotis, and J.W. Burdick, “Weighted Line Fitting Algorithms for Mobile Robot Map Building and Efficient Data Representation,” *ICRA 2003*, pp. 14-19, 2003.

[29] S.T. Pfister, “Weighted line fitting and merging,” Tech. Rep., California Institute of Technology (2002), Available: http://robotics.caltech.edu/~sam/TechReports/LineFit /linefit.pdf

[30] A.K. Qin, and P.N. Suganthan, “Self-adaptive differential evolution algorithm for numerical optimization,” *The 2005 IEEE Congress on In Evolutionary Computation*, vol. 2, pp. 1785-1791, 2005.

[31] A. Siadat, A. Kaske, S. Klausmann, M. Dufaut, and R Husson, “An Optimized Segmentation Method for a 2D Laser-Scanner Applied to Mobile Robot Navigation,” in *Proceedings of the 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications*, 1997.

[32] M. Schmidt, “Least Squares Optimization with L1-Norm Regularization,” CS542B Project Report, 2005.

[33] P. Skrzypcynsky, “Simultaneous Localization and mapping: a feature based probabilistic approach,” *Int. Journal of Applied Mathemahics and Computer Science*, vol. 19, pp. 575-588, 2009.

[34] P. Skrzypczyňski, “Building Geometrical map of environment using IR range finder data,” *Intelligent Autonomous Systems*, U.Rembold et. al. IOS Press, 1995.

[35] R. Storn, “On the Usage of Differential Evolution for Function Optimization,” *NAFIPS’96*, pp. 519–523, 1996.

[36] R. Storn, and K. Price, “Differential Evolution – a Simple and Efficient Heuristic for Global Optimization,” *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[37] J. Teo, “Exploring dynamic self-adaptive populations in differential evolution,” *Journal of Soft Computing*, vol. 10, pp. 673–686, 2006.

[38] C.K. Wong, J. Schmidt, and W.K. Yeap, “Using a Mobile Robot for Cognitive Mapping,” *International Joint Conference on Artificial Intelligence (IJCAI),* Hyderabad, India, 2007.

[39] W.K. Yeap, “Towards a computational theory of cognitive maps,” *Artificial Intelligence,* vol. 34, pp. 297-360, 1988.

[40] R. Kummerle, G. Grisetti, H. Strasdat, K.Konolige, and W. Burgard, „g2o: A General Framework for Graph Optimization,“, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3607-3613, 2011.

[41] N. Metropolis, and S. Ulam, “The Monte Carlo method,” *Journal of the American Statistical Aassociation*, vol. 44, 335-341, 1949.

[42] D. Whitley, S. Rana, and R.B. Heckendorn, „The Island Model Genetic Algorithm: On Separability, Population Size and Convergence,“ *Journal of Computing and Information Technology*, vol. 7, pp. 33-47, 1998.

[43] R. Tanese, „Distributed Genetic Algorithms,” *Proc. of the Third International Conference on Genetic Algorithms*, MorganKaufmann, J.D.Schaffer editor, pp. 434-439, 1989.

[44] M. Gorges-Schleuter, “Explicit parallelism of Genetic Algorithms through Population Structures,” *Paraller Problem Solving from nature*, Springer Verlag, H.P. Schwefel and Reinhard Manner, editors, pp. 150-159, 1991.

[45] M. Begum, G.K.I. Mann, and R.G. Gosine, “An Evolutionary Algorithm for Simultaneous Localization and Mapping of Mobile Robots,“ *Proc. of The IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 4066-4071, 2006.

[46] M. Begum, G.K.I. Mann, and R.G. Gosine, „A Fuzzy-Evolutionary Algorithm for Simultaneous Localization and Mapping of Mobile Robots,” *Proc. of IEEE Congress on Evolutionary Computation*, pp. 1975-1982, 2006.

[47] M. Begum, G.K.I. Mann, and R.G. Gosine, „Intergrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots,” *Journal of Applied Soft Computing*, vol. 8, no. 1, January, 2008.

[48] D.E. Goldberg, “Simple genetic algorithms and the minimal deceptive problem,” *Genetic Algorithms and Simulated Annealing, London*, Pitman, pp. 74-88, 1987.

[49] D.E. Goldberg, “*Genetic algorithms in search, optimization, and machine learning,*” Addison-Wesley New York, 1989.

[50] K. Price, D. Corne, M. Dorigo, and F. Glover, “An Introduction to Differential Evolution,” Eds. London, McGraw-Hill, pp. 79–108, 1989.

[51] K. Price, and R. Storn, “Minimizing the Real Functions of the ICEC’96 contest by Differential Evolution,” *IEEE Int. Conference on Evolutionary Computation (ICEC’96)*, pp. 842–844, 1996.

[52] M. Dorigo, “Optimization, Learning and Natural Algorithms”, Ph.D. dissertation, Politecnico di Milano, Italy, 1992.

[53] J. Kennedy, and R.C. Eberhart, “Particle swarm optimization,” *Proc. of the 1995 IEEE International Conference on Neural Networks*, 1995.

[54] J. Moravec, “Cascaded Evolutionary Estimator for Robot Localization,” *International Journal of Applied Evolutionary Computation (IJAEC)*, vol. 3, no. 3, pp. 33-61, 2012.

[55] J. Moravec, “Continuous Robot Localization in Known Environment Using Genetic Algorithms,” *The 10th IEEE Int. Conference on Fuzzy Systems FUZZ IEEE 2001*, Melbourne, Australia, p.6, 2001

Redes de palabras alineadas como recurso en la extracción de equivalencias léxicas de traducción y su aplicación en la alineación

Eduardo Cendejas, Grettel Barceló, Gigori Sidorov, Alexander Gelbukh, and Liliana Chanona-Hernandez

***Resumen*—La equivalencia léxica de traducción se define mediante correspondencias establecidas entre dos lenguas, comúnmente denominadas lengua de origen y lengua meta. Este artículo propone un método de extracción de dichas equivalencias en palabras no funcionales. El algoritmo se basa en dos recursos principales: 1) MultiWordNet como léxico especializado para cada uno de los idiomas involucrados y 2) textos paralelos como información adicional para proporcionar diversas lexicalizaciones de las palabras a corresponder. Utiliza como fundamento principal el hecho de que las redes de palabras que conforman MultiWordNet están alineadas. Además, se presenta la reutilización del repositorio de pares léxicos obtenidos, señalando la forma en que esta información es susceptible de ser usada en un sistema de alineación a nivel de palabras. Para realizar los experimentos se emplearon textos paralelos bilingües sin notación morfosintáctica alguna, alineados a nivel de oración en los pares de idiomas español / inglés y español / italiano.**

***Palabras clave*—Equivalencias léxicas, alineación, redes de palabras, textos paralelos.**

Aligned Word Networks as a Resource for Extraction of Lexical Translation Equivalents, and their Application to the Text Alignment Task

***Abstract*—The notion of lexical translation equivalent is defined via correspondence established between two languages conventionally called source and target languages. We propose a method for extraction such equivalents for non-functional words. Our algorithm is based in two main resources: (1) MultiWordNet as a specialized lexicon for each one of the two languages in question and (2) a parallel text corpus as a source of additional information that provides various lexicalizations of the words that are being aligned. Our method is based on the fact that the word networks that form MultiWordNet are aligned. In addition, we discuss an application of the obtained list of word pairs in a word-level text alignment system. In our experiments we used bilingual sentence-level aligned parallel texts, without any morphosyntactic annotation, for word pairs Spanish / English and Spanish / Italian.**

***Index Terms*—Translation equivalents, alignment, word networks, parallel texts.**

I. Introducción

EN LA DÉCADA de los ochentas se introdujo la idea de almacenar electrónicamente traducciones pasadas en un formato bilingüe. El concepto consistía en la construcción de una concordancia bilingüe teniendo un operador de datos que manualmente introdujera el texto y su traducción [1], creando así una herramienta de referencia valiosa para los traductores. Como consecuencia, se originó un gran interés en la construcción automática de tales bases de datos a mayor escala.

En años recientes, se ha suscitado un progreso considerable en el campo de la equivalencia y alineación paralela de textos. El creciente interés en éstos viene dado básicamente de la popularización de Internet que ha hecho de la red una enorme colección documental bilingüe. La expresión «texto paralelo» por sí misma es ahora bien establecida dentro de la comunidad de la lingüística computacional.

Un texto paralelo es la unión de dos o más textos que poseen el mismo contenido semántico, pero expresados en lenguajes diferentes [2]. El término paralelo no implica que los textos tengan una correspondencia exacta entre palabras, oraciones y/o párrafos; es decir, dos textos pueden estar completamente desalineados sin dejar de ser textos paralelos [3].

I-A. *Equivalencia vs. alineación*

Uno de los problemas clásicos y tradicionales de la teoría de la traducción ha sido, y lo sigue siendo, el de la equivalencia. En esta tarea, un par de traducción es considerado correcto si existe al menos un contexto en el cual éste ha sido acertado. Usualmente la extracción sólo está interesada en las categorías principales (sustantivos, adjetivos y verbos).

Sin embargo, la alineación a nivel de palabras es diferente y más difícil que el problema de extracción de equivalencias [4]. Consiste en indicar qué palabra del lenguaje origen (LO), se corresponde con una palabra en el lenguaje meta (LM), hasta encontrar todas las correspondencias entre las palabras de los textos que conforman el corpus bilingüe [5]. Por tanto, requiere que cada palabra (sin importar el POS –*Part of speech*) o signo de puntuación en ambas partes del bitexto sean asignados a una traducción o a un nulo en su contraparte.

Manuscript received December 9, 2011. Manuscript accepted for publication March 8, 2012.

E. Cendejas, G. Barceló, G. Sidorov and A. Gelbukh are with the Center for Computing Research, Instituto Politécnico Nacional, Mexico City, Mexico (web: cic.ipn.mx/šidorov, www.gelbukh.com); L. Chanona-Hernandez is with the ESIME, Instituto Politécnico Nacional, Mexico City, Mexico.

II. Trabajos previos

Tanto la extracción de equivalencias léxicas, como la alineación de textos, han demostrado ser fuentes inestimables de datos de traducción para los bancos de terminología y los diccionarios bilingües. Actualmente, ambas tareas están proporcionando la base para el desarrollo de una nueva generación de herramientas de asistencia para los traductores humanos, que permitan mejorar la calidad y productividad de su trabajo.

Los métodos propuestos para encontrar equivalentes y alineaciones en textos paralelos, se han clasificado generalmente en dos tipos de aproximaciones: los estadísticos y los lingüísticos.

Los métodos *estadísticos* clásicos utilizan información no-léxica, como la correlación esperada de longitud y posición de las unidades de texto (párrafos u oraciones), la frecuencia de co-ocurrencia, la proporción del tamaño de las oraciones en los diferentes idiomas, etc. [6]. Intentan establecer la correspondencia entre las unidades del tamaño esperado [7], [8], [9], el cual puede medirse en el número de palabras o caracteres [10].

Por otro lado, los métodos *lingüísticos* se apoyan en recursos léxicos existentes, como diccionarios bilingües de gran escala y glosarios [6], para establecer la correspondencia entre las unidades estructurales. Por la disponibilidad cada vez mayor de recursos bilingües, se invierte más esfuerzo en la investigación de la efectividad de los acercamientos basados en léxicos [11], [12], [13].

La reestructuración que se realiza durante la alineación, independientemente del método que se requiera, puede ser realizada entre párrafos, sentencias o palabras, del contenido expresado en lenguaje original y su traducción [14].

Por otra parte, a pesar de que la extracción de equivalencia puede realizarse en cualquiera de los niveles de emparejamiento, del mismo modo que la alineación, ésta ha estado enfocada a la identificación de pares de palabras o secuencias de palabras (con patrones establecidos [15] o colocaciones [16]).

La correspondencia de los textos paralelos, en una primera fase a nivel de párrafo, es la más simple, pues casi siempre están en una relación 1:1, lo cual resulta fortuito ya que la correspondencia a nivel de oraciones mejora mucho si se realiza primero la relación a nivel de párrafos [3].

El nivel de resolución de oraciones presentó un desafío mayor, descubriendo que en él, las correspondencias uno a muchos y muchos a uno, no son raras. Se han publicado muchos algoritmos para alinear oraciones en textos paralelos: algunos se basan en la observación de la correlación entre la longitud de un texto y la de su traducción [7], [8], otros emplean técnicas estadísticas basadas en cognados [17] o que maximizan el número de las correspondencias sistemáticas entre las palabras [11], [18]. Aunque los resultados obtenidos por algunos de los métodos antedichos han sido absolutamente exactos cuando están probados en recopilaciones relativamente limpias y extensas, siguen siendo alineaciones "parciales", pues ocultan el grado más fino de resolución debajo del nivel de oración: el nivel de palabra.

La correspondencia a nivel de palabras tiene mayor dificultad que el de oración, puesto que la relación 1:1 llega a ser cada vez más rara. Se han propuesto varios métodos para encontrar equivalencias y alineaciones entre palabras en textos paralelos. Algunas técnicas dependen de un conjunto de parámetros que son aprendidos mediante un proceso de entrenamiento de datos [19], [20], [21]. Otros se basan en técnicas estadísticas, clasificadas principalmente en dos categorías [22]: métodos de prueba de hipótesis y métodos de estimación. Los primeros extraen los candidatos de equivalencia de las unidades de traducción y se someten a una prueba estadística, que miden la co-ocurrencia y/o similaridad de las cadenas [16], [23], [24]. Los enfoques de estimación hacen uso de modelos de alineación probabilística, calculados de corpus paralelos [25], [26]. Estos modelos son a menudo derivados de la traducción automática estadística [19] y las probabilidades son obtenidas de la observación de pares en función de los parámetros del modelo empleado.

Pero a pesar de la existencia de diversos métodos a nivel de palabra, las tareas de equivalencia y alineación aún están lejos de ser un trabajo trivial debido a la diversidad de idiomas naturales. Por ejemplo, la correspondencia de palabras dentro de las expresiones idiomáticas y traducciones libres son problemáticas. Además, cuando dos idiomas difieren ampliamente en el orden de las palabras, resulta muy difícil encontrar las relaciones. Por consiguiente, es necesario incorporar información lingüística útil para aliviar estos problemas.

III. Nuestro acercamiento

El propósito de nuestra investigación en una etapa inicial consistió en el diseño de un algoritmo efectivo de alineación de palabras. La idea de obtener equivalencias léxicas en una primera fase, vino con la disponibilidad del recurso MultiWordNet y su concepción.

Se han utilizado al menos dos metodologías de construcción de redes de palabras multilingües. La primera consiste en la construcción independiente de wordnets específicas del lenguaje, con una fase posterior de búsqueda de correspondencias entre ellas. Este enfoque se basa en la hipótesis de que traducciones recíprocas en textos paralelos deben tener los mismos significados y utiliza un índice interlingua (ILI) para materializar las relaciones entre idiomas. EuroWordNet [27] y BalkaNet fueron desarrolladas empleando este enfoque. La segunda metodología consiste en la construcción de las de wordnets específicas del lenguaje manteniendo, tanto como sean posibles, las relaciones semánticas disponibles en Princeton WordNet (PWN). Este acercamiento se empleó para el desarrollo de MultiWordNet [28].

MultiWordNet (MWN) es una base de datos léxica multilingüe, en la cual se ha realizado una alineación estricta entre PWN y redes de palabras para el español y el italiano,

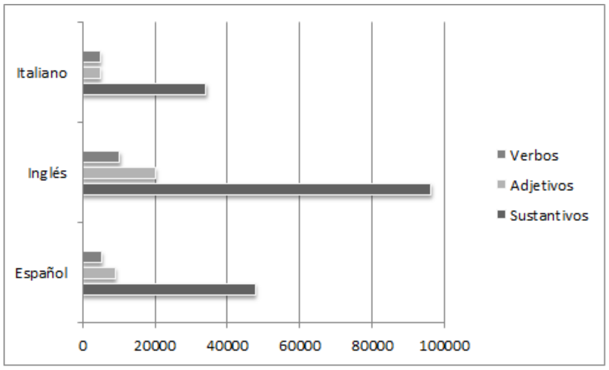


Fig. 1. Composición de las redes de palabras según la categoría gramatical

entre otros lenguajes. Estas redes describen las relaciones léxicas y semánticas existentes entre las palabras y recopilan sus sentidos al igual que un diccionario monolingüe.

La Figura 1 muestra la composición, por categoría gramatical, de cada una de las redes de palabras empleadas en el estudio. Con ello se tiene una perspectiva más clara de la completitud de las mismas, comparando los tres idiomas involucrados.

Los synsets, para cada uno de los idiomas alineados, fueron creados en correspondencia con los synsets de PWN en la medida de las posibilidades. Las relaciones semánticas también fueron importadas de los synsets ingleses correspondientes. Es decir, se asume que si existen dos synsets relacionados en PWN, la misma relación existe en los synsets pertinentes en los otros idiomas [29].

Esta estructura es la que nos permite realizar agrupamiento de sentidos para extraer pares de equivalentes de traducción, mismos que son usados en la fase de alineación, como información lingüística útil para reforzar los enlaces obtenidos. Aunque los experimentos han sido realizados usando MWN, se pueden extender las nociones intuitivas del algoritmo propuesto, introduciendo un marco formal y conceptual para el uso de los ILI. De esta forma, se podrían realizar estudios con redes de palabras construidas bajo el enfoque de EuroWordNet.

III-A. Pre-procesamiento de los textos

Para extraer las equivalencias léxicas, el algoritmo propuesto requiere que los textos se encuentren alineados a nivel de oración. Además, los enlaces pueden producirse entre dos palabras como entradas básicas, dos lemas obtenidos de la aplicación de reglas morfológicas sobre las palabras, o la combinación de éstas. Por ejemplo, el establecimiento de la equivalencia entre: *clothes* (en inglés) y *ropas* (en español) se representa por medio de un enlace entre una palabra y su traducción en plural, es decir, entre una entrada léxica básica y otra a la que se le ha aplicado la regla morfológica de formación del plural. Por tanto, todas las palabras implicadas en ambos contextos deben ser lematizadas.

Como el algoritmo propuesto se basa en la concordancia de synstes en redes de palabras alineadas, una vez que se cuenta con los lemas, se extraen los synsets de las palabras no funcionales. Es importante recalcar que no se lleva a cabo ningún proceso de etiquetado durante esta fase.

III-B. Extracción de equivalencias léxicas

MWN fue elegido como léxico especializado, porque contiene redes de palabras para cada uno de los idiomas involucrados y por su alineación exacta con PWN. Esta última característica constituye la clave en el método que se plantea.

El hecho de que las redes de palabras están alineadas, implica que para expresar un determinado significado, todos utilizan el mismo synset, independiente del idioma. La constitución de los synsets, permite incorporar un grupo de formas sinónimas, por tanto, todos los wordnets en dicho synset almacenan las formas de palabras que pueden representar el significado específico.

Haciendo énfasis en el proceso de extracción de equivalencias, lo que se hace a grandes rasgos, es buscar todos los synsets de la palabra a corresponder, es decir, todos sus posibles sentidos. Este conjunto se compara entonces con los conjuntos extraídos del contexto paralelo (texto en el idioma meta), y aquella palabra que posea mayor coincidencia o intercepción de synsets, será la palabra que se relaciona con la palabra en el origen.

Aquí es necesario considerar los niveles de ambigüedad de los idiomas, pues ello podría afectar en el desempeño del algoritmo. Se realizó entonces el cálculo de los promedios de sentidos por palabra almacenada en su red correspondiente. El análisis comprende la observación por categoría gramatical.

TABLA I
ESTADÍSTICAS DE SENTIDOS EN MWN

	Español	Italiano	Inglés
Sentidos asignados	93425	64384	171018
Sentidos sustantivos	63028	48376	119050
Sentidos adjetivos	17999	6228	29883
Sentidos verbos	12398	9780	22085
Promedio de sentidos	**1.55612**	1.50009	1.42733
Promedio sustantivos	1.32046	1.41351	1.23591
Promedio adjetivos	1.98818	1.26972	1.48148
Promedio verbos	2.34057	1.99755	2.13815

La primera fila en la tabla I muestra la cantidad de sentidos que han sido atribuidos a los lemas de entrada en las tablas de índice. De esta forma, si un mismo sentido ha sido asignado a dos palabras diferentes, se toma dos veces en cuenta en este conteo. Por tanto, no se está haciendo referencia al total de synsets del idioma, sino a la cantidad de veces que éstos han sido asignados. Las siguientes tres filas, se corresponden con la división de esta cantidad por POS.

Con la medida de asignación y el número de lemas, se puede determinar el promedio de sentidos. Mientras mayor sea el resultado obtenido, mayor será el grado de polisemia en el lenguaje. Por tanto, el español es, de los tres idiomas

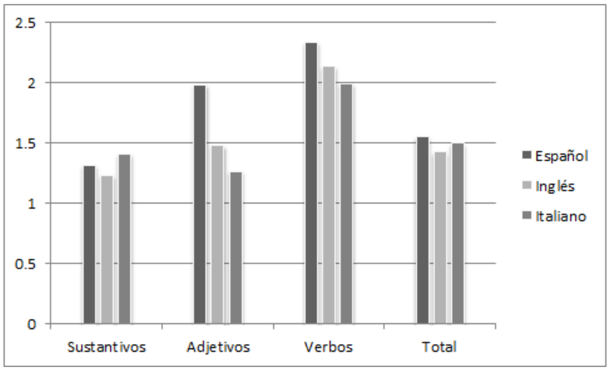


Fig. 2. Promedio de sentidos asociados en MWN

estudiados, el más ambiguo. Sin embargo, en este punto es necesario considerar que aspectos como la incompletidud de los wordnets influyen en los resultados.

En la Figura 2 se presentan los valores de la tabla anterior con formato de gráfica de barras.

Por otra parte, el mecanismo desarrollado está enfocado en la extracción de equivalencias con limitación uno a uno [24]. Es decir, todas las palabras participan en una única equivalencia, con excepción del nulo. Esto para evitar correspondencias no deseadas, como por ejemplo, tener muchas palabras en español alineadas con la misma palabra en inglés. Así, en cada fase del algoritmo, si una traducción potencial pasa a ser parte de un par de equivalencia, la lista de traducciones potenciales se va reduciendo.

En MWN se manejan elementos léxicos simples, por tanto las expresiones multipalabras no son encontradas y consideraremos sólo equivalencias 1:1, durante la fase de extracción. De este modo, en el texto quedan descartadas todas aquellas frases indivisibles con un sentido específico.

III-C. El programa de linkeo

El sistema toma como entrada los textos paralelos preprocesados para realizar el emparejamiento entre las palabras. Para ello, por cada par de sentencias alineadas $< SO_i/SM_j >$, se recorre desde la primera hasta la última palabra en SO_i, donde SO_i representa la i-ésima oración en el texto origen y SM_j la j-ésima oración en el texto meta. Si la palabra considerada (W_x) pertenece a alguna de las clases gramaticales contempladas en el estudio, es comparada entonces con todas las palabras (T_x) en su contexto paralelo.

En la extracción de equivalencias y la alineación a nivel de palabras no existe restricción alguna de la clase gramatical (POS) de las palabras en un par, pues es muy frecuente que durante la traducción se cambie el POS para expresar la misma idea. Por ejemplo, en los fragmentos: *...che non voglio ricordare come si chiami...*(italiano) y *...de cuyo nombre no quiero acordarme...*(español), *nombre* puede ser alineado con *si chiami*.

Basados en el supuesto de que los textos origen y meta se encuentran alineados a nivel de oración, el contexto paralelo de cada palabra en SO_i será aquella oración SM_j en el par alineado. Hasta este momento, todas las palabras no funcionales en SM_j constituyen traducciones potenciales de la palabra W_x. La comparación consiste en la búsqueda del conjunto de intersección entre los synsets de los lemas de W_x y los synsets de los lemas de cada palabra en el contexto paralelo, obtenidos en la fase de preprocesamiento. La tabla II representa este análisis. $S()$ y $L()$ representan funciones de extracción de synsets y lemas respectivamente.

TABLA II
BÚSQUEDA DE COINCIDENCIAS DE SYNSETS ENTRE W_x Y LAS PALABRAS DEL TEXTO META

	W_x
T_1	$S(L(W_x)) \cap S(L(T_1))$
T_2	$S(L(W_x)) \cap S(L(T_2))$
...	...
T_n	$S(L(W_x)) \cap S(L(T_N))$

Para determinar la equivalencia léxica de W_x se calcula la cardinalidad de los conjuntos de intersección producidos. A partir de estos valores puede presentarse alguna de las tres siguientes situaciones:

1. Que una de las palabras en el contexto paralelo pueda ser directamente asignada como equivalencia léxica de W_x por ser partícipe del conjunto de intersección de mayor cardinalidad.
2. Que varias palabras en el contexto paralelo poseen la misma cardinalidad.
3. Que ninguna de las palabras en el contexto paralelo constituya la equivalencia de W_x

En el primer caso el proceso de extracción de equivalencia queda concluido para la palabra W_x.

Cuando no es posible seleccionar una de las traducciones potenciales por haber muchas que poseen el mismo valor de cardinalidad, se utiliza el sentido más cercano. Por ejemplo, suponga que la palabra *quiero* ha tenido un synset en común con *have* y uno con *desire*. Para determinar cuál de las dos traducciones será elegida como equivalente, se compara la posición que ocupa en cada una, el synset coincidente. Aquella palabra cuya posición del synset coincidente sea menor, será la ganadora. La Figura 3 muestra este procedimiento para el ejemplo citado.

Para la forma de palabra *have*, el synset coincidente (v#01530096) ocupa la posición 15, mientras que para *desire*, el synset coincidente (v#01245362) está en la posición 1. Lo anterior se resumen en que es mucho más común utilizar *desire* en el sentido de querer, que *have*.

En el último caso, donde todas las palabras en el contexto paralelo tienen intersección nula con el conjunto de synstes de W_x, se han aplicado cuatro medidas de similitud semántica: Leacock and Chodorow [30], Hirst and St-Onge [31], edge [32] y random. Estas medidas han sido implementadas en WordNet::Similarity package [33] y todas ellas se basan, de algún modo, en la estructura de PWN. A continuación se describe cada una de estas cuatro medidas:

$< SO_i$ / $SM_j >$ = <*"En un lugar de la Mancha, de cuyo nombre no* ***quiero*** *acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor."* /
"In a village of La Mancha, the name of which I ***have*** *no* ***desire*** *to call to mind, there lived not long since one of those gentlemen that keep a lance in the lance-rack, an old buckler, a lean hack, and a greyhound for coursing."*>
W_x = *quiero*
posibles traducciones = {*have*, *desire*}
$L(quiero)$ = {$querer$[Verb]}
$L(have)$ = {$have$[Verb]}
$L(desire)$ = {$desire$[Verb]}
$S(querer$[Verb]) = {v#00472243, v#00479719, v#00479841, v#00808096, v#01211759, v#01245362, **v#01530096**}
$S(have$[Verb]) = {v#01508689, v#01794357, v#01443215, v#01509295, v#01857688, v#00080395, v#00786286, v#01620370, v#01185771, v#01509557, v#01876679, v#00080645, v#00045715, v#00523422, **v#01530096**, v#01513366, v#00045966, v#01608899, v#00039991, v#00978092}
$S(desire$[Verb]) = {v#01245362, v#01246466, v#01246175}

Fig. 3. Búsqueda de coincidencias

– *Leacock and Chodorow* (*LCH*): Se basa en la longitud $len(s_1, s_2)$ de la ruta más corta entre dos synsets y la profundidad máxima D de la taxonomía:

$$LCH(s_1, s_2) = \frac{-log\ len(s_1, s_2)}{2D}$$

El valor máximo de esta medida dependerá de la profundidad de la taxonomía y se obtiene al comparar un synset consigo mismo.

– *Hirst-St-Onge* (*HSO*): La idea en esta medida de similitud es que dos conceptos lexicalizados son semánticamente cercanos si sus synsets en PWN están conectados por una ruta que no es tan larga y que no cambia de dirección frecuentemente. Las direcciones de los enlaces en la misma ruta pueden variar (entre: hacia arriba –hiperonimia y meronimia–, hacia abajo –hiponimia y holonimia– y horizontal –antonimia–). La cercanía de la relación está dada por:

$$HSO(s_1, s_2) = C - \mathrm{longituddelaruta} - k \times d,$$

donde C y k son constantes (en la práctica se usa C = 8 y k = 1) y d es el número de cambios de dirección en la ruta. Si no existe dicha ruta, entonces HSO es cero y los synsets se consideran independientes.

– *Edge*: Realiza un conteo de la ruta más corta con respecto a la distancia entre synsets.

– *Random*: Asigna un número aleatorio entre 0 y 1 como medida de similitud, donde 0 indica que no existe similitud entre los conceptos y 1 que poseen el mismo sentido.

Para todas las medidas anteriores, con excepción de Edge, mientras más corta es la ruta entre los synsets, mayor será el valor de similitud.

La utilidad *similarity.pl* permite al usuario introducir pares de conceptos en la forma $word\#pos\#sense$ para medir qué tan parecidos son semánticamente. Por ejemplo, en la tabla III aparecen los sentidos de la palabra *breed* como sustantivo.

TABLA III
SYNSETS ASOCIADOS A CADA SENTIDO DE LA PALABRA *breed* COMO SUSTANTIVO

Entrada en similarity.pl	Synset correspondiente
$breed\#n\#0$	$n\#06037479$
$breed\#n\#1$	$n\#06037015$
$breed\#n\#2$	$n\#07308064$
$breed\#n\#3$	$n\#03852666$

Esta forma de especificar las entradas presenta una desventaja, pues sólo se aceptan palabras inglesas y las posiciones que ocupan los sentidos definidos de dichas palabras en PWN. Es decir, *similarity.pl* no se puede aplicar directamente si se están comparando palabras de idiomas diferentes al inglés.

Para solucionar este problema se obtiene la traducción inglesa para cada uno de los synset de la palabra española o italiana implicada en el par de comparación, aprovechando nuevamente la ventaja de que las redes de palabras en MultiWordNet están alineadas.

Así por ejemplo, si se tiene la palabra *generación*, cuyos synstes como sustantivo son: n#06196326, n#06195881, n#10955750 y n#00546392; las posibles traducciones para dichos synstes serían las palabras inglesas que poseen dichos offsets. Sin embargo, como las posibles traducciones para cada synset se emplean para hacer referencia al mismo concepto (mismo sentido), se puede elegir la primera traducción para cada synset.

TABLA IV
SELECCIÓN DE LA TRADUCCIÓN PARA CADA SYNSET DE LA PALABRA *generación* COMO SUSTANTIVO

Synset	Posibles traducciones	Traducción elegida
$n\#06196326$	*coevals* *contemporaries* *generation*	*coevals*
$n\#06195881$	*generation*	*generation*
$n\#10955750$	*generation*	*generation*
$n\#00546392$	*generation* *multiplication* *propagation*	*generation*

Ahora sólo bastaría determinar la posición que ocupa el sentido buscado sobre la lista de sentidos de la traducción y se podrían realizar comparaciones entre las palabras *breed* y *generación* con el formato aceptado por el paquete *similarity.pl*, por ejemplo usando las entradas: breed#n#0 / generation#n#1

Una vez que se tienen los valores de similitud[1] de todas las posibles parejas de synstes que se pueden formar con las dos palabras que se comparan, se toma como similitud del par, el mayor valor obtenido.

$coevals\#n\#0$ - $breed\#n\#0$ = 0
$coevals\#n\#0$ - $breed\#n\#1$ = 0
$coevals\#n\#0$ - $breed\#n\#2$ = 2
$coevals\#n\#0$ - $breed\#n\#3$ = 2

$generation\#n\#1$ - $breed\#n\#0$ = 0
$generation\#n\#1$ - $breed\#n\#1$ = 0
$generation\#n\#1$ - $breed\#n\#2$ = 2
$generation\#n\#1$ - $breed\#n\#3$ = 2

$generation\#n\#2$ - $breed\#n\#0$ = 0
$generation\#n\#2$ - $breed\#n\#1$ = 0
$generation\#n\#2$ - $breed\#n\#2$ = 4
$generation\#n\#2$ - $breed\#n\#3$ = 0

$generation\#n\#3$ - $breed\#n\#0$ = 0
$generation\#n\#3$ - $breed\#n\#1$ = 0
$generation\#n\#3$ - $breed\#n\#2$ = 0
$generation\#n\#3$ - $breed\#n\#3$ = 0

Fig. 4. Similitud de *breed* y las traducciones de *generación* (*coevals* y *generation*)

Este valor se coloca en una matriz de similitud como se muestra en la Figura 5. Tal matriz posee el valor -1 en todas las palabras que fueron directamente asignadas por poseer la mayor cardinalidad absoluta, con el objetivo de que no sean tomadas nuevamente en cuenta en la actual etapa de extracción. Así, de la matriz se advierte que W_1 y T_2 fueron previamente relacionadas en un par de equivalencia.

	T_1	T_2	*breed*	...	T_n
W_1	-1	-1	-1	...	-1
generación	2	-1	4	...	0
W_3	0	-1	3	...	2
...	...	...	...	...	...
W_m	3	-1	2	...	0

Fig. 5. Matriz de similitud

Finalmente, los pares de equivalencia se forman comenzando por la mayor similitud y así sucesivamente. En el caso de la matriz anterior los pares que se obtendrían con los valores definidos son:

$$(generación, breed)$$

$$(W_3, T_n)$$

$$(W_m, T_1)$$

IV. Evaluación

Para realizar nuestros experimentos, empleamos fragmentos elegidos aleatoriamente de la novela *Don Quijote de la Mancha*, en sus versiones paralelas español / inglés y español / italiano. Los fragmentos usados en ambos pares de idiomas fueron los mismos. Los conjuntos de prueba estuvieron formados por 23 sentencias alineadas. El texto en español está constituido por 828 palabras. La tabla V muestra la composición de cada uno de los fragmentos de los textos meta empleados como corpus de prueba.

TABLA V
Composición de los textos meta

	Inglés	Italiano
# palabras	796	866
Promedio de sentidos	6.8653	4.4276

Para producir el gold estándar de los pares de alineación de todas las palabras no funcionales, dos anotadores fueron instruidos con procedimientos específicos de cuándo asignar un equivalente nulo. No se incluyeron etiquetas de probabilidades. En caso de que hubiera un desacuerdo para un par específico, un tercer anotador definía el correcto.

La tabla VI muestra la cantidad de pares de alineación determinados por los anotadores (estándar de oro). El topline indica el número máximo de equivalencias que podrían ser extraídas por el sistema, considerando relaciones 1:1 exclusivamente, la incompletitud de las redes de palabras que conforman MWN y las brechas léxicas (gaps) de los lenguajes implicados.

TABLA VI
Número de equivalencias sugeridas por los anotadores y topline / recall del sistema

Texto	Gold estándar	Topline
Don Quijote de la Mancha (español / inglés)	389	329
Don Quijote de la Mancha (español / italiano)	333	277

IV-A. *Medidas de evaluación*

Realizamos la evaluación respecto a tres diferentes medidas: precisión, recall y F-measure. La precisión es calculada como el número de equivalencias extraídas correctamente entre el número de equivalencias sugeridas por el sistema. El recall se corresponde al número de equivalencias extraídas correctamente entre el número de equivalencias sugeridas por los anotadores.

Sin embargo, ni la precisión ni el recall pueden, de manera independiente, determinar la calidad del emparejamiento. Por lo general, la maximización del recall compromete la precisión y viceversa [34]. Por tanto, se requiere una medida que combine ambos parámetros. La F-measure posee esta característica y se determina como:

[1] En este ejemplo obtenidos empleando el método Hirst-St-Onge

$$F\text{-measure} = \frac{2 \times \text{precisión} \times \text{recall}}{\text{precisión} + \text{recall}}$$

En este caso, la precisión y el recall poseen el mismo peso, pero la fórmula anterior puede ajustarse si se desea otorgar mayor peso a algunas de estas dos medidas.

IV-B. Resultados y discusión

Los resultados fueron obtenidos con una iteración del algoritmo, por tanto, los tiempos de respuesta son muy pequeños. Además, se establecieron umbrales para la asignación de similitudes en cada método. De esta forma, se evitan despropósitos lingüísticos, al establecer qué valores numéricos de las medidas se consideran cercanos. Los umbrales en cada método fueron los siguientes: LCH $\Rightarrow$ 2, HSO $\Rightarrow$ 2, Edge $\Rightarrow$ 0.2 y Random $\Rightarrow$ 0.2. De esta forma, un valor de 0.166667 para el par (*astillero*, *village*) en el método Edge es descartado y se asigna similitud 0, o sea, sólo se toman valores de similitud mayores a 0.2 como fue señalado en el umbral. Esto evita la extracción de equivalencias erróneas y por tanto, tiene influencia en la precisión.

La tabla VII muestra la cantidad de pares extraídos por el sistema (total), la cantidad de éstos que son correctos (considerando equivalencia –EQ– y alineación –AL–) y los valores de las tres medidas empleadas. Los mismos valores han sido graficados en la Figura 6. La entrada *Sólo coincidentes*, se refiere al caso donde no se ha aplicado ninguna medida de similitud semántica para efectuar la extracción cuando no hay synsets comunes entre la palabra a corresponder y sus traducciones potenciales.

Si se comparan los valores de las tres medidas, independientemente del método de similitud empleado, existe una diferencia aproximada de 3.76 % en promedio entre el procedimiento de extracción de equivalencias (EQ) y la alineación (AL). Es comprensible que en EQ siempre se obtengan mejores resultados, pues, por definición, un par se considera correcto al existir al menos un contexto en el que las palabras que conforman el par sean traducciones entre sí.

El desempeño del sistema se ve afectado durante la alineación, ya que no sólo se deben considerar pares de traducción, sino también la correspondencia entre las palabras según la función que realizan en la oración y en este proceso, la inserción de nulos es requerida.

En términos generales, se puede advertir que el inglés consigue mejores resultados, en su alineación con el español, para todos los métodos aplicados, a pesar de tener un mayor grado de polisemia para el corpus de prueba (véase la tabla V).

La precisión del método de “Sólo coincidentes” tiene que ver con la asertividad de las alineaciones de las redes de palabras que conforman MWN. Sin embargo, los valores de recall obtenidos para este método son pobres en ambos casos (usando el gold standard y el topline). Esto se debe a dos razones fundamentales: 1) el hecho de que en MWN se almacenan elementos léxicos simples y por tanto, es imposible asignar un sentido específico a las expresiones multipalabra y 2) la incompletitud de las redes (véase Figura 1 para observar la desproporción entre el inglés y el resto de los idiomas).

Por otra parte, la medida LCH es comparable en términos de F-measure con el método de sólo coincidencia, a pesar de la diferencia significativa si se analizan los valores de precisión. Los métodos de similitud HSO y Edge, aumentan el recall, aunque no de manera significativa (entre 2 y 6 % aproximadamente), pero lo hacen a costa de una disminución considerable de la precisión (entre 16 y 18 %). En este sentido, es lógico que el método HSO posea el mayor valor de recall, puesto que su definición toma en cuenta cuatro tipos de relaciones semánticas (hiperonimia, meronimia, hiponimia y holonimia) y una léxica (antonimia) de PWN, en tanto que LCH sólo se basa en la hiperonimia.

La baja precisión del método Random, con respecto al resto de las medidas, está relacionada con el simple marcaje basado en la aleatoriedad de la asignación del valor de similitud.

El comportamiento anterior, es también notorio si se toman los valores del topline en vez de los del gold estándar (descartando las equivalencias establecidas por pertenencia a frases, incompletitud de MWN y gaps), como se observa en la tabla VIII.

La Figura 7 muestra la mejora en los valores de F-measure, comparando los resultados obtenidos con el topline y el gold estándar.

Si se colocan en el mismo gráfico la precisión y el recall, como se muestra en la Figura 8, se puede advertir que la precisión disminuye a medida que aumenta el recall, independientemente de la base (gold standard o topline). Para cuantificar la relación que existe entre estas medidas se ha determinado el coeficiente de correlación producto o momento de Pearson, r, un índice adimensional acotado entre -1 y 1 que refleja el grado de dependencia lineal entre dos conjuntos de datos.

$$r = \frac{\sum_{i=1}^{n}(x-\overline{x})^2(y-\overline{y})^2}{\sqrt{\sum_{i=1}^{n}(x-\overline{x})^2 \sum_{i=1}^{n}(y-\overline{y})^2}}$$

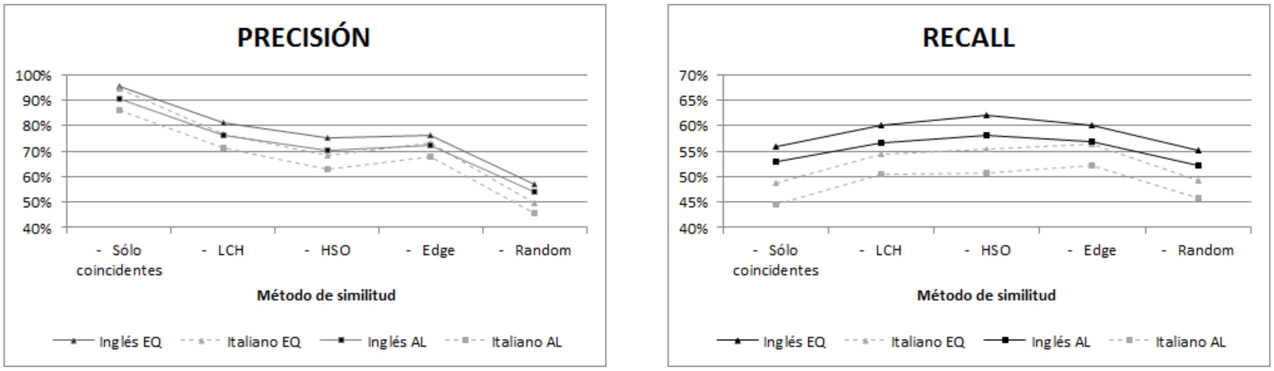
donde:

- n es el número de métodos de similitud empleados
- x el valor de precisión para el método i (conjunto de valores independientes)
- y el valor de recall para el método i (conjunto de valores dependientes)
- $\overline{x}$ e $\overline{y}$ son las medias de muestra promedio para el conjunto de valores independientes y dependientes respectivamente.

El resultado de aplicar el coeficiente de correlación a los valores de precisión y recall obtenidos en la tabla VII para la equivalencia son $r = -0{,}9675$ (inglés) y $r = -0{,}9497$ (italiano) y para la alineación $r = -0{,}9838$ (inglés) y $r =$

TABLA VII
EVALUACIÓN DE LOS RESULTADOS EN EL CORPUS DE PRUEBA

			Correctos		Precisión		Recall		F-measure	
Versión (Gold-estándar)	Método similitud	Total	EQ	AL	EQ	AL	EQ	AL	EQ	AL
español/inglés (389)	Sólo coincidentes	228	217	206	95.18 %	90.35 %	55.78 %	52.96 %	70.34 %	66.78 %
	LCH	289	234	220	80.97 %	76.12 %	60.15 %	56.56 %	68.94 %	64.88 %
	HSO	321	241	226	75.08 %	70.40 %	61.95 %	58.10 %	67.89 %	63.66 %
	Edge	307	234	221	76.22 %	71.99 %	60.15 %	56.81 %	67.24 %	63.51 %
	Random	377	214	203	56.76 %	53.85 %	55.01 %	52.19 %	55.87 %	53.01 %
español/italiano (333)	Sólo coincidentes	172	162	148	94.19 %	86.05 %	48.65 %	44.44 %	64.16 %	58.61 %
	LCH	236	181	168	76.69 %	71.19 %	54.35 %	50.45 %	63.62 %	59.05 %
	HSO	269	184	169	68.40 %	62.83 %	55.26 %	50.75 %	61.13 %	56.15 %
	Edge	257	188	174	73.15 %	67.70 %	56.46 %	52.25 %	63.73 %	58.98 %
	Random	332	164	152	49.40 %	45.78 %	49.25 %	45.65 %	49.32 %	45.71 %



(a)

(b)

Fig. 6. Valores de precisión y recall por versión, procedimiento y método de similitud

TABLA VIII
EVALUACIÓN DE LOS RESULTADOS EN EL CORPUS DE PRUEBA

			Correctos		Precisión		Recall		F-measure	
Versión (Topline)	Método similitud	Total	EQ	AL	EQ	AL	EQ	AL	EQ	AL
español/inglés (329)	Sólo coincidentes	221	217	206	98.19 %	93.21 %	65.96 %	62.61 %	78.91 %	73.96 %
	LCH	271	234	220	86.35 %	81.18 %	71.12 %	66.87 %	78.00 %	73.33 %
	HSO	299	241	226	80.60 %	75.59 %	73.25 %	68.69 %	76.75 %	71.98 %
	Edge	288	234	221	81.25 %	76.74 %	71.12 %	67.17 %	75.85 %	71.64 %
	Random	346	214	203	61.85 %	58.67 %	65.05 %	61.70 %	63.41 %	60.15 %
español/italiano (277)	Sólo coincidentes	165	162	148	98.18 %	89.70 %	58.48 %	53.43 %	73.30 %	66.97 %
	LCH	216	181	168	83.80 %	77.77 %	65.34 %	60.65 %	73.43 %	68.15 %
	HSO	236	184	169	77.97 %	71.61 %	66.43 %	61.01 %	71.74 %	65.89 %
	Edge	232	188	174	81.03 %	75.00 %	67.87 %	62.82 %	73.87 %	68.37 %
	Random	291	164	152	56.36 %	52.23 %	59.21 %	54.87 %	57.75 %	53.52 %

$-0{,}9194$ (italiano), lo que indica que estas medidas poseen un alto vínculo y son inversamente dependientes.

REFERENCIAS

[1] E. Macklovitch and M. Hannan, “Line ’em up: Advances in alignment technology and their impact on translation support tools,” *Machine Translation*, vol. 13, no. 1, pp. 41–57, 1998.

[2] C. . Nevill and T. Bell, “Compression of parallel texts,” *Information Processing and Management: an International Journal*, vol. 28, no. 6, pp. 781–793, 1992.

[3] J. Vera and G. Sidorov, “Proyecto de preparación del corpus paralelo alineado español-inglés,” in *Memorias del Encuentro Internacional de la Ciencias de la Computación*, Mexico, 2004.

[4] D. Tufiş, A. Barbu, and R. Ion, “Treq-al: a word alignment system with limited language resources,” in *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond*, Canada, 2003, pp. 36–39.

[5] R. Mihalcea and T. Pedersen, “An evaluation exercise for word alignment,” in *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Canada, 2003, pp. 1–10.

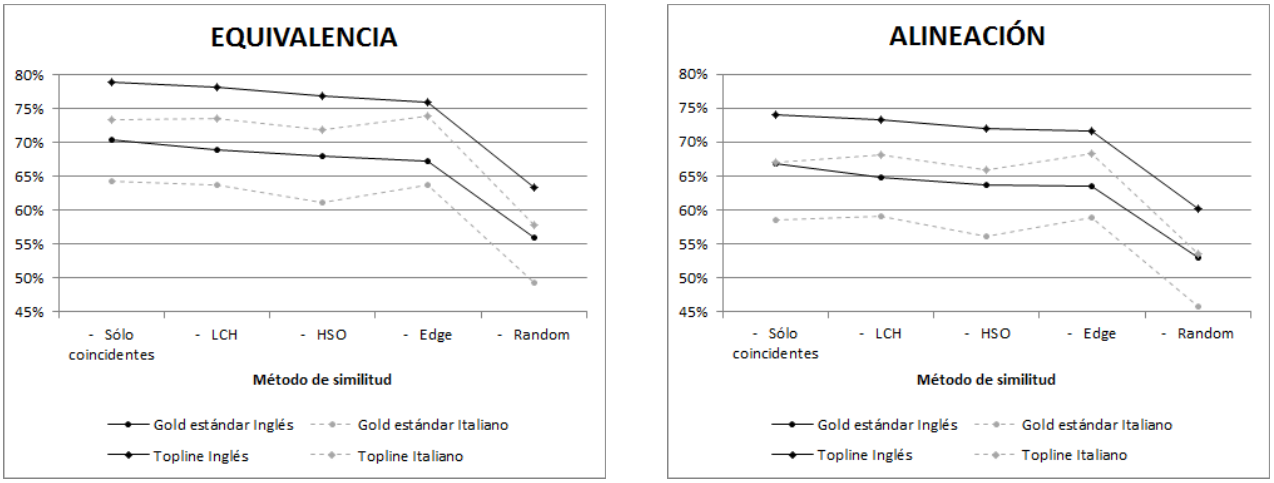
[6] C. Kit, J. Webster, H. P. K. Sin, and H. Li, “Clause alignment for bilingual hong kong legal texts with available lexical resources,” in *Proceedings of the 20th International Conference on Computer Processing of Oriental Languages*, China, 2003, pp. 286–292.

[7] W. Gale and K. Church, “Program for aligning sentences in bilingual corpora,” *Computational Linguistics*, vol. 19, no. 1, pp. 75–102, 1993.

[8] P. Brown, J. Lai, and R. Mercer, “Aligning sentences in parallel corpora,” in *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, EUA, 1991.

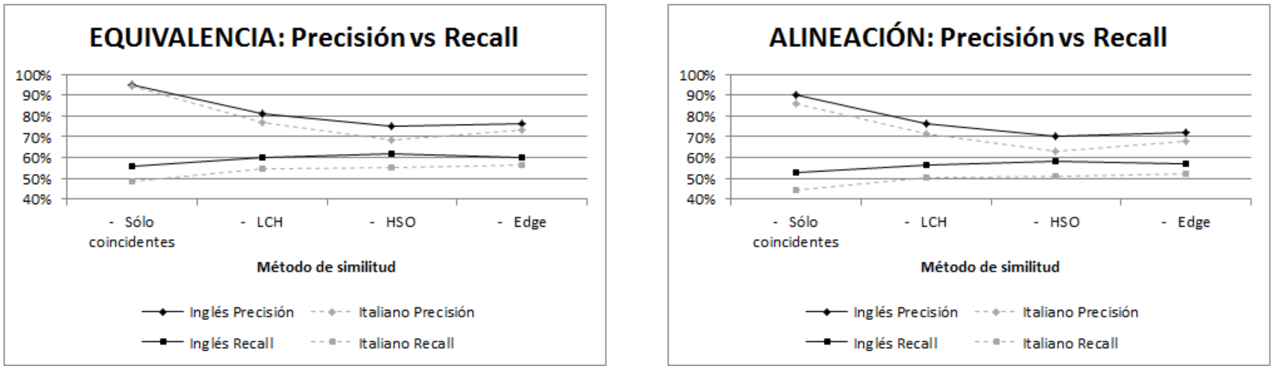
[9] D. Wu, “Aligning a parallel english-chinese corpus statistically with lexical criteria,” in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, EUA, 1994, pp. 80–87.

[10] A. Gelbukh, G. Sidorov, and J. Vera, “A bilingual corpus of novels aligned at paragraph level,” in *Proceedings of the 5th International*



(a) (b)

Fig. 7. Comparación de los valores de F-measure con base en el gold estándar y el topline



(a) (b)

Fig. 8. Comparación de los valores de precisión y recall con base en el gold estándar

Conference on NLP, Finlandia, 2006, pp. 16–23.

[11] M. Kay and M. Roscheisen, “Text-translation alignment,” *Computational Linguistics*, vol. 19, no. 1, pp. 121–142, 1993.

[12] S. Chen, “Aligning sentences in bilingual corpora using lexical information,” in *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, EUA, 1993, pp. 9–16.

[13] M. Haruno and T. Yamazaki, “High-performance bilingual text alignment using statistical and dictionary information,” in *Proceedings of the Annual Conference of the Association for Computational Linguistics*, EUA, 1996, pp. 131–138.

[14] M. Mikhailov, “Parallel corpus aligning: Illusions and perspectives,” *The Austrian Academy Corpus*, 2002.

[15] T. Tanaka and Y. Matsuo, “Extraction of translation equivalents from non-parallel corpora,” in *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation*, England, 1999, pp. 109–119.

[16] F. Smadja, V. Hatzivassiloglou, and K. McKeown, “Translating collocations for bilingual lexicons: A statistical approach,” *Computational Linguistics*, vol. 22, no. 1, pp. 1–38, 1996.

[17] M. Simard, G. Foster, and P. Isabelle, “Using cognates to align sentences in parallel corpora,” in *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, Canada, 1992, pp. 67–81.

[18] F. Debili and E. Sammouda, “Appariement des phrases de textes bilingues français-anglais et français-arabe,” in *Proceedings of the 14th Conference on Computational Linguistics*, Francia, 1992.

[19] P. Brown, S. Della, V. Della, and R. Mercer, “The mathematics of statistical machine translation: Parameter estimation,” *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[20] S. Vogel, H. . Ney, and C. Tillmann, “Hmm-based word alignment in statistical translation,” in *Proceedings of the 16th International Conference on Computational Linguistics*, Denmark, 1996, pp. 836–841.

[21] K. Sato and H. Saito, “Extracting word sequence correspondences with support vector machines,” in *Proceedings of the 19th international conference on Computational linguistics*, Taiwan, 2002, pp. 1–7.

[22] D. Tufis, “A cheap and fast way to build useful translation lexicons,” in *Proceedings of the 19th International Conference on Computational Linguistics*, Taiwan, 2002, pp. 1030–1036.

[23] S. Ker and J. Chang, “A class-based approach to word alignment,” *Computational Linguistics*, vol. 23, no. 2, pp. 313–343, 1997.

[24] D. Melamed, “Models of translational equivalence among words,” *Computational Linguistics*, vol. 26, no. 2, pp. 221–249, 2000.

[25] D. Hiemstra, “Deriving a bilingual lexicon for cross language information retrieval,” in *Proceedings of Gronics*, Netherlands, 1997, pp. 21–26.

[26] J. Kupiec, “An algorithm for finding noun phrase correspondences in bilingual corpora,” in *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, EUA, 1993, pp. 17–22.

[27] “Eurowordnet,” http://www.illc.uva.nl/EuroWordNet/, 2001, consultado 29/12/08.

[28] “Multiwordnet,” http://multiwordnet.itc.it/english/home.php, 2004, consultado 29/12/08.

[29] E. Pianta, L. Bentivogli, and C. Girardi, “Multiwordnet: developing an aligned multilingual database,” in *Proceedings of the First International Conference on Global WordNet*, India, 2002, pp. 21–25.

[30] C. Leacock and M. Chodorow, “Combining local context andwordnet similarity for word sense identification,” *WordNet: An electronic Lexical Database*, pp. 265–283, 1998.

[31] G. Hirst and D. St-Onge, “Lexical chains as representations of context for the detection and correction of malapropisms,” *WordNet: An electronic Lexical Database*, pp. 305–332, 1998.

[32] R. Rada, H. Mili, E. Bicknell, and M. Bletner, “Development and application of a metric on semantic nets,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 1, pp. 17–30, 1989.

[33] T. Pedersen, S. Patwardhan, and J. Michelizzi, “Wordnet::similarity - measuring the relatedness of concepts,” in *Proceedings of the 19th National Conference on Artificial Intelligence*, EUA, 2004, pp. 144–152.

[34] S. M. H.H. Do and E. Rahm, “Comparison of schema matching evaluations,” in *Proceedings of the GI-Workshop Web and Databases*, Erfurt, 2002, pp. 221–237.

Lexical Disambiguation of Arabic Language: An Experimental Study

Laroussi Merhben, Anis Zouaghi, and Mounir Zrigui

***Abstract*—In this paper we test some supervised algorithms that most of the existing related works of word sense disambiguation have cited. Due to the lack of linguistic data for the Arabic language, we work on non-annotated corpus and with the help of four annotators; we were able to annotate the different samples containing the ambiguous words. Since that, we test the Naïve Bayes algorithm, the decision lists and the exemplar based algorithm. During the experimental study, we test the influence of the window size on the disambiguation quality, the derivation and the technique of smoothing for the (2n+1)-grams. For these tests the exemplar based algorithm achieves the best rate of precision.**

***Index Terms*—Supervised algorithms, training data, Naïve Bayes, decision list, exemplar based algorithm, window size.**

I. Introduction

Human language is ambiguous; many words can have more than one sense that is dependent on the context of use. The word sense disambiguation (WSD) allows us to find the most appropriate sense of the ambiguous word.

The benefits of WSD were exploited by many NLP applications such as machine translation, information retrieval, grammatical analysis, speech processing as well as text processing.

The task of identifying the correct sense for the ambiguous word is not simple as it appears. What should be done to disambiguate a word? We must find a way to define the possible meanings of the word, since that we have to assign each occurrence of the ambiguous word to the appropriate sense.

In this work, we use the Naïve Bayes method, the decision lists and the exemplar based algorithm. These methods are based on a training phase during this part of work, we use an annotated training corpus (we extract from a non-annotated corpus the different samples containing the ambiguous word and we tag them with their senses).

Since that, a testing phase will classify a word into senses [1, 2]. In the most WSD works that was evaluated in the conference Semeval 2007, the supervised methods achieve the best disambiguation quality (about 80% precision and recall for coarse-grained WSD).

The paper is structured as follows. We describe in Section II how we tag the samples. After that, in Section III, we give a detailed account of the used supervised methods applied for the Arabic language. In Section IV, we present the results and discuss the difference with some related works in Section V. Finally, Section VI concludes the paper.

II. Related works

A. Review Stage

We can cite the work of Mona Diab that uses a supervised learning approach called "bootstrap" [15]. This approach is highly accurate in the average of 90% of the evaluated data items based on Arabic native judgment ratings and annotations. Also, we find the work of Elmougy [16], where the Naïve Bayes algorithm was applied for the Arabic language. Some pre-treatement steps were applied like word rooting and eliminating stopwords, since that they use the net and a dictionary to collect ten training samples to each word for the testing phase. This work achieves a rate of precision of 73%. Compared to our work the amount of data is more less, and collecting the testing samples from the net is a hard task and not sufficient.

Finally, Soha M. Eid [17] compared the Rocchio Classifier to Naïve Bayesian classifier, the most frequent sense and the support vector machine using arabic lexical samples. The Rocchio classifier achieves an overall accuracy of 88% as the best rate and reduces the error by over 14%. But they test only five ambiguous words and they haven’t explained how tagging the samples of the training phase.

Compared to our work we obtain a less rate of precision because of the important number of ambiguous tested words (fifteen ambiguous words). Also as a comparative study there is no test for the influence of the window size, the stemming and the smoothing on the quality of disambiguation.

For the other English related works, we can cite the experimental study that com-pares some supervised algorithms to disambiguate six senses of the word line [18] and [19]. Also the work of Pedersen where he compared the Naïve Bayes with Decision tree, Rule based learner, etc., to disambiguate the word line and 12 other words [20]. All these works, found that the Naïve Bayes algorithm performed as well as the other supervised algorithms, which is the same results founded in this work. Compared to the number of tested words by the English related works, we have to point that we test fifteen words for a derivational language that suffers from the lack of resources.

We can also compare the obtained results by some works of unsupervised Arabic word disambiguation, where the same samples and the same words were tested. In the first work

Manuscript received June 18, 2012. Manuscript accepted for publication July 24, 2012.

Laroussi Merhben Anis Zouaghi, and Mounir Zrigui are with the Unité de Recherche en Technologies de l'Information et de la Communication of the Réseau National Universitaire Tunisien, Tunisia (e-mail: aroussi_merhben@hotmail.com; Anis.Zouaghi@gmail.com; mounir.zrigui@fsm.rnu.tn).

[21], it was proposed to use some information retrieval measures with the Lesk algorithm and it achieves a rate of 73%. In the second one [22], a Context matching algorithm returns a semantic coherence score corresponding to the context of use that is semantically closest to the original sentence. This algorithm achieves a precision of 78%. In this work, we obtain a less rate of precision. We can presume that the supervised works are more satisfactory for the task of Arabic Word Sense Disambiguation.

III. Methodology

This study experiments some supervised methods for the Arabic Word Sense Disambiguation. It compares the use of the Naïve Bayes algorithm, the decision list and the k nearest neighbor. These methods were used previously in many related works (that will be discussed in the section 5).

We have applied some pre-processing steps to the words belonging to the original sentence and the training sets.

A. Pre-processing

1) Extraction of stopwords

Over the past ten years several methods have been proposed for the extraction of stopwords that have no influence on the meaning of a given word [5], [6] and [7]. These methods are used to evaluate the significance of a word in a document, which also varies depending on the frequency of the word in the corpus. Thus allowing us to eliminate the stopwords (words that have no influence on the meaning of the ambiguous word) such as (حتى, من, قد, بها, في, كان, له, فوق) (even, of, may, by, in, was, to him, over the).

These words will be removed from the sentence containing the ambiguous word, to decrease the number of compared words.

In this part of our work, we use the tf-idf metric [8] that use the term frequency (tf) and the inverse document frequency (idf) (see equation 1).

$$\text{Tf -idf}_{i,j} = \text{tf}_{i,j} \times \text{idf}_{i} \quad (1)$$

The frequency of a word (Term Frequency) is the number of occurrences of a word in a given document. Let the document dj and the word ti, the frequency of ti in dj is measured as follows (see equation 2).

$$\text{tf}_{i,j} = n_{i,j} / \sum_k n_{k,j} \quad (2)$$

Where $n_{i,j}$ is the number of occurrences of the word w_i in the document d_j. The denominator is the number of occurrences of all words in the document d_j.

The inverse document frequency gives the importance of a word in the corpus. It's measured as follows (see equation 3).

$$\text{idf}_{i} = log \frac{|D|}{|\{dj : ti \in dj\}|} \quad (3)$$

where | D | is the total number of documents in the corpus and $|\{d_j : t_i \in d_j\}|$ is the number of documents where the word t_i appears. We have to note, that the elimination of stopwords, will decrease the number of compared words in the testing phase.

2) Stemming

Each Arabic word, nouns or verbs, is based on three letters, or more rarely four or two letters. These three letters are the root of the Arabic word, they are the most important letters used to be compared with other letters used for the derivation of the word (added to the right or left of the root).

In this work, we use the Khoja stemmer that removes the longest suffix and the longest prefix. It then combines the remaining words with verbal and nominal patterns, to extract the root [9].

The stemming were applied to the words contained in the original sentence, to find there occurrences in the extracted training samples.

B. Tagging Samples

The supervised methods need a training phase that used a tagged corpus. The examples obtained by the training phase, must contain as many words surrounding the ambiguous words as it will be needed in the test phase. We chose to work on texts dealing with multiple domains (sport, politics, religion, science, etc.). These texts were recorded in the corpus of Latifa Al-Sulaiti [3].

Using this corpus, we tag the founded ambiguous words (used in the testing phase) by their senses, this step was achieved with the help of four annotators (Arabic language teachers), that choose the ambiguous words by the important number of senses out of context. Using the dictionary Lissan al arab [4] which is one of the most famous Arabic dictionary, we were able to tag the words with their corresponding senses.

We haven't found an important difference between the sense tags, the arrangement between the annotators is in the average of 95%. In table 1, we give the statistics of the extracted samples.

Table I
Distribution of the senses in the extracted samples.

Avg. # of words per sentence	Avg. # of senses per word	Avg. # of senses per ambiguous words	Avg. dominant sense for the ambiguous word
9,42	1,56	6,32	74%

Fifty words have been chosen. For each one of these ambiguous words, we evaluate 20 examples per sense. This number of words is judged as sufficient compared to the Senseval evaluation that put into practice 15 nouns, 13 verbs, 8 adjectives and 5 words that the grammatical tags wasn't taken into consideration. Totally there are 206 tests for every word.

C. Supervised methods

During the training phase, we tag the words surrounding the ambiguous word with $C_{i,j}$ (which is the local collocation that will indicate the position of two words given the ambiguous word). Let the ambiguous sentence is: " الفرار من الواقع إلى العلم والكتب هو سبيل اللقاء بين هذين العالمين المختلفين" "Escape from reality to science and books is the way of the meeting between these two different worlds".

Let “الكتب” “books” is the ambiguous word, in this sentence we can find 156 collocations. For example the collocation $C_{-2,-4}$: “الواقع_العلم” “fact _science”, and $C_{-2,\,3}$: “العلم_ اللقاء” “science_ meeting”.

For the original sentence, we define m features that correspond to the m collocations surrounding the ambiguous words. The supervised methods cited in what follows, will use these features for each sense of the ambiguous word.

1) The Naïve Bayes rithmalg

This method is one of the most popular and performant probabilistic method [10], it was used in different works of natural language processing including the word sense disambiguation. In fact the comparative works of Perdersen [11] found that the Naïve Bayes gives sufficient results compared to the other methods.

After tagging the samples containing the ambiguous word (AW) (section 3), we have to measure the probability of the collocations (F_j) contained in the same context of use of the ambiguous word for the sense (S) (see equation 4).

$$P = \sum_{j=1}^{m} \frac{\text{Number of occurrences of } F_j \text{ with the sense } S_i}{\text{number of occurrences of } F_j} \quad (4)$$

Where; N is the number of collocations in the original sentence.

This step is followed by the measure of the probability for each sense in the corpus (see equation5).

$$P(S_i) = \sum_{i=1}^{k} \frac{\text{Number of occurrences of AW with the sense Si}}{\text{Total number of occurrences of the ambiguous word}} \quad (5)$$

For the different collocations contained in the original sentence, we add the logarithm of the probability. The score is the sum of the obtained results (see equation 6).

$$\text{Score } (S_i) = \text{argmax}_{S_i\, \in\, senses(w)}\; log\; P(S_i) \prod_{j=1}^{m} P\ (F_j|S_i) \quad (6)$$

The sense with the highest score is the correct sense.

2) The decision List

The decision list algorithm was adopted for the WSD, in the work of Yarowsky [12]. In this part of our work, we need to compute the conditional probability of each sense for every collocation contained in the local context, p (Si | F1,F2,…, Fm) (see equation 7), for that we use :

- The probability of each observed collocation given the sense of the ambiguous word p ($F_1,F_2,\ldots, F_m |S_i$) (see equation 1),
- The probability of each sense in the corpus p(S_i) (see equation 5) ,
- The unconditional probability of features (collocations surrounding the ambiguous word) (see equation 7).

$$P = p\ (F_1,F_2,\ldots, F_m |S_i) \times p(S_i)\ /\ p\ (F_1,F_2,\ldots, F_m) \quad (7)$$

Since that to construct the decision list, we have to sort the different obtained results (given by the equation 7 for the different senses of the ambiguous word) using the log of the conditional probability of two compared senses for the tested word (see equation 8).

Finally the surrounding collocations that obtain the highest score, will be attributed to that sense, and will be ranked in the top of the decision list. After this step of classification, we will obtain an ordered list of S_i given the obtained score.

$$\text{Score } (w_i) = \text{Abs } (\log\ (P(S_1|w_i)\ /\ P(S_2|w_i))) \quad (8)$$

Given the score obtained in the decision list, we can judge the significance of the words contained in the original sentence.

***3)** The exemplar based (K Nearest Neighbor) algorithm*

The k nearest neighbor algorithm (KNN) is one of the highest performing methods in WSD [2]; [8]. The KNN algorithm is based on the k nearest similar instances to the tested instance. The classification phase is achieved by measuring the distance between the new example x = (F_1, $F_2,\ldots, F_m$), and the previously stored examples x_i = (F_{i1} , . . . , F_{im}), to do that in this work we use the hamming distance (see equation 9).

$$\Delta(x, x_i) = \sum_{j=1}^{m} d_j\ \delta\big(F_j, F_{ij}\big) \quad (9)$$

Where (d_j = Number of occurrences of the j^{th} collocation in the previously stored examples / Total number of collocations) and $\delta\big(F_j, F_{ij}\big)$ is 0 if F_j = F_{ij} and 1 otherwise. Since that, we can establish the set of the k most nearest examples. The most frequent sense between those k obtained samples is considered as the correct sense.

IV. Experimental Results

A. Encountered problems

Many problems have been encountered during the process of disambiguation cited in what follows:

- For the Naïve Bayes algorithm, we have the problem of the zero counts. As a solution, we replace the zero with P(S_k)/N, where N is the total size of the training sets. This solution is called smoothing.
- The important number of glosses given by a dictionary for the ambiguous word and the difference between the founded senses in the corpus. In the table 2, we give the number of senses for some words and their corresponding founded senses in the training corpus. As a solution, we try to collect from the net some texts containing the missing senses and add them to our corpus.
- Finding the samples for the tests (that can be judged effective and adequate for the process of disambiguation) is a hard task and differs between works for the obtained results.
- For some considered words, we have found senses that appear in the corpus and don’t exist in the dictionary. These senses were added to the list of candidate senses. For the word “ayn” we extract about ten sentences from the training corpus where it means a name of a city in United Arab Emirates. A sample is given in what follows:

“تستقبلنا مدينة العين ببهاء يختلف تماما عما ألفناه في أبوظبي” → “The ayn city receives us brightly completely different than abu-dhabi”.

– The difference between the number of occurrences in the corpus for each sense. For example for the word “كتب” “kataba”, we have found about 452 samples where the considered sense is “write” which is the most frequent sense and about 23 samples where the considered sense is “predestined”.

TABLE II
DIFFERENCE OF SENSES BETWEEN THE DICTIONARY AND THE EXTRACTED SAMPLES.

Ambiguous words	transcription	Number of senses in the dictionary	Number of senses in the extracted samples
حسب	hassaba	15	6
كتب	kataba	8	5
عين	ayn	20	8
شعر	chaar	8	4
عقل	aakl	18	6

1) *Obtained results*

To test the effectiveness and the impact of the different methods (presented in the previous section 4) on Arabic word sense disambiguation, we performed some experiments. In the work of Yarowsky [14], a study of the influence of the window size on WSD, shows that the most useful keywords for the WSD are included in a micro-context from six to eight words.

However, we have to point out that in a so large context; it is difficult to discern the key elements for determining the meaning of a word. It seems obvious that a fixed size of the context window is not adapted for all the words.

In order to solve this problem, we suggest determining the optimal size of the appropriate context for each test. Tests were conducted by measuring the performance (Precision) of each method varying the window size (the tested sentence containing the ambiguous word).

In Table 3, we give the rate of precision (Correct answers obtained / Obtained results) obtained using the bigram ($n = 2$), where we test only one word after or before the ambiguous word. In the case of trigram ($n = 3$) two words will be considered (one after the ambiguous word and another one before). Finally in the ($2n + 1$)-grams, we take into consideration n words surrounding the target word, in this experiments $n = 3$, because it give us the better results.

During the experimentation phase, we change the number of samples taken into consideration from the training phase. For example the 25% of samples means that we have taken into account 25% from the total number of samples that we have obtained in the training phase. The Figure 1 shows how the rate of precision varies across the percentage of samples. We conclude that the lowest rate of disambiguation is mainly due to the insufficient number of samples, which result in the failure to meet all possible events. For that we try to collect as many texts as we can, to extend the number of samples.

TABLE III
RESULTS OBTAINED BY DIFFERENT METHODS VARYING THE WINDOW SIZE.

Methods / tests	Naïve Bayes		Decision List		KNN	
	P	MFS	P	MFS	P	MFS
Bigram	23.04	29.17	21.43	25.29	26.33	30.21
Trigram	34.19	40.29	31.11	39.40	43.97	47.02
(2n+1)-grams	47.89	54.70	43.21	53.68	51.32	53.46

The rate of precision is increased for the most frequent sense, it is explained by the fact that the number of samples containing the most frequent sense is more important that the number of samples containing the other senses.

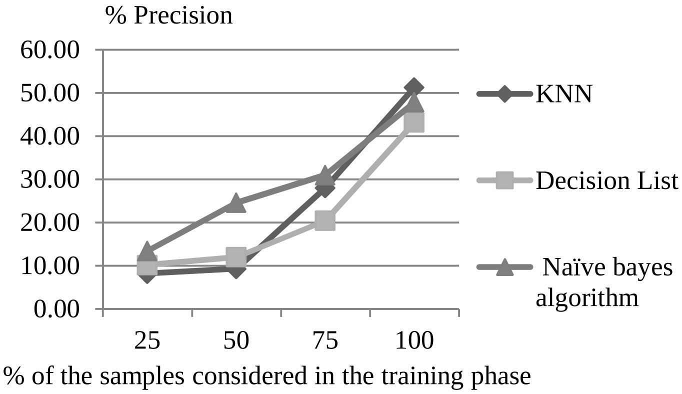


Fig. 1. Obtained results by the different algorithm depending on the amount of data considered in the training phase.

Since that during the step where we have to count the number of occurrences of each collocation contained in the original sentence, we take into consideration all the derivation of the words using the khoja stemmer. For example for the word “قرأ” “karaa” that occurs with the word “كتب” “kataba”, we have to count the number of occurrences of “قرأت, يقرؤون يقرأ تقرأ, قرأتنّ, قرأت, …” in the extracted samples.

We detail in Table 4, the rate of precision obtained with and without the stemming. The most supervised methods estimate the probability of each word using the context of the previous n-1 words. The problem of those methods is that unfortunately it assigns zero to n-grams that have not been observed, in the training phase. To avoid this problem we have to smooth the zero counts (see Section 4.1).

In Table 4, we give also the results of the different methods without and with the use of the smoothing for the (2n+1) grams experiment.

From the results cited in table 4, we find that the stemming increase the precision by a percentage that varies between 9% and 21% for the different methods. It was supposed that the smoothing will decrease the rate of precision, because it increases all probabilities for unseen words.

TABLE IV
RATE OF PRECISION OBTAINED BY CONSIDERING THE STEMMING AND THE SMOOTHING STEPS.

Tests Methods	Without stemming and smoothing	Without Stemming and with smoothing	With stemming and without smoothing	With smoothing and stemming
Naïve Bayes	31.25	31.75	47.89	48.23
Decision List	22.06	22.56	43.21	43.86
KNN	42.19	42.69	51.32	52.02

Also the smoothing increases the precision and recall of about 0.5%, and this increase is encouraging to perform the disambiguation quality.

As we have cited in the beginning of this section, the supervised methods needs a highly amount of data. The results varies from a word to another one and for the majority of the tested words the k nearest neighbor algorithm gives the better results and for some other words (“عين،كتب ،شعر” and their corresponding transcription is “ayn, kataba, chaar”) the Naïve Bayes algorithm is the best one. The rate of precision obtained by the decision list is in all the case is more less than the rate obtained by the Naïve Bayes algorithm. We may explain these results by the fact that the other supervised algorithm needs a so large training sets than the Naïve Bayes algorithm.

V. COMPARISON WITH SOME RELATED WORKS

We can cite the work of Mona Diab that uses a supervised learning approach called "bootstrap"[15]. This approach is highly accurate in the average of 90% of the evaluated data items based on Arabic native judgment ratings and annotations. Also, we find the work of Elmougy[16], where the Naïve Bayes algorithm was applied for the Arabic language. Some pretreatement steps were applied like word rooting and eliminating stopwords, since that they use the net and a dictionary to collect ten training samples to each word for the testing phase. This work achieves a rate of precision of 73%. Compared to our work the amount of data is more less, and collecting the testing samples from the net is a hard task and not sufficient.

Finally, Soha M. Eid [17] compared the Rocchio Classifier to Naïve Bayesian classifier, the most frequent sense and the support vector machine using arabic lexical samples. The Rocchio classifier achives an overall accuracy of 88% as the best rate and reduce the error by over 14%. But they test only five ambiguous words and they haven’t explain how tagging the samples of the training phase.

Compared to our work we obtain a more less rate of precision because of the important number of ambiguous tested words (fifteen ambiguous words). Also as a comparative study there is no test for the influence of the window size, the stemming and the smoothing on the quality of disambiguation.

For the other English related works, we can cite the experimental study that compares some supervised algorithms to disambiguate six senses of the word line [18] and [19]. Also the work of Pedersen where he compared the Naïve Bayes with Decision tree, Rule based learner, etc, to disambiguate the word line and 12 other words [20]. All these works, found that the Naïve Bayes algorithm performed as well as the other supervised algorithms, which is the same results founded in this work. Compared to the number of tested words by the English related works, we have to point that we test fifteen words for a derivational language that suffers from the lack of resources.

We can also compare the obtained results by some works of unsupervised Arabic word disambiguation, where the same samples and the same words were tested. In the first work [21], it was proposed to use some information retrieval measures with the Lesk algorithm and it achieves a rate of 73%. In the second one [22], a Context matching algorithm returns a semantic coherence score corresponding to the context of use that is semantically closest to the original sentence. This algorithm achieves a precision of 78%. In this work, we obtain a less rate of precision. We can presume that the supervised works are more satisfactory for the task of Arabic Word Sense Disambiguation.

VI. CONCLUSION

This paper has presented an experimental study of some supervised algorithms that were applied to perform word sense disambiguation in Arabic. These algorithms are based on tagged samples and a very important amount of data in the used corpus.

For a sample of fifty ambiguous Arabic words that are chosen by their number of senses out of contexts, the KNN achieves the best performance. We conclude that the supervised methods need an important amount of tagged data to achieve satisfactory results. We propose in future works to integrate some other resources and experiment some other supervised methods.

REFERENCES

[1] R. Mihalcea, “Word Sense Disambiguation Using Pattern Learning and Automatic Feature Selection”, in Journal of Natural Language and Engineering (JNLE), December 2002, p.p: 348–358.

[2] H. T. Ng and H. B. Lee, “Integrating multiple knowledge sources to disambiguate word senses: An examplar-based approach”. In Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA, 1996, p.p: 40–47.

[3] L. Al-Sulaiti, E. Atwell, “The design of a corpus of contemporary Arabic”. International Journal of Corpus Linguistics, vol. 11, 2006, pp. 135-171.

[4] M. Ben Mukarram and al-Ifriqi al-Misri ibn MANZUR, “ Lisàn al-'arab”, Ibn Manzûr, 15 volumes, 1956, Beyrout.

[5] J. Savoy, Y. Rasolofo, “Report on the TREC-11 Experiment: Arabic, Named Page and Topic Distillation Searches”. Eleventh Text Retrival Conference TREC, 2002.

[6] C. Fox, “A stop list for general text”. SIGIR Forum, 1990, Vol. 24, No. 1-2, pp. 19-35.

[7] A. Chen, F. Gey, translation Term Weighting and Combining Translation Resources in Cross-Language retrieval, Tenth text retrieval conference, 2001, TREC.

[8] S. Gerard, M.J. McGill, “Introduction to modern information retrieval”, ISBN: 0070544840, 1983, p.p: 448.

[9] K. Shereen and G. Roland, “Stemming Arabic text”, Computer Science Department, Lancaster University, Lancaster, UK, 1999.

[10] R. Navigili, “Word Sense Disambiguation: A Survey”. ACM Computing Surveys, Vol. 41, No. 2, Article 10, Publication date: February 2009.

[11] T. Pedersen, “Learning probabilistic models of word sense disambiguation”, Ph.D. dissertation. Southern Methodist University, Dallas, TX. 1998.

[12] D. Yarowsky, “Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French”. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (Las Cruces, NM), 1994, p.p: 88–95.

[13] A. Zouaghi, L. Merhbene, M. Zrigui, “Word Sense disambiguation for Arabic language using the variants of the Lesk algorithm”, WORLDCOMP’11, Las Vegas, juillet 2011, p.p. 561-567.

[14] D. Yarowsky, “One sense per collocation”. In Proceedings of the ARPA Workshop on Human Language Technology, Princeton,1993, pp. 266-7.

[15] M. Diab and P. Resnik, “An unsupervised method for word sense tagging using parallel corpora”. Proceedings of the ACL40th Meeting of the Association for Computational Linguistics, Philadelphia, U.S.A. 2002, pp. 255-262.

[16] S. Elmougy, H. Taher and H. Noaman “Naïve Bayes Classifier for Arabic Word Sense Disambiguation”. In proceeding of the 6th International Conference on Informatics and Systems, 2008, pp: 16-21.

[17] M. Soha Eid, et al., “Comparative Study of Rocchio Classifier Applied to supervised WSD Using Arabic Lexical Samples”. Proceedings of the tenth conference of language engeneering (SEOLEC’2010), Cairo, Egypt, December 15-16, 2010.

[18] C. Leacock, G. Towell and E. Voorhees, “Corpus based statistical sense resolution”. In Proceedings of the ARPA Workshop on Human Language Technology, 1993, p.p. 260-265.

[19] R.J. Mooney, “Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. Proceedings of EMNLP, 1996, p.p: 82-91.

[20] T. Pedersen, “Learning Probabilistic Models of Word Sense Disambiguation”. Ph.D. Dissertation. Southern Methodist University, 1998.

[21] A. Zouaghi, L. Merhbene and M. Zrigui, “Combination of information retrieval methods with LESK algorithm for Arabic word sense disambiguation”. Journal Article published in the Artificial Intelligence , Online First, 30 May 2011, Review; DOI: 10.1007/s10462-011-9249-3.

[22] L. Merhbene, A. Zouaghi and M. Zrigui, Ambiguous Arabic Words Disambiguation. In Proceeding of The 11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD’10), The University of Greenwich, London, United Kingdom, 9-11 June, 2010, p.p. 157-164.

A Hybrid Approach for Event Extraction

Anup Kumar Kolya, Asif Ekbal, and Sivaji Bandyopadhyay

***Abstract*—Event extraction is a popular and interesting research field in the area of Natural Language Processing (NLP). In this paper, we propose a hybrid approach for event extraction within the TimeML framework. Initially, we develop a machine learning based system based on Conditional Random Field (CRF). But most of the *deverbal* event nouns are not correctly identified by this machine learning approach. From this observation, we came up with a hybrid approach where we introduce several strategies in conjunction with machine learning. These strategies are based on semantic role-labeling, WordNet and handcrafted rules. Evaluation results on the TempEval-2010 datasets yield the precision, recall and F-measure values of approximately 93.00%, 96.00% and 94.47%, respectively. This is approximately 12% higher F-measure in comparison with the best performing system of SemEval-2010.**

***Index Terms*—About Event, TimeML, Conditional Random Field, TempEval-2010, WordNet.**

I. Introduction

TEMPORAL information extraction is, nowadays, a popular and interesting research area of Natural Language Processing (NLP). Generally, events are described in different newspaper texts, stories and other important documents where events happen in time and ordering of these events are specified. One of the important tasks of text analysis clearly requires identifying events described in a text and locating these in time. This is also important in a wide range of NLP applications that include temporal question answering, machine translation and document summarization.

In the literature, relation identification based on machine learning approaches can be found in [1, 2, 3] and some of the TempEval-2007 participants [4]. Most of these works tried to improve classification accuracies through feature engineering.

The performance of any machine learning based system is often limited by the amount of available training data. Mani *et al*. [2] introduced a temporal reasoning component that greatly expands the available training data. The training set was increased by a factor of 10 by computing the closures of the various temporal relations that exist in the training data. They reported significant improvement of the classification accuracies on event-event and event-time relations. However, this has two shortcomings, namely feature vector duplication caused by the data normalization process and the unrealistic evaluation scheme. The solutions to these issues are briefly described in [5]. In TempEval-2007 task, a common standard dataset was introduced that involves three temporal relations. The participants reported F-measure scores for event-event relations ranging from 42% to 55% and for event-time relations from 73% to 80%.

In TempEval-2007, the event-event relations were not considered discourse-wide like [2, 5]. Here, the event-event relations are restricted to events within the maximum of two consecutive sentences. Thus, these two frameworks produce highly dissimilar results for solving the problem of temporal relation classification.

One most common trend to apply machine learning algorithm for temporal information extraction is to formulate temporal relation as an event paired with a time or another event, and to transform these into a set of feature values. In most of the previous attempts, researchers have used some popular machine learning techniques like Naive-Bayes, Decision Tree (C5.0), Maximum Entropy (ME) and Support Vector Machine (SVM). Machine learning techniques alone cannot always yield good accuracies. In order to achieve reasonable accuracy, some researchers [6] used hybrid approach, where a rule-based component was added with machine learning. The system [6] was designed in such a way that they can take the advantage of rule-based as well as machine learning during final decision making. But, for a given instance, whether machine learning or rule-based component will be used, was not explained. They used either of the components in different situations in order to enjoy the advantage of the both the components.

In this work, we propose a hybrid approach for event extraction from the text under the TempEval-2010 framework. Initially, we develop a method for event extraction based on machine learning. We use Conditional Random Field (CRF) as the underlying machine learning algorithm. We observe that this machine learning based system often makes the errors in extracting the events denoted by *deverbial* entities. This observation prompts us to employ several strategies in conjunction with machine learning. These strategies are implemented based on semantic role labeling, WordNet and handcrafted rules. We experiment with the TempEval-2010 evaluation challenge setup [7]. Evaluation results yield the precision, recall and F-measure values of approximately 93.00%, 96.00% and 94.47%, respectively. This is approximately 12% higher F-measure in comparison to the best system [8] of TempEval-2010.

We use semantic role labels for event nominalizations. Events can be analyzed by these kinds of nominalizations. As our goal is on nominal Semantic Role Label (SRL), we concentrate on the event/target/results class. SRL for nominalization represents semantic roles to extract high level information that are more independent from the word tokens. On the other hand on verbal SRL [9, 10] there is relatively little work that specifically addresses nominal SRL. Nouns are generally treated like verbs. The task is split into two

Manuscript received November 2, 2010. Manuscript accepted for publication January 15, 2011.

A. Kolya, A. Ekbal, and S. Bandyopadhyay are with the Computer Science and Engineering Department , Jadavpur University, Kolkata; A. Ekbal is also with Computer science and engineering department,IIT Patna, India (e-mail: anup.kolya@gmail.com,asif.ekbal@gmail.com, sivaji_cse_ju@yahoo.com).

classification steps, argument recognition (telling arguments from non-arguments) and argument labelling (labelling recognized arguments with a role). Nominal SRL also typically draws on feature sets that are similar to those for verbs, i.e. comprising mainly syntactic and lexical-semantic information [11]. On the other hand, there is converging evidence that nominal SRL is somewhat more difficult than verbal SRL.

Hence, semantic roles may aid in learning a more general model. This learning model could improve the results of the approaches that are solely focused on lower-level information. Two frameworks for semantic roles have found wide use in the community, PropBank [12] and FrameNet [13]. Their corpora are used to train supervised models for semantic role labelling of new text [9][14]. The resulting analysis can benefit a number of applications, such as Information Extraction [15] or Question Answering [16].

The rest of the paper is structured as follows. Section 2 describes our Conditional Random Field (CRF) based event extraction approach. We describe our event extraction approaches with the use of semantic roles in Section 3, WordNet in Section 4 and hand-crafted rules in Section 5. Evaluation results under the experimental set up of TempEval-2010 evaluation challenge are reported in Section 6. Finally, Section 7 concludes the paper

II. CRF BASED APPROACH FOR EVENT EXTRACTION

Conditional Random Field (CRF) [17] is an undirected graphical model, which is a special case of which corresponds to conditionally trained probabilistic finite state automata. Being conditionally trained, these CRFs can easily incorporate a large number of arbitrary, non-independent features while still having efficient procedures for non-greedy finite-state inference and training. CRFs have shown success in various sequence modeling tasks including noun phrase segmentation [18] and table extraction [19]. The main advantage of CRF comes from that it can relax the assumption of conditional independence of the observed data often used in generative approaches, an assumption that might be too restrictive for a considerable number of object classes. Additionally, CRF avoids the label bias problem.

CRF is used to calculate the conditional probability of values on designated output nodes given values on other designated input nodes. The conditional probability of a state sequence $S =< s_{1,} s_{2,...,} s_T >$ given an observation sequence $O =< o_1, o_{2,}....., o_T)$ is calculated as:

$$P_\Lambda(s \mid o) = \frac{1}{Z_o}\exp(\sum_{t=1}^{T}\sum_{k=1}^{K}\lambda_k f_k(s_{t-1}, s_{t,} o, t))$$

where, $f_k(s_{t-1}, s_{t,} o, t)$ is a feature function whose weight λ_k is to be learned via training. The values of the feature functions may range between $-\propto .....+\propto$, but typically they are binary. To make all conditional probabilities sum up to 1, we must calculate the normalization factor,

$$Z_0 = \sum_s \exp(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(s_{t-1}, s_{t,} o, t)),$$

This, as in HMMs, can be obtained efficiently by dynamic programming.

Here, the CRF parameters are optimized using Limited-memory BFGS[16], a quasi-Newton method that is significantly more efficient, and results in only minor changes in accuracy due to changes in σ . CRFs generally can use real-valued functions but it is often required to incorporate the binary valued features. A feature function $f_k(s_{t-1}, s_{t,} o, t)$ has a value of 0 for most cases and is only set to 1, when s_{t-1}, s_t are certain states and the observation has certain properties. We have used the C^{++} based CRF++ package[1], a simple, customizable, and open source implementation of CRF for segmenting /labeling sequential data.

A. Features of CRF

We extract the gold-standard TimeBank features for events in order to train/test the CRF model. In the present work, we mainly use the various combinations of the following features:

(i). **Part of Speech (POS) of event terms**: It denotes the POS information of the event. The features values may be either of ADJECTIVE, NOUN, VERB, and PREP.

(ii). **Event Tense**: This feature is useful to capture the standard distinctions among the grammatical categories of verbal phrases. The tense attribute can have values, PRESENT, PAST, FUTURE, INFINITIVE, PRESPART, PASTPART, or NONE.

(iii). **Event Aspect**: It denotes the aspect of the events. The aspect attribute may take values**,** PROGRESSIVE, PERFECTIVE and PERFECTIVE PROGRESSIVE or NONE.

(iv). **Event Polarity**: The polarity of an event instance is a required attribute represented by the boolean attribute, polarity. If it is set to ’NEG’, the event instance is negated. If it is set to ’POS’ or not present in the annotation, the event instance is not negated.

(v). **Event Modality**: The modality attribute is only present if there is a modal word that modifies the instance.

(vi). **Event Class**: This is denoted by the ‘EVENT’ tag and used to annotate those elements in a text that mark the semantic events described by it. Typically, events are verbs but can be nominal also. It may belong to one of the following classes:

REPORTING: Describes the action of a person or an organization declaring something, narrating an event, informing about an event, etc.

PERCEPTION: Includes events involving the physical perception of another event. Such events are typically expressed by verbs like: *see*, *watch*, *glimpse*, *behold*, *view*, *hear*, *listen*, *overhear* etc.

ASPECTUAL: Focuses on different facets of event history.

[1] http://crfpp.sourceforge.net

I_ACTION: An intentional action. It introduces an event argument which must be in the text explicitly describing an action or situation from which we can infer something given its relation with the I_ ACTION.

I_STATE: Similar to the I_ACTION class. This class includes states that refer to alternative or possible words, which can be introduced by subordinated clauses, nominalizations, or untensed verb phrases (VPs).

STATE: Describes circumstances in which something obtains or holds true.

Occurrence: Includes all of the many other kinds of events that describe something that happens or occurs in the world.

Event Stem: It denotes the stem of the head event.

III. USE OF SEMANTIC ROLES FOR EVENT EXTRACTION

We use Semantic Role Label (SRL)[9] [20] to identify different features of the sentences of a document. These features help us to extract the events from the text. For each predicate in a sentence acting as event word, semantic roles extract all constituents, determining their arguments (agent, patient, etc.) and their adjuncts (locative, temporal, etc.). Some of the others features like predicate, voice and verb sub-categorization are shared by all the nodes in the tree. In the present work, we use predicate as an event. Semantic roles can be used to detect the events that are the nominalizations of verbs such as *agreement* for *agree* or *construction* for *construct*. Event nominalizations often afford the same semantic roles as verbs, and often replace them in written language [21]. Nominalisations (or, *deverbal nouns*) are commonly defined as nouns, morphologically derived from verbs, usually by suffixation [22]. They can be classified into at least three categories in the linguistic literature, event, result, and agent/patient nominalisations [23]. Event and result nominalisations constitute the bulk of *deverbal* nouns. The first class refers to an event/activity/process, with the nominal expressing this action (e.g., killing, destruction etc.). Nouns in the second class describe the result or goal of an action (e.g., agreement, consensus etc.). Many nominals have both an event and a result reading (e.g., selection). A smaller class is agent/patient nominalizations that are usually identified by suffixes such as *-er*, *-or* etc., while patient nominalisations end with *-ee*, *-ed* (e.g. employee). Let us consider the following example sentence to understand how semantic roles can be used for event extraction.

All sites were inspected to the satisfaction of the inspection team and with full cooperation of Iraqi authorities, Dacey said.

The output of SRL for this sentence is as follows:

[ARG1 All sites] were [TARGET inspected] to the satisfaction of the inspection team and with full cooperation of Iraqi authorities, [ARG0 Dacey] [TARGET said]

The sentence is traversed to find the argument-target relations. A sentence is scanned as many times as the number of target words in the sentence. In the first traversal, *inspected* is identified as the event. In the second pass, *said* is identified as an event. All the extracted target words are treated as the event words. We observed that many of these target words are identified as the event expressions by the CRF model. But, there exists many nominalised event expressions (i.e., *deverbal nouns*) that are not identified as events by the supervised CRF. These nominalised expressions are correctly identified as events by SRL. We observe performance improvement with the inclusion of this module.

IV. USE OF WORDNET FOR EVENT EXTRACTION

WordNet [23] features have been widely used to extract different lexical categories, such as *part-of-speech (POS)*, *stem*, *hypernym*, *meronym*, *distance* and *common-parents*, and demonstrated its worth in many tasks. Here, WordNet is mainly used to identify *non-deverbal event nouns*. We observed from the outputs of CRF and SRL that the event entities like '*war*', '*attempt*', '*tour*' etc. are not properly identified. These words have noun (NN) POS information, and the previous approaches, i.e. CRF and SRL can only identify those event words that have verb (VB) POS information. We know from the lexical information of WordNet that the words like '*war*' and '*tour*' are generally used as both *noun* and *verb* forms in the sentence. We design two following rules based on the WordNet:

Rule 1: The word tokens having Noun (NN) PoS categories are looked into the WordNet. If it appears in the WordNet with noun and verb senses, then that word token is also considered as an event. For example, *war* has both noun and verb senses in the WordNet, and thus considered as an event.

Rule 2: The *stems* of the noun word tokens are looked into WordNet. If one of the WordNet senses is verb then the token will be identified as verb. For example, the stem of *proposal*, i.e. *propose* has two different senses, noun and verb in the WordNet, and thus it is considered as an event.

We observe significant performance improvement on event extraction with the above mentioned two rules.

V. USE OF RULES FOR EVENT EXTRACTION

We used WordNet to extract the event expressions that appear in the WordNet with both noun and verb senses. Here, we mainly concentrate to identify the specific lexical classes like '*inspection*' and '*resignation*'. These can be identified by the suffixes such as ('*-ción*'), ('*-tion*') or ('*-ion*'), i.e. the morphological markers of deverbal derivations.

Initially, we run the CRF based Stanford Named Entity (NE) tagger[2] on the TempEval-2 test dataset. The output of the system is tagged with *Person*, *Location*, *Organization* and *Other* classes. The words starting with the capital letters are

[2] http://nlp.stanford.edu/software/CRF-NER.shtml

also considered as NEs. Thereafter, we came up with the following rules for event extraction:

Cue-1: Nouns which are morphologically derived from verbs are commonly distinguished as nominalizations (or, deverbal nouns). The deverbal nouns are usually identified by the suffixes like '*-tion*', '*-ion*', '*-ing*' and '*-ed*' etc. The nouns that are not NEs, but end with these suffixes are considered as the event words.

Cue 2: The verb-noun combinations are searched in the sentences of the test set. The non-NE noun word tokens are considered as the events.

Cue 3: Nominals and non-deverbal event nouns can be identified by the complements of aspectual PPs headed by prepositions like *during*, *after* and *before*, and complex prepositions such as *at the end of* and *at the beginning of* etc. The next word token(s) appearing after these clue word(s)/phrase(s) are considered as events.

Cue 4: The non-NE nouns occurring after the expressions such as *frequency of*, *occurrence of* and *period of* are most probably the event nouns.

Cue 5: Event nouns can also appear as objects of aspectual and time-related verbs, such as *have begun a campaign* or *have carried out a campaign* etc. The non-NEs that appear after the expressions like "*have begun a*", "*have carried out a*" etc. are also most probably the events.

VI. EVALUATION RESULT

We use the TempEval-2010 datasets to report the evaluation results. We start with the development of a CRF based system. We develop a number of CRF models depending upon the various features included into it. We have a training data in the form W_i, T_i, where, W_i is the i^{th} pair along with its feature vector and T_i is its corresponding output label (i.e., *Event* or *Other*). Models are built based on the training data and the feature template. The procedure of training is summarized below:

1. Define the training corpus, C.
2. Extract the <*token, output*> relations from the training corpus.
3. Create a file of candidate features derived from the training corpus.
4. Define a feature template.
5. Compute the CRF weights λ_k for every f_K using the CRF toolkit with the training file and feature template as input.
6. Derive the best feature template depending upon the performance.
7. Select the best feature template obtained from Step 6.
8. Retrain the CRF model

We use various subsets of the template as shown in Figure 1 during our experiment. In the figure, w_i : Current <*token, output*> pair, $w_{(i-n)}$: Previous nth pair, $w_{(i+n)}$: Next nth pair, $t_{i-1:}$ previous pair.

The test data had 373 verbal and 125 non-deverbal event nouns. Overall evaluation results are reported in Table 1. The CRF based system shows the precision, recall and F-measure values of 75.3%, 78.1% and 76.87%, respectively. The performance increases by 1.39 percentage F-measure points with the use of semantic roles. Table shows very high performance improvement (i.e., 11.01%) with the use of WordNet. The rule-based component also shows the effectiveness with the improvement of 5.20 F-measure percentage points. Finally, the system achieves the precision, recall and F-measure values of 93.00%, 96.00% and 94.47%, respectively. This is actually an improvement of approximately 12% F-measure value over the best reported system [8].

$w_{(i-2)}$
$w_{(i- 1)}$
w_i
w_{i+1}
$w_{(i+2)}$
Combination of w_{i-1} and w_i
Combination of w_i and w_{i+1}
Dynamic output tag (t_i) of the previous token
Feature vector of w_i of other features

Figure 1: Feature template used for the experiment

TABLE 1.
EVALUATION RESULTS OF EVENT EXTRACTION (PERCENTAGES)

Model	precision	Recall	F-measure
CRF	75.30	78.10	76.87
CRF+SRL	76.60	80.00	78.26
CRF+SRL+WordNet	88.56	90.00	89.27
CRF+SRL+WordNet+Rules	93.00	96.00	94.47

VII. CONCLUSION

In this paper, we have reported our work on event extraction under the TempEval -2010 evaluation exercise. Initially, we developed a CRF based supervised system for event extraction. This CRF based systems suffer mostly in identifying the deverbal nouns that denote the event expressions. Thereafter, we came up with several proposals in order to improve the system performance. We proposed a number of techniques based on SRL, WordNet and handcrafted rules. Evaluation results yield the precision, recall and F-measure values of 93.00%, 96.00% and 94.47%, respectively. This is an improvement of approximately 12 percentage F-measure points over the best performing system of TemEval-2010 evaluation challenge.

Future works include the identification of more precise rules for event identification and multiword events. Future works also include experimentations with other machine learning techniques like maximum entropy and support vector machine.

ACKNOWLEDGEMENTS

The work was partially supported by a grant from English to Indian language Machine Translation (EILMT) funded by Department of Information and Technology (DIT), Government of India.

REFERENCES

[1] Boguraev, B. and R. K. Ando. 2005. TimeMLCompliant Text Analysis for Temporal Reasoning. *Proceedings of Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pp. 997–1003, Edinburgh, Scotland.

[2] Mani, I., Wellner, B., Verhagen, M., Lee C.M., Pustejovsky, J. 2006. Machine Learning of Temporal Relation. *Proceedings of the 44th Annual meeting of the Association for Computational Linguistics*, Sydney, Australia.

[3] Chambers, N., S. Wang, and D. Jurafsky. 2007. Classifying Temporal Relations between Events. *Proceedings of the ACL 2007 Demo and Poster Sessions*, pp. 173–176, Prague, Czech Republic.

[4] Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., Pustejovsky, and J.: SemEval-2007 Task 15: TempEval Temporal Relation Identification. *Proceedings of the 4th International Workshop on Semantic Evaluations (semEval-2007)*, pp. 75-80, Prague.

[5] Mani, I., B. Wellner, M. Verhagen, and J. Pustejovsky. 2007. Three Approaches to Learning TLINKs in TimeML. *Technical Report CS-07-268, Computer Science Department, Brandeis University*, Waltham, USA.

[6] Mao, T., Li. T., Huang, D.,Yang,Y. 2006. Hybrid Models for Chinese Named Entity Recognition. *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing.*

[7] A. Kolya, A. Ekbal and S. Bandyopadhyay. 2010e. JU_CSE_TEMP: A First Step towards Evaluating Events, Time Expressions and Temporal Relations. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, ACL 2010, July 15-16, Sweden, PP. 345–350.

[8] Hector Llorens, Estela Saquete, Borja Navarro.2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles. *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010*, pages 284–291, Uppsala, Sweden, 15-16 July 2010.

[9] Gildea, D. and D. Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245–288.

[10] Fleischman, M. and E. Hovy. 2003. Maximum Entropy Models for FrameNet Classification. *Proceedings of EMNLP*, pages 49–56, Sapporo, Japan.

[11] Liu, C. and H. Ng. 2007. Learning Predictive Structures for Semantic Role Labeling of NomBank. *Proceedings of ACL*, pages 208–215, Prague.

[12] Palmer, M., D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

[13] Fillmore, C., C. Johnson, and M. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16:235–250.

[14] Carreras, X. and L. M`arquez, editors. 2005. Semantic Role Labeling. *Proceedings of the CoNLL-05 Shared Task*.

[15] Moschitti, A., P. Morarescu, and S. Harabagiu. 2003. Open-domain information extraction via automatic semantic labeling. *Proceedings of FLAIRS*, pages 397–401, St. Augustine, FL.

[16] Frank Schilder, Graham Katz, and James Pustejovsky. 2007. Annotating, Extracting and Reasoning About Time and Events (Dagstuhl 2005), *volume 4795 of LNCS. Springer*.

[17] Lafferty, J., McCallum, A., Pereira, F. 2001 Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of 18th International Conference on Machine Learning*, pp.282-289.

[18] Sha, F., Pereira, F. 2003. Shallow Parsing with Conditional Random Fields. *Proceedings of HLT-NAACL*.

[19] Pinto, D., McCallum, A.,Wei, X., Croft, W.B. 2003. Table Extraction Using Conditional Random Fields. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 235-242.

[20] Sameer S. Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, Daniel Jurafsky, Shallow Semantic Parsing using Support Vector Machines. *Proceedings of the Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL-2004*), Boston, MA, May 2-7, 2004.

[21] Gurevich, O., R. Crouch, T. King, and V. de Paiva.2006. Deverbal Nouns in Knowledge Representation. *Proceedings of FLAIRS*, pages 670–675, Melbourne Beach, FL.

[22] Quirk, R., S. Greenbaum, G. Leech, and J. Svartvik. 1985. A Comprehensive Grammar of the English Language. *Longman*.

[23] Grimshaw, J. 1990. Argument Structure. MIT Press.

[24] George A. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4): 235–312

Graphical Description of Soft Fault on Manufacturing Systems Using FDI Strategy: a SCL Approach

E. Lebano-Perez, C. A. Gracios-Marin, J. F. Guerrero-Castellanos and G. A. Munoz-Hernandez

***Abstract*—This work shows the benefits of a virtual graphical environment to model soft faults behavior in the resources of typical manufacturing processes applying Fuzzy Filter Time Series. It is shown in this work that using the programming tool called Scheduling Control Language (SLC), it is possible to improve the level of abstraction to introduce non-deterministic characteristics in the structural and functional description for each resource and the whole definition for the model. A graphical representation is proposed to generate an on-line platform as a virtual manufacturing laboratory. This instrument will be useful for academic, research and industrial applications. This tool can be used for validating and evaluating models, simulation of scheduling tasks and verification of control algorithms.**

***Index Terms*—Graphics utilities, Virtual device interfaces, Virtual instrumentation**

I. Introduction

The design in rapid prototyping approach for modeling, simulation, fault analysis and control of flexible manufacturing systems, have interested to academic and industrial communities in the last 10 years. Several solutions have been proposed to design using rapid prototyping; however these solutions have not contemplated the insertion of intelligence in the elements and devices included in the process.

Besides, requirements such as: the active interaction of the human with software and hardware, agility, fault tolerance and the adaptability in general, are decisive characteristics that any intelligent manufacturing system should satisfy [1].

Therefore, new methods and tools for designing, simulating and controlling, must include a unified modeling language that allow a direct translation between the parameters of the process and the different strategies of intelligent control.

The present work begins with a brief description of modeling, simulation and control of manufacturing systems. After that, fundamental theorems for modeling manufacturing systems using C.A.D. environments are discussed. Subsequently, the use of SCL language to describe structural and behavioral is explained. Finally, conclusions are drawn.

II. Fundamental Theory

When the standard of "Intelligent" is desired to be accomplished current manufacturing processes must achieve new requirement of high performance characteristics, such as: interoperability, open and dynamic structures, interaction between human beings, software and hardware and fault tolerance [1].

The Integration of human beings with software and hardware means that people and computers need to be incorporated to work collectively at various stages of the developed product, and even the whole product life cycle with rapid access to require knowledge and information. Heterogeneous sources of information must then be integrated to support these needs and to enhance the decision capabilities of the system. Bi-directional communication environments are required to allow effective and quick communication between human beings and computers to facilitate the interaction.

In the case of Fault Tolerance requirement, the system should be fault tolerant in both cases: at the system level and at the subsystem level. This characteristic is important to be able to detect and recover from system failures and minimize the impact on the working environment. When the two requirements are needed to be verified in a Manufacturing Process, it is adequate to use the Computer Environment in modeling, simulating and Model Based Direct Control Generating [2].

In many applications in which a malfunction of the system can cause significant losses or even cause danger to the environment or human life, a fault analysis model is required to evaluate the performance in order to anticipate possible faults. Examples of such areas are material transporting, process controlling and instrumentation. The systems used in such or similar application areas are expected to exhibit always an acceptable behavior. This system property is often referred as “dependability”. Any deviation from the acceptable behavior is considered a system failure. Failures are caused by faults, which can arise in different phases of the process system lifecycle [3].

By definition, a fault represents an unexpected change of system function, although it may not represent a physical failure. The term failure indicates a serious breakdown of a

Manuscript received October 10, 2011. Manuscript accepted for publication November 20, 2012.

E. Lebano-Perez is with Universidad Popular Autonoma del Estado de Puebla, 21 sur 1103 Barrio Santiago, Puebla, México C.P. 72410 (phone: +52-222-229-94-00; e-mail: eduardo.lebano@upaep.mx).

C.A. Gracios-Marin and J.F. Guerrero-Castellanos are with the Benemerita Univerisdad Autonoma de Puebla. C. U. Av. San Claudio and 18 Sur. Puebla, Pue. Mexico C.P. 72570 (phone: +52-222-229-55-00 ext. 7400; e-mail: cgracios@ece.buap.mx; fguerrero@ece.buap.mx).

G.A. Munoz-Hernandez is with Instituto Tecnologico de Puebla, Av. Tecnologico 420 Puebla, Mexico C.P. 72220 (phone: +52-222-229-88-24; fax: +52-222-222-21-14; e-mail: gmunoz@ieee.org).

system component or function that leads to a significantly deviated behavior of the whole system. The term fault rather indicates a malfunction that does not affect significantly the normal behavior of the system [4].

An incipient (soft) fault represents a small and often slowly developing continuous fault. The effects on the system are in the early stages almost unnoticeable. A fault is called hard or abrupt if its effects on the system are larger and bring the system very close to the limit of acceptable behavior. A fault is called intermittent if its effects on the system are hidden for discontinuous periods of time. Although a fault is tolerable at the moment that it occurs, it must be diagnosed as early as possible because it may lead to serious consequences after some time [5].

A fault diagnosis system is a monitoring system that is used to detect faults and diagnose their location and significance in a system. The system performs the following tasks:

- Fault detection: to indicate if a fault occurred or not in the system.
- Fault isolation: to determine the location of the fault.
- Fault identification: to estimate the size and nature of the fault.

The first two tasks of the system -fault detection and isolation are considered the most important.

Fault diagnosis is often named as Fault Detection and Isolation (FDI). A fault -tolerant scheme is a robust system that continues to operate acceptably after faults are presented in the system or in the controller. An important feature of such system is the automatic reconfiguration, once a malfunction is detected and isolated. Fault diagnosis contributes to develop a fault-tolerant control system, which will be capable of detecting and isolating the faults in order to decide how to perform reconfiguration [6].

Using a modeling Soft Fault Graphical Environment, the model based fault diagnosis can be defined as the determination of the faults in a system by comparing available system measurements with a priori information represented by an analytical/mathematical model system, through generation of residuals quantities and their analyses [6].

When an analytical model is used to represent any system under diagnosis, it cannot perfectly model uncertainties due to disturbances and noise. The differences provoked by the non-complete description of the model, cause residual values, which are instruments to indicate faults. Vasile [5] and Lakhmi [6] presented a robust FDI scheme that provides satisfactory sensitivity to faults, while being robust (insensitive or even invariant) to modeling uncertainties.

The principal challenge in designing a robust FDI scheme is to make it able to diagnose incipient faults. The effects of an incipient fault on a system are almost unnoticeable in the beginning, thus effects of uncertainties on the system could hide these small effects. Gracios et al [7] presented a work where any Manufacturing System can be described as a Discrete Event System (DES) because when digital sensors and actuators are included in each resource, using this approach typical methods of modeling Discrete Event System can be used to model, simulate and obtain the control for the system. Therefore it is possible to include faults in the activity or availability for the resources such as Robots, C.N.C., Conveyor Belts, etc.

Cassandras [8] has proposed Three Levels of Abstraction in the Study of Discrete Event Systems, i.e. untimed (or logical), timed, and stochastic.

Queuing models provide a convenient framework to describe manufacturing systems because their real physical limitations (Fig. 1), buffers in a manufacturing system usually have finite size. However when a type of fault is inserted for tolerance analysis the size of the model could be increased exponentially. If this procedure is inserted as an algorithm developed in a C.A.D. platform, describing the structural and behavioral level of the system, then the computing time is difficult to reduce [9]. Munoz-Hernandez et al [10] have reported some novel schemes to describe efficiently, on size and computing time, the use of Graphical C.A.D. environments.

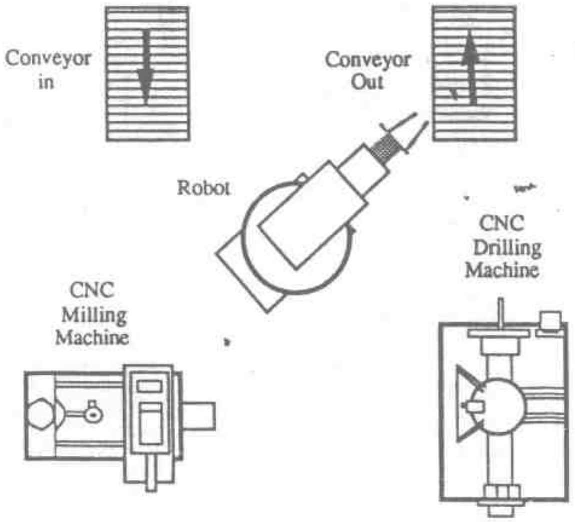


Fig. 1. Typical resources for a Manufacturing System.

Examples as VHDL, Matlab® and others platform have been applied with good results. Recent solution using explicitly environment have been also reported. In the issue of manufacturing process, Nasr and Kamrani [11] established the use of C.A.D. modeling as part of a framework collaborative environment.

Design problems can be decomposed into models such as physical components, parametric models, or analysis procedures. The important aspect of the proposed framework is the integration of these models used during the design process in the collaborative environment. Thus, the proposed collaborative framework allows the integrated model to be revised with any change made by individual involved in the modeling process. The individual does not have to analyze the scenario repeatedly for every change in the design of variables and to validate it for each instance. This framework allows the designer to collaborate with the vendors, and other design team members to speed up and optimize the design process considering the relationship within these models. Parametric modeling can also be introduced to take its advantages of quicker response, accuracy, consistency, documentation, etc.

Several companies around the world have developed frameworks with programmable tools to describe real resources with libraries and capabilities to change the structural and behavioral description.

III. Graphical Approach

There are several methods proposed to develop the description for the structural and behavior definition of each resource included in a typical manufacturing system. In this study, the method proposed by Gracios [9] was chosen to demonstrate the reduction of size and timing in the internal environment scheduling. The application of the method was developed by Lebano [12], the model is currently used for teaching and researching in postgraduate courses at the "Universidad Popular Autonoma del Estado de Puebla".

The first level of application is: to determine the layout for the system to establish the physical distribution for each resource. Figure 2 shows this abstraction level.

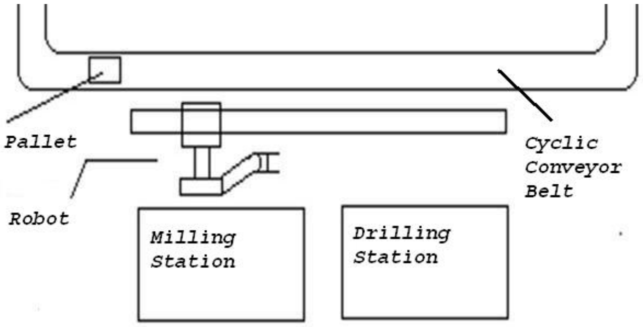
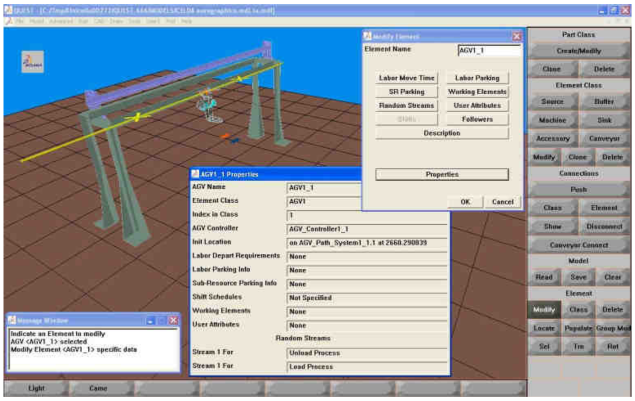


Fig. 2. Layout for the system.

For the particular case of the Cyclic Conveyor Belt, only the interaction with the robot, pallets, and machines is described.

The second level of abstraction for the system is: to describe the structural part of each resource. For this step, Quest® by Delmia® (Dassault System) was used as the C.A.D. environment.

Figure 3 shows the initial graphical description for the robot in the system. Quest® provides a complete graphical tool that accepts files from other platforms, as Solid Works®.



Modify Element

Element Name: AGV1_1

AGV1_1 Properties

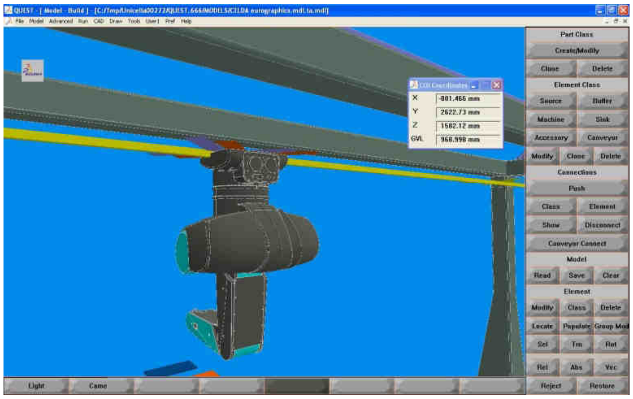
AGV Name	AGV1_1
Element Class	AGV1
Index in Class	1
AGV Controller	AGV_Controller1_1
Init Location	on AGV_Path_System1_1.1 at 2660.290839
Labor Depart Requirements	None
Labor Parking Info	None
Sub-Resource Parking Info	None
Shift Schedules	Not Specified
Working Elements	None
User Attributes	None
Random Streams	
Stream 1 For	Unload Process
Stream 1 For	Load Process

Message Window

```
Indicate an Element to modify
AGV <AGV1_1> selected
Modify Element <AGV1_1> specific data
```

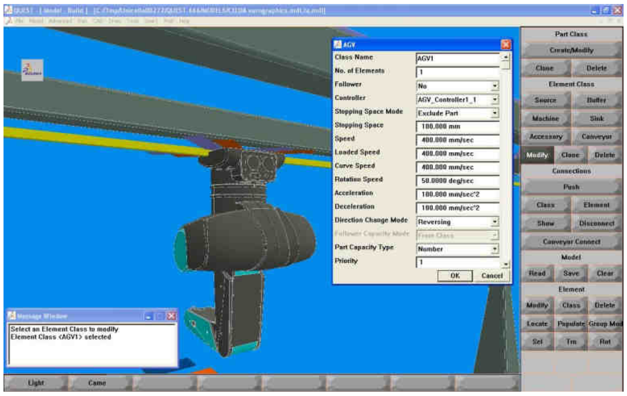
Fig. 3. Structural graphical description for the robot.

The main characteristic provided by Quest® is the possibility of inserting the source code to increase the level of description (open programming). The Simulation Control Language (SCL) [13] is a procedural language used to construct logic, which can be employed to govern the actions and behavior of individual model entities in a simulation. SCL incorporates conventions commonly used in high-level computer languages with specific enhancements for discrete event logic processing and simulation environment inquiries. In Figures 4 and 5, samples of the capability to describe parametric details for each resource is showed, in this case, the functional parameters for the robot, in its particular behavior.



X	-801.466 mm
Y	2622.73 mm
Z	1582.12 mm
GVL	968.998 mm

Fig. 4. Graphical detail for the robot.



AGV

Class Name	AGV1
No. of Elements	1
Follower	No
Controller	AGV_Controller1_1
Stopping Space Mode	Exclude Part
Stopping Space	100.000 mm
Speed	400.000 mm/sec
Loaded Speed	400.000 mm/sec
Curve Speed	400.000 mm/sec
Rotation Speed	50.0000 deg/sec
Acceleration	100.000 mm/sec^2
Deceleration	100.000 mm/sec^2
Direction Change Mode	Reversing
Follower Capacity Mode	From Class
Part Capacity Type	Number
Priority	1

Message Window

```
Select an Element Class to modify
Element Class <AGV1> selected
```

Fig. 5. Parametrical information.

The term "logic" is used to describe the decision-making activities that occur at specific instances during the simulation. QUEST® provides large numbers of built-in logic; however, because unique and complex requirements may not be met with the standard built-in logic, the user may compose custom SCL logic programs to control any or all aspects of element behavior. An SCL file may contain several SCL logic programs. SCL logic may be written to control:

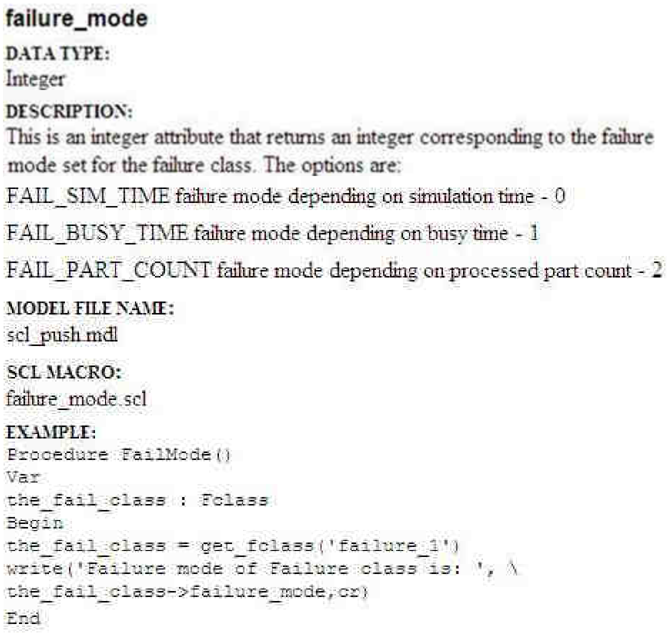
- Routing.
- Processing.
- Queuing.
- AGV/labor Motion.
- Decision Point Activity.
- Initialization, Termination, Simulation.
- Pre and Post event actions.
- Defined user behavior (buttons/macros).

SCL defines several data types that allow the programmer to access the specific information of each element. It also provides a rich collection of built-in procedures and routines that can be used to call actions in the model such as routing or starting a process.

The solid models of the components for the complete system are generated and can be displayed at the user interface in graphical format. This gives to the designer the opportunity to visualize the different alternatives and the optimum configuration of the component. These CAD models as a repository can further be used for FDI analysis, NC code generation, manufacturing documentation, reuse for new products, and several other applications.

IV. Inserting Soft Fault.-Simulation and Control steps

When the insertion of any fault in the behavior is made, the "normal" activity of the logic for the resource must be modified. In Figure 6, the SCL code is listed in terms of a minimal variation when a fault is inserted. In this section, the complete structural and functional behavior for the whole system is shown.



failure_mode

DATA TYPE:
Integer

DESCRIPTION:
This is an integer attribute that returns an integer corresponding to the failure mode set for the failure class. The options are:

FAIL_SIM_TIME failure mode depending on simulation time - 0

FAIL_BUSY_TIME failure mode depending on busy time - 1

FAIL_PART_COUNT failure mode depending on processed part count - 2

MODEL FILE NAME:
scl_push.mdl

SCL MACRO:
failure_mode.scl

EXAMPLE:

```
Procedure FailMode()
Var
the_fail_class : Fclass
Begin
the_fail_class = get_fclass('failure_1')
write('Failure mode of Failure class is: ', \
the_fail_class->failure_mode,cr)
End
```

Fig. 6. SCL code example for fault insertion.

When the submenu of Simulation is selected, new tools are available to insert the parametrical restrictions for the FDI analysis. Figure 7 shows these new characteristics. The tabs which will be more used are: Simulation, SCL program and the subtab of experimentation. The subtab is used to obtain the graphical report of simulation without faults and soft faults routines.

Figure 8 indicates the graphical representation of the Logical Sequence Assigned for this example. The figure shows an application of graphical modeling to Detect Soft Faults in the real system.

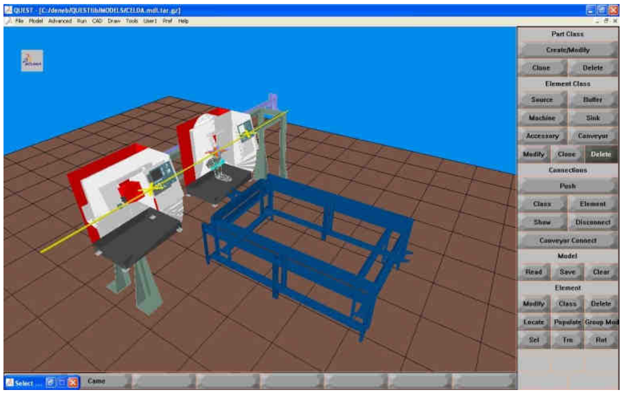
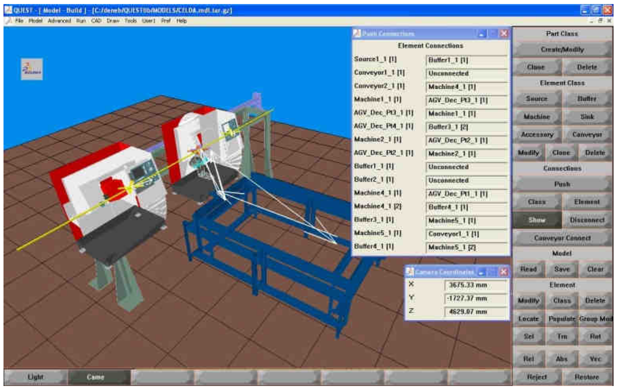


Fig.7. Complete graph definition for the Manufacturing System.



Element Connections

Source1_1 [1]	Buffer1_1 [1]
Conveyor1_1 [1]	Unconnected
Conveyor2_1 [1]	Machine4_1 [1]
Machine1_1 [1]	AGV_Dec_Pt3_1 [1]
AGV_Dec_Pt3_1 [1]	Machine1_1 [1]
AGV_Dec_Pt4_1 [1]	Buffer3_1 [2]
Machine2_1 [1]	AGV_Dec_Pt2_1 [1]
AGV_Dec_Pt2_1 [1]	Machine2_1 [1]
Buffer1_1 [1]	Unconnected
Buffer2_1 [1]	Unconnected
Machine4_1 [1]	AGV_Dec_Pt1_1 [1]
Machine4_1 [2]	Buffer4_1 [1]
Buffer3_1 [1]	Machine5_1 [1]
Machine5_1 [1]	Conveyor1_1 [1]
Buffer4_1 [1]	Machine5_1 [2]

Camera Coordinates

X	3675.33 mm
Y	-1727.37 mm
Z	4629.07 mm

Fig. 8. Logical Sequence description.

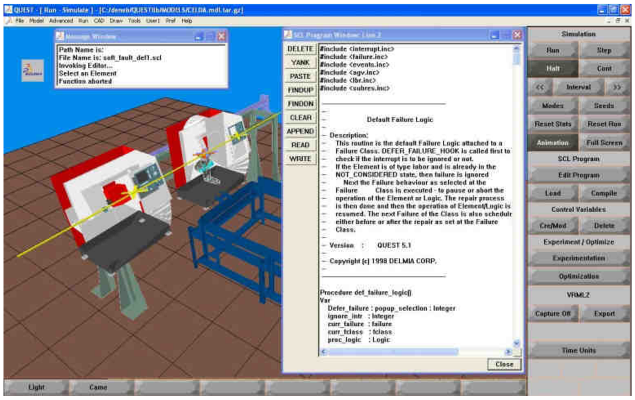
SCL environment is used to insert the Soft Fault considering four restrictions:

- The number and type of soft faults were obtained by Lebano [12] and was reported in [14].
- The graphical model is expressed in the present article with restrictions by Copyright Laws of the institution (Universidad Popular Autonoma del Estado de Puebla).
- The results obtained with this model have only academic and research application.
- The soft fault model is focused in terms of robot interaction with the other resources.

The preliminary description of a fault in the activity of the robot is presented in Figure 9 using SCL code. The insertion of soft fault is developed by the used of research results presented by Munoz et al where Fuzzy Time Series (FTS) are applied as a stochastic model in [15].

V. Conclusions

This work has shown the use of a C.A.D. environment inserting the ability of fault detection and isolation modeling where a typical manufacturing system has included soft faults. The level of abstraction obtained is a good approach to develop a virtual graphic environment to model, simulate and control manufacturing system using online interfaces. The quality of the graphical models is adequate to be inserted into the real structural environment and functional behavior.



```
Path Name is:
File Name is: soft_fault_def1.scl
Invoking Editor...
Select an Element
Function aborted
```

```
#include <interrupt.inc>
#include <failure.inc>
#include <events.inc>
#include <agv.inc>
#include <lbr.inc>
#include <subres.inc>

----------------------------------------
--
--            Default Failure Logic
--
-- Description:
--    This routine is the default Failure Logic attached to a
--    Failure Class. DEFER_FAILURE_HOOK is called first to
--    check if the interrupt is to be ignored or not.
--    If the Element is of type labor and is already in the
--    NOT_CONSIDERED state, then failure is ignored
--       Next the Failure behaviour as selected at the
--    Failure       Class is executed - to pause or abort the
--    operation of the Element or Logic. The repair process
--    is then done and then the operation of Element/Logic is
--    resumed. The next Failure of the Class is also schedule
--    either before or after the repair as set at the Failure
--    Class.
--
-- Version   :     QUEST 5.1
--
-- Copyright (c) 1998 DELMIA CORP.
--
----------------------------------------

Procedure def_failure_logic()
Var
   Defer_failure : popup_selection : Integer
   ignore_intr   : Integer
   curr_failure  : failure
   curr_fclass   : fclass
   proc_logic    : Logic
```

Fig. 9. SCL description for the robot on soft fault conditions.

Scheduling Control Language is a novel approach to describe non-deterministic activities in each part of the system, and the inclusion of soft fault in the whole system offer new alternative on the modeling, simulations and verification of control strategies where a virtual laboratory is required.

Perhaps the advances in this work, the model of other type of resources, Automatic Guided Vehicle, Controllers are needed. The application of different types of controllers requires the description of a tool for the inclusion of dataports or buses and to add personal behavior like operators or supervisors can be described using new attributes which can be defined in the platform.

The complete system has been installed and used for the students and researchers of the Universidad Popular Autónoma del Estado de Puebla, where practical experiences are developed, those experiences are mainly: the interaction of humans with software and hardware, in presence of soft fault in the line of knowledge generation, and the application of Intelligent Manufacturing Systems in under- and post-graduate levels [12, 14].

ACKNOWLEDGMENT

The authors thank the Integrated Manufacturing Systems Office and Government Board of the Universidad Popular Autónoma del Estado de Puebla for the authorization to use and report the results of this research. We are also thankful to the Dassault System Company for the permission to use the Quest® environment.

REFERENCES

[1] Shen W., Norrie D., Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. Knowledge and Information Systems, an International Journal, 1(2), 129–156, 1999.

[2] Zhou M., Venkatesh K., Modeling, Simulation and Control of Flexible Manufacturing Systems.-A Petri Net Approach, Series in Intelligent Control and Intelligent Automation Vol. 6, World Scientific, 1999.

[3] Rzevski G., Artificial Intelligence in Manufacturing, Computational Mechanics Publications, Springer, Proceedings of the 4th International Conference on the Applications of Artificial Intelligence on Engineering, Cambridge, U. K. July 1989.

[4] Dagli C. H., Artificial Neural Networks for Intelligent Manufacturing, First Edition, Chapman and Hall, U. K., 1994.

[5] Palade V., Bocaniala C., and Jain L. (Eds), Computational Intelligence in Fault Diagnosis Advanced Information and Knowledge Processing, 2006.

[6] Jain L.; Martin N., Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications, CRC Press, CRC Press LLC, ISBN: 0849398045 Pub Date: 11/01/98

[7] Gracios C., Vargas E., Diaz A., Describing an IMSby a FNRTPN definition: a VHDL approach, Robotics and CIM, Vol. 21, Issue 3, Elsevier June 2005.

[8] Cassandras C. and LaFortune S., Introduction to Discrete Event Systems, Second Edition, Springer, 2008.

[9] Gracios C., Munoz G., Diaz A., Nuno P., Estevez J. and Vega C., Recursive decision-making feedback extension (RDFE) for fuzzy scheduling scheme applied on electrical power control generation, International Journal of Electrical Power and Energy Systems Volume 31, Issue 6, July 2009, Pages 237–242.

[10] Munoz G., Gracios C., Diaz A., Mansoor P. and Jones D., Neural PDF Control Strategy for a Hydroelectric Station Simulator, Automation Control -Theory and Practice, A D Rodick (Ed.), INTECH, Available from: http://sciyo.com/articles/show/title/neural-pdfcontrol-strategy-for-a-hydroelectric-station-simulator, 2009.

[11] Nasr E. and Kamrani A., Computer-Based Design and Manufacturing: An Information-Based Approach, Springer, LLC, 2007.

[12] Lebano E., Modeling Soft Faults in Flexible Manufacturing Systems using Stochastic Petri Nets and Virtual Models, Doctoral Dissertation, Universidad Popular Autonoma del Estado de Puebla, April, 2009.

[13] Quest c@SCL Reference Manual

[14] Lebano E., Gracios C., Determining the degree of adaptability of a Flexible Manufacturing System under uncertainty situations, Proceedings of the 8o. National Congress of mechatronics, Veracruz, Ver. Mexico, November 26–27, 2009.

[15] Munoz G., Gracios C., Diaz A., Scheduling scheme of electrical power generation using Recursive Decision making Feedback Extension (RDFE). 6º Congreso Internacional Tendencias Tecnologicas en Computacion, 18–22 October. Mexico, D.F.

VirtUATx: A Virtual Reality and Visualization Center

Marva Angélica Mora Lumbreras, Álvaro Jair Martínez Varela, Julio Cesar Calva Plata, Rubén Alfredo Mejorada Lira, Brian Manuel González Contreras, and Alberto Portilla

***Abstract*—The construction of a Virtual Reality and Visualization Center in a public or private University results in a series of benefits; it allows the students to interact with modern technology, to learn with different tools. Specifically, this paper focuses on a Virtual Reality and Visualization Center built at the Bachelor in Computer Engineering at the Autonomous University of Tlaxcala, this center is called VirtUATx. VirtUATx allows building different virtual environment multi-screen considering the head movements through a professional head tracker, a full navigation system by means of recognition of body movement's using Kinect or manual using keyboard. Furthermore, VirtUATx incorporates 3D object manipulation, anaglyphic and active stereoscopy and 3D sound. This Center is used in different courses, such as Research Seminars, Computer Graphics, Research Methodology, and Interaction Human-Computer, but in the future this center will benefit other bachelors.**

***Index Terms*—Virtual Reality, virtual world, immersion, stereoscopy, 3D sound, manipulation**

I. Introduction

A VIRTUAL Reality and Visualization Center allows producing research and the development of multi-disciplinary projects, combining the talents of researchers in many areas. Furthermore the construction of these environments involves the use of different courses, such as: Operative Systems, Computer Networks, Interaction Human-Computer, Computer Graphics, and Mathematics, so its construction is ideal for Bachelor of Computer Engineering.

Currently there exist several Institutions and Research Centers with their own virtual environments. One of the most popular virtual environments is the CAVE (Cave Automatic Virtual Environment), this concept was developed at the Electronic Visualization Laboratory at the University of Illinois at Chicago since 1992 [1].

In Mexico there are some Universities use Virtual Reality. Universidad Nacional Antónoma de México (UNAM) is an example of this. UNAM has built a visualization observatory called Ixtli, which uses virtual reality tools [2]. Colima University focuses on teaching using virtual reality [3, 4].

II. VirtUATx

An Advanced Visualization Center involves many resources, such as hardware, software, installations, and specific applications. Moreover, there are others issues that must be considered such as: immersion, projection types and the user interaction.

VirtUATx is composed mainly of two parts:

- The physical part, which includes gaming computers, monitors, three back and front projection screens, four 3D projectors, six loud-speakers and input devices like a head-tracker and two Kinects
- The virtual part, which includes the software for building different flexible virtual environments and virtual worlds.

In order to have an effective immersion experience, VirtUATx allow for the creating of involve virtual environments, when it is cubic the size is 3X2X3 meters.

III. VirtUATx Design

In this section, we show the general scheme of VirtUATx. The system allows:

- Displaying virtual environments across multiple coplanar and not coplanar screens (monitors and/or projection screens).
- Using three input devices: keyboard, head-tracker and kinect.
- Different virtual environments using OpenGL, 3DS, and Java.
- Anaglyphic and active stereoscopy
- Full Navigation.

VirtUATx uses a client-server structure, which allows for the use of a set of devices connected via net. In Fig. 1 the white squares represent the server and the clients. The gray squares represent the virtual worlds and the manipulation kind. A client is a computer, which display part of a virtual world on a screen or monitor.

The server manages one or more clients (screens) and is responsible for doing a set of tasks. The server obtains the screen configurations, updates the viewer position through an input device, which can be a keyboard or a head-tracker. A user choices a virtual world through the server.

Each client knows the information about its screen. This information includes the real position, the size, and information about whether the virtual world is stereoscopic or not. The client establishes communication with the server via messages. Then the client sends its screen configuration to the

Manuscript received October 14, 2012. Manuscript accepted for publication December 5, 2012.

Marva Angélica Mora Lumbreras. Álvaro Jair Martínez Varela, Julio Cesar Calva Plata, Rubén Alfredo Mejorada Lira, Brian Manuel González Contreras, and Alberto Portilla are with the Research Center of the College of Engineering and Technology, Calzada Apizaquito s/n, 90300, México (e-mail: marva.mora@gmail.com).

server, and the server returns the information necessary to determine the section of the world that corresponds to the display. The client frequently asks the server for the viewer position as well as the camera position to be able to navigate.

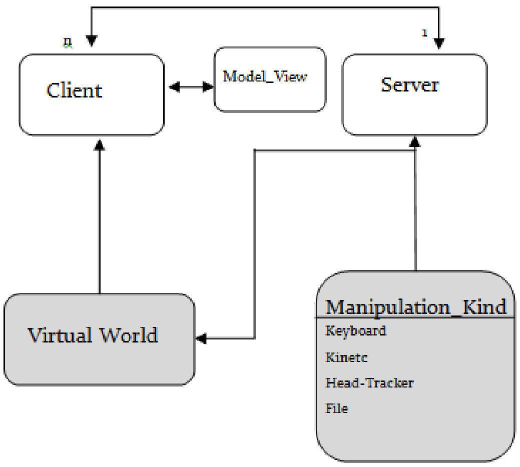


Fig. 1. General Scheme

IV. Interacting in VirtUATx

Typical interfaces use only keyboard and mouse, which is insufficient to generate immersion because in a virtual environment is important a natural interaction, therefore, a virtual environment requires appropriate devices to interact. VirtUATx uses different interaction methods. The first method consider the viewer position to calculate the vision volume through of generic frustum (described in this section), the second interaction method is the navigation in the world using translations and rotations movements and a virtual camera, finally the specific 3D object manipulation, such as scaling, rotations and translations.

A. Head tracking

A *view volume* is a perspective projection called *frustum, view frustum and viewing frustum*. VirtUATx uses frustums to display virtual worlds in multiples screens. *Frustum* determines the region of a virtual world that is going to be displayed on a screen [5]. It is defined by the *near, far, left, right, top, and bottom* distances between the screen and the viewer position. *Near* and *far* are the planes that cut the frustum and are perpendicular to the *Z* axis. The *projection center* is the pyramid apex and a projection plane is parallel plane to the pyramid base. The *vision line*, also called *frustum axis*, if this line is perpendicular to the screen center and is spread on *Z* axis the frustum is symmetric, in any other case is asymmetric.

Frustum uses viewer's and the screens's positions, the viewer position is located in the pyramid apex and every screen is a projection plane, see Fig. 3. Using this method, VirtUATx is dynamic, and every user movement produces different views in the virtual world.

When a *viewing frustum* is miscalculated, the viewer sees the incorrect information, although the *frustum* only differs some millimeters.

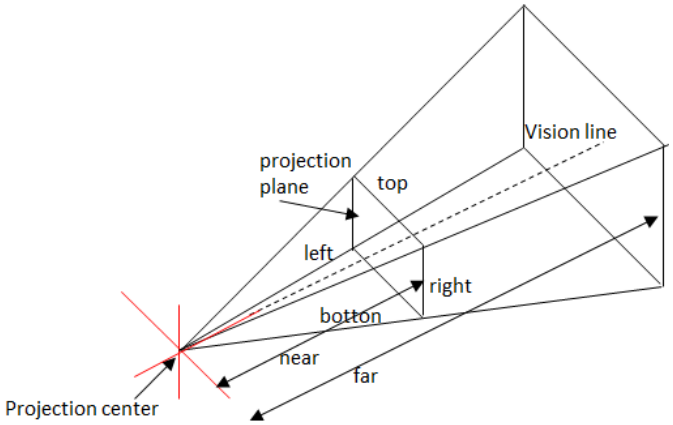


Fig. 2. Frustum

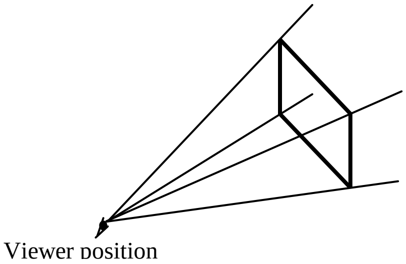


Fig. 3. Viewing Frustum

VirtUATx uses *n* screens, therefore use *n frustums*. Producing a virtual world divided into regions, these are displayed on individual screens. The projection center for all screens is the viewer position. Fig. 4 shows an example built with four screens and Fig. 5 is other configuration using four screens too.

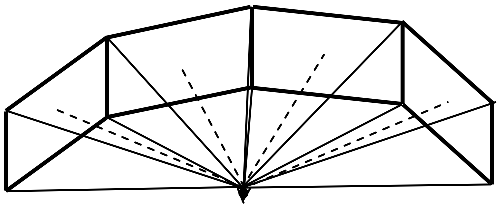


Fig. 4. A virtual environment built with four screens

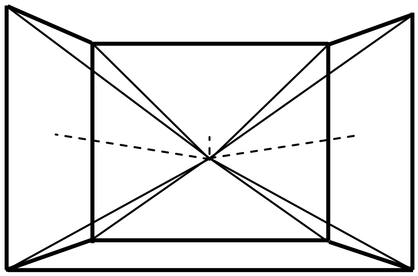


Fig. 5. An involve environment built with four projections

When the frustums are miscalculated in a virtual environment built with *n* screens, the information does not look continuous, on some occasions the objects can look wider, thinner or even folded. Therefore, it is necessary to synchronize all measures involved in the environment and take into consideration the screen frames.

The result on apply frustums is shown in Fig. 6–9. These configurations were built with two computers.

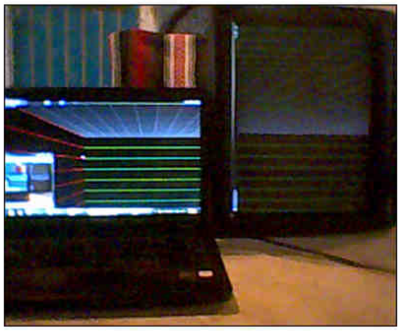


Fig. 6. Two monitors, one rotated 90 degrees

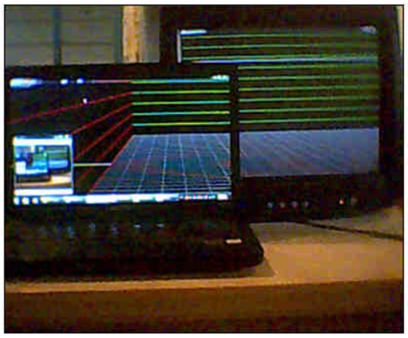


Fig. 7. Two overlapping monitors

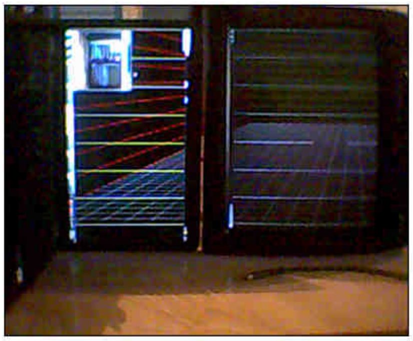


Fig. 8. Monitors rotated 90 degrees on z axis

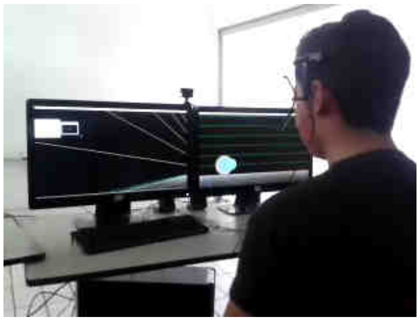


Fig. 9. Two monitors and a Head-Movement-Tracker

B. *Navigating a virtual world using a kinect*

VirtUATx displays a virtual world and uses a method for navigating in first and third person [6]. Furthermore, VirtUATx uses a Kinect, which allows navigating the world in six different directions. This device has the advantage of providing full freedom of movement to the viewer.

- If the viewer *turns* his body to his left, right, up or down direction, the virtual environment will turn an opposite angle that the viewer turns.
- The moves in the virtual world are made in accordance with body movements, when the viewer *moves* his body left, right, upward, downward, forward, or backward, see Fig. 10.

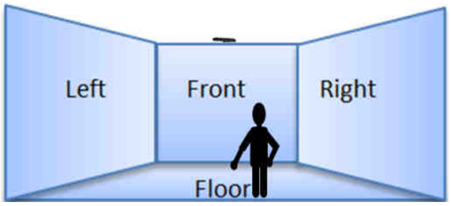


Fig. 10. Navigation using a Kinect

C. *Manipulating 3d Objects using gesture recognition*

VirtUATx incorporates a better interaction in its virtual world on using a system for manipulating 3D objects through hand movements, the operations used are translation, rotation and scaling. The recognition is done using a Kinect, see Fig. 11.

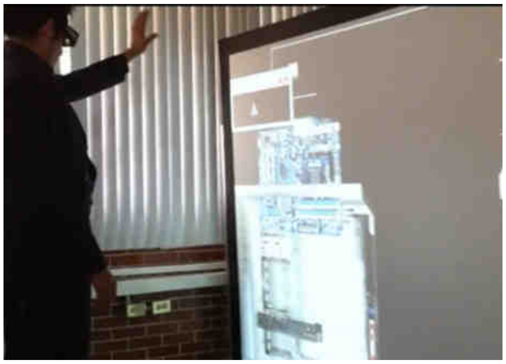


Fig. 11. Manipulating 3d Objects using hand movements

V. 3D Sound

Digital sound has evolved [7], currently, the use of sound is becoming a more common tool in virtual environments, because it adds a certain level of realism to any virtual environment [8], in addition, sound allows for having a better interface, because indicate the reception of commands or confirm some activities [9]. Even music helps to manipulate user’s emotions, including happiness, sadness, nostalgia, peace, etc.

3D sound means that a listener hears sounds from any direction; this sound is generally simulated by a computer. 3D sound has many characteristics that can provide advantages in virtual environments. VirtUATx incorporates 3D due to the following characteristics:

– 3D Sound provides extra help for the user to find objects when he is navigating, because the hearing system can determine the location of the sound sources.
– The 3D Sound produces a high immersion level in a virtual environment.
– The 3D Sound helps to interpret distances among objects.
– The 3D Sound facilitates a more natural interaction because it is similar to the sound in the real world.
– Sound can provide additional information to a graphic world, by helping users to understand extra information without extra effort.

3D sound is similar to 3D graphics; it uses the positions of the sound sources, besides the listener orientation and position for creating a real effect. The rotations that the listener is permitted to make are called Elevation and Azimuth. See Fig, 12, Elevation is the angle along the vertical plane. Azimuth is the angle along the horizontal plane. With these rotations user is able to see a virtual world in its entirety and perceive the sound from the new positions. In VirtUATx these rotations are included.

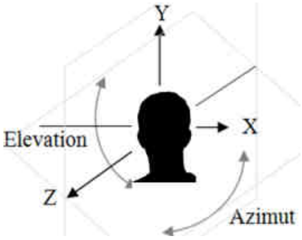


Fig. 12. Elevation and Azimuth rotation.

The allowed translations in the navigation of this work include *left, right, upward, downward, forward, and backward*.

Three important coordinates are used in virtUATx: the positions of the sound sources see (1); the listener position, see (2); and the orientation, see (3).

$$SourcePosition_n = x_n y_n z_n, \quad (1)$$

where n ≥ 1,

$$LPosition = L_{xyz} \quad (2)$$

$$LOrientation = R_x, R_y \ , \quad (3)$$

VI. Stereoscopy

Although the human beings are able to see with depth in a natural way, some scientists have worked on the artificial simulation of this perception since the nineteenth century. Sir Charles Wheatstone who built the first stereoscope in 1838 is considered one of the pioneers in stereoscopy [10].

The stereoscopy consists in using two photographs or images, one for the left eye and one for the right. When they are combined they provide an illusion of depth by allowing us to see three-dimensional views in an artificial way. There are different stereoscopic techniques such as Parallel Viewing or Wall Eyed, Crossed-Eye Viewing, View-Master, Head-Mounted Displays, active stereoscopy, passive stereoscopy, Two-color Anaglyph, Wiggle Stereoscopy, Autostereograms, Pulfrich Effect, Displays with Filter Arrays, etc.

VirtUATx includes active and Two-color Anaglyph stereoscopy, both techniques allows for seeing stereoscopy in giant virtual environments.

Fig. 13 shows the assembly of a computer using active stereoscopy. Fig. 14 shows Research Center of the College of Engineering and Technology using anaglyphic technique.

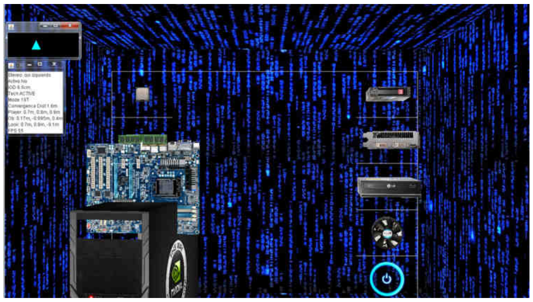


Fig. 13. Computer with stereoscopic view

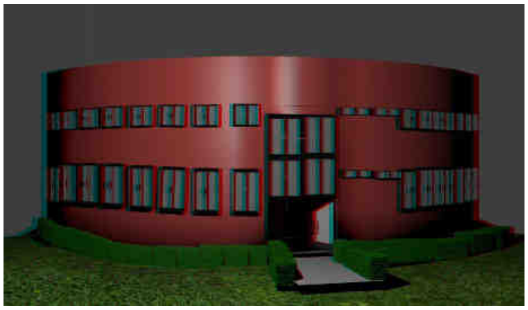


Fig. 14. Research Center of the College of Engineering and Technology

VII. Results

In this paper VirtUATx is presented, VirtUATx is used with undergraduate students; furthermore it has been presented events of different academic levels such as high schools, bachelors and masters in the Tlaxcala city.

Every technique has been tested in the courses and in the academic events from manipulation, stereoscopy, sound, continuity and coherence between screens, including immersion level, etc.

Fig. 15 shows VirtUATx in the Computer Graphics course.

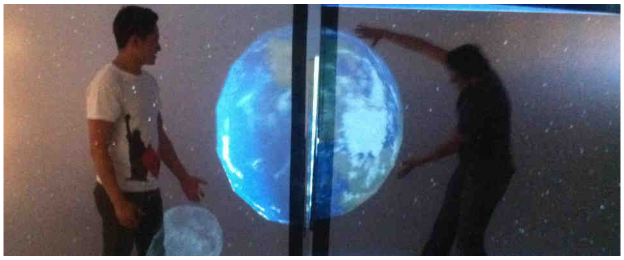


Fig. 15. VirtUATx built with two screens, a class demonstration

Fig. 16 shows a section of VirtUATx presented in a high school.

Fig. 16. A section of VirtUATx presented in a high school

VIII. Conclusions

This paper presented a Virtual Reality and Visualization Center built at the Bachelor in Computer Engineering at the Autonomous University of Tlaxcala. VirtUATx can change in size, shape (angle between screens), monitors, projections, etc., it is totally flexible, allowing it to be employed in varied projects.

Different students have used VirtUATx in courses such as Research Seminars, Computer Graphics, Research Methodology, and Interaction Human-Computer, furthermore academic events, with satisfactory results. VirtUATx allows the students to interact with modern technology; in addition, it allows working in research complex projects, although the construction of virtual environments is a complex work.

Acknowledgments

This work was supported in part by the Universidad Autónoma de Tlaxcala. This research project would not have been possible without the support of PROMEP (Programa de Mejoramiento del Profesorado)**.**

References

[1] C. N. Carolina., J. S. Daniel. And A. D Thomas, “Surround screen projection-based virtual reality: the design and implementation of the cave”. Proceedings of the 20th annual conference on Computer graphics and interactive techniques. ACM Press. 1993. pp. 135–142.

[2] G. Lucet. Una experiencia de aprendizaje inmersiva. Learning Review Latinoamérica Number 10. 2006.

[3] G. R. Miguel Ángel, .A., V. V. Laura, G. S. Zeferino. Integration of molecular scientific visualization in the classrom (in Spanish). Quimica Nova, 31(8), Pp 2184-2189. 2008.

[4] G. R. Miguel A., E. Arthur, "Collaborating and learning a second language in a Wireless Virtual Reality Environment", Mobile Learning and Organisation, Vol. 2, No. 4, 2008, pp 369–377.

[5] J. Ken “User’s Guide: SGI OpenGL Multiple SDK”, Ver. 3.2. 007-4239, 2005

[6] M. L. Marva, Evaluating Virtual Environments into the University System, Proceeding of International Congress of Technological Innovation 2010, Vol. 1, pp.143–147.

[7] C. L. Martha Rosa, Simulación de escenarios naturales tridimensionales, Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, 2008.

[8] G. Matti. Doctoral thesis: Application of spatial sound reproduction in virtual environments-experiments in localization, navigation, and orientation. ISSN: 0787-7498, Espoo, Finland: Department of Computer Science and Engineering, Helsinki University of Technology, 2006.

[9] B. Grigore and C. Philippe, Virtual Reality Technology. New Jersey: Wiley-Interscience , vol. 2, 2003.

[10] Wheatstone, C. On some remarkable, and hitherto unobserved, phenomena of binocular vision. Philosophical Transactions of the Royal Society of London (Vol. 128), 1838, pp. 371–394.

Anonymizing but Deteriorating Location Databases

Tran Khanh Dang and Tuan Anh Truong

Abstract—**The tremendous development of location-based services and mobile devices has led to an increase in location databases. Through the data mining process, valuable information can be discovered from such location databases. However, the malicious data miner or attackers may also extract private and sensitive information about the user, and this can create threats against the user location privacy. Therefore, location privacy protection becomes a key factor to the success in privacy protection for the users of location-based services. In this paper, we propose a novel approach as well as an algorithm to guarantee k-anonymity in a location database. The algorithm will maintain the association rules that have significance for the data mining process. Moreover, there may appear new significant association rules created after anonymization, they maybe affect the data mining result. Therefore, the algorithm also considers excluding new significant association rules that are created during the run of the algorithm. Theoretical analyses and experimental results with real-world datasets will confirm the practical value of our newly proposed approach.**

Index Terms—**k-anonymity, location databases, data mining, privacy protection.**

I. Introduction

TODAY, advances in location technologies and wireless communication technologies enable the widespread development of location-based services (LBSs). When using services, the user may face with risks because the location information of the user can disclose some private information. Therefore, we need to protect the location information of the user from attacking of malefactors.

The user's location privacy should be protected in two stages. In the first stage, the location privacy should be protected at the time of using services. One popular method used in this stage is to obfuscate the user's exact location with respect to service providers in order to hide the user's location information. These solutions focus on preventing the user's location from an illegal observation at the time of service calls. We also proposed some approaches to obfuscate the user's location in [2, 3, 11, 22–25]. In the second stage, the location privacy of the user should be protected as the user's location information is stored in the database for data mining purposes. In this stage, the location information of the user will be anonymized before these data are published to other organizations or companies.

In this paper, we will focus on protecting the user's location at the second stage when the location data is stored in the database for data mining purposes. We assume that when the user uses services, he/she will provide his/her true location to service servers and the service servers will save all information about the location of the user. Then, many organizations, companies or individuals may collect these location data.

Through the data mining process, some valuable information can be obtained. However, these location data maybe disclose some private information of the user. For example, the attacker queries the database and receives results, but he also has some knowledge about the service and links the knowledge with the results to obtain some sensitive information. Therefore, we should protect these location data before they are collected by organizations, companies or individual. Fortunately, we have some techniques to protect user data before publishing these data as randomization, k-anonymity, etc. Among them, k-anonymity is an important method for privacy de-identification. The motivating factor behind the k-anonymity technique is that many attributes in the data can often be considered pseudo-identifiers which can be used in conjunction with public records in order to uniquely identify the records [1, 5].

This paper will improve the approach which was proposed in [4] and will use this improved approach to anonymize the location database to achieve an effective k-anonymous version. This approach does not use two popular techniques (generalization and suppression) because data after anonymizing by these techniques may not be significant to the data mining processes. The approach will use a technique which is called *Migrate Member technique* to anonymize the database [4]. The approach also considers the result of data mining process by maintaining association rules that are significant to the data mining process.

The rest of this paper is organized as follows: in Section II, we briefly summarize related works. Section III introduces definitions and calculating methods of crucial values for the algorithm. Next, Section IV presents the proposed algorithm to guarantee k-anonymity in location databases. Experimental results are shown in Section V. Finally, Section VI presents concluding remarks as well as future works of our approach.

Manuscript received March 14, 2012. Manuscript accepted for publication December 12, 2012.

T.K. Dang is with the Faculty of Computer Science & Engineering, HCMC University of Technology, VNUHCM, Ho Chi Minh City, Vietnam (e-mail: khanh@cse.hcmut.edu.vn).

T.A. Truong is with the Department of Information Engineering and Computer Science, University of Trento, Italy (e-mail: anhtt@cse.hcmut.edu.vn).

II. Related Work

A. k-anonymity

The notion of k-anonymity, proposed by Samarati [7], is an approach to protect data from individual identification. k-anonymity is a property that models the protection of released data against possible re-identification of the respondents to which the data refer. Intuitively, k-anonymity states that each release of data must be such that every combination of values of released attributes, which are also externally available and therefore exploitable for linking, can be indistinctly matched to at least k respondents.

k-anonymous data mining has been recently introduced as an approach to ensuring privacy-preservation when releasing data mining results [7]. With this approach, the author defined the set of attributes whose values may be used, possibly together with external information, to re-identify the user's data. These attributes are called *Quasi-Identifiers* (QI). For example, even if data about the ZIP code, date of birth and sex do not explicitly identify an individual, they may be linked to external information (for example: public voter lists) to obtain name, address. Intuitively, the greater the value of k, the better the protection of privacy. However, if value of k is too great, data quality for the data mining process is not good. Therefore, how to keep the balance between data privacy and data quality is an important factor in privacy preserving in data mining. In this paper, we propose an algorithm not only to anonymize the location database but also to consider the result of data mining processes.

B. k-Anonymity Techniques, M3AR algorithm and problems

Today, we have some algorithms which guarantee k-anonymity in a database. These algorithms usually use one of two techniques: Generalization or Suppression. In the method of generalization, attribute values are generalized to a range in order to reduce the granularity of representation [10]. For example, date of birth could be generalized to a range such as year of birth, so as to reduce the risk of identification. In the method of suppression, the value of an attribute could be removed completely to guarantee k-anonymity. Clearly, these methods reduce the risk of identification with the use of public records and also reduce the accuracy of data mining applications on the transformed data. They only concentrate on guaranteeing k-anonymity for the database and do not consider data mining processes.

Normally, after data is collected, they will be analyzed by data mining applications to enucleate some value information. Therefore, if input data is not correct, the result of data mining applications may be invaluable. With these methods, transformed data is generalized and suppressed too much. Consequently, the results, which are received after mining, may not bring some value information. Moreover, most data mining applications use association rule mining as their main technique to enucleate value information from the input data. Therefore, association rules, which are supported in the data, should be maintained. However, it is difficult to maintain all association rules because the number of association rules may be big. Moreover, only association rules, which are significant to the data mining process, may enucleate some value information. Therefore, we should only maintain the significant association rules to reduce the number of rule maintained and also to reduce the complexity of work.

In [4, 21], the authors proposed the *Migrate Member technique* to anonymize the database to achieve a k-anonymous version. The technique first groups tuples of original data into separate groups by the similarity in quasi-identifier values. Then, the groups, which have less than k tuples, will be transformed into the satisfied ones by performing some Migrate Member operations. A satisfied group will have at least k tuples in it. The database achieves a k-anonymity version if all groups must be satisfied after the processing. The authors also proposed an algorithm called M3AR (Migrate Member Maintenance Association Rules) to concretize the approach.

With M3AR, we guarantee k-anonymity for the database while still maintaining the significant association rules. However, it remains many unsatisfied groups, which the algorithm can not transform them into the satisfied ones, after processing. Therefore, the algorithm may need more time and pay the "cost" to anonymize these unsatisfied groups. The cause of this is that M3AR selects a random unsatisfied group for each process step and thus this group may not be good for this step. As a result, this group can receive more tuples than its need, thus we may have no tuples to anonymize other groups. Moreover, M3AR did not also consider reducing new significant association rules that are generated during the process. Because these new significant association rules can interfere in the input data of the data mining process, it can make the result of the data mining process less valuable.

In next sections, we will propose some solutions to solve the problems of the algorithm M3AR. We also propose a new algorithm to anonymize the location database to achieve an effective k-anonymous version. Moreover, the algorithm also reduces new significant association rules generated during the run of the algorithm.

III. Definitions and Values

As discussed above, a database satisfies k-anonymity if any tuple in this database can be indistinctly matched to at least k respondents. Moreover, this approach also defined a set of attributes whose values may disclose some sensitive information. For the location database, we will consider the *QI* will include a location attribute and a time attribute. For simplification, we will only consider the location attribute in this paper. The time attribute will leave as future works. In this section, we will give some definitions and calculate the values that are used for the proposed algorithm.

A. Definitions

This section will give essential definitions that will be used in the algorithm:

Definition: A group is a set of tuples. Moreover, all tuples in a group must have the same *QI* values. A group satisfies k-anonymity if it has at least k tuples or has no tuples in it. Otherwise, we call this group as an unsatisfied group.

We will consider the following example. The *QI* attributes are: **Sex**, **ZIP code**, **Salary** and **Status**. We will have groups: Group *A* includes tuples: *1*, *7*, *10* and *13*. Group *B* includes tuple: *2*. Group *C* includes tuples: *3*, *9*. Group *D* includes tuples: *4*, *8* and so on. If we assume that *k* is equal to 4, group *A* will satisfy 4-anonymity while group *B*, *C*, *D* will be unsatisfied groups.

TABLE 1: AN EXAMPLE TABLE

No.	ID	Sex	ZIP code	Salary	Status	Data
1	u1	Male	70000	2000	Married	…
2	u2	Male	10000	1500	Single	…
3	u1	Female	48000	1000	Married	…
4	u5	Male	48000	2000	Married	…
5	u7	Female	70000	1500	Single	…
6	u8	Female	10000	1000	Single	…
7	u6	Male	70000	2000	Married	…
8	u4	Male	48000	2000	Married	…
9	u5	Female	48000	1000	Married	…
10	u3	Male	70000	2000	Married	…
11	u9	Male	25000	1500	Single	…
12	u11	Male	54000	1500	Married	…
13	u10	Male	70000	2000	Married	…

As discussed in the previous section, the algorithm will try to retain the association rules while guaranteeing k-anonymity. However, it is difficult to retain all association rules because the number of the association rules may be very big. Normally, the data mining process will consider association rules which occur frequently in the database. Therefore, the algorithm should try to retain these rules. We call these rules as significant rules. In the algorithm, two thresholds will be provided to specify whether an association rule is significant or not. We call them as *t_s* and *t_c*. An association rule is significant if its support value is greater than *t_s* and its confidence value is also greater than *t_c*. Conversely, the association rule is insignificant.

Definition: A change between two groups *a→b*, where *a* and *b* are groups, will change all *QI* values of some tuples in *a* to the correlative values in *b*. For example, group *a* has two tuples with *QI* is *(x1, y1, t1)* and group *b* has three tuples with *QI* is *(x2, y2, t2)*, the change *a→b* will form group *b* which has five tuples. The additional tuples are from group *a* and their *QI* attributes are changed to *(x2, y2, t2)*.

Definition: a change *a→b* is total if all tuples in group *a* are transferred to group *b*. Conversely, if several of them are transferred, the change will be partial.

B. *Values Calculation*

With our algorithm, we will try to transform unsatisfied groups into satisfied ones. To do this, the algorithm will find the changes which will be performed to transform these unsatisfied groups to satisfied groups. Moreover, the algorithm also maintains significant association rules of the database. Thus, the algorithm should find the suitable changes in order that when performing these changes, these significant association rules will not be lost. In this section, we will calculate some values which will be used in the algorithm to find these changes.

Assume that we have a significant association rule *A→B* that needs to be maintained. It means that the support and confidence values of this rule are greater than thresholds. When we perform the changes, they maybe alter some values of QI attributes of tuples supporting this significant association rule. The result is that this tuple may no longer support the association rule. Clearly, if we alter too more tuples, the association rule *A→B* may not be significant. Therefore, for each significant association rule, we should calculate the maximal number of tuples which we can alter so that the significant association rule is still significant. Moreover, when performing the changes, an insignificant association rule may become a significant one. As discussed above, the algorithm also guarantees that no new significant rule will be generated because the new significant rules may affect the result of the data mining process. Therefore, we also calculate the maximum number of tuples which we can alter without generating new significant association rules. The algorithm will use these maximal numbers to calculate cost for each change. The cost of a change will be mentioned in next sections. From the costs, the algorithm will find the best changes that will be used to transform an unsatisfied group into a satisfied one.

In following parts, we will calculate the maximal number of tuples which we can alter so that the association rule is still significant: We have a significant association rule *A→B*, *s* is the support value and *c* is the confident value of this rule. We have: *s>=t_s* and *c>=t_c*. When we perform a change *a→b*, some tuples in *a* will be altered. These tuples may support the association rule *A→B*. Therefore, when they are altered, the rule may be affected. We will consider the following cases:

Case 1: *A* will be changed:

We call *n* is the number of tuples which are anonymized, *s'* is the support value and *c'* is the confident value of the rule after performing the change. The rule is significant, therefore, we must have *s'>=t_s* and *c'>=t_c*:

$$c' = \frac{number(A \to B) - n}{number(A) - n} \qquad (1)$$

$$s' = \frac{number(A \to B) - n}{total} \qquad (2)$$

where *number*(*A*→ *B*) is the number of tuples which have both *A* and *B*, *number(A)* is the number of tuples which only have *A*, *total* is the number of tuples in the database. Besides, we also have:

$$s = \frac{number(A \to B)}{total} \qquad (3)$$

$$c = \frac{number(A \to B)}{number(A)} \qquad (4)$$

The maximal number of tuples, which can be altered, is:

$$n = MIN\big(total * (s - t\_s), \left\lfloor \frac{s * total * (c - t\_c)}{c * (1 - t\_c)} \right\rfloor \qquad (5)$$

Case 2: *B* will be changed:

Similarly, we have:

$$c' = \frac{number(A \to B) - n}{number(A)} \qquad (6)$$

$$s' = \frac{number(A \to B) - n}{total} \qquad (7)$$

Moreover, we also have:

$$s = \frac{number(A \to B)}{total} \qquad (8)$$

$$c = \frac{number(A \to B)}{number(A)} \qquad (9)$$

The condition are *s'>=t_s* and *c'>=t_c*. Therefore, we have:

$$n = MIN\left(total \times (s - t\_s), \left\lfloor \frac{s * total * (c - t\_c)}{c} \right\rfloor \right) \qquad (10)$$

Case 3: Both *A* and *B* will be changed: We notice that this case is similar to the case 1. Therefore, we have:

$$n = MIN\left(total \times (s - t\_s), \left\lfloor \frac{s * total * (c - t\_c)}{c * (1 - t\_c)} \right\rfloor \right)$$

In short, when A is changed, we have:

$$n = MIN\left(total \times (s - t\_s), \left\lfloor \frac{s * total * (c - t\_c)}{c * (1 - t\_c)} \right\rfloor \right),$$

and when B is changed, we have:

$$n = MIN\left(total \times (s - t\_s), \left\lfloor \frac{s * total * (c - t\_c)}{c} \right\rfloor \right).$$

For insignificant association rule *A*→*B*, the algorithm must guarantee that this rule will not become a significant one when performing any changes. When a tuple is altered, the value of its attributes may be changed to other values. These values may be *A* or *B*. Therefore, the support value and confidence value of the rule may be affected and this rule can become a significant rule. In this part, we will calculate the maximal number of tuples which can be altered so that the rule does not become a significant one. *s* and *c* are the support value and the confident value of this rule before performing the change, *s'* and *c'* are the corresponding values after performing the change. *n* is the number of additional tuples. We will consider following cases:

Case 1: *A* will be added:

In this case, we always have:

$$c' = \frac{number(A \to B)}{number(A) + n} \qquad (11)$$

$$c = \frac{number(A \to B)}{number(A)} \qquad (12)$$

where *number(A→B)* is the number of tuples which have both *A* and *B*, *number(A)* is the number of tuples which only have *A*. Clearly, *c'* is smaller than *c*. We also have:

$$s = \frac{number(A \to B)}{total} = s' \qquad (13)$$

where *total* is the number of tuples in the database. Therefore, if only value A will be added, this rule can not become a significant rule.

Case 2: *B* will be added: Similar to the case 1, the rule can not be a significant rule.

Case 3: Both *A* and *B* will be added:

Similarly, we will have:

$$c' = \frac{number(A \to B) + n}{number(A) + n} \qquad (14)$$

$$s' = \frac{number(A \to B) + n}{total} \qquad (15)$$

Moreover, we also have:

$$s = \frac{number(A \to B)}{total} \qquad (16)$$

$$c = \frac{number(A \to B)}{number(A)} \qquad (17)$$

The condition are *s'<=t_s* and *c'<=t_c*. Therefore, we have:

$$n = MIN\left(total \times (t\_s - s), \left\lfloor \frac{s * total * (t\_c - c)}{c * (1 - t\_c)} \right\rfloor \right) \quad (18)$$

In summary, when A and B are added, the maximum number of tuples which we can alter without generating new significant association rules:

$$n = MIN\left(total \times (t\_s - s), \left\lfloor \frac{s * total * (t\_c - c)}{c * (1 - t\_c)} \right\rfloor \right).$$

IV. Algorithm

Clearly, the objectives of the proposed algorithm are to perform the changes to transform unsatisfied groups into satisfied ones, and to maintain the significant association rules. Moreover, the algorithm should also reduce the number of new significant association rules that are created while running the algorithm. We call the maintaining significant association rules and reducing the number of new significant association rules as proposed algorithm's goals. During the anonymization, a group can be in two statuses, receiving tuples or distributing tuples. When we perform a change *a*▯*b* between two groups *a* and *b*, we consider *a* is the group that distributes tuples and *b* is the group that receives tuples. Furthermore, the algorithm should guarantee that the goals will not be violated when the changes are performed.

The algorithm will perform some changes to transform each unsatisfied group into the satisfied one. Thus, for each unsatisfied group, the algorithm will choose a/some group(s), which is the other unsatisfied group or the satisfied group, to form the changes. However, the algorithm does not choose these groups randomly; it will choose the best "compatible" groups so that when performing the changes between the unsatisfied group and these "compatible" groups, they have the least effect on the algorithm's goal. To do this, the algorithm will calculate "cost" for each change. Then it will choose the changes which have the least cost. While seeking these best "compatible" groups, the algorithm should concern the following issues:

- Consider two-way for the changes between two groups. It means the algorithm will consider the changes *a→b* and *b→a* and then choose the best one when considering the changes between group *a* and *b*.
- For each unsatisfied group, the algorithm will choose the changes which have the least effect on the association rules when performing it.
- A group can receive or distribute tuples more than one time.
- A group can receive tuples from different groups.
- Prioritize the combination of two unsatisfied groups when we have some combinations that have same cost.
- For unsatisfied groups, prioritize the receipt of tuples from satisfied groups and the distribution of tuples to another unsatisfied groups.

Moreover, as discussed above, the algorithm should assign a priority degree for each unsatisfied group in order to determine which groups will be processed first. First of all, the algorithm will try to transform unsatisfied groups which have higher priority degree. Then, it will work with the lower ones. In the previous papers [4, 21], their algorithm chose the current transformed unsatisfied group randomly. Therefore, this group may receive all of tuples that are available for distribution and we will not have enough tuples for other unsatisfied groups. As a result, we may get more unsatisfied groups after finishing the algorithm.

For example, we have three unsatisfied groups: first group has 1 tuple, the second group has three tuples and the third one has four tuples. We also have k = 5 and the number of tuples, which are available for distributing, is 3. If the first one is processed first, it will receive all of these tuples and when the other groups are processed, we have no tuple for them. Therefore, we still have three unsatisfied groups after the processing. Conversely, if we process the third group first, it will receive one tuple to guarantee k-anonymity; the second one will be processed then and receives two tuples. Finally, we have two satisfied groups and one unsatisfied group. The second result is better. In this algorithm, we will try to transform many more unsatisfied groups into the satisfied ones by assigning a priority degree for each unsatisfied group. To assign the priority degree for unsatisfied groups, the algorithm will base on criteria:

<u>Criteria:</u>

- Prioritize unsatisfied groups in which the number of tuples is closer to k: because the algorithm will try to receive satisfied groups as many as possible, it will give priority to the unsatisfied groups which are closer to gain the satisfied ones. Clearly, unsatisfied groups, which the number of its tuples is closer to k, will be transformmed to the satisfied ones more easily.
- Prioritize unsatisfied groups which can not distribute tuples.

The algorithm will try to finish the anonymization of current unsatisfied group before working with next unsatisfied groups. An unsatisfied group can be transformed into a satisfied one if one of two following cases can be performed without affecting the goals: (i) all its tuples are distributed to other groups; (ii) it adds some tuples from other groups so that the number of its tuples is greater than k. In the second case, if a great number of tuples can be added to current unsatisfied group without affecting the goals, the group should only add enough tuples. It means that the number of group's tuples after processing should be equal to k. The remaining tuples will be left for other unsatisfied groups which are processed later.

Clearly, the greater the number of unsatisfied groups is, the more slowly the algorithm may run. Therefore, the algorithm should first reduce the number of unsatisfied groups. Moreover, the number of significant association rules is also affect the run of the algorithm because the algorithm always considers these rules during the transformation of unsatisfied groups. In this paper, we also propose the grid based solution to apply to the location attribute of the location database. Normally, when mining the location data, data mining applications usually try to find some valuable values in an area rather than at an exact location. Therefore, the idea of this solution is that the exact location values will be anonymized into grid cells. With this solution, the algorithm will create a grid which covers the space containing the locations of the users in the database. After that, the locations of the users will be anonymized into this grid's cell. We will consider the following example: we have 11 location values which will be anonymized into a grid. The grid, which covers the space containing the locations of all users, is in the following figure (*S* is a starting point):

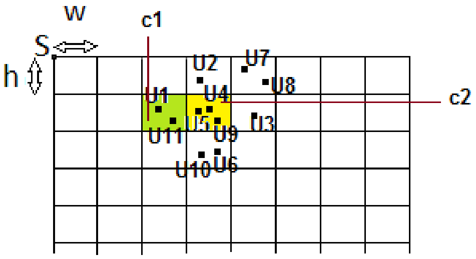


Fig. 1. An example of grid based solution.

The values *U1* and *U11* will be anonymized into cell *c1*, the values *U4, U5, U9* will be anonymized into cell *c2* and so on. Clearly, if we have two association rules *A*→*U1* and *A*→*U11*, they may become an association rule *A*→*c1*. Thus, the number of significant association rules, which need to maintain, can be reduced. Moreover, the number of unsatisfied groups may also be reduced because the number of tuples in each group may be increased. As a result, the algorithm will run more quickly.

The proposed algorithm can be described as the following pseudocode:

Name: *k_anoymization()*

Input: Set *R* includes the significant association rules which need to maintain, *k*, original table *T*, *QI*, the grid cell size.

Output: anonymous version table *T'*

Method:

```
1. Create a grid and anonymize all
   location values into this grid.
2. Construct a set S which contains
   satisfied groups and a set US which
   contains unsatisfied groups.
3. Sort the set US by above criteria.
4. Calculate the number of tuples which
   can be moved for each rule in R
5. a set cannotProcess=⊘, it contains
   groups that can not be transformed into
   a satisfied one.
   While (US is not empty) {
6. Select an unstaisfied group proUS from
   US by its priority degree
7. US = US \ proUS
   While (proUS is unsatisfied group) {
8. Run find_best_can_group() function to
   find a best change to transform proUS.
   A candidate group can and a set of
   tuples W containing tuples, which can
   be anonymized without affecting the
   goals, will be returned by this
   function.
9. Exclude can from US or S
   if (can == null){
10.cannotProcess = cannotProcess U proUS
11.Give back all tuples, which are
   anonymized during the transformation of
   the current unsatisfied group, to their
   original groups.
12.Unmark all examined groups in S and US
13.break;
   } Else {
14.Perform the change.
15.Update the support and confidence
   values of each rule in R
16.Mark can as be examined
17.if(can is satisfied group) S = S U can
18.Else US = US U can
19.S = S U proUS
20.Unmark all examined groups in S and US
   } } }
   if (cannotProcess is not empty) {
21.final_process() }
```

The algorithm will try to transform each unsatisfied group into a satisfied one. For each unsatisfied group, the algorithm will try to finish the transformation for it before working with next unsatisfied groups. After the processing, if the algorithm can not transform this unsatisfied group into a satisfied one, all tuples, which are anonymized during the processing of this unsatisfied group, will be given back to their original groups and this unsatisfied group will be added to the set *cannotProcess*. The algorithm will try to solve this set at the final step.

During the transformation of an unsatisfied group *proUS*, the algorithm will try to find changes which will apply to this unsatisfied group to transform this group into a satisfied one. Each change will have its cost which reflects the effect of this change on the goals. The cost for each change will be calculated in the *find_best_can_group()* function. From the costs of these changes, this function will also find the best changes for current unsatisfied group. A candidate group *can* and a set of tuples *W* containing tuples, which can be anonymized without affecting the goals, will be returned by this function. The set *W* will contain tuples from *can* if we have the change *can->proUS*. Otherwise, *W* will contain tuples from *proUS*. After receiving results from the *find_best_can_group()* function, the algorithm will perform the change, which is in accord with the results, for current unsatisfied group. After performing each change, if the unsatisfied group is not still satisfied, the algorithm will try to find additional changes to transform this unsatisfied group into the satisfied one. If the algorithm can not find any additional changes to transform the group without affecting the goals, this group will be moved to the set *cannotProcess*.

Moreover, as discussed above, in the case we have too more tuples can be added to an unsatisfied group to transform it into the satisfied one, this unsatisfied group should add enough tuples to guarantee k-anonymity. The remains of tuples will be reserved for other unsatisfied groups. Therefore, when the algorithm performs a change for current unsatisfied group *proUS*, following cases will be considered:

– *proUS* receives tuples: if *(the number of tuples in W + the number of tuples in proUS)* is smaller than *k*, all tuples in *W* will be anonymized into *proUS*. Otherwise, the number of tuples in *W*, which will be anonymized into *proUS*, is *(k – number of tuples in proUS)*.

– *proUS* distributes tuples: if *can* is an unsatisfied group and *(the number of tuples in W + the number of tuples in can)* is greater than *k*, the number of tuples in *W*, which will be anonymized into *can*, is *(k – number of can)*. Otherwise, all tuples in *W* will be anonymized into *can*.

Clearly, the most important function in the algorithm is *find_best_can_group()*, which will try to find the best changes to transform current unsatisfied group into the satisfied one. In this function, we will provide 2 thresholds *t_s* and *t_c*. As discussed above, the algorithm will maintain the significant association rules which their support values are greater than *t_s* and their confident value are also greater than *t_c*. Moreover, the algorithm will not generate additional significant association rules, which their support values are greater than *t_s* and their confident value are also greater than *t_c*, during the running of it. This function will be described as the below pseudo code:

Name: *find_best_can_group()*

Input: unsatisfied group *proUS*, threshold *t_s*, threshold *t_c*

Output: a group *can* and a set *W* contains tuples that can be moved, the direction of the change (*proUS->can* or *can->proUS*)

Method:

```
1. A group can = null
   For each group temp from US U S
   (exclude proUS and examined groups) {
2. Calculate the cost for the changes
   proUS-> temp and temp-> proUS
3. Generate set W containning tuples,
   which will not affect the goals when
   anonymizing them.
   }
   If (exist the changes that do not
   violate the goals when performming
   them) {
4. Choose a best change so that: (i) when
   performing it, the goals are not
   violated and (ii) it has the lowest
   cost. The change will include a group
   temp, a set W and a direction which
   determines proUS->temp or temp->proUS
5. Assign can = temp.
   }
   Return can and W
```

With this function, it will calculate cost for each change at first step. Moreover, as mentioned above, we always consider two-way for the changes between two groups, thus, if we have two groups *proUS* and *temp*, the algorithm will consider two changes: *proUS->temp* and *temp->proUS*. The cost, which is calculated, will base on following criteria (Notice that upper criterion has higher priority):

– The number of significant association rules which will be insignificant after performing the change.

– The number of significant association which will be generated after performing this change.

– Danger degree of significant rules after performing the change: for example, a significant rules has *support=0.7* and *confidence=0.6*. Assume that after performing the change number 1, this rule will have *support=0.64* and *confidence=0.53* and after performing the change number 2, the corresponding values will be *support=0.67* and *confidence=0.59*. The change number 2 will be better because it make the rule less dangerous.

– The number of tuples in the set *W*: the algorithm prefers set *W* which has greater number of its tuples because the more the number of tuples in the set *W*, the more satisfied an unsatisfied group.

Intuitively, we will choose the change that has the lowest cost. Moreover, the function should return the set *W* containing the tuples which can be anonymized. As discussed above, if the algorithm chooses the change *proUS ->temp*, *W* will contain some tuples from *proUS*. Otherwise, it will contain tuples from *temp*.

After anonymization, there are some unsatisfied groups which the algorithm can not find the changes to transform these unsatisfied groups into satisfied ones. These groups will be added to the set *cannotProcess*. We also notice that before an unsatisfied group will be added to the set *cannotProcess*, all tuples, which are anonymized during the processing of this unsatisfied group, will be back to their original groups. It means that all groups will return the statuses which they had before transforming current unsatisfied group. In the case the set *cannotProcess* is not empty, the algorithm will run some additional steps to transform groups in this set into satisfied ones, these addition steps are in the *final_process()* function:

– At the first step, the algorithm will try to transform unsatisfied groups, which are in the set *cannotProcess*, into the better groups that are more satisfied than the original group. It also means that the number of tuples in each better group will be closer to *k* or 0. To do this step, the algorithm will choose the best changes, which will not affect the goals when performming them, to transform the unsatisfied group into a better one. The function *find_best_can_group()* can be used to find these best changes in this step.

– At the second step, the algorithm will try to transform these better unsatisfied groups into the satisfied ones. At this step, the goals may be violated.

In order to transform the better unsatisfied groups into the satisfied ones. The algorithm will find changes that have the least effect on the goals. After that, it will perform these changes to transform the better unsatisfied groups into the satisfied ones.

In contrast with the previous steps, the goals will be violated if changes, which are found in the second step, are performed. It means that some significant association rules may be no longer significant and/or new significant association rules may be generated after these changes are performed. This is “cost” which we must pay to guarantee k-anonymity for the database because with these unsatisfied groups, the algorithm can not find any changes to transform them without effect on the goals.

V. Evaluations

In this section, we show the evaluation we conducted in order to evaluate the effectiveness of our algorithms. We will verify the proposed algorithm with three other algorithms: M3AR [4], KACA [20], OKA [19] in both criteria: the percentage of lost significant association rules and the percentage of new significant association rules that are generated during the run of algorithms. Intuitively, the smaller two values, the more effective the algorithm. We call them as *p_s* and *p_n*:

$$p\_s = \frac{l\_r}{t\_r}, \tag{19}$$

where *l_r* is the number of significant association rules that are lost during the run of the algorithm and *t_r* is the total of significant association rules.

$$p\_n = \frac{n\_r}{t\_r}, \tag{20}$$

where *n_r* is the number of significant association rules that are generated during the run of the algorithm and *t_r* is the total of significant association rules.

The real database, which is used for the evaluation, will be extracted from GeoLife project [16, 17], which is collected in (Microsoft Research Asia) GeoLife project by 165 users in a period of over two years (from April 2007 to August 2009) and Adult database from the UC Irvine Machine Learning Repository [18]. This database will include 34827 records. The *QI* will include status, age, sex and location attribute. The grid cell size, which is used to anonymize the location attributes, is 500m*500m. For each value of *k*, we will execute each algorithm in five times; the achieved result is the average of five tests. The following figures show the result of the evaluation.

These results show that with our proposed algorithm, the percentage of significant association rules, which are lost during the run of the algorithm, is minimal. Similarly, the percentage of new significant association rules, which is generated during the processing, is also minimal. It also means that our algorithm will generate an effective k-anonymous version of the database. The reason of these results is that our algorithm tries to transform the unsatisfied groups with the changes that will cause least effect on the goals. Therefore, the result of data mining process may be more effective.

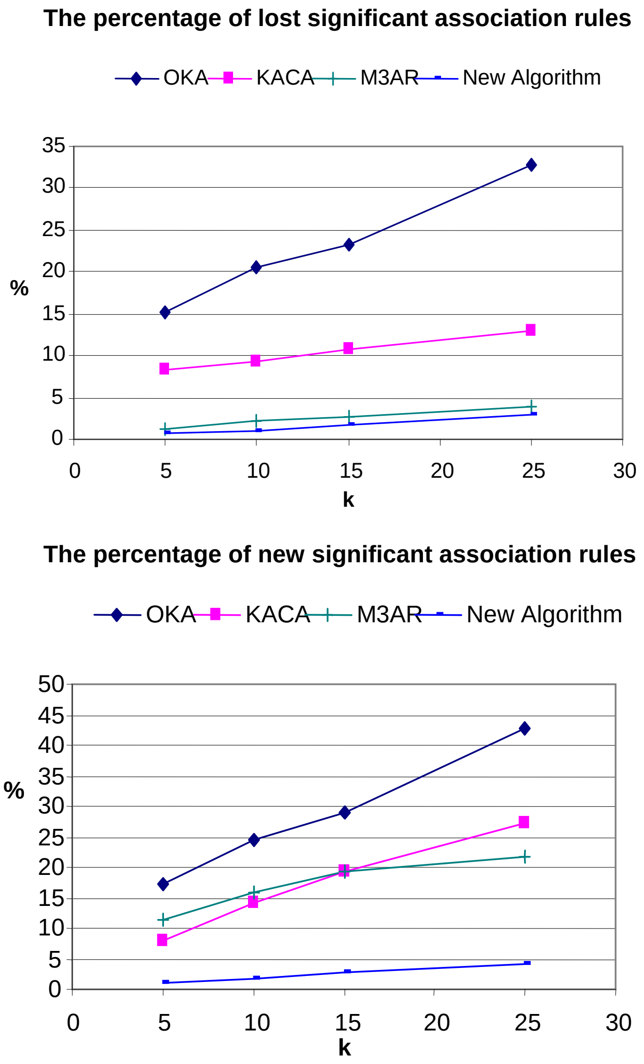


Fig. 2. The evaluation results.

VI. Conclusion and Future Works

In this paper, we proposed an algorithm that anonymizes the location database to an effective k-anonymous version. The algorithm solves some problems in the M3AR algorithm that was proposed before to guarantee k-anonymity for general databases. With the algorithm, the number of significant association rules, that are lost during the anonymization, is reduced. Thus, the results generated by the data mining process, which input data is the k-anonymous version of the database, are more valuable.

The paper also proposed the solution to reduce the significant association rules which are generated during the anonymization. Clearly, if new significant association rules are generated, they may interfere negatively in the results of the data mining process. With the newly proposed algorithm, the number of new significant association rules, which is generated spuriously, is also reduced and hence the result of the data mining process is more effective.

In this paper, we applied the grid based solution to reduce the number of significant association rules and also reduce the number of unsatisfied groups. Thus, the algorithm is more efficient. In the future, we will focus on investigating additional solutions to improve the performance of the algorithm. On the other side, we should assign a priority degree for each unsatisfied group to determine which group will be processed first. The priority degree will be based on some criteria that are mentioned above. We can improve these criteria so that the algorithm can return a more effective k-anonymous version of the database. Moreover, the location of the user is usually accompanied with a time value. Therefore, the algorithm should also consider the time value when anonymizing the location database.

Acknowledgments

This work was supported in part by D-STAR Lab (www.dstar.edu.vn). We appreciate the helpful supports from all members of D-STAR Lab (www.dstar.edu.vn) during this research**.**

References

[1] Sergio, M., Claudio, B., Wang, S.X. and Sushil, J.: k-anonymity in Databases with Time Stamped Data. In: 13th International Symposium on Temporal Representation and Reasoning, pp. 177--186, IEEE Press, Budapest, Hungary (2006).

[2] Truong, T.A., Truong, Q.C. and Dang, T.K.: An Adaptive Grid-based Approach to Location Privacy Preservation. In: 2nd Asian Conference on Intelligent Information and Database Systems, pp. 133--144, Springer Verlag, Vietnam (2010)

[3] Truong, Q.C., Truong, T.A. and Dang, T.K.: The Memorizing Algorithm: Protecting User Privacy in Location-Based Services using Historical Services Information. In: International Journal of Mobile Computing and Multimedia Comm., 2(4), pp. 65--86, IGI-Global (2010)

[4] Dang, T.K, Kueng, J., Huynh, V.Q.P: Protecting User Privacy while Discovering and Maintaining Association Rules. In: 4th IFIP International Conference on New Technologies, Mobility and Security, IEEE Computer Society, Paris, France (2011)

[5] Ciriani, V., De Capitani di Vimercati, S., Foresti, S. and Samarati, P.: k-Anonymous Data Mining: A Survey. In: Michael, G., Sushil, J. (eds.), Handbook of Database Security– Applications and Trends, pp. 105--136, Springer Science, LLC (2008)

[6] Sweeney, L.: Achieving k-anonymity Privacy Protection using Generalization and Suppression. In: International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, 10(5), pp. 571--588, World Scientific (2002)

[7] Samarati, P. and Sweeney, L.: Protecting Privacy When Disclosing Information: k-anonymity and its Enforcement through Generalization and Suppression. Technical Report SRI-CSL-98-04, Computer Science Laboratory, SRI International (1998)

[8] Gedik, B. and Ling, L.: Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. In: IEEE Trans. on Mobile Computing, 7(1), pp. 1--18 (2008)

[9] Gruteser, M. and Grunwald, D.: Anonymous Usage of Location-Based Services through Spatial and Temporal Cloaking. In: ACM International Conference Mobile Systems, Applications, and Services, pp 31--42, ACM New York, USA (2003)

[10] Bettini, C., Mascetti, S. and Wang, X.S.: Privacy Protection through Anonymity in Location-based Services. In: Michael, G., Sushil, J. (eds.), Handbook of Database Security – Applications and Trends, pp. 509--530, Springer Science, LLC (2008)

[11] To, Q.C., Dang, T.K., Kueng, J.: A Hilbert-based Framework for Preserving Privacy in Location-based Services. Intl. Journal of Intelligent Information and Database Systems (IJIIDS), Inderscience Publisher, ISSN 1751-5858 (2012) (*to appear*)

[12] Cuellar, J.R.: Location Information Privacy. In: B. Srikaya (Ed.), Geographic Location in the Internet, pp. 179--208, Kluwer Academic Publishers (2002)

[13] Gidófalvi, G., Huang, X. and Pedersen, T.B: Privacy-Preserving Data Mining on Moving Object Trajectories. In: 8th International Conference on Mobile Data Management, pp. 60--68, Mannheim, Germany (2007)

[14] Bettini, C., Wang, X. and Jajodia, S.: Protecting Privacy against Location-based Personal Identification. In: 2nd VLDB Workshop on Secure Data Management, pp. 185--199, Trondheim, Norway (2005)

[15] Ardagna, C.A., Cremonini, M., Vimercati, S.D.C. and Samarati, P.: Privacy-enhanced Location-based Access Control. In: Michael, G., Sushil, J. (eds.), Handbook of Database Security – Applications and Trends, pp. 531--552, Springer Science, LLC (2008)

[16] Zheng, Y., Li, Q., Chen, Y. and Xie, X.: Understanding Mobility Based on GPS Data. In: ACM conference on Ubiquitous Computing, pp. 312--321, ACM Press, Seoul, Korea (2008)

[17] Zheng, Y., Zhang, L., Xie, X. and Ma, W.Y.: Mining Interesting Locations and Travel Sequences from GPS Trajectories. In: International conference on World Wild Web, pp. 791--800, ACM Press, Madrid, Spain (2009)

[18] Newman, D.J., Hettich, S., Blake, C.L. and Merz, C.J.: UCI Repository of Machine Learning Databases, available at *www.ics.uci.edu/mlearn/MLRepository.html*, University of California, Irvine (1998)

[19] Jun, L.L. and Meng, C.W.: An Efficient Clustering Method for k-anonymization. In: the 2008 International Workshop on Privacy and Anonymity in Information Society, pp. 46--50, ACM New York, Nantes, France (2008)

[20] Li, J., Wong, R.C.W., Fu, A.W.C and Pei, J.: Achieving k-Anonymity by Clustering in Attribute Hierarchical Structures. In: Tjoa, A.M., Trujillo, J. (eds.) Data Warehousing and Knowledge Discovery, LNCS 4081, pp. 405--416, Springer Verlag, Heidelberg (2006)

[21] Huynh, V.Q.P. and Dang, T.K.: eM^2: An Efficient Member Migration Algorithm for Ensuring k-Anonymity and Mitigating Information Loss. In: 7th VLDB Workshop on Secure Data Management, pp. 26--40, Springer Verlag, Singapore (2010)

[22] To, Q.C., Dang, T.K., Kueng, J.: OST-tree: An Access Method for Obfuscating Spatio-Temporal Data in Location Based Services. In: 4th IFIP International Conference on New Technologies, Mobility and Security, IEEE Computer Society, Paris, France (2011)

[23] To, Q.C., Dang, T.K., Kueng, J.: B^{ob}-Tree: An Efficient B+-Tree Based Index Structure for Geographic-aware Obfuscation. In: 3rd Asian Conference on Intelligent Information and Database Systems, LNAI 6591, pp. 109--118, Springer Verlag, Korea (2011)

[24] To, Q.C., Dang, T.K., Kueng, J.: A Hilbert-based Framework for Preserving Privacy in Location-based Services. In: International Journal of Intelligent Information and Database Systems, Inderscience Publisher (2013, to appear)

[25] Le, T.B.T, Dang, T.K.: Semantic Bob-Tree: A New Obfuscation Technique for Location Privacy Protection. In: 10th International Conference on Advances in Mobile Computing & Multimedia, ACM, Bali, Indonesia (2012)

A QoS App-SLO Manager for Virtualized Infrastructure

Fernando Rodríguez-Haro, Felix Freitag, and Leandro Navarro

***Abstract*—The management of infrastructure for supporting Cloud Computing presents the challenge of automated service provisioning, which addresses the problem of mapping high-level requirements expressed in end-user terms to low-level resources such as CPU, memory, and network bandwidth. Current infrastructure is supported through virtualization via hypervisors. In this paper, we describe the formal specification of a high-level component for enhancing hypervisors. With this component, applications running in a Virtual Machine can receive a Quality of Service defined by Service Level Objectives. The manager is aware of the application's needs and requests the CPU resources through the lifetime of the Virtual Machine. The implementation of our proposal achieves to manage computing-oriented and net-oriented applications.**

***Index Terms*—Hypervisor, QoS, SLO.**

I. Introduction

NOWADAYS, virtualization infrastructure is a common solution for supporting Cloud Computing, Grid, and High Performance Computing. An important challenge in these infrastructures is the automated service provisioning of Virtual Machine (VM) based resource providers for the execution of applications. When we review the literature [1], some interesting questions arise from end users willing to deploy applications in VM-based resource providers.

– How can we predict (or have some degree of certainty) that deadline execution time requirements for a given job will be met?

– How can we compute the amount of resources that are needed to increase (or decrease) the number of transactions to a certain required level?

– Moreover, how can we provide the needed resources and at the same time minimize the degradation of the externally-perceived response times?

In this paper, we describe the formal specification of a high-level component for enhancing hypervisors. The component is named QoS App-SLO manager and allows end users to express the application's requirements in terms of a Service Level Objective (SLO).

With our component, applications running in a virtual machine can receive a Quality of Service (QoS) defined by two types of SLOs (slo_2, slo_3).

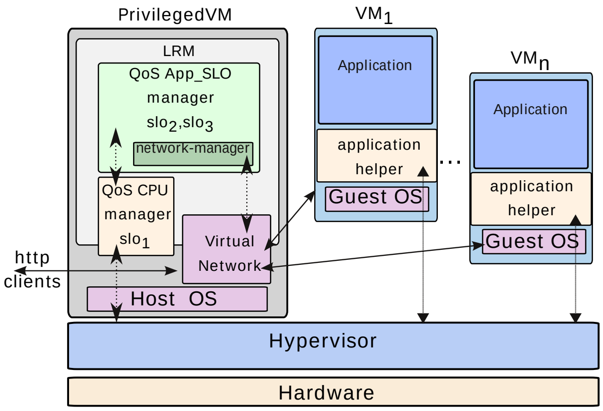


Fig. 1: Interactions of the QoS App-SLO manager in a VM-based resource provider

The manager is aware of the application's needs and requests the CPU resources through the lifetime of the Virtual Machine. The implementation of our proposal achieves to manage computing-oriented and net-oriented applications, i.e. it meets at least the agreed application requirements, and provides self-management for external modifications in the application's SLO (e.g. a user requests more transactions per secod (tps)).

The proposal relies on the services offered by the hypervisor, the host OS, and the low-level component QoS CPU manager [2] which manages the SLO slo_1.

II. QoS App-SLO manager

In this section, we describe a QoS App-SLO manager. Figure 1 shows the interactions of this manager in a VM-based resource provider. It allows handling the high-level requirements of user's applications, i.e. it runs in the scope of the privilegedVM and manages the resources needed by each guest VM.

Additionaly, the definitions of the proposal are build upon two systems to acquire knowledge about information of each application:

– An inter-VM messaging system, which is a communication system that allows writing and reading information in both ways: privilegedVM from/to guest virtual machines.

– An application-helper which runs in the guest virtual machine with an independent period τ_{ah}. It measures

Manuscript received on September 19, 2012; accepted for publication on December 10, 2012

Felix Freitag and Leandro Navarro are with the Department of Computer Architecture (DAC) at the Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, 08034 Barcelona, Spain, and Fernando Rodríguez-Haro is with the University of Colima (e-mail: felix, leandro, frodrigu@ac.upc.edu)

the metric *tps* transactions per second and writes the measured metrics through the inter-VM messaging system.

A. Characteristics

The formal specification that we propose has the following features: Application rate (app-rate) guarantees for granting a requirement expressed in transactions per second tps, user-perceived service guarantees for providing a SLO in (user-perceived) response times, net-rate guarantees for granting a requirement expressed in requests per second rps, and a request admission control for applying a policy in the incoming requests to meet a SLO.

B. Definitions

Definition 2.1: Let ϱ be the set of online guest virtual machines identifiers and let $Apptype_{vm}$ be the sequence indexed by ϱ that defines the type of application that is running in the guest virtual machine so that

$$Apptype_{vm} \in \{\text{computing-oriented}, \text{net-oriented}\} \forall vm \in \varrho.$$

Definition 2.2: Let $slotype_{vm}$ be a set that has the type of resquested SLOs slo_2 or slo_3 for a given vm. Let slo_2 be a service level objective of the form "ensure that the application running in a guest virtual machine vm will be able to achieve an application rate of at least 95% of the target tps or rps". Let slo_3 be a service level objective of the form "ensure that the application running in a guest virtual machine vm will be able to achieve a % of the served requests with response times below a given threshold expressed in seconds". Let $appslo$ be the set of parameters of the requested service level objectives for all managed virtual machines

Inspecting the throughput that was achieved by each computing-oriented application can be done as follows.

Definition 2.3: Let tps_{vm} be the throughput achieved by the computing-oriented application between the time interval Cah_t, Cah_{t-1} and let T be the number of composite transactions measured by the application-helper so that

$$tps_{vm} = \frac{T}{Cah_t - Cah_{t-1}}.$$

C. Network-manager

With this subsystem we aim to measure the burst behaviour of the net-oriented application. Web technologies are based on the transport protocols Hypertext Transfer Protocol (HTTP) and the secure HTTP protocol (HTTPS), both main function is to move data between Web servers and browsers. Despite the fact that it is a stateless protocol it is nowadays the facto transport protocol for technologies based on web services. Common protocols for deployment of web services are SOAP (Simple Object Access Protocol), REST (Representational state transfer), and XML-RPC.

The approach for measuring the throughput of a web application assumes that requests are atomic and represents a unit of work which ends with the successful transfer of the results. As Web servers are the target application, for managing the net-oriented applications we propose the following approach.

Trace the http requests of each VM by inspecting the connection states of the incoming HTTP packets. Depending of the hypervisor [3], [4], tracing could be done in the privilegedVM or the guest VM. Thus, we can measure the response time of each connection and build a distribution array $http$ at every trace period of length $\tau_{http} = Cht_t - Cht_{t-1}$. This array has n elements, each element is a counter of successful (served) requests according to its response time rt. The granularity of n depends of the rt ranges that need to be grouped. We propose a granularity of $n = 9$ which is mapped to the ranges shown in Table I. The metric λ_{vm} is the arrival rate observed in the virtual machine vm for the net-oriented application.

TABLE I: SERVICE TIME GROUPS OF HTTP REQUESTS USED TO BUILD THE DISTRIBUTION ARRAY.

array position	response time range
0	$rt < 1\mu s$
1	$1\mu s \leq rt < 10^{-5}s$
2	$10^{-5}s \leq rt < 10^{-4}s$
3	$10^{-4}s \leq rt < 1ms$
4	$1ms \leq rt < 10^{-2}s$
5	$10^{-2}s \leq rt < 10^{-1}s$
6	$10^{-1}s \leq rt < 1s$
7	$1s \leq rt < 10s$
8	$10s \leq rt < 100s$

In order to measure the metric mean response time of the served requests it makes use of a circular buffer CB with a history length l for each vm so that

Definition 2.4: Let CB_{vm} be the history of response times of the last l served requests for a given vm. Let $meanRT_{vm}$ be the computed mean response time observed for virtual machine vm so that

$$meanRT_{vm} = \frac{\sum_{k=l}^{1} CB_{vm,k}}{l} \forall vm \in \varrho.$$

Using the distribution array $http$ the inspection of throughput achieved by each net-oriented application can be done as follows.

Definition 2.5: Let rps_{vm} be the throughput achieved during the last period τ_{http} and R the number of successful completed requests measured by the *network-manager* so that

$$R = \sum_{k=1}^{n} http_{vm,k},$$

$$rps_{vm} = \frac{R}{Cht_t - Cht_{t-1}}.$$

Finally, the metrics are advertised through the inter-VM messaging system.

D. App-rate metrics collector

It is in charge of getting the observed metrics of the running application. It interacts with the inter virtual machine messaging system and the network manager. Its function is to keep a snapshot of the measured metrics for the learning component.

Definition 2.6: Let TPSapp$_{vm}$ be the set of measured metrics app-rate expressed in requests/transactions per second so that TPSapp$_{vm} = m(vm)$

$$m(x) = \begin{cases} tps_{vm} & Apptype_{vm} = \text{computing-oriented} \\ rps_{vm} & Apptype_{vm} = \text{net-oriented} \end{cases} \quad (1)$$

E. Learning component

Its function is to compute online parameters that profiles the current application rate. We use a multi queue system. Each virtual machine is modeled using Little's law from queueing theory.

By having the CPU consumption of each virtual machine and the application rate we obtain the service demand SD as follows

Definition 2.7: Let $avgmets$ be a set of mean CPU metrics for each virtual machine computed by the QoS CPU manager during the previous controller period. Let SD_{vm} be the mean CPU time spent per transaction/request during the previous period

$$SD_{vm} = \frac{avgmets_{vm}}{TPSapp_{vm}}.$$

In order to obtain the learned service demand we apply a forecasting method. First, we obtain a trend of the pasts service demands by applying exponential moving average (EMA) [5] which technically can be classified as an Auto-Regressive Integrated Moving Average ARIMA(0,1,1) model with no constant term [6]. Second, the method enhances the trend by measuring the volatility of the sampled metrics using a trading mechanism with a configurable parameter ϖ. We propose to apply Bollinger bands [7] in order to capture the burst behavior of the running applications and improve the reactiveness of the QoS CPU manager.

Definition 2.8: Let $SDtrend$ be an Exponential Moving Average function to compute the trend for the service demand of a given vm so that $SDtrend_{vm,t}$ is defined as follows

$$SDtrend_{vm,t} = SD_{vm,t}, t = 0,$$

$$SDtrend_{vm,t} = \alpha * SD_{vm,t} + (1-\alpha) * SDtrend_{vm,t-1}, t > 0.$$

Definition 2.9: Let N be the length of history needed to forecast the next service demand. Let $SDforecast$ be a forecasting function to compute the next service demand of a given vm so that $SDforecast_{vm,t}$ is defined as follows

$$\sigma_{vm,t} = \sqrt{\frac{1}{N}\sum_{i=t}^{t-N}(SD_{vm,i} - \overline{SD_{vm}})^2},$$

$$\overline{SD_{vm}} = \frac{1}{N}\sum_{i=t}^{t-N} SD_{vm,i},$$

$$SDforecast_{vm,t} = SDtrend_{vm,t} + \varpi * \sigma_{vm,t} \qquad 0 \leq \varpi \leq 2.$$

The service demand forecasted is a snapshot of the needed resources for the next controller period. However, we introduce the notion of Number of Rounds To Learn $NRTL$ parameter in order to find out a tradeoff between reactiveness and disturbance. A $NRTL = 1$ means that the $SDforecast_{vm,t}$ will be used at each controller period in order to compute a new CPU requirement for the application, the reactiveness of the learning phase is high but the accuracy of the forecast is affected by the disturbances of so frequent changes in the slo_1 (CPU resource). On the contrary, for $NRTL > 1$ we introduce the notion of learning phase (or window) which helps in the smoothing of the service demand forecasted and also improves the accuracy of computed values.

Definition 2.10: Let NRTL be the length of the controller window needed to learn a smoothed value of $SDforecast$. The length of the period of each NRTL window is given by controller period times. Let NRTL be a counter which decreases at each controller period.

Now we use an approach to find out the burst behaviour of the requests in the net-oriented application. It is proposed to use an array of percentiles $perc$ of $p = n - 1$ elements (see section II-C) for each vm with a net-oriented application. Each element has a circular buffer of length $NRTL$. The position of the element in the array accounts, in the circular buffer, the percentage of requests that were served below the threshold defined in the position of each element (response times) in the array $http$.

Definition 2.11: Let $perc$ be an array of circular buffers. Each circular buffer of length NRTL for each virtual machine vm so that

$$\text{put}\left(\frac{\sum_{i=0}^{j} http_{vm,i}}{rps_{vm}}\right) \text{in}\, perc_{vm,j}, 1 \leq j \leq n.$$

For instance, $perc_{vm,6}$ is a circular buffer with the percentage of requests that were successful server below 1 second, see table I.

The QoS CPU manager uses β to set the reactiveness of the manager in order to climb and achieve the requested slo_1 for each vm. The default value of the parameter β is set to 0, though can be dynamically configured by the type of application running in the virtual machine, e.g. computing intensive applications have a value of 0. However, net-oriented applications need mechanisms to detect the behavior of bursty applications and configure a properly value of β. Therefore, we use $perc$ to measure the bursty behavior as follows.

Definition 2.12: Let β be the degree of burst behaviour detected in the (served) requests of the traced application running in the virtual machine vm so that

$$burst_{vm} = \sum_{j=1}^{n} \sigma_{vm,j},$$

$$\sigma_{vm,j} = \sqrt{\frac{1}{N} \sum_{i=1}^{NRTL} (perc_{vm,i} - \overline{perc_{vm}})^2},$$

$$\beta_{vm} = b(burst_{vm}) \forall vm \in \varrho \quad \text{iff NRTL} = 0.$$

$$b(x) = \begin{cases} 0.5 & 0.00 \leq x \leq 0.03 \\ 0.6 & 0.03 \leq x \leq 0.06 \\ 0.7 & 0.06 \leq x \leq 0.09 \\ 0.8 & 0.09 \leq x \leq 0.12 \\ 0.9 & 0.12 \leq x \leq 0.15 \\ 1.0 & 0.15 \leq x \end{cases} \tag{2}$$

F. CPU-rate estimator

When a learning phase ends, that is $NRTL = 0$, this component sets the $newQoS$ estimated for the running application. The approach of the process involves asking an increase or decrease in the amount of assigned resources, which in fact is and admission control procedure. Due that higher decisions (such as migration of VMs) are leave to the global resource manager, the CPU-rate estimator works as follows.

- It stores the forecasted service demand.
- It stores the modification of the QoS.
- It computes and stores the percentage of QoS granted with the current state of the CPU resources.
- It uses the notion of premium services, via differentiated service, to prioritize the assignment of the CPU resources.

The CPU-rate estimator computes the needed raw CPU power for all virtual machines with computing-oriented and net-oriented applications defined in the sets tps and rps. The needed raw CPU power is transformed into a SLO of type slo_1 and requested via the QoS CPU manager.

Definition 2.13: Let S be the current state (available resources) of the CPU capacity of the resource provider at time t. Let ϑ be a parameter required by the end user for the SLO which express the degree of tolerable (soft,...,hard) reduction in the requested slo_1. Let ac an admission control mechanism that is defined in the QoS CPU manager as follows:

$$ac(MHzSLA, \vartheta) = \begin{cases} 1 & MHzSLA \leq S_t \bigwedge \vartheta = 1 \\ \frac{S_t}{MHzSLA} & MHzSLA * \vartheta \leq S_t \bigwedge 0.5 \leq \vartheta < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

This function helps to manage the admission control of new virtual machines and online virtual machines that request internal updates of slo_1, i.e. all virtual machines with $ac(MHzSLA, \vartheta) = 0$ can be rejected and the status is informed to the QoS App-SLO manager.

Definition 2.14: Let $\hat{S}_\varrho$ be an array of virtual machines ordered by differentiated service. Let *mhz* be the needed CPU in order to achieve a given application rate (i.e. MHz to serve the target reference $appslo$ or λ) for a given vm. Let $newQoS$ be the CPU that is granted by the QoS CPU manager according to ϑ, if is not specified then $\vartheta = 1$. Let φ be the configurable node capacity, and let Φ be the raw CPU capacity of the resource provider. Let ψ the minimum reservation of φ for the guest virtual machines. Let Ξ be the absolute CPU capacity of the node, i.e. for 4 processors $\varphi = 4 * 100 = 400$. The sets $appslo$ and $slotype$ were defined in Def. 2.2. Now, we define $negotiateslo1$ as the function that sets the raw CPU (in % of φ) for each VM. In other words, an slo_1 is computed so that the running application receives slo_2 or slo_3.

$$newQoS_{vm} = negotiateslo1(mhz(vm), \vartheta) \forall vm \in \hat{S}_\varrho \quad \text{iff NRTL} = 0 \tag{4}$$

$$negotiateslo1(MHzSLA, \vartheta) = max(\frac{MHzSLA * ac(MHzSLA, \vartheta)}{\varphi}, \psi) \tag{5}$$

$$mhz(vm) = \begin{cases} \Phi * min\left(\frac{appslo_{vm} * SDforecast_{vm,t}}{\Xi}, 1.0\right) \\ \quad \text{if } vm \in \text{tps} \bigwedge slotype_{vm} = slo_2 \\ \Phi * min\left(\frac{appslo_{vm} * SDforecast_{vm,t}}{\Xi}, 1.0\right) \\ \quad \text{if } vm \in \text{rps} \bigwedge slotype_{vm} = slo_2 \\ \Phi * min\left(\frac{\lambda_{vm} * SDforecast_{vm,t}}{\Xi}, 1.0\right) \\ \quad \text{if } vm \in \text{rps} \bigwedge slotype_{vm} = slo_3 \end{cases} \tag{6}$$

In the third case of Equation 6, we have introduced the notion of automatic sizing for net-oriented applications which request an slo_3. In this case, the requested slo_3 takes into account the observed arrival rate λ of each traced net-oriented application and it increases or decreases its demand according to the observed behaviour.

G. Net-rate estimator

For net-oriented applications we follow an approach to implement an admission control mechanism. It is applied at the end of a learning phase. Only if the target net-oriented application has a computing-oriented behaviour it is likely, applying queueing theory, to find a relation between the number of requests served and the CPU consumed. However, net-oriented applications have a burst behaviour with different resource consumption patterns. Therefore, we propose to use the user-perceived service as a measure of the quality served by the net-oriented application. Even if it can be seen as a black-box that ignores the inner bottlenecks which can cause a bad perceived service, with this approach we aim to size the resources according to the current configuration of the net-oriented application and scale up the aggregation of VMs. We assume that a load balancer can manage the external requests and distribute them to the online virtual machines, thus by aggregating VMs we increase the number of served requests. However, additional inner optimizations in the configuration of the web application server can be applied out of the band and the effect of this optimizations will be seen as an increase/reduce of the resources assigned. Finally, the $capacity$ is granted according to the differentiated service.

Definition 2.15: Let γ be the CPU resources, expressed in percentage of the full node capacity, that are available for all VMs. Let $capacity_{vm}$ be the granted capacity, expressed in rps, of the virtual machine so that

$$capacity_{vm} = n(vm) \forall vm \in rps.$$

$$n(vm) = \frac{newQoS_{vm} * \gamma}{SDforecast_{vm,t}} \tag{7}$$

Now we compute the admission control parameter for the net packets. We use the metrics of the network manager, i.e. the dynamics of the external arrival rate of the net-oriented application clients and the queue length of the current pending requests for each traced net-flow that was observed during the last controller period.

Definition 2.16: Let $perc_{vm,6}$ be a circular buffer with the percentage of requests that were successful server below 1 second. Let $slo_{vm,target}$ the service level objective requested by the user. Let Δ_{slo} be the adjustment positive/negative in the number of requests per second admitted to reach the virtual machine. Let $reqadmission_{vm}$ be the required admission control parameter that limits the amount of accepted requests.

$$abovecapacity_{vm} = \lambda_{vm} - rps_{vm},$$

$$slo_{vm,level} = \left(\frac{\sum_{i=0}^{NRTL} perc_{vm,6}}{NRTL} \right),$$

$$\Delta_{vm} = (slo_{vm,level} - slo_{vm,target}) * abovecapacity_{vm},$$

$$reqadmission_{vm} = capacity_{vm} + \Delta_{vm}.$$

Next step is to adjust the admission control according to the number of waiting requests in the system.

Definition 2.17: Let QL be an array of circular buffers. Each circular buffer for each virtual machine vm has a length of NRTL elements and it stores the observed queue length of waiting/pending requests inside the system during the last controller period. Let maxQL a sort-term memory value of the maximum queue length observed in the NRTL samples. Let $admission_{vm}$ be the rps that will be admitted in the next learning phase.

$$\text{put}\,(ObservedQL_{vm,t})\ \text{in}\, QL_{vm,t}, \forall vm \in rps,$$

$$admission_{vm} = steadystate(vm) \forall vm \in rps.$$

$$steadystate(vm) = \begin{cases} max\,(reqs_{vm}, reqadmission_{vm} - maxQL) \\ \qquad \text{if } \frac{\text{QL}_{vm}}{reqs_{vm}} > 0.05 \\ max\left(reqs_{vm}, reqadmission_{vm} - \frac{maxQL}{2}\right) \\ \qquad otherwise \end{cases} \tag{8}$$

The next step is the application of the network level admission-control through the network manager. By controlling the admission of incoming net packets before arriving the target application we aim to ensure a given level of user-perceived service in the response times. Therefore, with this approach we do not only size the resources according to a given external demand but also according to an expected level in the quality of service.

III. Evaluation

We implement the abovementioned proposal in a Local Resource Manager (LRM) to test the resource management of computing-oriented and net-oriented applications. In the following, we have the characteristics of the QoS App-SLO manager.

- Application-aware. It uses inspection of high-level application metrics in order to learn the CPU needed (slo_1) to achieve app-rate level requirements (tps, rps).
- Service negotiation. It acts on behalf of administrators in order to request the new slo_1 through the QoS CPU manager interfaces. However, it depends on reservations (leases) to grant or revoke the assigned resources as well as policies to detect and limit the misbehavior of virtual machines. If a new slo_1 can not be fully granted then it is informed via a VMstate information system. The VM-state agent is in charge of informing about this issue to external agents so that decisions about the migration of virtual machines can be managed by, for instance, global resource managers.
- Learning. It acquires online knowledge about the consumed resources. It also constructs a CPU profile for the resource consumed by the running application.

We set up three experiments in two physical machines with Fedora Linux and Xen interconnected through a gigabit switch, both with Intel Quad CPU Q6700 with 8GB of memory and a 750GB SATA Disk.

A. Evaluation of computing-oriented applications

The experimental setup test the following festures:

- Create four virtual machines which request, as a bootstrap mechanism, different CPU-rate SLO guarantees.
- Deploy a math-application in all VMs.
- Handle virtual machines which request different app-rate guarantees with fixed differentiated services.

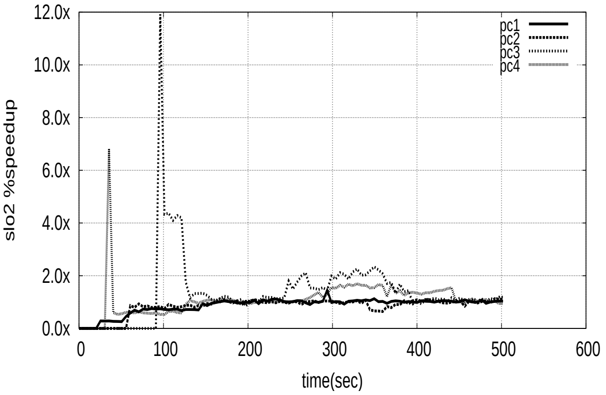


Fig. 2: Application throughput relative to agreed SLO. An user's view of the application performance.

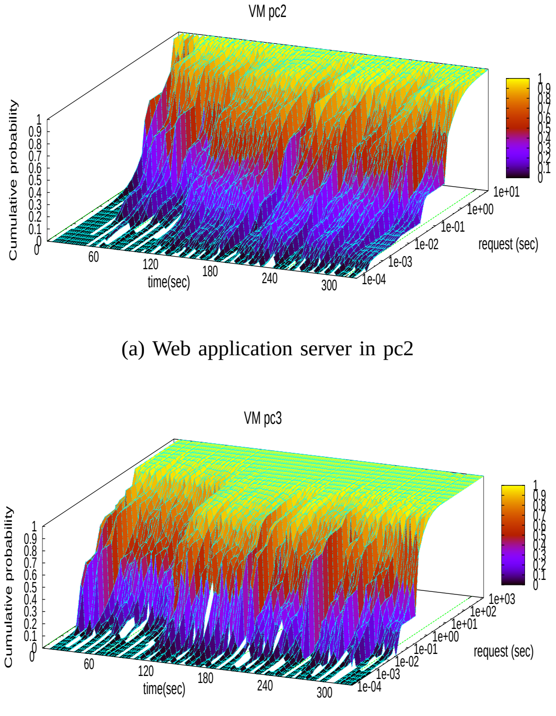
– Provide dynamic management by responding to external agents that change the initial VM's differentiated service (pc3 and pc4).
– Provide dynamic management by responding to external agents that change the initial VM's app-rate.

The agreed app-rate requirement is managed according to each requested parameter. The initial CPU-rate parameter is an initial guess of the needed resources though it can be obtained from previous executions. The learning procedure obtains and requests new CPU-rates which are managed according to their respective differentiated service. Applications can benefit from having hard and soft guarantees about the expected performance (tps). Additionally, each user has a real view of its application throughput. Figure 2 shows results for this experiment. The speedup graph represents the transactions per second relative to its agreed app-rate. From the results for the SLO type 2 (slo_2) we obtained the following mean relative errors: pc1 -0.01, pc2 -0.01, pc3 -0.07, and pc4 -0.03. Therefore, we can observe that each VM achieves its agreed SLO, and additionally we can observe that VMs with premium services receive their corresponding aggregated resources, thus they achieve a better throughput.

B. Evaluation of net-oriented applications

We setup a web application server that renders 3D images and a web image server in two VMs, and for http benchmarking we use Siege [8]. Figure 3 shows the results of this experiment. In this experiment we see the controller changing the CPU resources of two web-based services which have different resource intensive requirements. Despite that pc3's workload is network intensive with low CPU consumption and pc2's workload is CPU intensive with low network consumption, the controller is able to manage both type of applications. At the same time, the network QoS policy adjusts the acceptance in the number of allowed connections that can reach each VM.

A closer look in Figure 3(b) shows that after time 95s more than 90% of the requests are served by the web image server with a response time below one second. The same behaviour is observed for the web application server in Figure 3(a).



(a) Web application server in pc2

(b) Web image server in pc3

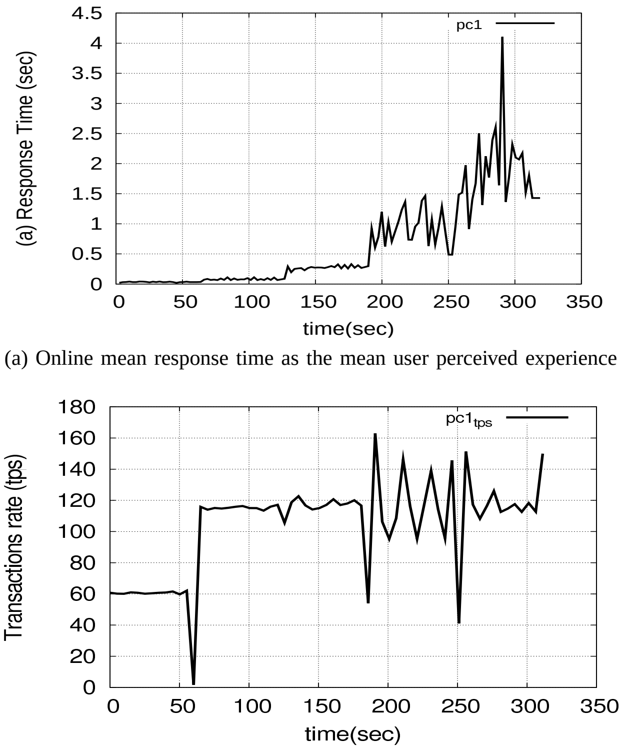
Fig. 3: Distribution vector of served requests. Http packets are controlled.

C. Automatic sizing

We test in this experiment a net-oriented application, i.e. a virtual machine with the web application server described in the previous experiment. The goal is to evaluate that the QoS App-SLO manager is able to find out the VM's resource configuration parameters so that the application, at any moment, can reach its maximum throughput and at the same time meet the user's perceived service-time requirements slo_3.

First, as a baseline experiment, we evaluate the web application server without our QoS App-SLO manager, for this experiment we launch the benchmarking tool Siege with an incremental load in the number of simulated web clients (2,8,32,128,256). Each incremental load has a think-time equal to zero and a duration of 60secs. The results show the saturation points which can be observed in metrics CPU consumption and successful requests. The surges correspond to the start and end times of the load generated by the benchmarking tool, each one with a duration of 60secs.

The maximum transaction rate is achieved with 8 clients starting at second 61 and the saturation point of response times at second 190 with 32 clients, which can be seen more clearly in Figure 4(b) and Figure 4(a) respectively. Therefore, the



(a) Online mean response time as the mean user perceived experience

(b) Online application metrics

Fig. 4: Application web server.

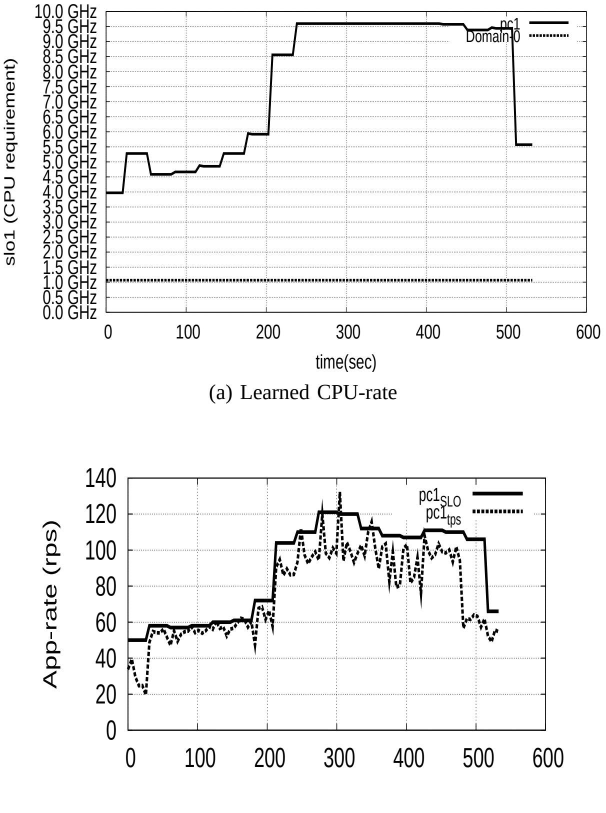
mean response times perceived by remote users start climbing at second 190.

In the second part of this experiment we enabled the QoS App-SLO manager in order to evaluate its automatic sizing capability. The workload used in this experiment has an incremental/decremental traffic pattern with the following number of remote clients: 2,8,32,128,256,128,32,8,2; each set of clients has a duration of 60secs giving an experiment of length 540secs.

The results can be seen in Figure 5 and Figure 6. Figure 5(a) shows that automatic sizing allows requesting CPU resources taking into account the dynamics of the observed requests. Therefore, it is achieved that the resources assigned to the VM can grow or shrink by tracing platform-independent metrics (http requests).

We achieve to trace accuracy http metrics that allow us to keep track of the pending requests in the system. Figure 6(b) shows the queue length of the mean pending requests observed during the last controller period.

We observe that, when managed, the queue of pending requests is less than when there is not admission control. Finally, the admission control applies the requested policy in order to meet the slo_3. Figure 6(a) compares the mean response times of the web application server as seen by the end-user. When managed, the response times are kept below 1 second.



(a) Learned CPU-rate

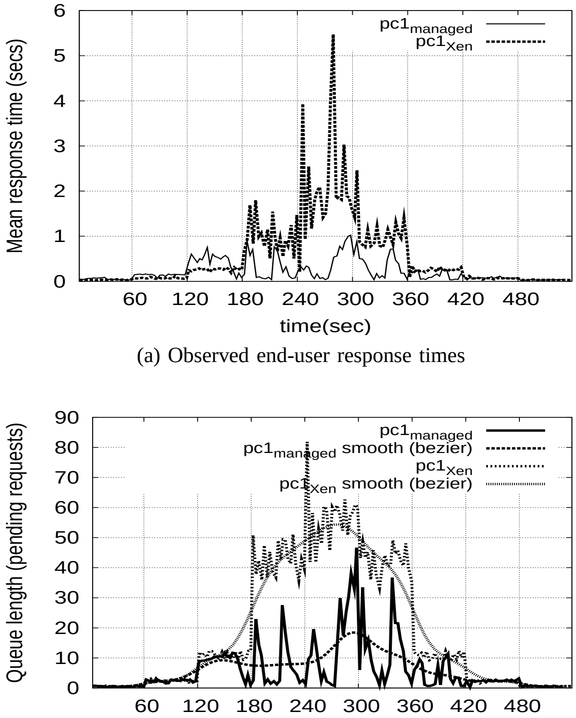
(b) App-SLO manager

Fig. 5: Automatic sizing of an application web server.

IV. Related work

Dongyan Xu *et al.* [9], [10] identified the following challenges that arise in realizing the vision of an *autonomic virtual environment adaptation* in a multi-domain share infrastructure:

1) *Live adaptation mechanisms*: The need to support application-transparent adaptation of Virtual Distributed Enviroments (VDEs). VMs supports runtime resource re-allocation and VM migration within LANs but, a multi-domain infrastructure needs live migration across networks domains without pausing or checkpointing the application. The solution has to meet two requirements: VMs need to retain the same IP address and remain connected to each other and migration mechanism cannot relay on NFS.
2) *Logistic service for VM migration*: Consisting of distributed *depots*. A depot is part of a infrastructure domain, in it, VM images are assembled using either local or transfered "parts". Optimization problem: how to compute a distributed schedule for VM parts delivery and assembly so that all VMs will be ready in their destination hosts no later than a certain deadline?.
3) *Adaptation decision making*: Mechanisms for monitoring, controling, and adjusting resource allocations and



(a) Observed end-user response times

(b) Observed rps in-process

Fig. 6: Comparing traced metrics of an application web server

locations of VDEs. The identified problems are related to find out when an application needs more resources to perform well (or better), how to conciliate when adaptation affects the virtual environments sharing the same resources?, and migration issues, i.e. decide which virtual environment and where should it go by solving tradeoffs between resource availability and overhead.

4) *Adaptation shepherding*: a intelligent component that takes decisions ("justify and approve") regarding adaptation requests, as a mechanism for preventing the abuse of adaptations.

The factors that drive the adaptation of *VDEs* are: availability of infrastructure resources that are dynamic and heterogeneous, and (2) the changing resources needs of the applications that run in a virtual environments.

Paul Ruth *et al.* [11] presents VioCluster, virtualization for dynamic computational domains. The problem is that each computational domain (e.g. cluster) faces the conflict between dynamic workload and static capacity. An opportunity to arises to resolve this conflict by dynamically adapting the capacity of clusters by borrowing idle machines of peer domains. Authors introduce the concept of *virtual computation domains* (or "*virtual domains*" for short) which allow a cluster to dynamically grow and shrink based on resource demand. VioCluster uses both machine and network virtualization techniques to logically move machines between virtual domains.

The HPC research community is particularly interested in using VMs. However, the main concerns widely discussed are the overhead caused by the virtualization layer, and the security [12], [13], [14]. On the other hand, large-scale scenarios such as HPC will benefit from fine-grain management tools to assign CPU resources.

The work of Kephart *et al.* [15] discusses the importance of self-management systems in the context of autonomic computing. These systems accept high-level objectives from administrators and apply self management policies.

Policy-based QoS control and learning have been proposed in non-VM contexts. Solutions based on QoS guarantees have been discussed using control theory [16], [17], online analytic performance models [18], regression-based analytic models [19], and statistical inference [20]. Some applications of these approaches are dynamic provisioning [21] and energy conservation [22]. With our proposal, we aim to provide a framework to meet the low-level VM requirements for the dynamic workload of the hosted application.

V. Conclusion

We have presented the conceptual design and theoretical foundation of a Quality of Service App-SLO manager which is in charge of managing the application goals.

The proposal has a set of definitions that captures the properties for the management of two types of SLOs: slo_2 expressed in the app-rate metric transactions (or requests) per second, and slo_3 expressed in the web metric response times. Additionally, the manager supports an admission control mechanism for the management of net-oriented applications.

Through experiments, we have presented results for different types of workloads: a math parallel application and a web-based application. We evaluated the management of two service level objectives: application rate, and response times.

The results show that the component is able to concurrently manage mixed workloads with their specific application's objectives at different levels with mixed workloads.

Future work includes extending the capabilities of the proposal to support distributed applications in Cloud Computing environments.

Acknowledgment

This work is supported in part by the Computer Architecture Department of the Technical University of Catalonia, the Ministry of Education of Mexico, the program PROMEP of the Public Education Secretary under Contract PROMEP/103.5/10/7336, and the University of Colima.

References

[1] L. Cherkasova, D. Gupta, and A. Vahdat, "When virtual is harder than real: Resource allocation challenges in virtual machine based IT environments," Hewlett Packard Laboratories, Tech. Rep. HPL-2007-25, Feb. 20 2007. [Online]. Available: http://www.hpl.hp.com/techreports/2007/HPL-2007-25.pdf

[2] F. Rodríguez-Haro, F. Freitag, and L. Navarro, "Enhancing virtual environments with qos aware resource management," *Annals of Telecommunications*, vol. 64, pp. 289–303, 2009, 10.1007/s12243-009-0106-1. [Online]. Available: http://dx.doi.org/10.1007/s12243-009-0106-1

[3] VMWare, "Understanding Full Virtualization, Paravirtualization, and Hardware Assist. Whitepaper," 2007, http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf.

[4] M. MSDN, "Hyper-V Architecture," 2008, http://msdn.microsoft.com/en-us/library/cc768520.aspx.

[5] "NIST/SEMATECH e-handbook of statistical methods," 2006, url-http://www.itl.nist.gov/div898/handbook/.

[6] R. F. Nau, "Introduction to ARIMA: nonseasonal models," 2005. [Online]. Available: http://www.duke.edu/~rnau/411arim.htm

[7] J. Bollinger, "Bollinger bands," 2012, http://en.wikipedia.org/wiki/Bollinger_bands.

[8] Siege, "An http regression testing and benchmarking utility," 2012. [Online]. Available: http://www.joedog.org/JoeDog/Siege

[9] D. Xu, P. Ruth, J. Rhee, R. Kennell, and S. Goasguen, "Short paper: Autonomic adaptation of virtual distributed environments in a multi-domain infrastructure," in *15th IEEE International Symposium on High Performance Distributed Computing (HPDC'06)*, June 2006, pp. 317–320.

[10] P. Ruth, J. Rhee, D. Xu, R. Kennell, and S. Goasguen, "Autonomic live adaptation of virtual computational environments in a multi-domain infrastructure," in *IEEE International Conference on Autonomic Computing, 2006. ICAC '06*, 2006, pp. 5–14.

[11] P. Ruth, P. Mcgachey, and D. Xu, "Viocluster: Virtualization for dynamic computational domains," *Proceedings of the IEEE International Conference on Cluster Computing (Cluster'05)*, 2005.

[12] L. Youseff, R. Wolski, B. C. Gorda, and C. Krintz, "Paravirtualization for hpc systems." in *ISPA Workshops*, 2006, pp. 474–486.

[13] M. F. Mergen, V. Uhlig, O. Krieger, and J. Xenidis, "Virtualization for high-performance computing," *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 2, pp. 8–11, 2006.

[14] W. Huang, J. Liu, B. Abali, and D. K. Panda, "A case for high performance computing with virtual machines," in *ICS '06: Proceedings of the 20th annual international conference on Supercomputing*. New York, NY, USA: ACM Press, 2006, pp. 125–134.

[15] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[16] T. Abdelzaher, J. Stankovic, C. Lu, R. Zhang, Y. T. A. Lu, J. Stankovic, C. Lu, R. Zhang, and Y. Lu, "Feedback performance control in software services," *Control Systems Magazine, IEEE*, vol. 23, no. 3, pp. 74–90, 2003.

[17] J. L. Hellerstein, "Challenges in control engineering of computing systems," IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Research Report RC23159 (W0309-091), sep 2003.

[18] D. A. Menascé, M. N. Bennani, and H. Ruan, "On the use of online analytic performance models in self-managing and self-organizing computer systems," in *Self-star Properties in Complex Information Systems*, ser. Lecture Notes in Computer Science, Ö. Babaoglu, M. Jelasity, A. Montresor, C. Fetzer, S. Leonardi, A. P. A. van Moorsel, and M. van Steen, Eds., vol. 3460. Springer, 2005, pp. 128–142. [Online]. Available: http://dx.doi.org/10.1007/11428589_9

[19] Q. Zhang, L. Cherkasova, and E. Smirni, "A regression-based analytic model for dynamic resource provisioning of multi-tier applications," in *ICAC '07: Proceedings of the Fourth International Conference on Autonomic Computing*. IEEE Computer Society, 2007, p. 27.

[20] L. Bertini, J. C. B. Leite, and D. Mosse, "Statistical qos guarantee and energy-efficiency in web server clusters," in *ECRTS '07: Proceedings of the 19th Euromicro Conference on Real-Time Systems*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 83–92.

[21] B. Urgaonkar and A. Chandra, "Dynamic provisioning of multi-tier internet applications," in *ICAC '05: Proceedings of the Second International Conference on Automatic Computing*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 217–228.

[22] C.-H. Tsai, K. G. Shin, J. Reumann, and S. Singhal, "Online web cluster capacity estimation and its application to energy conservation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 932–945, 2007.

A Novel Method of Beamforming to Improve the Space Diversity

Marco Antonio Acevedo Mosqueda, Emmanuel Martínez Zavala,
María Elena Acevedo Mosqueda, and Oleksiy Pogrebnyak

***Abstract*—At present, the systems of mobile communications demand major capacity in the services, therefore optimum space diversity is needed. In this paper, we propose a new method of beam-forming to improve the space diversity and achieve frequency reuse. Our proposal consists in solving a system of linear equations from an 8-elements linear antenna array that increases the size of the side lobes. This method is compared with the algorithm of Dolph-Chebyshev. Our method shows competitive results in increasing the space diversity.**

***Index Terms*—Beamforming, Space diversity, Linear antenna array, Dolph-Chebyshev.**

I. Introduction

In recent decades, the interest in the antenna array has grown significantly due to it plays an important role in mobile communications systems. Now these systems demand more services due to in the constant increase in the number of users, and as a possible solution to this problem several studies have focused on a new mechanism known as division multiple access space (Space Division Multiple Access, SDMA) [1–7] which provides the ability to reuse resources (frequency bands, slots time, codes or any combination) simultaneously within the same service area. The goal of this paper is to propose a method to optimize the space diversity of a linear antenna array. A new technique of solving linear equations of a linear array of 8 elements to increase the magnitude of the side lobes and improve space diversity is proposed. This method is compared with the array factor of Dolph-Chebyshev [8] which is used to improve space diversity and it is considered important because it has the largest amplitude of side lobes around the main lobe side.

In Section II the method for designing a weighted and symmetrical array factor (AF) [9] is described. The array has a total of elements equal to *8* and an equidistant separation between the elements of *d= λ/2.* Section III will show the algorithm for calculating the factor of Dolph-Chebyshev array [8], with a Chebyshev polynomial order *m = 8* and a side lobe level *R = 15* dB [10].

The radiation pattern of Section II is used for a first comparison with the pattern of Dolph-Chebyshev radiation in Section III.

The method to increase secondary lobes is described in Section IV and finally, Section V presents different simulations to obtain a radiation pattern by increasing the magnitude of side lobes, which is comparable to the radiation pattern of Dolph-Chebyshev which has a polynomial order of *m = 8* and a maximum ratio of 4 dB side lobes.

II. Array factor for linear weighted uniform symmetrical arrays

An antenna array is a set of simple antennas which are connected under certain conditions, and they are usually equal and oriented in the same direction. They are arranged in a specific physical layout and relatively close with respect to each other. The combination of all these parameters provides a unique radiation pattern with desirable features such as width of the main lobe, the location of zero and maximum angles, among others [11].

Figure 1 shows the linear weighted uniform symmetrical array used in this work.

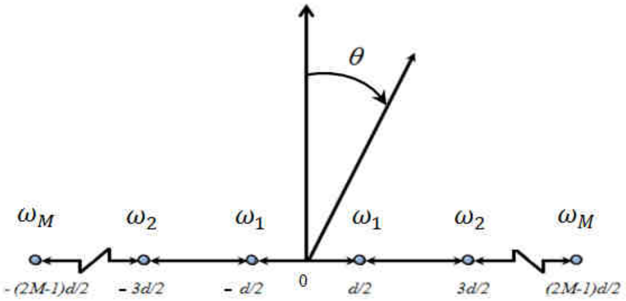


Fig 1. Linear weighted uniform symmetrical array.

In Figure 1, d is the distance between adjacent elements, θ is the angle of measurement and w_M are the respective weights for each element of the linear array.

The array factor is obtained by adding the result of the weighing of each element such that:

$$FA = \omega_M e^{-j\frac{(2M-1)}{2}kd\,\sin\theta} + \ldots + \ \omega_1 e^{-j\frac{1}{2}kd\,\sin\theta} + \omega_1 e^{j\frac{1}{2}kd\,\sin\theta} + \ldots + \omega_M e^{j\frac{(2M-1)}{2}kd\,\sin\theta}\ , \qquad (1)$$

where k= $\frac{2\pi}{\lambda}$. We can express the above formula in terms of the cosine function using Euler's identity so that:

$$FA = \sum_{n=1}^{8} \omega_n \cos\big((2n-1)u\big), \qquad (2)$$

Manuscript received April 17, 2012. Manuscript accepted November 20, 2012.

M.A. Acevedo, E. Martinez, and M.E. Acevedo are with the Department of Telecommunications Section of Postgraduate Studies and Research, School of Mechanical and Electrical Engineering Zacatenco, DF, Mexico (e-mail: macevedo@ipn.mx, emz06@hotmail.com, eacevedo@ipn.mx).

O. Pogrebnyak is with the Center for Computing Research, National Polytechnic Institute (CIC-IPN), DF, México (e-mail: olek@cic.ipn.mx).

where $u = kdsin\theta$, note that the array factor is maximum if the angle $\theta = 0$.

In this paper we use the following array factor for a number of elements equal to 8 with an equidistant distance between elements of *d* = *λ/2*.

$$\begin{aligned} FA = \ & \omega_1 e^{-j\frac{(2\cdot4-1)}{2}kdsin\theta} + \omega_2 e^{-j\frac{(2\cdot3-1)}{2}kdsin\theta} + \\ & \omega_3 e^{-j\frac{(2\cdot2-1)}{2}kdsin\theta} + \omega_4 e^{-j\frac{1}{2}kdsin\theta} + \\ & \omega_5 e^{j\frac{1}{2}kdsin\theta} + \omega_6 e^{j\frac{(2\cdot2-1)}{2}kdsin\theta} + \\ & \omega_7 e^{j\frac{(2\cdot3-1)}{2}kdsin\theta} + \omega_8 e^{j\frac{(2\cdot4-1)}{2}kdsin\theta} \end{aligned} \quad (3)$$

Figure 2 shows the radiation pattern of the array factor used in this article.

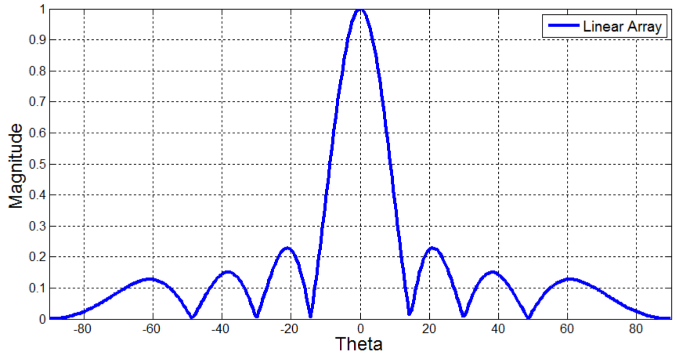


Fig 2. Radiation pattern of linear uniform symmetric array of 8 elements and equidistant distance of *d*=*λ*/2.

Null or zero angles of the array factor is obtained when:

$$\theta_{nulls} = \sin^{-1}\left(\frac{1}{kd}\left(\pm\frac{2n\pi}{N} - \delta\right)\right), \qquad n = 1, 2, 3, \ldots \quad (4)$$

where N is element array, δ is the difference of the electrical phase between two adjacent elements. For real angles sin θ is less than or equal to one, therefore, the argument of the equation must be less than or equal to one, this means that only a finite set of values satisfy the equality.

The angles of the side lobes are the greatest when:

$$\theta_s = \sin^{-1}\left(\frac{1}{kd}\left(\pm\frac{(2n+1)\pi}{N} - \delta\right)\right) \qquad n = 1, 2, 3, \ldots \quad (5)$$

The width of the main beam is defined as the angular aperture of the main lobe measured at a given constant power level and it is obtained when:

$$\theta_{\pm} = \sin^{-1}\left(\frac{1}{kd}\left(\frac{\pm 2.782}{N} - \delta\right)\right), \quad (6)$$

where $\theta_+ \ and \ \theta_-$ are half-power points and they are normalized to a $FA = 0.707$. The bandwidth of medium power is determined as follows:

$$HPBW = |\theta_+ - \theta_-| \quad (7)$$

For an 8-elements array factor, the positive and negative angles where maximums and zeros appear are shown in Table I.

TABLE I. POSITIVE AND NEGATIVE ANGLES FOR MAXIMUM AND NULL.

Maximum	*Nulls*
±22.02 º	*±14.47º*
±38.68 º	*±30.00 º*
±61.04 º	*±48.59 º*
±119.8 º	*±90.00 º*

III. DOLPH-CHEBYSHEV ARRAY FACTOR

The method Dolph-Chebyshev Array was initially investigated by Dolph and it is based on the studies by Chebyshev.

Dolph in 1946 proposed a method to calculate the array factor; this method is based on the approximation to the structure of the linear array by means of the Chebyshev polynomials of order *m* [9].

1) The Chebyshev polynomials (ChP)

The ChP of order *m* are defined as:

$$T_m(z) = \begin{cases} (-1)^m \cosh(m \cdot arccosh|z|) & z \leq 1 \\ \cos(m \cdot arccos(z)) & -1 \leq z \leq 1 \\ \cos h(m \cdot arccosh(z)) & z \geq 1 \end{cases} \quad (8)$$

Chebyshev polynomials can also be generated using the following recurrent formula:

$$T_m(z) = 2zT_{m-1}(z) - T_{m-2}(z). \quad (9)$$

We find the first ChP by evaluating equation (9), and the first eight polynomials are:

$$\begin{aligned} & t_0(z) = 1 \\ & t_1(z) = z \\ & t_2(z) = 2z^2 - 1 \\ & t_3(z) = 4z^3 - 3z \\ & t_4(z) = 8z^4 - 8z^2 + 1 \\ & t_5(z) = 16z^5 - 20z^3 + 5z \\ & t_6(z) = 32z^6 - 48z^4 + 18z^2 - 1 \\ & t_7(z) = 64z^7 - 112z^5 + 56z^3 - 7z \end{aligned} \quad (10)$$

It can be observed that if the module of *z* is greater than or equal to 1 then the ChP are directly related to the cosine function.

2) Properties of Chebyshev polynomials

We must take into account some of the properties of Chebyshev polynomials:

a) *All polynomials of any m order will pass through the point* (1,1).

b) *Within the range* $-1 \leq z \leq 1$, *the polynomials have values in* [-1, 1].

c) *All null values occur within the range* $-1 \leq z \leq 1$.

d) *The maximum and minimum values within the range z* $\in$ [-1, 1] *have values* $+1$ *and* -1 *respectively.*

e) *The higher the order of the polynomial, the steeper the slope of* $|z| \geq 1$

The zeros of Chebyshev polynomials are calculated as follows:

$$\cos\left[\frac{(2m-1)\cdot\pi}{2N}\right], \qquad m = 0, 1, \ldots, N-1 \tag{11}$$

3) *The designing of the Dolph-Chebyshev Array Factor*

The approximation of the linear array of antennas through the ChP is accomplished by the proposal of the number of elements in the array, in this case we proposed $N = 8$.

$$m = N - 1, \tag{12}$$

where m is the order of Chebyshev polynomial. Chebyshev polynomials are identical to equation (2), if we write them as follows:

$$T_{N-1}(z) = \sum_{n=1}^{M} a_n\, cos[(2n-1)u] \tag{13}$$

where $= (\frac{\pi \cdot d}{\lambda}) cos\ \theta$, a_n= Chebyshev coefficients.

The level of side lobes in dB and the rectangular shape can be obtained by:

$$R_o = \frac{1}{|FA_{ls}|}, \quad R = 20 log_{10}(R_o), \quad R_o = 10^{R/20}, \tag{14}$$

where FA_{ls} is array factor of side lobes, R_o is the level of the main lobe and R is the level of the side lobes with respect to the main lobe.

Figure 3 shows the radiation pattern generated by a Dolph-Chebyshev-type amplitude distribution with a total number of elements $N = 8$ and a ratio of lobes of $R = 15$ dB.

A comparison between the radiation patterns from the original array factor and the Dolph-Chebyshev method is shown Figure 4.

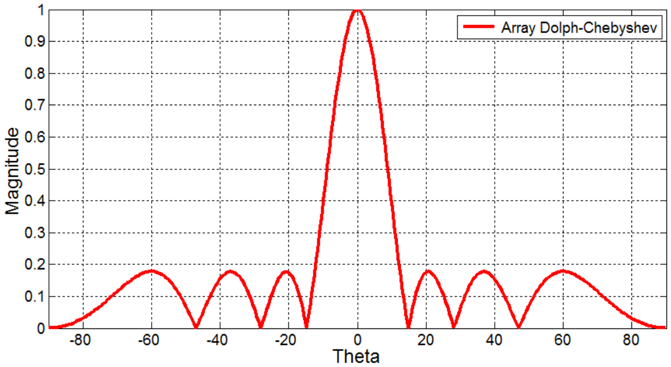


Fig 3. Radiation Pattern of 8-element Dolph-Chebyshev and a ratio of lobes of 15dB.

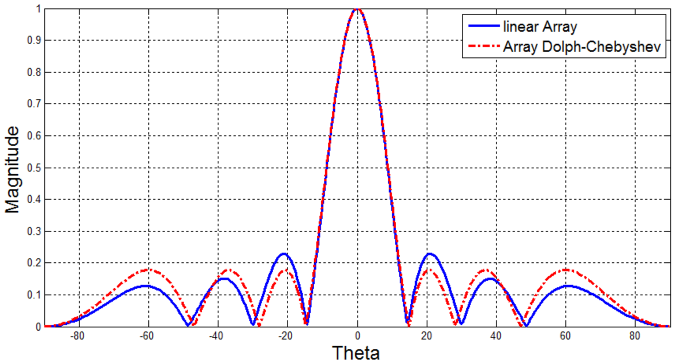


Fig. 4. Radiation Patterns from Linear Array and Dolph-Chebyshev.

In Figure 4 we can observe that the width of the main lobe is the same in both radiation patterns. The amplitude of the first side lobe of the radiation pattern from the linear array is greater than the amplitude of the first side lobe of the radiation pattern from Dolph-Chebyshev method.

The level of the amplitude of the second and third secondary lobes of the radiation pattern from Dolph-Chebyshev are greater than the second and third side lobes of the radiation pattern from linear array. The first null in both radiation patterns is located at the same angle, but the second and third nulls from Dolph-Chebyshev are shifted with respect to those from the linear array.

This comparison suggests that there is not a significant difference between the two radiation patterns.

IV. Method for Increasing Side Lobes

For increasing the side lobes the weighing of the antenna array has to be chosen so that a null value will be placed in calculated directions and a maximum value will be placed in the angles of the side lobes and leaving the peak of the main lobe in the direction of interest [10].

For a set of 8 elements the vector of the array is given by:

$$\bar{a}(\theta) = [e^{-j\frac{7}{2}kd \sin\theta}\, e^{-j\frac{5}{2}kd \sin\theta}\, e^{-j\frac{3}{2}kd \sin\theta}\, e^{-j\frac{1}{2}kd \sin\theta}\, e^{j\frac{1}{2}kd \sin\theta}\, e^{j\frac{3}{2}kd \sin\theta}\, e^{j\frac{5}{2}kd \sin\theta}\, e^{j\frac{7}{2}kd \sin\theta}]^T \tag{15}$$

The weighting vector is described as follows:

$$\bar{w}^T = [\omega_1\ \ \omega_2\ \omega_3\ \omega_4\ \omega_5\ \omega_6\ \omega_7\ \omega_8] \tag{16}$$

Therefore the total array is the sum of the vector array plus the weighting vector obtained:

$$S = \bar{w}^T \cdot \bar{a} = \omega_1 e^{-j\frac{7}{2}kd \sin\theta} + \omega_2 e^{-j\frac{5}{2}kd \sin\theta} + \omega_3 e^{-j\frac{3}{2}kd \sin\theta} + \omega_4 e^{-j\frac{1}{2}kd \sin\theta} + \omega_5 e^{j\frac{1}{2}kd\sin\theta} + \omega_6 e^{j\frac{3}{2}kd \sin\theta} + \omega_7 e^{j\frac{5}{2}kd \sin\theta} + \omega_8 e^{j\frac{7}{2}kd \sin\theta} \tag{17}$$

Since there are 8 variables we have 8 conditions to satisfy:

$$S_1 = \omega_1 e^{-j\frac{7}{2}kd\ sin\ \theta_1} + \omega_1 e^{-j\frac{5}{2}kd\ sin\ \theta_1} + \omega_1 e^{-j\frac{3}{2}kd\ sin\ \theta_1} + \omega_1 e^{-j\frac{1}{2}kd\ sin\ \theta_1} + \omega_1 e^{j\frac{1}{2}kd\ sin\ \theta_1} + \omega_1 e^{j\frac{3}{2}kd\ sin\ \theta_1} + \omega_1 e^{j\frac{5}{2}kd\ sin\ \theta_1} + \omega_1 e^{j\frac{7}{2}kd\ sin\ \theta_1} = \boldsymbol{c1}$$

$$S_2 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_2} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_2} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_2} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_2} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_2} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_2} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_2} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_2} = \boldsymbol{c2}$$

$$S_3 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_3} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_3} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_3} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_3} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_3} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_3} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_3} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_3} = \boldsymbol{c3}$$

$$S_4 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_4} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_4} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_4} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_4} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_4} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_4} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_4} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_4} = \boldsymbol{c4}$$

$$S_5 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_5} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_5} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_5} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_5} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_5} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_5} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_5} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_5} = \boldsymbol{c5}$$

$$S_6 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_6} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_6} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_6} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_6} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_6} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_6} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_6} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_6} = \boldsymbol{c6}$$

$$S_7 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_7} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_7} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_7} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_7} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_7} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_7} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_7} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_7} = \boldsymbol{c7}$$

$$S_8 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_8} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_8} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_8} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_8} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_8} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_8} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_8} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_8} = \boldsymbol{c8}$$

(18)

where $c1,\cdots,c8$ are constant.We have a system of equations where each equation is equal to the magnitude of the desired level of secondary lobes which is trying to keep the width of the main lobe in the half-power points. Solving this system of equations the weights can be obtained for each condition.

V. Simulation

In this section, we describe the process for increasing the secondary lobes in a linear array of 8-element antenna with an equidistant distance between elements of $d=\lambda/2$.

Case I

Increase the first positive side lobe at magnitude of 0.7. For this we will use a combination of maximum and zero angles which are shown in Table II.

Table II. Angle combination for increasing the first positive side lobe

*θ1=-90*r*	*θ2=-48.59*r*	*θ3=-30*r*	*θ4=-14.47*r*
*θ5=14.47*r*	***θ6=22.02*r***	*θ7=48.59*r*	*θ8=90*r*

In Table II, $r = \pi/180$ and the angle θ_6 represents the direction in which we require to increase the side lobe so that the equation will be matched to a desired maximum magnitude value. The angles θ_4 and θ_5 will not change because they are required to keep the width of the main lobe, while the remaining angles will be null for other directions.

These angles are replaced in the following equation system as follows:

$$S_1 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_1} + \omega_1 e^{-j\frac{5}{2}kd\,sin\,\theta_1} + \omega_1 e^{-j\frac{3}{2}kd\,sin\,\theta_1} + \omega_1 e^{-j\frac{1}{2}kd\,sin\,\theta_1} + \omega_1 e^{j\frac{1}{2}kd\,sin\,\theta_1} + \omega_1 e^{j\frac{3}{2}kd\,sin\,\theta_1} + \omega_1 e^{j\frac{5}{2}kd\,sin\,\theta_1} + \omega_1 e^{j\frac{7}{2}kd\,sin\,\theta_1} = \mathbf{0}$$

$$S_2 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_2} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_2} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_2} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_2} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_2} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_2} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_2} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_2} = \mathbf{0}$$

$$S_3 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_3} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_3} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_3} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_3} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_3} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_3} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_3} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_3} = \mathbf{0}$$

$$S_4 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_4} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_4} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_4} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_4} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_4} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_4} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_4} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_4} = \mathbf{0.7}$$

$$S_5 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_5} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_5} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_5} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_5} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_5} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_5} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_5} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_5} = \mathbf{0.7}$$

$$S_6 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_6} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_6} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_6} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_6} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_6} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_6} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_6} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_6} = \mathbf{0.7}$$

$$S_7 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_7} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_7} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_7} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_7} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_7} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_7} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_7} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_7} = \mathbf{0}$$

$$S_8 = \omega_1 e^{-j\frac{7}{2}kd\,sin\,\theta_8} + \omega_2 e^{-j\frac{5}{2}kd\,sin\,\theta_8} + \omega_3 e^{-j\frac{3}{2}kd\,sin\,\theta_8} + \omega_4 e^{-j\frac{1}{2}kd\,sin\,\theta_8} + \omega_5 e^{j\frac{1}{2}kd\,sin\,\theta_8} + \omega_6 e^{j\frac{3}{2}kd\,sin\,\theta_8} + \omega_7 e^{j\frac{5}{2}kd\,sin\,\theta_8} + \omega_8 e^{j\frac{7}{2}kd\,sin\,\theta_8} = \mathbf{0}$$

(19)

The result from this equation system will be 8 weights that will satisfy each condition. The obtained weights are substituted in the array factor in the formula (17).

This process will be held in every case to obtain the radiation pattern. Figure 5 shows the radiation pattern obtained from the first case:

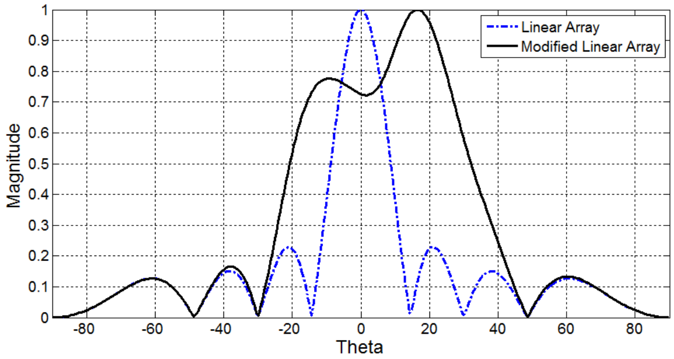


Fig 5. Radiation pattern of the first linear array by increasing the positive side lobe.

In Figure 5, we can observe the increase of the first positive side lobe, and due to there is not a null between the two maximum angles then a continuity between the main lobe and side lobe of the first maximum is shaped; therefore, a spread of the main lobe is generated by the joint of the highs from the two lobes.

We can observe that a null appears at the first negative angle which is at -30°, the second angle is at -48.59° and the third is at 90°. The first null at a positive angle is found at 48.59° and the second angle is at 90° as it is indicated in the equations.

Table III shows the combinations of angles which generate maximum and null values.

TABLE III. COMBINATION OF ANGLES FOR INCREASING THE SIDE LOBES.

Case II			
*θ₁=-90*r*	*θ₂=-48.59*r*	*θ₃=-30*r*	*θ₄=-14.47*r*
0	0	0	0.7
*θ₅=14.47*r*	*θ₆=30*r*	***θ₇=38.68*r***	*θ₈=90*r*
0.7	0	0.7	0
Case III			
*θ₁=-90*r*	*θ₂=-48.59*r*	*θ₃=-30*r*	*θ₄=-14.47*r*
0	0	0	0.7
*θ₅=14.47*r*	*θ₆=30*r*	*θ₇=48.59*r*	***θ₈=61.04*r***
0.7	0	0	0.7
Case IV			
*θ₁=-61.04*r*	*θ₂=-48.59*r*	*θ₃=-30*r*	*θ₄=-14.47*r*
0.7	0	0	0.7
*θ₅=14.47*r*	*θ₆=30*r*	*θ₇=48.59*r*	***θ₈=61.04*r***
0.7	0	0	0.7
Case V			
*θ₁=-61.04*r*	*θ₂=-48.59*r*	***θ₃=-22.02*r***	*θ₄=-14.47*r*
0.7	0	0.7	0.7
*θ₅=14.47*r*	***θ₆=22.02*r***	*θ₇=48.59*r*	***θ₈=61.04*r***
0.7	0.7	0	0.7

Case II

We increase the second positive side lobe with a magnitude of 0.7. Once the equation system has been solved and taking into account the obtained weight in case I, then the weights will be multiplied as follows:

$$\omega^T = \begin{array}{l}[\omega_1^I\cdot\omega_1^{II} = \omega_1;\ \omega_2^I\cdot\omega_2^{II} = \omega_2;\\ \omega_3^I\cdot\omega_3^{II} = \omega_3;\ \omega_4^I\cdot\omega_4^{II} = \omega_4;\\ \omega_5^I\cdot\omega_5^{II} = \omega_5;\ \omega_6^I\cdot\omega_6^{II} = \omega_6;\\ \omega_7^I\cdot\omega_7^{II} = \omega_7;\ \omega_8^I\cdot\omega_8^{II} = \omega_8;]\end{array} \tag{21}$$

These new weights are replaced in the linear array factor and the result is the radiation pattern that is shown in Figure 6. Equation (21) will be applied in the same way in each and every of the later cases.

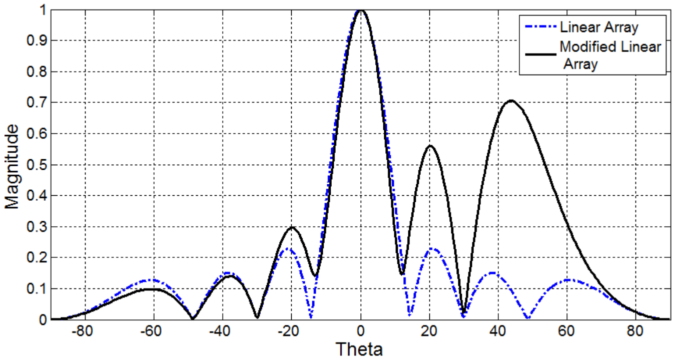


Fig 6. Radiation pattern from the linear array when the positive second side lobe is increased.

In Figure 6 we observe an increase in the positive side-lobe level, as well as the widening of it. We also observed an increase of the first secondary lobe. The angular opening in the half-power points is preserved.

Case III

We increase the third positive side lobe to an amplitude of 0.7.

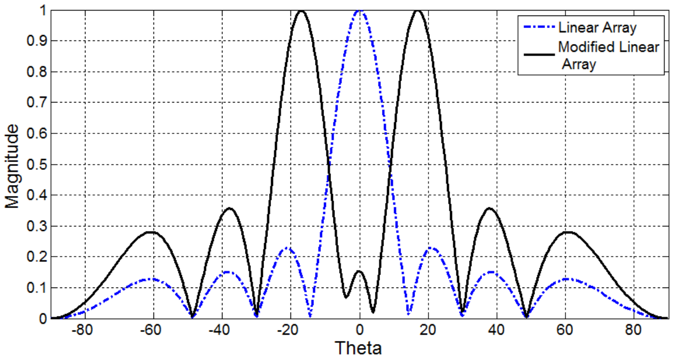


Fig 7. Radiation pattern from linear array when the third positive side lobe is increased.

In Figure 7 we note an increase in amplitude of the positive and negative side lobes, but the direction of interest does not present the desired change of amplitude. We can also observe that there is not a main lobe.

The increase in negative side lobes are not illustrated because the results are symmetrical with respect to the radiation patterns of case I, II, III respectively. In the following cases we only present the most important results to obtain the final radiation pattern.

Case IV

In this case we use a combination of two maximum angles at different positions. Figure 8 shows the resulting radiation pattern.

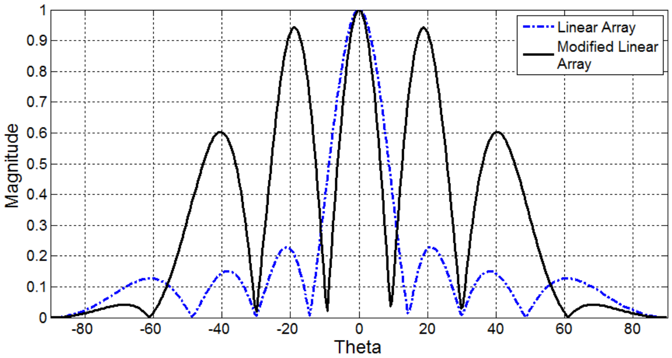


Fig 8. Radiation pattern of linear array by increasing lobes

The width of the main lobe is narrower than the original; this result causes a displacement of null angles. All side lobes had a significant increment in the amplitude except the last two side lobes.

Case V: Comparison with Dolph-Chebyshev

Finally, we performed the comparison between the resulting radiation pattern from linear array with the radiation pattern from Dolph-Chebyshev method. The Dolph-Chebyshev polynomial will be of order $m = 8$ and a maximum ratio between side lobes of 4 dB. Figure 9 shows the comparison among the three methods that we discussed in this work: Linear array, Modified linear array and Dolph-Chebyshev. It performs the change in the relative side lobes for comparison.

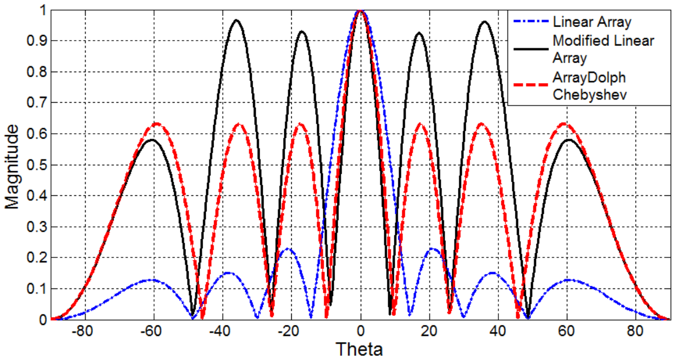


Fig 9. Radiation pattern from original linear array, Modified linear array and Dolph-Chebyshev.

In Figure 9 we can observe that the resulting modified radiation pattern present the highest amplitude in the first two secondary lobes, but this behavior is not uniform because Dolph-Chebyshev amplitude is the highest in the last two side lobes. The amplitude of side lobes from Dolph-Chebyshev is uniform throughout the radiation pattern. In both radiation patterns, Modified linear array and Dolph-Chebyshev, the angular aperture is narrow which causes the null angles to be shifted.

VI. Conclusions

The comparison between the method with increased side lobes and the Dolph-Chebyshev method showed similar results, but the amplitude of side lobe of our proposal presented higher level than the amplitude obtained by Dolph-Chebyshev method. The latter method is considered important because it is one that shows the greatest amplitude in side lobes near the main lobe.

Our proposal showed a more directive main lobe and the side lobes have the same level of amplitude as the main lobe. The increase of lobes can be the key to improve the capacity in mobile communication systems, using the frequency reuse concept based on the diversity of space.

Future research aims to achieve uniform increase of all side lobes using adaptive methods.

References

[1] M. Alanis, I. Elshafiey, A. Al-Sanie, “Multi-User Detection for SDMA-OFDM Communication Systems,” *IEEE*, 2011.

[2] A.l. Sulyman, M. Hefnawi, “Capacity-aware linear MMSE detector for OFDM-SDMA Systems”, *Institution of Engineering and Technology* , 2010.

[3] Ch. Santhi Rani, P. V. Subbaiah, K. Chennakesava Reddy, S. Sudha Rani, “LMS and RLS Algorithms for smart antennas in a W-CDMA mobile communication environment, ” *ARPN Journal of engineering and applied sciences*, vol. 4, No. 6, 2009.

[4] S. Bellofiore, J.Foutz, R. Govindarajula,C.A.Balanis, “ Smart antenna system analysis, integration and performance for mobile ad-hoc networks, ” *IEEE*, vol. 50, No. 5, 2002.

[5] I. Villordo, I.E. Zaldivar, “An Overview of SDMA in Communications Systems,” *IEEE*, 2006.

[6] M. P. Lotter, P.Rooyen, “Space Division Multiple Access for Cellular CDMA,” *IEEE*, 1998.

[7] G.M. Galvan, M. Botello, “Acceso multiple por division de espacio:una nueva dimension, ” *ESIME-IPN*, 2004.

[8] J.L Ramos, M. J. Martínez, G. A. Vega, M. S. Ruiz, “Software para el cálculo de patrones de radiación de arreglos lineales de antenas,” *Conferencia de Ingeniería Eléctrica*, 2001.

[9] M. A. Acevedo, R.Castañeda, Oleksiy-Pogrebnyak,“Diseño de antenas de ranura resonante para su aplicación en redes WiFi,” *ESIME-IPN*, vol. 13, No.1, 2009.

[10] Gross, Frank, *Smart Antennas for Wireless Communication*, McGraw-Hill, 2005.

[11] C, A, Balanis, *Antenna theory: Analysis and Design*, John Wiley and Sons, New York, 1982.

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, and e-revistas.

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 to 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, English title, author names, abstract, and keywords must be provided; in recent issues of the journal you can find examples of how it is done.

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 TeX 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. While we make every reasonable effort for fast review and publication, we cannot guarantee any specific time for this.

B. Format

The journal uses the IEEE Template for all Transactions except IEEE Transactions on Magnetics, www.ieee.org/web/publications/authors/transjnl/index.html (while the journal uses this format for submissions, it is in no way affiliated with, or endorsed by, IEEE).

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.