Automatic Handling of Expert Knowledge Using Artificial Immune Systems Applied to Combinatorial Optimization Problems

Francisco Javier Díaz Delgadillo, Oscar Montiel-Ross, Roberto Sepúlveda, Patricia Melin

Abstract— **This paper presents a methodology that reduces computing time of combinatorial optimization problems that can be applied to any population based algorithm such as: genetic algorithms, artificial immune systems, ant colony optimization, etc. To demonstrate the usefulness of the proposed methodology an Artificial Immune System was used since we used concepts derived from this paradigm. The proposed methodology was applied to solve the Traveling Salesman Problem (TSP) and it is introduced the concept of vaccination to reduce cities in the TSP. It is able to work practically with any optimization population based method in order to obtain optimal and suboptimal routes of the original problem. Comparative experimental results of several experiments with large number of cities (711 cities) are shown.**

Index Terms— **Artificial Immune System, AIS, Automatic Handling of Expert, AHEK, Combinatorial Optimization Problem, COP, Knowledge, Optimization, Traveling Salesman Problem, TSP, Vaccines.**

I. Introduction

COMBINATORIAL optimization problems (COPs) are until these days open problems. The amount of computational resources needed to solve these problems that go beyond a few elements is truly astonishing; hence the emphasis given to the development of algorithms that outperform the existing ones; providing us with faster and better solutions is a very important field of research in optimization [14].

The inspiration to develop new algorithms has been obtained from various sources, including nature. In this work, computational paradigms inspired in the natural functions of the human immune system are explored, they are named Artificial Immune Systems (AIS) [3]. A novel methodology that improves the application of AIS to solve combinatorial problems (CPs), as well as concepts that help the AIS to perform better solving huge CPs, are presented.

The Traveling Salesman Problem (TSP) is a classic example of a COP. The problem states that a salesman has to travel to an *n* number of cities in the shortest route length possible, starting from his hometown visiting each city only once, and finally returning home. This CP at plain sight seems to have a very simple solution, but in practice, the solution itself presents a very complex problem with multiple implications regarding planning and execution that can become intractable for today's computers.

A solution for one case of the TSP can be represented as a permutation of the city's indices, and the process of finding the optimal solution is an iterative process. The use of different techniques to develop new algorithms to solve more efficiently the TSP is very varied, and until these days is still an important topic of research. In general, there are two major methodologies to solve the TSP: by exact solution and by metaheuristics.

In the field of finding exact solutions for the TSP there exist methods such as branch-and-bound and lineal optimization or programming [13][14]. In this field one of the most important algorithms is presented as a program named Concorde, which is considered by many as one of the best TSP solvers [6][7]. Concorde's algorithm has many strengths when compared to other methods and has been able to handle a very large number of cities providing excellent results [7].

On the other hand, metaheuristics have contributed with many ways to solve the TSP, some of the main algorithms are genetic algorithms [8], ant colony optimization [9], neural networks [12]. In this branch, there are several works inspired in AIS metaheuristics that have been used successfully to solve the TSP, they use *Clonal Selection* [10] or *cooperation mechanisms of the Th cells* [11] mainly.

In this paper a new methodology inspired in AIS is presented, the concept of vaccination to reduce complexity, hence computational time is presented. The proposal is supported with several experiments.

This work is organized as follow: In Section II the Human and Artificial Immune Systems are described. In Section III we define mathematically a Vaccine for COPs, and we explain how to implement this concept. The criteria of designing the conducted experiments are explained in Section IV. The results are presented using graphs and tables in Section V. Finally, in Section VI the conclusions are given.

II. Human And Artificial Immune Systems

The human immune system is in charge of keeping our body healthy. It is monitoring our body constantly searching for any anomalies or foreign agents called pathogens. If the immune system is confronted with such threats it starts what is called an immune response. There are different types of

Manuscript received July 10, 2011; and accepted August 15, 2011.

Francisco Javier Díaz Delgadillo is a Ph.D. student in Intelligent Systems in Instituto Politécnico Nacional - CITEDI. Av. del Parque 1310, Otay, Tijuana, B.C., México. e-mail: frajadiade@gmail.com.
Oscar H. Montiel Ross is with Instituto Politécnico Nacional - CITEDI. (Corresponding author phone: 52(664)-623-1344;
e-mail: o.montiel@ieee.org.)
Roberto Sepúlveda is with Instituto Politécnico Nacional - CITEDI.
e-mail:sepulveda@ieee.org.
Patricia Melin is with Tijuana Institute of Technology in the Department of Computer Science in the Graduate Division. Calzada Tecnológico S/N, Tijuana, B. C., México. e-mail: pmelin@tectijuana.mx

immune responses: the innate immune response and the adaptive immune response [1].

The innate response constitutes the immune system ability to response to the imminent threat, even if it has never been dealt with it before. After it has started, the cells that have been activated undergo a series of cloning and mutations in order to better attack the pathogen and eventually exterminate it.

The adaptive immune response constitutes the immune system's ability to remember previously encountered pathogens. In order to do this, the immune system keeps alive for a period of time the cells responsible for eliminating a pathogen generated in innate response, thus providing a better more efficient immune response.

Although in reality the human immune system is much more complex than the general description given, we can see that it can be a very interesting source of inspiration for obtaining data, information and knowledge for present and future research in the fields of mathematics in computer science and optimization. This is in part because the human immune system, as well as other biological systems, exhibits extremely sophisticated capabilities for learning and processing data in its operations, which are characteristics we search to extrapolate into solving engineering problems.

Because of the biological and medical importance of the immune system a great deal of information is available in regards to its operation. Even though the operation of the human immune system is highly complex, the general functions are very clear to us and because of this we have been able to learn from them, providing us with new insight into solving classic and new problems.

The AIS are the extrapolation of the works of human immune system into the fields of mathematics, computer science and engineering [2]. The idea behind them is to take the main elements of the immune system and use them as a source of inspiration to solve problems.

Some of the main characteristics of the immune system that are of particular interest to us are [3]:

- Pattern recognition
- Singularity
- Diversity
- Autonomy
- Multilayer system
- Fault detection
- Adaptive capabilities
- Robustness
- Immune learning and memory
- Distributed system

Although we know the immune system has these attributes, how the immune system is able to achieve these characteristics is still a main area of research for immunologists everywhere. The idea that a natural system exists with such characteristics is a great source of inspiration, as our understanding of the immune system grows, extrapolation of the workings to our daily scientific and engineering lives will increase.

Continuously researches everywhere are finding new ways and areas to apply AIS paradigms. There are some areas where AIS are particularly good because their direct relationship between the natural function and a computational task is natural. Some of the main applications of the AIS are [3]:

- Pattern recognition
- Fault detection and anomalies
- Data mining and classification
- Agent based systems
- Scheduling
- Machine learning
- Autonomous control and navigation
- Optimization and search
- Artificial life

The Vaccines are the immune terminology proposed in order to obtain Automatic Handling of Expert Knowledge (AHEK). This concept is taken from the idea that vaccines offer reinforcement to the immune system, providing agents that resemble the disease that you are trying to combat. The use of Expert Knowledge applied to combinatorial problems is not new [4], it has been utilized by other researchers in order to provide better solutions to optimization problems; however, the main disadvantage of them is that they require the human expert understanding of the problem, the way it behaves, how it is composed, and others. This presents a big disadvantage considering that as the CP escalates, the involvement of the expert will also escalate to an extent where it will be impossible to provide this knowledge without turning itself into a problem. For this reason, this article proposes that this expert knowledge should be handled automatically, which takes the burden off the expert.

III. Vaccine Definition and Implementation

The performance of the vast majority of algorithms designed to solve COPs decrease when the number of working elements grows. Usually this performance degradation is presented either in a reduction on the quality of the solutions themselves or in a substantial increase of the execution time needed to find a viable solution. This situation presents a very important challenge for all the developed algorithms and to find a methodology that helps alleviate this problem is the focus of this work.

This paper presents a methodology that provides improvement in the performance of population based algorithms designed to solve COP in such a way that these algorithms do not need to be modified conceptually (they retain their original terminologies), or structurally (it is not necessary to modify the algorithm). In order to do this, the proposed methodology consists of creating an infrastructure around the original COP solving algorithm, the TSP in our case of study.

The proposal is a three steps methodology, which first step is to reduce the number of nodes of a COP, for the TSP is to reduce the original city list. This modification is achieved by grouping and replacing elements (cities) which effectively provide a reduction of the original city list. In other words, the proposed algorithm tries to capture the essence of having an artificial "Expert" which decides what cities would take part in the grouping and replacement process.

Once that the reduction has been carried out, in the second step, the COP solving algorithm works with this reduced set of nodes.

The goal of the third step is to recover the original quantity of nodes; i.e., the number of cities for the TSP, and connecting the omitted nodes (cities) to the path; i.e., the traveling path for the TSP.

This methodology is inspired on the concept of vaccination applied to COP where there are a set of subpaths grouping a specific number of nodes using some criteria, and once the repertoire of vaccines has been created, the involved cities are removed from the original list, and then a new node that represents the removed nodes is created.

Fig. 1 shows the conceptual representation of an artificial vaccine. For the step one, two different methods for creating vaccines are proposed: Vaccine Generation by Random Selector (VRS), see Fig. 2; and Vaccine Generation by Elitist Selector VES, see Fig. 3. Once the reduced set of nodes has been optimized, we proceed with step 3 expanding the vaccinated route using in order to recover the original set of nodes; this is shown in Fig. 4.

The application of the aforementioned method applied to the TSP is straightforward, and in the next sections we are going to apply it to solve this classical optimization computational problem.

A. Vaccine

The *vaccine* is the building block for the proposed algorithm, we define it as a 7-tuple as is shown in (1)

$$Vacc = (VID, NL, NN, IE, FE, GC, L) \tag{1}$$

where,

- **Vaccine Identifier (VID):** Each vaccine must have a unique identifier.
- **Node List (NL):** Vaccines must have either the data that represents each node such as its coordinates, the original node list index and a unique identifier.
- **Number of nodes (NN):** Additionally to the node list, the number of nodes must be available.
- **Initial Element (IE):** The initial element of the vaccine must be clearly defined in terms of the original city list (original node list index).
- **Final Element (FE):** The final element of the vaccine must be clearly defined in terms of the original city list (original node list index).
- **Geometrical Center (GC):** This represents the location of the vaccine on the map. Because the nodes that are part of the vaccine are removed, it is necessary to add an equivalent representation of the vaccine on the map. For a 2D map the coordinates are calculated using (2) and (3).

$$\text{Center in x: } X_c = \frac{\sum_{i=0}^{n} x_i}{n} \tag{2}$$

$$\text{Center in y: } y_c = \frac{\sum_{i=0}^{n} y_i}{n} \tag{3}$$

- **Length (L):** The tour length (sub-path length) of the cities contained in the vaccine. It can be calculated by the Euclidian distance using (4),

$$L = \sum_{i=0}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \tag{4}$$

Fig. 1 shows a scheme of how a vaccine should be organized from a computational point of view.

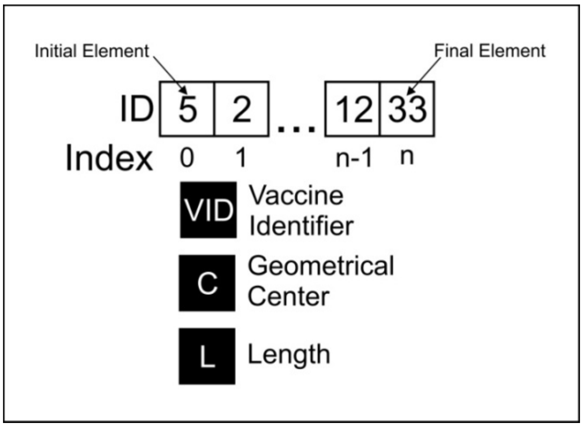


Fig. 1 Conceptual representation of an artificial vaccine. The string is the original city list and ID is the city list index.

B. Vaccination Process

We have named Vaccination Process to the systematic generation, application and removal of vaccines from a list of elements of a COP.

The vaccination process consists of three steps. The first step is the generation of the vaccines and the application of them to the original node (city) list in order to produce a reduced (node) city list; in this way, we are proposing two algorithms for achieving such generation of vaccines, one is using the algorithm called VCR, and the second option is using the algorithm called VES, both algorithms are described in depth in the next section.

Step two consists of using the reduced city list created on step one and solving the COP with an already established TSP optimization algorithm. Once an optimized route is created by this algorithm, it will be in terms of the reduced city list. Therefore, the aim of step three, which is called expansion of the vaccinated route, is to reconstruct this route but in terms of the original city list replacing the vaccinated elements with their original cities; this algorithm is also described in depth in a later section. Fig. 5 shows a general diagram that illustrates the whole vaccination process and how it is used as a shell of an optimization algorithm that helps to reduce computing time of the problem.

C. Vaccine Generation by Random Selector (VRS)

This is an efficient and fast method to generate vaccines. It is useful when information about node distribution is poor or nonexistent.

To apply the Random Selector (RS) algorithm, we choose randomly the initial elements of the vaccines from which the rest of the vaccine is made. Once the element is selected, the Random Selector algorithm requires calculating the distance from the chosen node (city) to the rest of nodes (cities) with the aim of choosing the nearest one. The distance is calculated by Euclidean distance. This is an iterative process that finishes when we reach the amount of required vaccines; i.e., we have reached the number NN indicated in (1). Fig. 2 shows the pseudo-code to this process.

```
1. Input: Original Node List (ONL), Num. of vaccines (NV), Num.
   of nodes per vaccine (NNV)
2. Copy ONL to Available node list (ANL)
3. Copy ONL to Reduced node List (RNL)
4. while NV is not met
   4.1. Randomly choose a node (NR) from ANL
   4.2. Calculate the route of the nearest nodes from the
        selected node. The route length is equal to NNV
   4.3. Generate a vaccine with the given route
   4.4. Calculate geometrical center of the vaccine
   4.5. Add vaccine to Vaccine List (VL)
   4.6. Remove the implicated nodes that form the vaccine
        from the RNL
   4.7. Add a new Node with the vaccine's geometrical center
        to the RNL
   4.8. Remove from the ANL the randomly selected node NR
5. Output VL and RNL
```

Fig. 2. Generation of vaccines using VRS. Note that for the TSP we can use city, cities, Num. of Cities instead of node, nodes, and Number of Nodes.

```
1. Input: Original Node List (ONL), Num. of vaccines (NV) and
   Num. of nodes per vaccine (NNV)
2. Copy ONL to Reduced node List (RNL)
3. Generate distance matrix for ONL by taking each node and
   calculating as many nearest neighbors are NNV states.
4. Sort the ONL by shortest distance to the nearest neighbor
5. for each Node in ONL
   5.1. if NV is not met
      5.1.1. Take the corresponding nearest neighbors for the
             Node
      5.1.2. Generate the two possible routes (from left-right
             and from right-left)
      5.1.3. if (left-right and right-left route) are not in Vaccine List
         5.1.3.1. Generate a vaccine with the given route
         5.1.3.2. Calculate geometrical center of the vaccine
         5.1.3.3. Add vaccine to Vaccine List (VL)
         5.1.3.4. Remove the implicated nodes that form the
                  vaccine from the RNL
         5.1.3.5. Add a new Node with the vaccine's
                  geometrical center to the RNL
      5.1.4. else
         5.1.4.1. Do nothing
   5.2. else
      5.2.1. Stop for each loop
6. Output VL and RNL
```

Fig. 3. Generation of vaccines using VES.

D. Vaccine Generation by Elitist Selector (VES)

This method generates a vaccine repertoire with decision criteria named Elitist Selector. The main objective of Elitist Select is to select the best initial elements from which a vaccine will be build. The application of this algorithm requires calculating the matrix of distances corresponding to the distance from each one of the elements to the rest with the aim of sorting by shortest distance. This guarantees that the shortest route vaccines are picked. Fig. 3 shows the pseudo-code to this process.

E. Expansion of vaccinated route

The expansion algorithm is needed to convert the optimized route of the reduced node list by VRS or VES, to the original node (city) list.

The temporally removed cities linked with sub-paths are reinserted into the list together with the reduced optimized set of nodes. Therefore, we obtain the whole set of nodes (cities) with an optimized path that may be optimal or suboptimal. Fig. 4 shows the pseudo-code to this process.

IV. Experiments

For testing the proposed method, we used the TSP because there are extensive researches with known results that have been stored in public repositories. We used five TSP problems from the public library TSPLIB with a range of cities from 131 to 711.

```
1. Input: Original Node List (ONL), Reduced Node List (RNL),
          Vaccine List (VL), Optimum Route (OR)
2. Select first Link (L) from (OR)
3. Obtain first Node (SN) from L
4. for each Link in the Optimum Route
   4.1. if SN is not Vaccine
      4.1.1. Add SN to Expanded Node List (ENL)
   4.2. else SN is Vaccine
      4.2.1. Find the Vaccine (V) in VL equal to SN
      4.2.2. Calculate distance d1 and d2 which represent the
             distance from the SN to the Initial
             Node (IN) and Final Node (FN) of V.
      4.2.3. if d1 ≤ d2
         4.2.3.1. Add to the ENL the IN (index 0)
         4.2.3.2. Add rest of the Nodes from V starting from
                  index 1 to n-1
      4.2.4. else
         4.2.4.1. Add to ENL the Final Node (index n)
         4.2.4.2. Add rest of the Nodes from V starting from
                  index n-1 to 0
      4.2.5. Select SN based on the following Node defined by
             the Link
5. Output ENL (order of elements is the route)
```

Fig. 4. Algorithm to expand the Reduced Node List to the Original Node List.

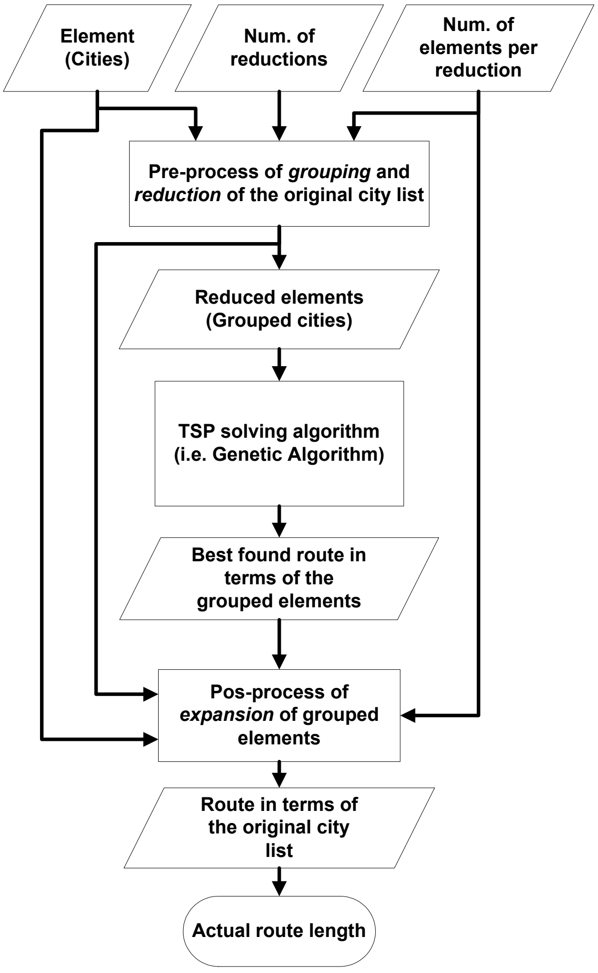

Element (Cities)
Num. of reductions
Num. of elements per reduction
Pre-process of grouping and reduction of the original city list
Reduced elements (Grouped cities)
TSP solving algorithm (i.e. Genetic Algorithm)
Best found route in terms of the grouped elements
Pos-process of expansion of grouped elements
Route in terms of the original city list
Actual route length

Fig. 5. Vaccination Process.

We are presenting three experiments that embrace a big diversity of comparative situations that will help to conclude about the usefulness of the method. This was achieved optimizing the TSP for 131, 237, 395 and 436 cities using a Genetic Algorithm (GA). With the aim of obtaining some benchmarks and be able of achieving comparisons, in Experiment 1 we did not use the proposed method. In Experiment 2 the vaccines were generated using the Random Selector, and in Experiment 3 the vaccines were generated using the Elitist Selector. For the three experiments the GA ran about 10 millions of generations, we recorded results every 500,000 generations, as it is shown in Fig.s 6 to 11. For the three experiments, we used a convergence value (stuck point), it was chosen such as the algorithm did not provided any significant advance (less than 1% of improvement compared the previous generation).

Experiment 1 - Obtaining benchmarks: This set of experiments consisted in obtaining the control data to be used as benchmarks in order to achieve comparisons. The behavior of the GA optimizing the TSP for different amount of cities can be seen in Figs. 6 to 11.

Experiment 2 - Vaccination with Random Selector (VRS): In this set of experiments vaccination using VRS on the City List was used; i.e., we applied the algorithm shown in Fig. 2.

Experiment 3 - Vaccination using the Elitist Selector (VES): The algorithm described in Fig. 3 was applied to obtain a reduced set of cities. Similarly to Experiment 2, after reducing the set of cities, the GA was used to obtain the shortest path, and then the expansion of the vaccinated route using the algorithm explained in Fig. 4 was achieved.

V. Experimental Results

In Fig. 6 the data for TSP with 131 cities is presented. We can observe that the convergence point for the GA is at 3,000,000 millions of generations while for the RS and ES are at 2,000,000. After these points, further iterations do not provide significant improvements in the solution. The final route value shows that the paths obtained by the application of the RS and ES are worse solutions than applying the GA alone, the percentage differences are -6.37% for the RS and -1.27% for the ES algorithm.

Fig. 7 shows data corresponding to the TSP with 237 cities. The convergence points are at 6,500,000 millions of generations for the GA and 4,000,000 for the RS and ES algorithms. Once again we notice a tendency of the vaccination algorithms to provide faster convergence points than the GA alone. The final route values obtained are better by 0.89% for the RS and 5.69% worse for the ES compared to the GA. At this point start to appear a tendency that shows that with a higher city count, the vaccination method (RS and ES) will provide better solutions than using the GA alone.

Fig. 8 shows the data for the TSP of 395 cities. Here the observation made on Fig. 7 is clearer, since the tendency about obtaining the best routes (shortest paths) using few generations becomes evident. The convergence point using the vaccination method with RS is 7,500,000 millions of generations, while for the ES algorithm is 6,500,000. The GA with no vaccination did not converge until 10 millions of generations. The final route value for the RS is 3.27% and ES is 8.78% better than the obtained using the standalone GA.

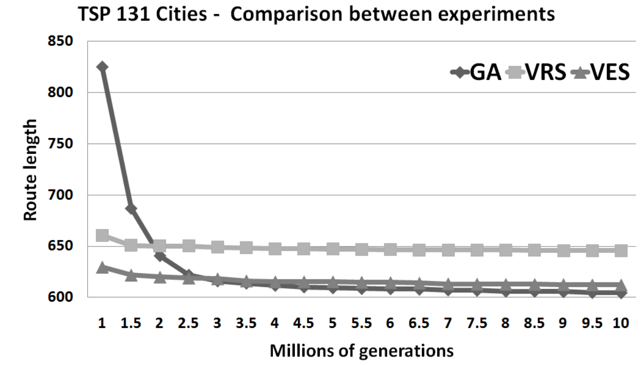


Fig. 6. Comparison of the 3 experiments for the TSP with 131 Cities.

Fig. 9 shows the data for the TSP 436 with cities. We observed that always the best results were obtained using any of the vaccination methods. For the RS algorithm, the point of convergence is about 8 millions of generations, while for the ES algorithm is 6.5 millions. Once again, the GA did not converged before 10 millions of generations giving the worst times. The final route solutions obtained by the use of the vaccination algorithms were better by 10.18% for the RS and 17.33% for the ES, than the obtained with the GA algorithm.

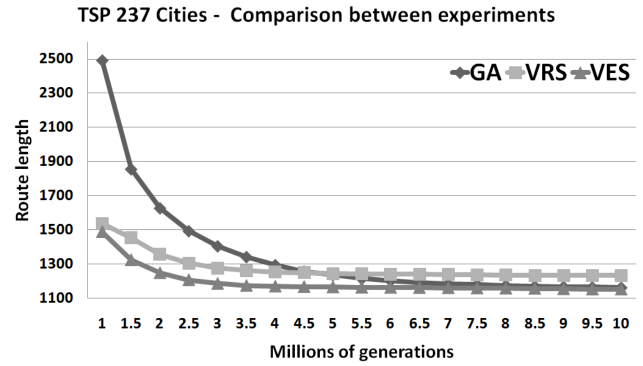


Fig. 7. Comparison of the 3 experiments for the TSP with 237 Cities.

Finally, we studied the TSP using 711 cities. This is the highest number of cities TSP studied and it shows very interesting results. At this number of cities none of the 3 experiments showed a convergence point before 10 millions of generations. However, by analyzing the graph in Fig. 10, it is clear that the vaccination algorithms, RS and ES, will converge faster than the alone GA. The final route values for the RS and ES were better than the GA by 37.65% and 44.71% respectively.

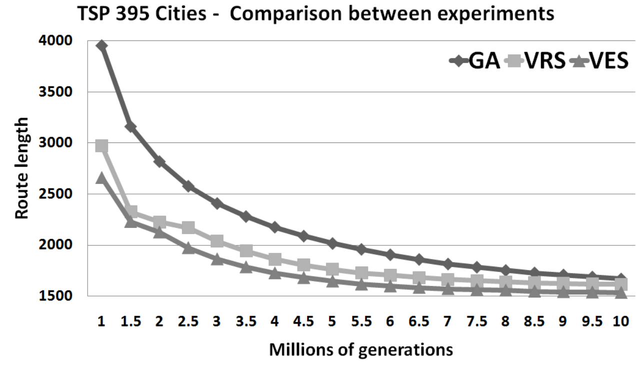


Fig. 8. Comparison of the 3 experiments for the TSP with 395 Cities.

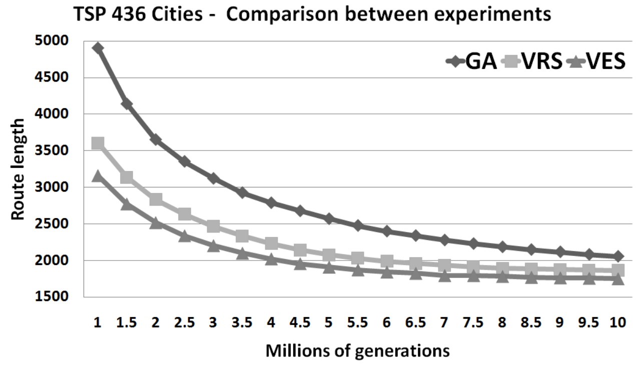


Fig. 9. Comparison of the 3 experiments for the TSP with 436 Cities.

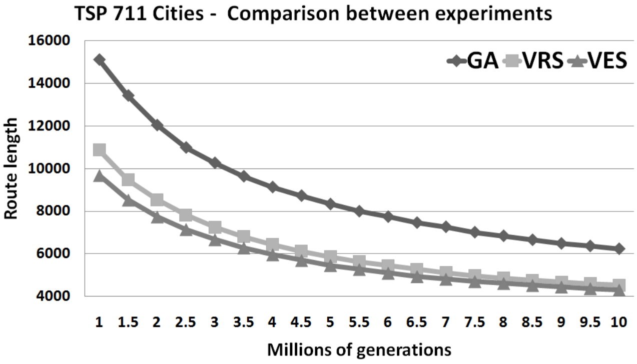


Fig. 10. Comparison of the 3 experiments for the TSP with 711 Cities.

Table I gives us the concentrated values of the length of each optimized final tour. For the first TSP which has 131 cities we can observe that the final value of the GA is better than using vaccination with RS and ES. However, this is not the case with the next TSP with 237 cities, as the results of ES are actually better, while the RS is slightly behind of the GA. As the number of cities per TSP grows, the final values obtained with the vaccination algorithms provide much better results, as can be noticed in the TSP with 395 and 436 cities. In the last TSP with 711 cities we have a very important difference between the GA and the vaccination algorithms with an 18.82% and 22.35% closer solution respectably.

TABLE I

CONCENTRATED VALUES OF THE THREE DIFFERENT ALGORITHMS AND PERCENTAGE OF IMPROVEMENT

	GA	RS		ES	
TSP	*L*	*L*	*%*	*L*	*%*
131	**604.50**	645.63	-6.37	612.25	-1.27
237	1161.40	1231.50	-5.69	**1151.19**	0.89
395	1668.35	1615.60	3.27	**1533.75**	8.78
436	2053.47	1863.80	10.18	**1750.15**	17.33
711	6232.00	4527.55	37.65	**4306.65**	44.71

In Table I, GA is Genetic Algorithm, RS is Vaccination by Random Selector and ES is vaccination by Elitist Selector. The length L denotes the best route for each experiment and the symbol "%" is the percentage of the corresponding final length value of experiments 2 and 3 in comparison with the final value of experiment 1.

Fig. 11 shows a graph that compares the execution time for each of the TSP solved by the three different algorithms GA, RS and ES. In this graph we can see that the execution times have been improved significantly in all the studied problems independently of the number of cities using the vaccination proposed method.

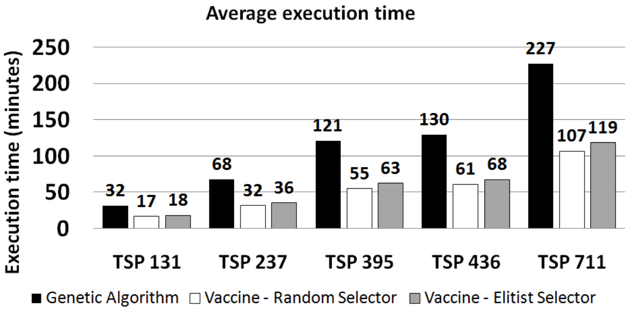


Fig. 11. Comparison of the average execution time for each TSP solved by the three different algorithms.

The improvements in execution time are attributed to the effective reduction of nodes (cities) that the optimization algorithm has to work in order to provide the optimal (or suboptimal) solution. The improvements are determined by (5) that produces a Set of Reduced Nodes (SRN) that will depend on the parameters selection.

$$SRN = NC - (NV - EV) + NV \tag{5}$$

Where *NC* is number of cities, *NV* is number of vaccines and *EV* is elements per vaccine. With this formula in Table II we present the effective reductions for each TSP with 40% vaccination and 2 elements per vaccine.

TABLE II

NUMBER OF ELEMENTS AFTER VACCINATION

Problem	Formula	Final Elements
TSP 131	131-((131*.4)*2)+(131*.4)	79
TSP 237	237-((237*.4)*2)+(237*.4)	142
TSP 395	395-((395*.4)*2)+(395*.4)	237
TSP 436	436-((436*.4)*2)+(436*.4)	256
TSP 711	711-((711*.4)*2)+(711*.4)	427

NV = NC * 0.4 and EV = 2.

VI. CONCLUSIONS

A new methodology to reduced computational time of COPs inspired in the concept of immunization and based on the artificial immune system was presented. In order to test the proposal we choose highly complex examples of the TSP.

Basically the methodology consists of three steps. The first step is to reduce the original quantity of node of a graph problem by the application of vaccination using random selector (VRS), or vaccination using elitist selector (VES). The second step consists in working with the reduced set of nodes, in our case for the TSP. The third step is to return to the original Set of nodes providing short-optimal-paths that introduced the global path.

This methodology can be applied to any population based COP without modifying the optimization algorithm to which is applied since it work as a shell. Hence, we applied it to a GA.

The achieved experiments were conducted to demonstrate the usefulness of the proposal by optimizing several complex examples of the TSP without using the method in order to obtain benchmarks about shortest paths and execution times.

Then we applied VRS and VES, and it was demonstrated that the vaccination methods improved results of execution time in all the cases; however, obtaining optimal paths only was possible for problems bigger than 131 cities for the TSP, in our test for the case of 237 cities.

For problems bigger than 237 cities this proposal always outperformed the GA alone considering obtaining the shortest path and execution time.

A formula to obtain the Reduced Set of Nodes (RSN) was developed; it is important because it provides a tool that allows estimating beforehand the actual reduction in the number of nodes therefore the reduction in execution time can be known before testing which can be tedious and time consuming. The formula takes into consideration vaccination parameters such as the Number of Vaccine (NV) and Elements per Vaccine (VE) which provides flexibility.

Finally, comparing VRS and VES, VRS is the faster algorithm, and VES provides the best routes. The difference in the experiments is substantially small but exists.

REFERENCES

[1] D. Dasgupta and L. F. Niño (2009): Immunological Computation: Theory and Applications. CRC Press Taylor & Francis Group. 8-9.

[2] J. Chen and M. Mahfouf (2009). Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies Chapter II Artificial Immune Systems as a Bio-Inspired Optimization Technique and Its Engineering Applications. Edited Hongwei Mo, Harbin Engineering University, China: IGI Global. 22-45.

[3] L. de Castro, J. Timmis (2002). Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Great Britain. 55-60.

[4] L. Jiao and L. Wang. A novel genetic algorithm based on immunity. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, Volume 8, 2000.

[5] J. Aitkin. Recovered July 2010, de Artificial Immune Systems available at: http://www.artificial-immune-systems.org/algorithms.shtml

[6] G. Gutin, H. Jakubowicz, S. Ronnen and A. Zverovitch. Seismic vessel problem. In Communic. In DQM 8, pages 13–20, 2005.

[7] S. A. Mulder and D. C. Wunsch II. Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks. Neural Networks 16, pages 827–832, 2003.

[8] P. Larrañaga, C.M.H. Kuijpers, R.H. Murga, I. Inza And S. Dizdarevic. Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artificial Intelligence Review, 13:129–170, 1999.

[9] L. M. Gambardella and M. Dorigo. Ant colonies for the traveling salesman problem. IRIDIA, 1997.

[10] M. Y. Zhao, et al. A novel clonal selection algorithm and its application. Apperceiving Computing and Intelligence Analysis, pages 385–388, 2008.

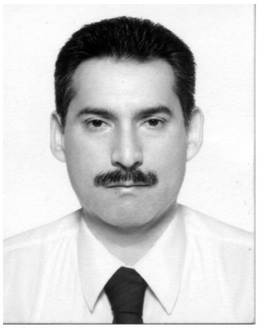
[11] Y. Zhu, Z. Tang, H. Dai and S. Gao. Cooperation artificial immune system with application to traveling salesman problem. ICIC Express Letters, pages 143–148, 2008.

[12] M. Budinich. A self-organizing neural network for the travelling salesman problem that is competitive with simulated annealing. Neural Computation Volume 8, 1996.

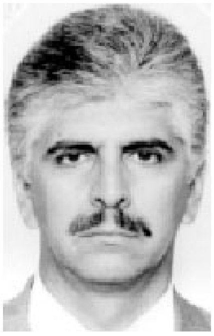
[13] G. Ausiello. Complexity and approximation: combinatorial optimization problems and their approximability properties. Springer, 1999.

[14] V. Vazirani. Approximation Algorithms. Springer, 2004.

Francisco Javier Díaz Delgadillo was born in Tijuana, B. C., México in 1987. He received his Engineering title in Cybernetic Electronics from Centro de Enseñanza Técnica y Superior (CETYS) and he obtained his M. Sc. from the Instituto Politécnico Nacional – CITEDI. Currently he is a Ph.D. student in Computer Science in the field of Intelligent Systems at this same institution. He has published papers about evolutionary computation, Artificial Immune Systems and Artificial Vaccination applied to combinatorial optimization problems. His research interests include optimization, intelligent systems, robotics, computer graphics and software design.



Oscar Humberto Montiel Ross was born in Mexicali, B.C., México in 1961, he received the M. Sc. From the Tijuana Institute of Technology, Tijuana, México, and the Ph. D. degree in the Universidad Autónoma of Baja California, Tijuana, México, both in Computer Science, in 2000 and 2006 respectively. He has published papers about evolutionary computation, Mediative Fuzzy Logic, Ant Colonies, type-2 fuzzy systems, embedded systems, and mobile robotic. He works as a researcher at the Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI) from the Instituto Politécnico Nacional. His research interests include optimization, intelligent systems, and robotics. Dr. Montiel is co-founder and an active member of HAFSA (Hispanic American Fuzzy Systems Association), and the Mexican Chapter of the Computational Intelligence Society (IEEE), he is member of the International Association of Engineers (IANG).



Roberto Sepúlveda Cruz was born in Chula Vista Ca., U.S.A., in 1952, he received the M. Sc. from the Tijuana Institute of Technology, Tijuana, México, and the Ph. D. degree in the Universidad Autónoma of Baja California, Tijuana, México, both in Computer Science, in 1999 and 2006 respectively. He has published papers about type-2 fuzzy systems, embedded type-1 and type-2 fuzzy logic controllers, mobile robot navigation. He works as a researcher at the Centro de Investigación y Desarrollo de Tecnología Digital from the Instituto Politécnico Nacional. His research interests include type-2 fuzzy systems, intelligent systems, and robotics. Dr. Sepúlveda is co-founder and an active member of HAFSA (Hispanic American Fuzzy Systems Association), and the Mexican Chapter of the Computational Intelligence Society (IEEE), he is member of the International Association of Engineers (IANG).

Patricia Melin was born in Tijuana, B. C., México in 1962, she holds the Doctor in Science degree (Doctor Habilitatus D.Sc.) in Computer Science from the Polish Academy of Sciences (with the Dissertation ”Hybrid Intelligent Systems for Pattern Recognition using Soft Computing). She is a Professor of Computer Science in the Graduate Division, Tijuana Institute of Technology, Tijuana, México. Additionally, she is Vice President of HAFSA (Hispanic American Fuzzy Systems Association). Prof. Melin is the founding Chair of the Mexican Chapter of the IEEE Computational Intelligence Society. She is Vice Chair of the IEEE Neural Network Technical Committee (2007), she also belongs to the IEEE CIS Educational Committee, the Committee of Women in Computational Intelligence of the IEEE and to the New York Academy of Sciences. She is member of NAFIPS, IFSA, and IEEE. She has published over 90 journal papers, 5 authored books, 6 edited books, and 160 papers in conference proceedings.