A Communication Scheme for an Experimental Grid in the Resolution of VRPTW using an Evolutionary Algorithm

Abelardo Rodríguez-León[1], Marco Antonio Cruz-Chávez[2], Rafael Rivera-López[3], Erika-Yesenia Ávila-Melgar[4], Fredy Juárez-Pérez[5] and Martín-Heriberto Cruz-Rosales[6]

[1,3] Departamento de Sistemas y Computación, Instituto Tecnológico de Veracruz
Calzada Miguel Ángel de Quevedo 2779, Veracruz, México
{arleon,rrivera}@itver.edu.mx
[2,4,5] CIICAp, [6]FC, Autonomous University of Morelos State, Av. Universidad 1001 Col. Chamilpa, Morelos, México
{mcruz,erika,juarezfreddy,mcr}@uaem.mx

Abstract

In this work is defined a scheme of two stages for sending population segments of one Parallel Genetic Algorithm (PGA) to the nodes of an experimental grid called "Tarántula miniGrid". The technique used to link the clusters and to configure the Tarántula miniGrid is described and the effects of latency in the communications between these clusters and the behavior of the speedup are discussed. The PGA is used for solving instances of the Vehicle Routing Problem with Time Windows (VRPTW). Results of this approach show that this communication scheme increases the efficiency of PGA in a grid environment and produces greater exploration and exploitation of the solution space of VRPTW.

1 Introduction

It is known that the computer is a powerful tool used to model and solve many complex problems in all branches of science and engineering, but has been shown that single computer resources are not sufficient to obtain results in reasonable time on these problems. Therefore it is necessary to use the resources of groups of computers distributed in clusters or grids. Grid computing is defined as a "*large scale geographically distributed hardware and software infra-structure composed of heterogeneous networked resources owned and shared by multiple administrative organizations which are coordinated to provide transparent, dependable, pervasive and consistent computing support to a wide range of applications*" [1]. Grid computing is a promising approach for solving complex problems but it has several challenges as the heterogeneous resource management, performance guarantees, fault management, optimization of distributed algorithms, control remote data access, etc.

The resolution of optimization problems using grid computing is an approach that has gained importance in recent years [2]. For example, in [3] is proposed a parallel evolutionary algorithm that uses simulated annealing for solving a job shop scheduling problem in a grid environment. In [4] is presented a parallel simulated annealing algorithm for solving the weighted unrelated parallel machines problem using the EELA-2 grid.

The objective of this work is the definition of a scheme to communicate the results of applying one genetic operator in several segments of a population of solutions of a genetic algorithm on the nodes of an experimental grid. This grid, called Tarántula miniGrid, is configured as a virtual private network such that all nodes are considered within a single network. This communication scheme has two stages: the first phase defines the procedure for combining the segments between all nodes in a cluster and the second phase deals with the transmission of group of segments between the clusters of the experimental grid using the FTP protocol to reduce traffic on network.

In order to test this communication scheme, is implemented a parallel genetic algorithm (PGA) for solving the Vehicle Routing Problem with Time Windows (VRPTW). This PGA divides the population of solutions into several segments that are distributed on the nodes of Tarántula miniGrid. Tests are designed to measure the efficiency of the algorithm in a grid environment, analyzing the effect of the latency in the communications and the behavior of the speedup of the algorithm when it is varied the number of nodes used in the Tarántula miniGrid.

2 Experimental Grid

The Tarántula miniGrid consists of two clusters installed in two educational institutions that are geographically distant (400 km): the Universidad Autónoma del Estado de Morelos (UAEM), located in Cuernavaca, Morelos and the Instituto Tecnológico de Veracruz (ITVer), located in Veracruz, Veracruz. The clusters are integrated as a single parallel machine using Internet 2. This integration allows local management of both clusters despite the fact that they are independent entities. The basic components of Tarántula minigrid are listed in table 1. An important element of Tarántula miniGrid is a link that joins the two clusters by a Virtual Private Network (VPN, network-to-network) using OpenVPN.

Table 1.- *Hardware and software for Tarántula miniGrid.*

Software for Tarántula miniGrid
Red Hat Enterprise Linux 4, Compiler gcc version 3.4.3, OpenMPI 1.2.8, MPICH2-1.0.8, Ganglia 3.0.6 , NIS ypserv-2.13-5, NFS nfs-utils-1.0.6-46, OpenVPN, Torque + Maui.
Hardware in UAEM cluster
1.- Switch Cisco C2960 24/10/100. 2.- Master node: Intel Pentium 4, 2793 MHz, 512 MB RAM, 80 GB HD, 2 network cards 10/100 Mb/s. 3.- 9 slave nodes: Intel Celeron Dual Core, 2000 MHz, 2 GB RAM, 160 GB HD, 1 network card 10/100 Mb/s.
Hardware in ITVer cluster
1.- Switch 3Com 8/10 2.- Master node: Intel Pentium 4 Dual Core, 3200 Mhz, 1 GB RAM, 80 GB HD, 1 network card 10/100 Mb/s 3.- 2 slave nodes: Intel Pentium 4 Dual Core, 3200 Mhz, 1 GB RAM, 80 GB HD, 1 network card 10/100 Mb/s

3 Vehicle Routing Problem with Time Windows

The Vehicle Routing Problem (VRP) is a combinatorial optimization problem whose resolution has a major impact on the processes of transportation, distribution and logistics of any enterprise. This problem was first studied by Dantzig [5] and it is known that its complexity places it within the group of NP-Hard [6]. VRP involves the designing of a set of routes for a fleet of vehicles based at one central depot that is required to service a number of geographically dispersed customers, while minimizing the total travel distance or the total distribution cost [7]. The VRP with Time Windows (VRPTW) is one variant of VRP that defines that every customer has to be supplied within a certain time window. The mathematical formulation of the VRPTW is presented in (1-11).

$$min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \tag{1}$$

subject to

$$\sum_{k \in K} \sum_{j \in \Delta^+(i)} x_{ijk} = 1 \qquad \forall i \in N \tag{2}$$

$$\sum_{j \in \Delta^+(0)} x_{0jk} = 1 \qquad \forall k \in K \tag{3}$$

$$\sum_{i \in \Delta^-(j)} x_{ijk} - \sum_{i \in \Delta^+(j)} x_{jik} = 0 \qquad \forall k \in K, j \in N \tag{4}$$

$$\sum_{i \in \Delta^-(n+1)} x_{i,n+1,k} = 1 \qquad \forall k \in K \tag{5}$$

$$x_{ijk}\left(w_{ik} + s_i + t_{ij} - w_{jk}\right) \le 0 \quad \forall k \in K, (i,j) \in A \tag{6}$$

$$a_i \sum_{j \in \Delta^+(i)} x_{ijk} \le w_{ik} \le b_i \sum_{j \in \Delta^+(i)} x_{ijk} \quad \forall k \in K, i \in N \tag{7}$$

$$E \le w_{ik} \le L \qquad \forall k \in K, i \in \{0, n+1\} \tag{8}$$

$$\sum_{i \in N} d_i \sum_{j \in \Delta^+(i)} x_{ijk} \le C \qquad \forall k \in K \tag{9}$$

$$x_{ijk} \ge 0 \qquad \forall k \in K, (i,j) \in A \tag{10}$$

$$x_{ijk} \in \{0,1\} \qquad \forall k \in K, (i,j) \in A \tag{11}$$

In this formulation $G = (V,A)$ is a complete graph, where $V=\{0,...,n\}$ is the vertex set and A is the arc set. K is the set of vehicles and $N = V \setminus \{0\}$ is the set of clients. In this case, the vertex 0 (also represented as the vertex $n+1$) is the vehicles depot. c_{ij} represents the travel cost spent to go from vertex i to vertex j, x_{ijk} is equal to 1 if arc (i,j) is used by vehicle k and 0 otherwise. d_i is the demand of node i and C is the vehicle capacity. w_{ik} specifies the start time of service at node i when serviced by vehicle k, s_i is the service time of vertex i, t_{ij} is the time travel for arc (i,j) and a_i and b_i are the limits of the time window of vertex i. E and L represent the earliest possible departure from the depot and the latest possible arrival at the depot, respectively. $\Delta^+(i)$ denote the vertices that are directly reachable from i and $\Delta^-(i)$ denote the vertices from which i is directly reachable.

4 PGA in a Grid Environment

Several studies have been developed on the implementation of the GA to solve VRP, where representation schemes and operators of crossover and mutation have been proposed ([8], [9], [10], [11]). In [12] is presented an evolutionary algorithm, named GA-VRPTW, that uses the combination of the "*the best*" selection operator and two new operators: crossover "*crossover-k*" and intelligent mutation "*mutation-s*" operators (table 2) for solving the VRPTW. Figure 1 shows the general scheme of the GA-VRPTW.

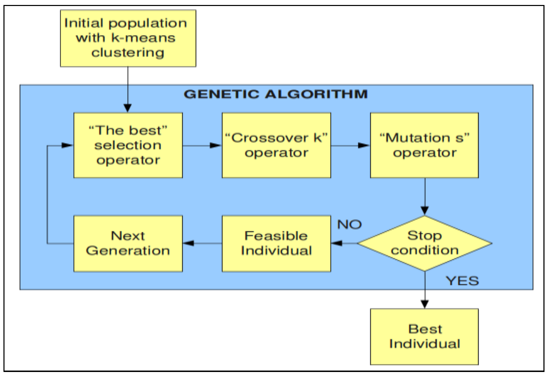
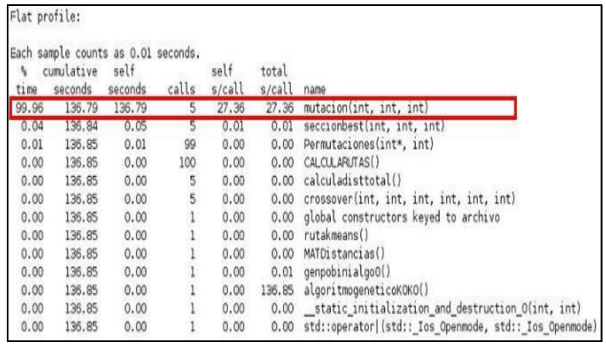


Fig. 1.- *General scheme for GA-VRPTW, proposed in [12].*

The initial population (feasible individuals) is generated with the k-means algorithm of clustering [13] for solving VRPTW. The "*mutation-s*" operator is called intelligent because it does not randomly make changes, instead it attempts to reduce the total distance traveled by only making changes that satisfy the time and capacity constraints. In order to use the GA-VRPTW in a grid environment, the "*mutation-s*" operator is applied in several segments of the population at the nodes of Tarántula miniGrid. This parallelization approach is due to the mutation operator is the process that consumes more time (figure 2).

Table 2.- *Genetic operators in GA-VRPTW.*

Operator	**Description**
The-best Selection	This operator consists of taking the best individual each time the cycle is repeated in GA.
Crossover-k	This procedure takes two numbers randomly and carries out the crossover exclusively in the node that corresponds to both individuals, in order to avoid repetition and not violate time and capacity restrictions.
Mutation-s	This operator searches for a gene with the greater distance (*gene-candidate*), and for another gene with a shorter distance (*gene-mutation*). Once the *mutation-s* operator has identified these genes, the *gene-mutation* makes the change only if the time and capacity constraints are not violated. With the movements of genes in an individual, *mutation-s* reduces the fitness.



Flat profile:

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self s/call	total s/call	name
99.96	136.79	136.79	5	27.36	27.36	mutacion(int, int, int)
0.04	136.84	0.05	5	0.01	0.01	seccionbest(int, int, int)
0.01	136.85	0.01	99	0.00	0.00	Permutaciones(int*, int)
0.00	136.85	0.00	100	0.00	0.00	CALCULARUTAS()
0.00	136.85	0.00	5	0.00	0.00	calculadisttotal()
0.00	136.85	0.00	5	0.00	0.00	crossover(int, int, int, int, int, int)
0.00	136.85	0.00	1	0.00	0.00	global constructors keyed to archivo
0.00	136.85	0.00	1	0.00	0.00	rutakmeans()
0.00	136.85	0.00	1	0.00	0.00	MATDistancias()
0.00	136.85	0.00	1	0.00	0.01	genpobinialgo0()
0.00	136.85	0.00	1	0.00	136.85	algoritmogeneticoKOKO()
0.00	136.85	0.00	1	0.00	0.00	__static_initialization_and_destruction_0(int, int)
0.00	136.85	0.00	1	0.00	0.00	std::operator|(std::_Ios_Openmode, std::_Ios_Openmode)

Fig. 2.- *Performance analysis of GA-VRPTW algorithm.*

The parallel version of GA-VRPTW, named Parallel Genetic Algorithm (PGA), uses equations (12)-(13) to define the lower and upper bounds of the division of the population between each node.

$$Lower_i = N_i*(L/NN + (N_i<res?1:0)) + (N_i<res?0:res) \quad \text{for } i=1, ..., NN \qquad (12)$$

$$Upper_i = Lower_i + L/NN + (N_i<res?1:0)-1 \quad \text{for } i=1, ..., NN \qquad (13)$$

In (12) and (13), L represents the population size, N_i is the node number, NN is the number of nodes in Tarántula miniGrid, and *res* represents the remainder of the division between L and NN. Figure 3 shows the general scheme of the parallelization of PGA.

In figure 3, an initial feasible population is created in each node of the Tarántula miniGrid. Each node carries out the selection and crossover operations. The population generated by these operators is divides in NN segments. Each node N_i executes the "*mutation-s*" operator in the segment of population delimited by the values $Lower_i$ and $Upper_i$ of the equations (12) and (13). Each node sends the mutated segment to the other nodes of the Tarántula miniGrid. All individuals collected are used for constructing a new population. This process continues until an optimal solution is found or the number of generations is reached.

5 Communication Scheme for Tarántula miniGrid

For all nodes that are part of the miniGrid can share their mutated segments, it is necessary to execute the PGA in parallel using all nodes of both clusters. It is in this moment that the use of a VPN becomes important, due to run a parallel program using MPI it is necessary that all nodes belonging to a local area network, which is achieved by setting the miniGrid as a VPN. VPN technology creates a tunnel between geographically distinct networks and uses cryptography to send all data securely through that tunnel. The performance implications of using VPN in grid computing are described in detail in [14].

To ensure that the mutated segments are shared between the nodes of the Tarántula miniGrid, a scheme in two stages to complete the communication between nodes is defined. The first stage is responsible for sending the mutated segments of the population, using the methods of the MPI library, between nodes that are shared by a cluster (UAEM or ITVer). The second stage defines the transmission of the mutated segments between the clusters of Tarántula miniGrid using the FTP protocol to reduce traffic on the network.

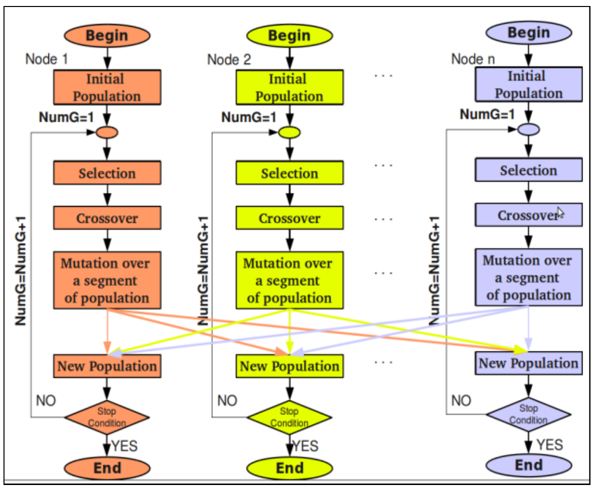


Fig. 3.- General scheme of the PGA.

Stage 1. Sending segments between nodes in a cluster: Figure 4 shows that each node (1, 2, ..., n) in a single cluster (ITVer or UAEM), after applying the mutation operator to a segment of the population, send it to the other nodes of the cluster, using the MPI_Send() and MPI_Recv() methods. In figure 4, s_1 represents the segment of the population mutated by node 1, s_2 is the segment of the population mutated by the node 2, and so on. The method MPI_Barrier() is used to ensure all nodes in one cluster have shared their mutated segments before applying, using the FTP protocol, the transmission of the group of segments between the clusters of Tarántula miniGrid.

Stage 2. Sending segments between clusters in a miniGrid: To combine the mutated segments on each cluster in the Tarántula miniGrid, it is necessary to share the mutated segments between all nodes of each cluster. If the segments are transmitted independently to each node in the miniGrid, would increase network traffic and this would reduce the performance of the algorithm. To reduce this traffic was determined to select a node of each cluster (ITVer and UAEM) as responsible for sending, using FTP, the group of mutated segments between the two clusters (figure 5). The selected nodes in each cluster are used to create a file with the mutated segments in this cluster. This file is sent to another cluster using the FTP protocol. Using one MPI message via the VPN, the selected node in a cluster communicates to another cluster that the file transmission is completed.

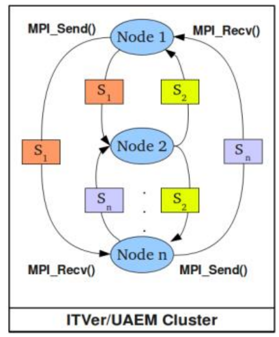


***Fig. 4.-** Sending mutated segments of the population to all nodes in the cluster.*

The mutated segments stored in the file sent by the other cluster, together with the mutated segments within the cluster itself, are used as the new population in the next iteration of the PGA. The combination of mutated segments for each node of the miniGrid, from a random initial generation of individuals in each node, produces an increase in the exploitation of the search space of GA.

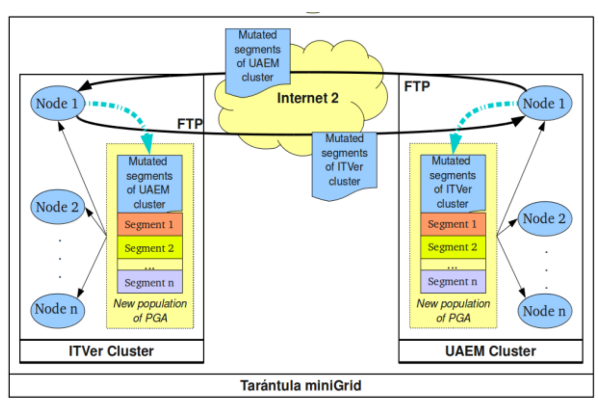


***Fig. 5.-** Sending a group of mutated segments within the Tarántula miniGrid.*

In [15] the exploration is defined has the mechanism to investigate new and unknown areas in the search space, and exploitation is defined has the technique to make use of knowledge found at points previously visited to help find better points. In PGA, the exploitation increases since each new node added in the Tarántula miniGrid uses the "*mutation-s"* operator to perform an iterative local search [12]. The increase in exploration is due to the new node also applies crossover operator, which can extend the search in different parts of the solution space. The performance of the PGA in Tarántula miniGrid also depends on the latency and speedup that arises between the clusters, which must be the most efficient.

5 Experimental Results

In order to measure the efficiency of the PGA in a grid environment, we analyze two factors: latency and speedup. Latency is a measure of time delay experienced in the miniGrid and speedup is the measure for the gain of the parallel program over the sequential version. The latency was analyzed to determine the effect of using our two stages approach for the transfer of information between the nodes of the miniGrid. On the other hand, the speedup is analyzed to determine the effect of increasing the number of nodes for the distributed application of mutation operator. The experimental basis used to measure the PGA is two instances of the test problems proposed by Salomon (C101 and C106) [16].

Latency: The latency computation is based on the transfer rates between the two clusters: UAEM and ITVer. The tests were performed with several pairs of nodes selected in the two clusters. Figure 6 shows the latency obtained throughout the day, from 12 AM to 6 PM. The results shown are the average values over a period of five days. It can be observed that the latency decreases as the day progresses. At 12 hours, higher latencies are reported, measuring between 45 and 40 seconds. At 18 hours, a latency of less than 0.4 seconds is observed. The latency tends to decrease as the day progresses. This indicates that in the afternoon the data transfer is more efficient. The latency that occurs makes the PGA efficiency decreases, so it is recommended that the latency has a low value when running the PGA. Due to the latency results produced in the Tarántula miniGrid, we recommend working with the PGA in the afternoon.

Speedup: Given that the clusters that form the Tarántula miniGrid are not homogeneous (as described in Table 1), tests to analyze the speedup were done in three blocks. First block is showed in figure 7 where the speedup obtained in the ITVer cluster is similar in the two problems discussed, but is far from the ideal speedup. This behavior occurs because the communication between nodes in the ITVer cluster is done using a switch of 10 Mb/s, which reduces the performance of the algorithm.

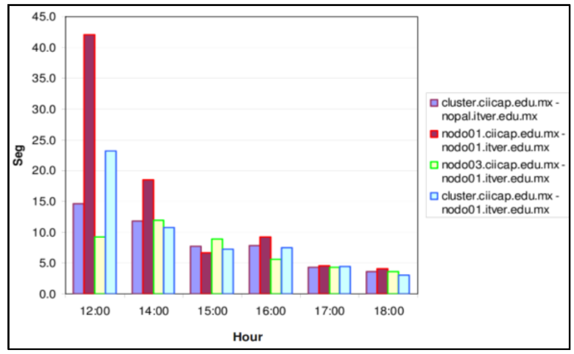


Fig. 6. *- Latency in Tarántula miniGrid.*

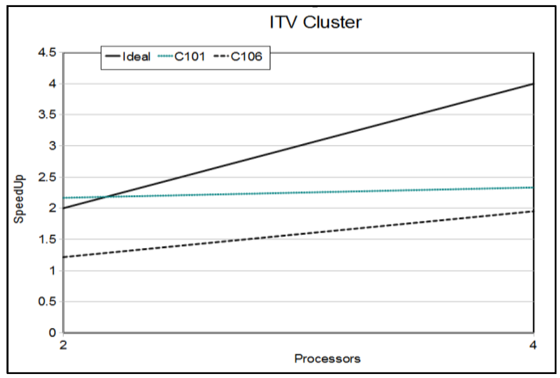


Fig. 7. *- Speedup of tests in ITVer cluster.*

Figure 8 shows that the speedup of the tests in UAEM Cluster s very close to ideal. This behavior is explained by the integrated architecture of the processors with two levels of cache and a common bus that improve efficiencies in the transfer of data. Figure 9 shows the speedup presented in the Tarántula miniGrid, where it is observed that the behavior of the algorithm is close to ideal. This is because the use of FTP-based approach reduces communication between clusters, and that the best performance of the UAEM cluster compensates for the low performance of the ITVer cluster.

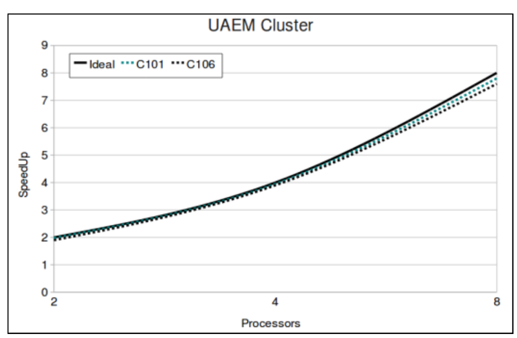


Fig. 8. *- Speedup of tests in UAEM cluster.*

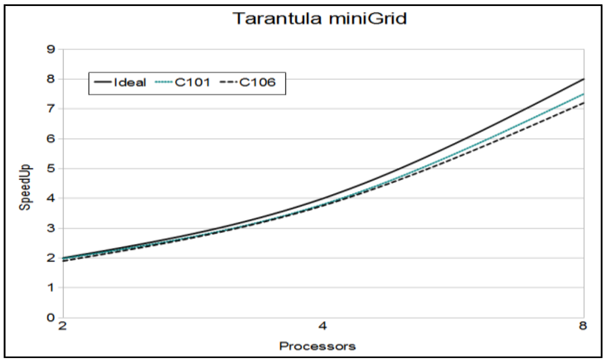


Fig. 9.- *Speedup of tests in Tarántula miniGrid.*

Conclusions and future work

The transmission using the FTP protocol of a group of mutated segments between the clusters of miniGrid, reduce the communications between clusters that are geographically distant, connected through the Internet 2. Furthermore, by applying the mutation on a segment of the population, there is a reduction in the time it is used to obtain a new population of a GA.

As future work, for reducing the negative impact that latency has on the PGA, will be increase the bandwidth between clusters and will be conduct a experimental study of message passing between nodes miniGrid with the aim of detecting possible improvements in communication between clusters. In order to increase the exploration and the exploitation of the solutions space of the VRPTW, there will be an increase in the number of nodes in the miniGrid.

References

[1] M.L. Bote-Lorenzo, Y.A. Dimitriadis, E. Gómez-Sánchez, "Grid Characteristics and Uses: A Grid Definition", *LNCS*, No. 2970, pp. 291-298, 2004.

[2] K. Fujisawa, M. Kojima, A. Takeda, M. Yamashita, "Solving Large Scale Optimization Problems via Grid and Cluster Computing", *Journal of the Operations Research Society of Japan*, 47(4), pp. 265-274, 2004.

[3] M.A. Cruz-Chávez, A. Rodriguez-León, E.Y. Ávila-Melgar, F. Juárez-Pérez, J.C. Zavala-Díaz, R. Rivera-López, "Parallel Hybrid Evolutionary Algorithm in Grid Environment for the Job Shop Scheduling Problem", *Proc. of the 2nd. EELA-2 Conf.*, pp. 227-234, 2009.

[4] M.A. Cruz-Chávez, I. Hernández-Báez, A. Rodriguez-León, E.Y. Ávila-Melgar, F. Juárez-Pérez, F., A. Martínez-Oropeza, "PSAUPMP Application in Grid EELA-2, status 5: Parallel Simulated Annealing algorithm for the weighted Unrelated Parallel Machines Problem", *Second EELA-2 Grid School, Qro., Mex*, (2009). http://applications.eu-eela.eu/app_list.php?l=20.

[5] G.B. Dantzig, R.H. Ramser, "The Truck Dispatching Problem", *Management Science*, vol. 6, pp. 80–91, 1959.

[6] M.R. Garey, D.S. Johnson, *Computers and Intractability, A Guide to the theory of NP-Completeness,* W.H. Freeman and Company, NY, 2003.

[7] P. Toth, D. Vigo, *The Vehicle Routing Problem*, SIAM, 2001.

[8] M. Rajmohan, P. Shahabudeen, "Genetic Algorithm based approach for Vehicle Routing Problem with Time Windows", *Int. Journal of Logistics Systems and Management*, 4 (3), pp. 338-365, 2008.

[9] O. Bräysy, "Genetic Algorithms for the Vehicle Routing Problem with Time Windows". *Arpakannus 1/2001,* 2001.

[10] M.A. Cruz-Chávez, O. Díaz-Parra, D. Juárez-Romero, M.G. Martínez-Rangel, "Memetic Algorithm Based on a Constraint Satisfaction Technique for VRPTW", *LNAI 5087*, Springer, pp. 376-387, 2008.

[11] M.A. Cruz-Chávez, O. Díaz-Parra, "Evolutionary Algorithm for the Vehicles Routing Problem with Time Windows Based on a Constraint Satisfaction Technique", *Computación y Sistemas*, 13(3), IPN, pp. 257-272, 2010.

[12] O. Díaz-Parra, M. Cruz-Chávez, "Evolutionary Algorithm with Intelligent Mutation Operator that solves the Vehicle Routing Problem of Clustered Classification with Time Windows", *Polish Journal of Environmental Studies*, 17(4C), pp. 91-95, 2008.

[13] S.P. Loyd, "Least squares quantization in PCM", *IEEE Trans. on Informartion Theory*, 28 (2), pp. 129-137, 1982.

[14] J. Mache, D. Tyman, A. Pinter, C. Allick, "Performance Implications of Using VPN Technology for Cluster Integration and Grid Computing", *Int. Conf. on Networking and Services*, pp. 75-80, 2006.

[15] D. Beasley, D.R. Bull, R.R. Martin, "An overview of genetic algorithms: Part 1, fundamentals". *University Computing*, 15(2), pp. 58-69, 1993.

[16] M.M. Solomon, "VRPTW Benchmark Problems". http://web.cba.neu.edu/~msolomon.