THE OPTIMAL ROUTING PROBLEM IN MULTICOMPUTER NETWORKS: AN EVOLUTIONARY APPROACH

Hernández J. L.
Facultad de Ingeniería
Universidad Nacional de Rio Cuarto
Ruta 36 Km. 601, 5800, Rio Cuarto, Argentina
e-mail: jlh@ing.unrc.edu.ar
Phone: (058) 676248

Gallard R.,
Proyecto UNSL-338403[1]
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
5700 - San Luis,Argentina
E-mail: rgallard@unsl.edu.ar
Phone: +54 652 20823
Fax : +54 652 30224

ABSTRACT

Optimal resource allocation is an important issue in computer network administration. One of these problems involves finding an optimal route to transport certain traffic from a source node to a destination node. For messages to get from the sender to the receiver it is necessary to make a number of hops choosing, at each of the intermediate nodes, an outgoing line to use. Selection of an outgoing link can depend on amount of traffic, type of link or other criteria based on the associated cost to each line. The total transportation cost through any of the possible routes is to be minimised.

Instead of facing the problem in a step by step decision making fashion, a global approach based on long term averages can be successfully used when network traffic is not extremely dynamic. Given the number of nodes in the network and the interconnection topology this later approach leads to a highly combinatorial problem.
Evolutionary Algorithms behave efficiently in searching optimal or near optimal solutions in a wide range of hard combinatorial problems. Moreover, when using an evolutionary approach, instead of a single optimal solution a set of near optimal solutions is provided. This property allows us to provide timely acceptable solutions when the network interconnectivity changes over time.
This paper describes a genetic algorithm using a sort of edge crossover, operating on variable length chromosomes. Also a macro-mutation operator is introduced by replacing an entire chromosome to avoid costly repair mechanisms.
A report on experiments and results contrasted against conventional approaches is also included.

KEYWORDS: Network routing, step by step node selection, evolutionary algorithms, edge recombination.

[1] The Research Group is supported by the Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology).

THE OPTIMAL ROUTING PROBLEM IN MULTICOMPUTER NETWORKS: AN EVOLUTIONARY APPROACH

1. INTRODUCTION

For a given topology, one of the problems to be faced in the field of Network Analysis is to determine an optimal route to transport certain amount of traffic from a source node to a destination node ([7], [8]). The objective function to minimize, in this case, is the cost function which correspond to the total sum of the partial costs associated to the intermediate links conforming the route to be followed by the traffic.

Cost allocation can be done according to different criteria. In some systems the cost is inversely proportional to the transfer speed, directly proportional to the transfer speed or computed as a function of transfer connection fares between links. Other proposals consider costs as a function of the main features of a link. King-Tim Ko [4] proposes to use the criterion of cost per distance with parameters based on transfer speeds between nodes (e.g. 6 Mbps cost one unit per kilometer, 45 Mbps cost 4 units per Km and 150 Mbps cost 9 units per Km, etc). Other metrics include more elements in the structure of costs in order to consider also dynamic characteristics of the system, such as the expected traffic at certain time intervals. These approaches attempt to predict the traffic demand and consequently the cost matrix is updated dynamically. Once a criterion to allocate costs to the links is chosen, the optimal route problem must be solved by some heuristic. The present paper shows the results when a genetic algorithm approach ([1], [2], [3]), is contrasted against a traditional greedy approach, proposed by Dijkstra and used by Bronson [9], [5], [6]).

2. THE TYPICAL APPROACH

For the classical optimal route problem (ORP) a cost is associated to each link between nodes in a connected network. The objective is to find a route between a pair of arbitrary chosen source and destination nodes in order to minimize the total cost of transport.

Dijkstra propose to assign costs to each node in the network. In his algorithm, the cost associated to a particular node is given by their neighbours nodes in the following simple fashion:

Costs are build as the sum of the partial path followed from the source node and if more than a neighbour contributes with a cost value then the lower cost is chosen.

To illustrate the algorithm we consider the network of figure 1, and we assume that nodes 1 and 8 are the source and destination nodes respectively.

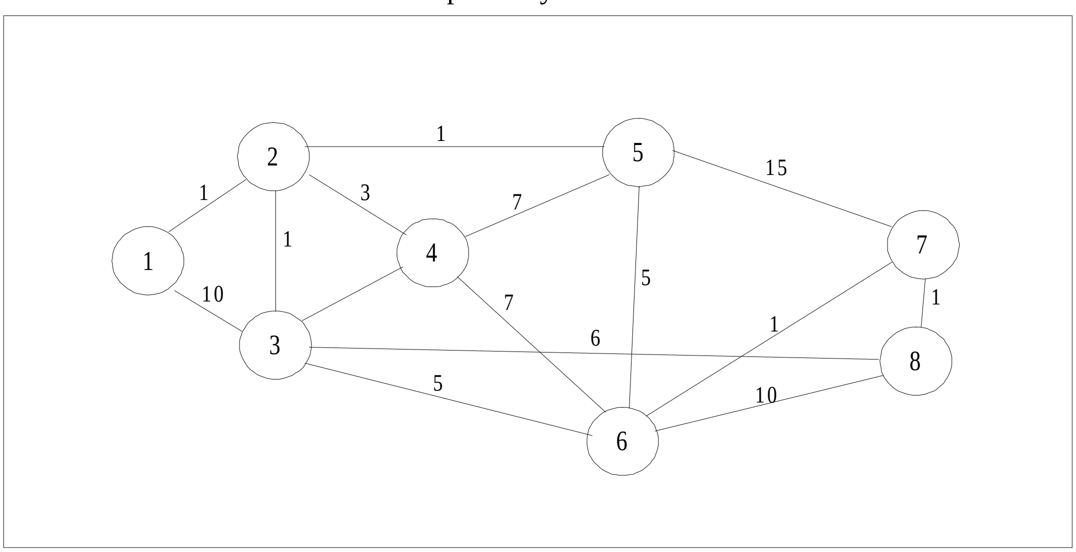

Fig. 1. Testing Net

At the beginning, node 1, being the source has no neighbour contributing with cost values then a zero value is assigned to it. From this initial assignment, only node 1 is available to assign costs to its neighbours, node 2 and 3. After assignment, node 2 have an associated cost of 1 and node 3, an associated cost of 10.

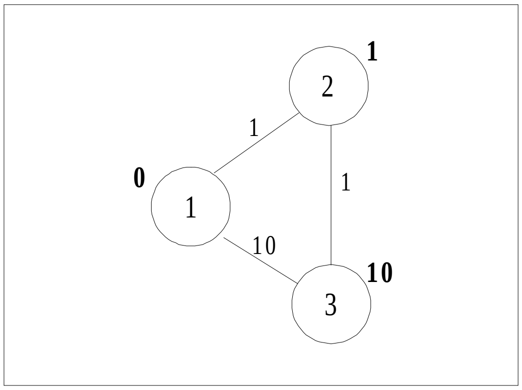

Fig 2. Assignment of costs to nodes

Proceeding in a similar fashion, during each step is necessary to consider all the nodes which can give costs to their neighbours and in the case that two or more neighbours assign the same lower value to a node the selection of the contributor is random.
For example, when considering nodes 1 and 2 as contributors, node 1 assigns 10 to node 3 and node 2 assigns 2 to node 3, 4 to node 4 and 2 to node 5.
The algorithm builds a path choosing as contributors those nodes with minimum cost associated in each step. In the case of the example nodes 3 and 5 have the same associated cost of 2 (node 3 following the partial path 1-2-3, and node 5 following the partial path 1-2-5) and the algorithm decides randomly which node (suppose node 3) is to be added to the contributor list.
As the nodes are covered the links are marked as used and the corresponding counts are updated.
When the assignment is done to the destination node the algorithm terminates and the route searched is built by traversing it towards the source node.

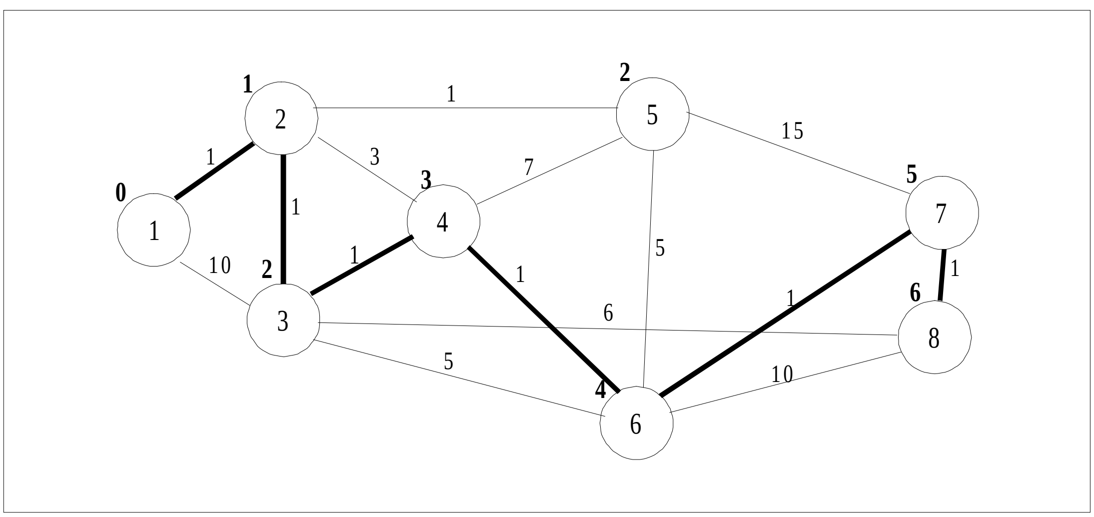

Fig. 3.Optimal route: 1-2-3-4-6-7-8. Minimum access cost to node in boldface.

If the graph of figure 3 is converted into a tree it can be observed that the algorithm prunes the tree branches which lead to higher costs and by doing it avoids intensive search and consequently excessive computing time.

3. THE EVOLUTIONARY APPROACH

There exists some similarities between the Travelling Salesman Problem (TSP) ([10], [11]) and ORP. Both of them are dealing with paths in a graph. But in TSP paths are Hamiltonian, that means that a path is a cycle and each city has to be visited exactly once . Consequently a solution includes each node of the graph and the length is always equal to the number of cities. The salesman must return to the starting city (source node matches destination node).

In ORP, the length of a route is variable depending on the topology of the network, it is not necessary to contain all the network nodes and source node differs from the destination node.

Nevertheless, TSP approaches serve as a base for representation of individuals *chromosomes*) in the population and for recombination operations.

About representation we can think of integers giving the sequence of nodes to be traversed, but chromosomes should be considered as having variable length.

In ORP, the problem is to find a sequence starting at the source node and ending at the destination node with minimum total cost associated.

In this case, for a network of n nodes it was considered a chromosome as an integer vector v of length n. In those cases where the length l of the route represented was such that $l < n$, then the vector elements v_{l+1} to v_n were set to 0. Instead of variable length strings, this sort of padding made easy data manipulation. This approach can be considered as a variant of the traditional *path representation* in genetic algorithms for TSP problems.

As an example the path P = 1-2-5-7-8, is represented by the vector v = (1,2,5,7,8,0,0,0).

The fitness of the chromosome is determined according the associated cost of the path and the selection of individuals for mating was done by using fitness proportional selection.

3.1. GENETIC OPERATORS

Under the problem restrictions, the representation adopted and in order not to include any procedure to repair unfeasible solutions, the genetic operators were carefully considered.

3.1.1. CROSSOVER

Even if the chromosome represents the order in which the nodes must be traversed for crossover it was decided to consider the arcs connecting nodes. This implies a reformulation of the crossover operator to produce a single offspring containing arcs, which are present in their parents.

To avoid creation of invalid solutions (e.g. node replication in an offspring), an auxiliary *connectivity table* is used. For example in the network of figure 1, if we have parents P1 and P2 as follows;

P1=(1, 2 ,4 ,6 ,8)
P2=(1,3,4,5,6,7,8)

According to the parents information, the corresponding connectivity table should be the following.

NODE	CONNECTED TO
1	2-3
2	4
3	4
4	5-6
5	6
6	7-8
7	8
8	-

From this table a single offspring is built by firstly choosing the source node as the first node in the new solution. The offspring begin as

$$o=(1)$$

The selected node is used to inspect the nodes connected to it and from this set, the node which in turn has the lowest number of nodes connected to it is chosen. This strategy minimise the probability of node isolation which would require a further chromosome repair including an arc absent in both parents.
In the case of more than one node with equal number of minimum connections then the selection is random. In our example we choose node 2 (nodes 2 and 3 have both only one link to other nodes).

To avoid a new selection of the same node, the node is deleted from the second column of the corresponding entry in the table. The new selected node is the next component of the offspring.

$$o=(1,2)$$

and the modified table is

NODE	CONNECTED TO
1	3
2	4
3	4
4	5-6
5	6
6	7-8
7	8
8	-

In the final stages when building the offspring we could have

$$h=(1,2,4,5,6)$$

with table content

NODE	CONNECTED TO
1	3
2	-
3	-
4	-
5	-
6	7-8
7	8
8	-

and finally

$$o=(1,2,4,5,6,8)$$

Note that the criterion of choosing the node with minimum number of links leads to a crossover operation where a single offspring is created. If all links are to be considered then a variable number of offspring could be created.

3.1.1. MUTATION

In this case a macro-mutation operator was used. If a chromosome undergoes mutation then a valid one replaces the whole chromosome. As mutation is applied with low probability, this do not disturbs local search while tries to maintain population diversity. The main goal here is to produce a valid chromosome without using any repair procedure, which are expensive in processing time. In the implementation this simply implies a call to the procedure which creates the initial population with population size set to one.

4. EXPERIMENTS AND ANALYSES OF RESULTS

A set of experiments were designed to study and contrast the genetic approach. In what follows a description of results for networks of 8, 12, 20 and more nodes until a total of 140 nodes are discussed. Main performance variables studied were quality of results and processing time. The following figures show the genetic algorithm performance when solving the ORP.

Figure 4 shows a single run in order to remark the behaviour of the genetic algorithm when better individuals are found in the population.

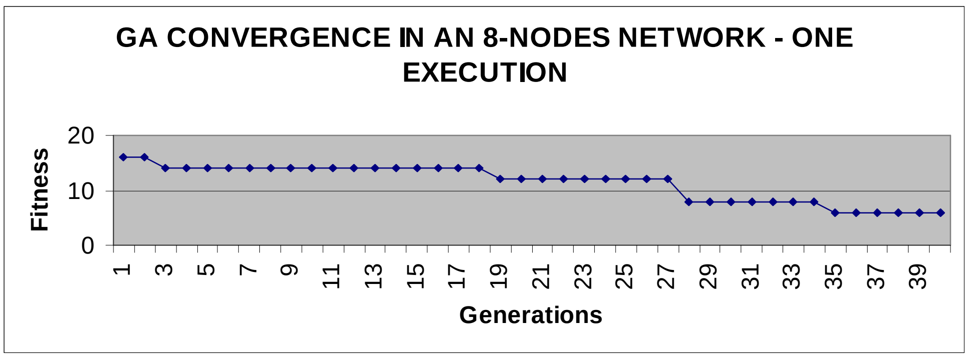

Fig. 4.

Nevertheless, the general behaviour is different: sometimes the convergence is immediate and other it is lengthy

Series of 40 runs were done with elitism, population size fixed to 10 individuals, probability of 0.75 for crossover and variable probability for mutation ranging from 0.0015 to 0.05. In figure 5, the minimum, mean and maximum value found, are shown.

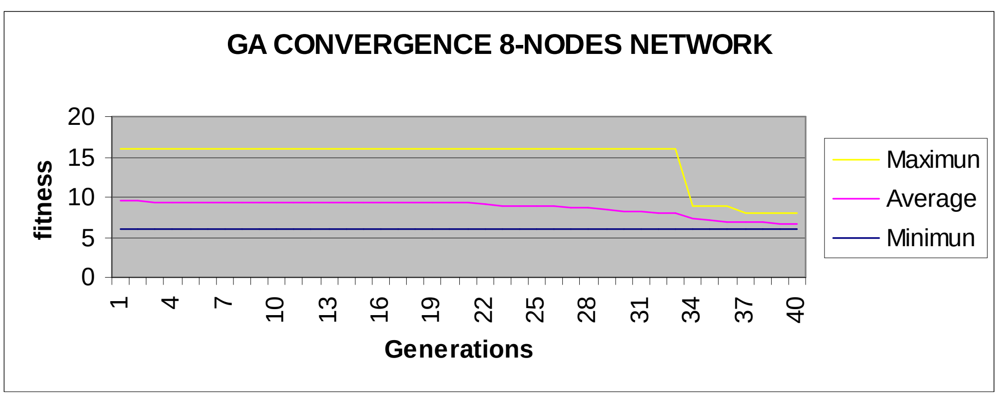

Fig. 5

The experiments evolved also through the experience gained so far. Initial experiments showed a better performance when the variable mutation probability was applied. This maintains genetic diversity and avoids being trapped into local optima. On the other hand excessive genetic diversity, which can slow down the search, was prevented by maintaining two elite individuals

(De Jong).

The following experiment was done on the 12 nodes network of figure 6. The minimum cost 9, corresponds to the route P* = 1-2-3-4-6-7-8-9-11-12.

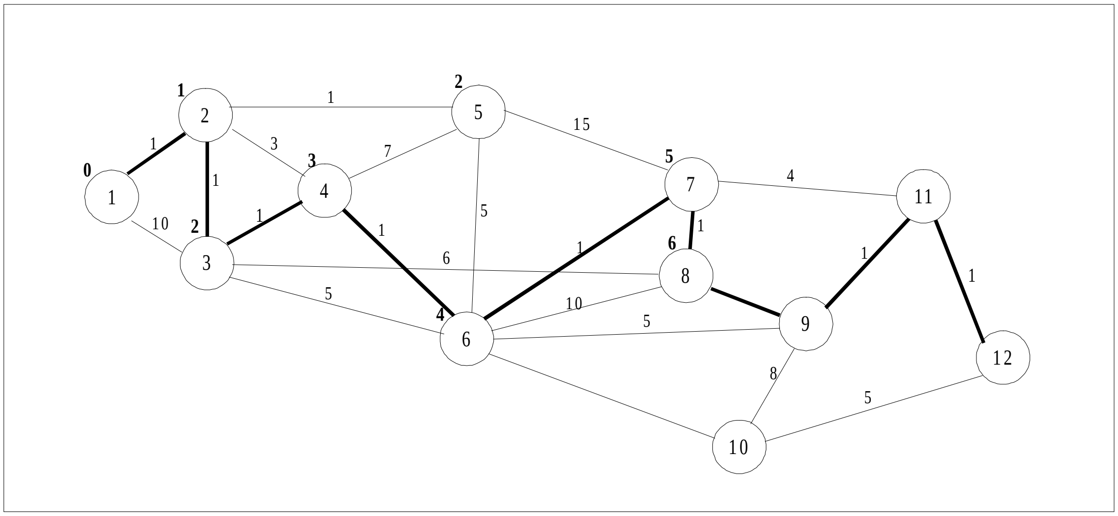

Fig. 6. Optimal route:1-2-3-4-6-7-8-9-11-12. Minimum access cost to node in boldface.

The corresponding minimum, mean and maximum values of each operation are plotted in figure 7. In this case, one elite individual was maintained, all other parameters remained with the previous settings.

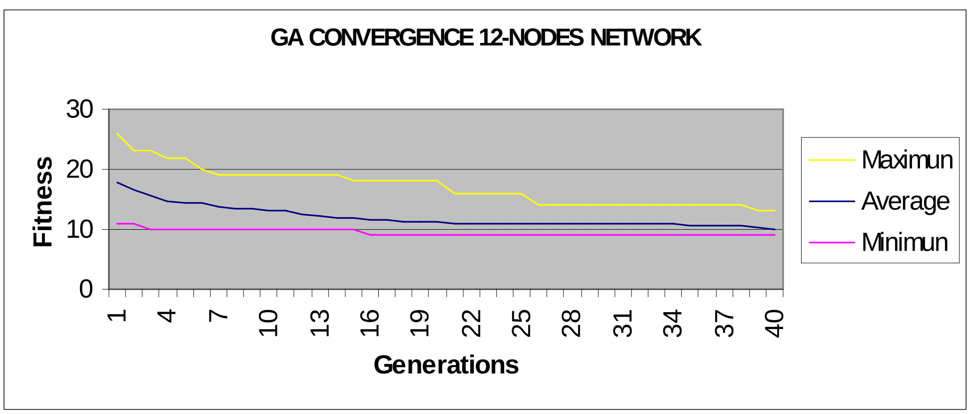

Fig. 7

Comparisons with the classical approach of Dijkstra were also performed. For this a network topology was designed in modules (See fig. 8) . The increment of complexity was associated to a concatenation of modules.

In the case of small networks the classic approach is the winner with precise optimal values and shorter running times. But when the size and complexity of the network increases the genetic algorithm is faster and the loss of quality in results is minimal.

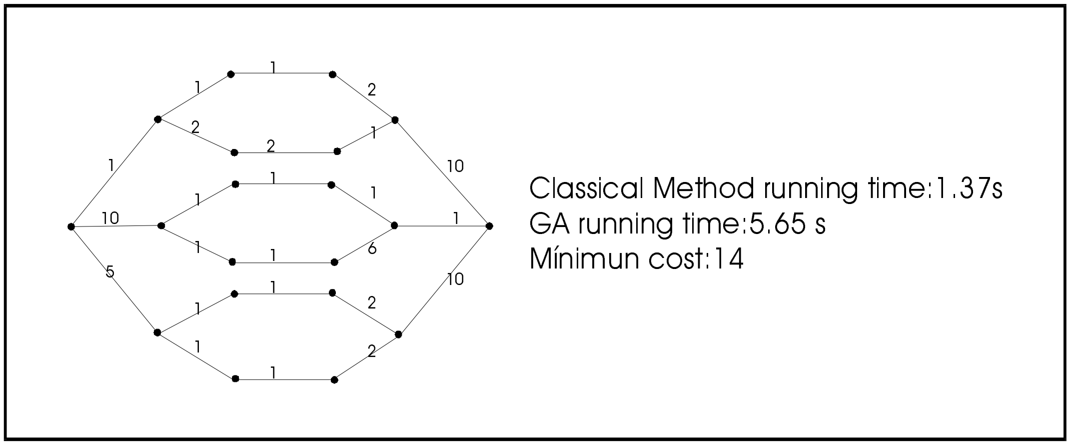

Fig. 8

About the quality of results, sometimes the genetic algorithm converges to a suboptimal but quite near optimal solution. This can be seen in figure 9.

On the other hand, the evolutionary approach has the advantage of providing a set of multiple, “good enough” solutions that can be used to face dynamical changes in system interconnectivity.

Figure 10 shows a comparison of running time for diverse degrees of complexity. Here can be observed a cross point determining when it is convenient to use the evolutionary approach.

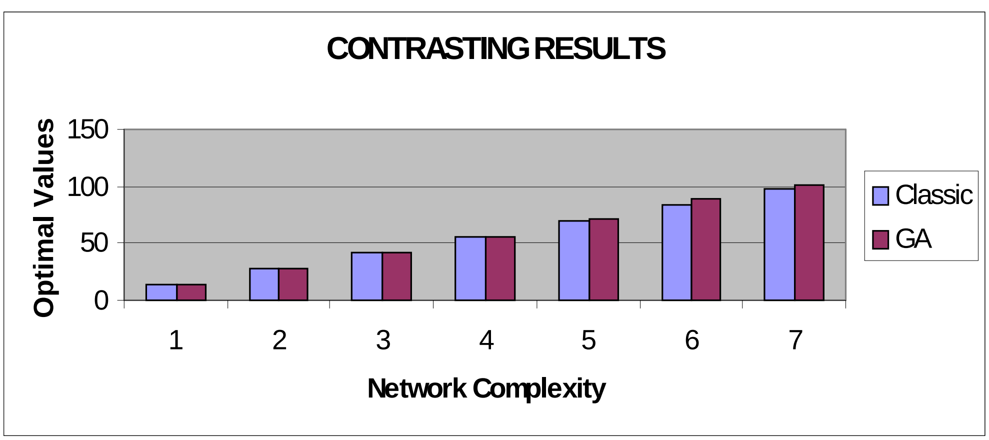

Fig. 9

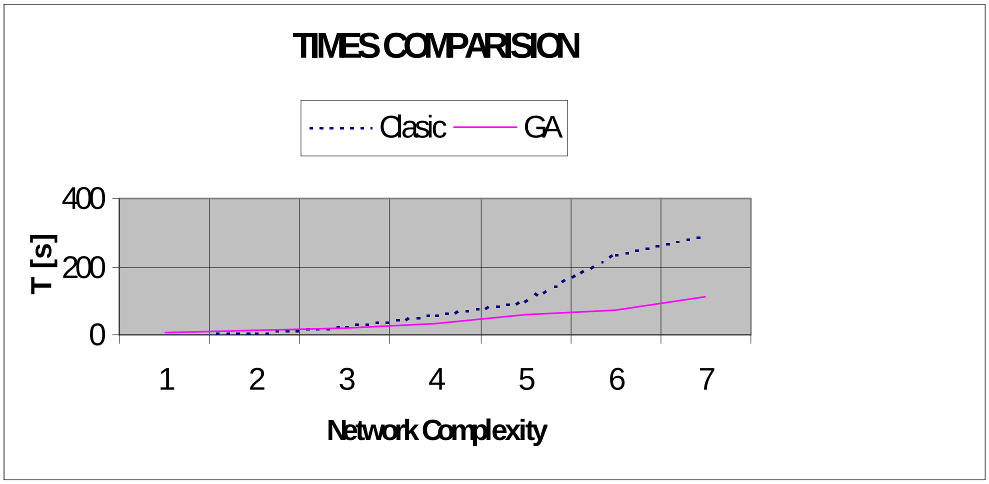

Fig. 10

CONCLUSIONS

Evolutionary computing techniques offer an alternative approach to solve the optimal route problem. Bearing this in mind, it was necessary to decide which representation and genetic operators were suitable to face the problem. Fitness proportional selection works properly in most problems and it was adopted here. In order to create legal offspring, a crossover operator exchanging arcs held by the parents and a macro-mutation operator, were devised.

Experimental results were contrasted against the Dijkstra's traditional method. For small networks the genetic algorithm performs as well as the Dijkstra's algorithm, when quality of results are considered, but it is more expensive in computing time. For large networks, even if sometimes the genetic algorithm does not find the optimal solution the best-found individuals are quite near to it. And this is done in shorter computing time. It is important to consider that, instead of providing a single and possible out of date optimal solution, the evolutionary approach offers a set of near optimal solutions which can be used as useful alternative solutions when due to the system dynamics the interconnectivity of the network changes.

Future work will investigate the possibility of creating multiple offspring by relaxing the criterion of node selection in the process of building the new child, together with varied selection mechanisms attempting to ensure a balance between genetic diversity and selective pressure.

BIBLIOGRAPHY

[1] David Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning. Addison Wesley. (1989).

[2] Zbigniew Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. Springer. 1995

[3] K.S. Tang, K.F. Man and S. Kwong. "Genetic Algorithms and their applications". IEEE signal processing magazine. Noviembre 1996.

[4] King-Tim Ko,Kit-Sang Tang, Cheung-Yau Chan Kim-Fung Man, Sam Kwong. “Using Genetic Algorithms to design mesh Networks”.Computer.V30 N°8. agosto 1997.

[5] Kamlesh Mathur, Daniel Solow. Investigación de Operaciones. Prentice Hall. 1996.

[6] Floudas Christodoulos, Nonlinear and Mixer-Integer Optimization, *Oxford University Press*, New York (1995).

[7] Douglas Comer. Internetworking with TCP/IP. Vol 1. Prentice Hall. 1995.

[8] Tanenbaum, A. S. Redes de ordenadores. Segunda Edición. Prentice Hall. 1991.

[9] Richard Bronson, Investigación de Operaciones, Mc. Graw Hill, 1993.

[10] Díaz V, Herrera N, Leguizamón G., Gallard R. “Algoritmos Genéticos aplicados a un problema de optimización combinatorial”. Anales congreso Infocom´95 pp 321-331. Buenos Aires, 1995

[11] Narshing Deo, Graph Theory with Applications to Engineering and Computer Science. Prentice-Hall, 1974.