View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Research Papers in Economics

38th CONGRESS OF THE EUROPEAN REGIONAL SCIENCE ASSOCIATION
B6 (MONDAY, 31.AUG.1998, 11:30 - 13:00).

A New Approach for Transport Network Design and Optimization

M. Bielli, P. Carotenuto
Italian Project on Transportation - National Research Council
Viale dell'Università 11 - 00185 Rome Italy
Tel. +39-6-4457127 Fax +36+6+4457186 E-Mail carotenuto@iasi.rm.cnr.it

G. Confessore
***Dept. of Computer Science, Systems and Production*
University of Rome "Tor Vergata" - Via di Tor Vergata – 00133 Rome Italy
Tel. +39-6-72597339 Fax +39-6-72597305 E-Mail: confessore@disp.uniroma2.it

ABSTRACT: The solution of the transportation network optimization problem actually requires, in most cases, very intricate and powerful computer resources, so that it is not feasible to use classical algorithms. One promising way is to use stochastic search techniques. In this context, Genetic Algorithms (GAs) seem to be - among all the available methodologies- one of the most efficient methods able to approach transport network design and optimization.

Particularly, this paper will focus the attention on the possibility of modelling and optimizing Public Bus Networks by means of GAs. In the proposed algorithm, the specific class of Simple GAs (SGAs) and Cumulative GAs (CGAs) will be used for solving the first level of the network optimization problem, while a classical assignment model ,or alternatively a neural network approach ,will be adopted for the Fitness Function (FF) evaluation.

CGAs will then be utilized in order to generate new populations of networks, which will be evaluated by means of a suitable software package. For each new solution some indicators will be calculated. A unique FF will be finally evaluated by means of a multicriteria method.

Altough the research is still in a preliminary stage, the emerging first results concerning numerical cases show very good perspectives for this new approach. A test in real cases will also follow.

1. INTRODUCTION

Urban development and the installation of new and manifold activities on a territory, with industrial, commercial or service character, cause an always increasing demand on mobility to which it is necessary to offer an efficient well-organized and well-distributed on the territory transportation system. The adjustment of old networks to new needs must occur across unitary and organic vision of the problem in order to obtain maximum economy for the resources employed and maximum functionality for transport users.

The *Network Design* is certainly the most recurrent matter, at all the decisional levels, when facing the problems of planning: at the strategic level, when it is necessary to decide about the financing of large facilities such as the construction of large infrastructure; at the tactical level, when it is necessary to reorganize the lines of an urban bus network; and in more general terms, when an urban traffic control with several transportation modes, has to be performed.

The aim of this work is to define an iterative scheme, based on *GAs*, that allows to build, from predefined *Bus Nerworks*, new ones in order to improve the old performance trying to reduce the average travel time and the management cost through the reduction of the number of vehicles employed in the network.

The algorithm is devoloped as follows: there is a starting phase where an initial collection of Bus Networks is assigned. The proposed GAs process each network in order to find new Bus Networks with best performances. In so doing, GAs play a role of an external envelope of an internal *assignment* phase. For each *n-th* generation the internal phase is defined by an assignment algorithm, that works on all the generated networks, and an aggregation procedure that takes as input the assignment values and produces, for each network, a set of predefined performance indicators. At the end of the assignment phases a *multicriteria choice procedure* runs and gives the *fitness function* values related to the performance indicator values just mentioned before.

2. GAs ALGORITHMS

GAs, described by Goldberg (1989), are based on the mechanism of the natural selection and genetics. They work starting from a population of strings which represents possible solutions of the problem; then all population of string members are evaluated using a *fitness function* (*f.f.*) and new populations are generated by genetic operators (*Selection*, *Reproduction* and *Mutation*). Several applications of GAs in optimization problems has been discussed by Michalewich (1992) and Chamberes (1995).

This method utilizes a population of strings called *chromosomes*, each representing a single member of the population and consequently representing a possible solution of the problem. Every position in the chromosome is called a *gene* and the value of a gene is called the *allelic* value. The allelic value may vary on an assigned set of values called *allelic alphabet*, most commonly the allelic alphabet is {0,1}. At each generation, this local search method uses a fitness function ($f.f. = f_i$) to evaluate the survival capacity of each string of the population and uses simple operators (Selection, Reproduction and Mutation) in order to create a new set of artificial creatures (population) which improve their *f.f.* by using pieces of the oldest ones. Figure 1 shows a simple layout of the GAs implementation.

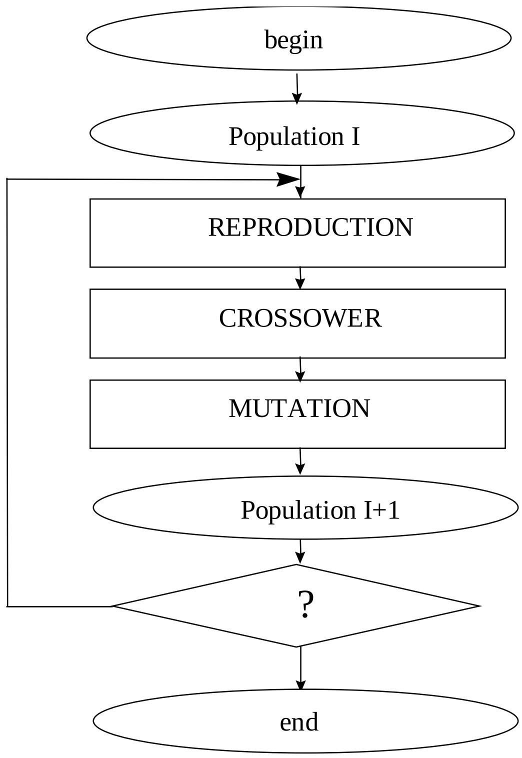

Figure 1 - Layout of the GAs implementation.

This method is different from the other local search techniques by at least four fundamental aspects:

1) it operates with codes of the parameter set and not with the parameters themselves;

2) it searches from a population of points and not a single point;

3) it uses objective function information and not derivative or auxiliary knowledge;

4) it uses probabilistic transition rules and not deterministic ones.

These particular aspects make this method applicable in a very general way, without the limitations imposed by other methods of local search (continuity, derivative existence, unimodality, and so on); they make it possible to exploit consequent information from more points than the dominion of the solutions, reducing the probability of find false peaks.

The working method of Simple Genetic Algorithms is very simple and involves nothing more complex than copying strings or swapping partial strings. Moreover, the simplicity of the operations and the power effect are two characteristics of the genetic algorithm approach which make this method very attractive.

The most commonly used operators are: *reproduction*, *crossover*, *mutation*.

2.1 Reproduction

Reproduction is a process in which individual strings are copied according to their objective function value. This represents a measure of the utility or goodness related to what we want to maximize.

Copying strings according to their fittness value means that strings with a high value have a higher probability of contribution to one or more offspring in the next generation. The easiest way to implement the reproduction operator is to create a biased roulette wheel where each string in the current population has a roulette wheel slot sized proportionally to its fitness function value.

Given a population with *n* items, to calculate the slot size of roulette wheel corresponding to the reproduction probability, we can use the following formula:

$$p_r(i) = \frac{f_i}{\sum_{i=1,\ldots,n} f_i}$$ (where *n* is the number of the total members of the population).

At each simple spin of the weighted wheel we can select a candidate and simply enter its exact copy into a mating pool for further genetic operator action (figure 2).

String Members	$f.f = (f_i)$
10110	(f_1)
11001	(f_2)
00110	(f_3)
10010	(f_4)

$$p_r(i) = \frac{f_i}{\sum_{i=1,\dots,n} f_i}$$

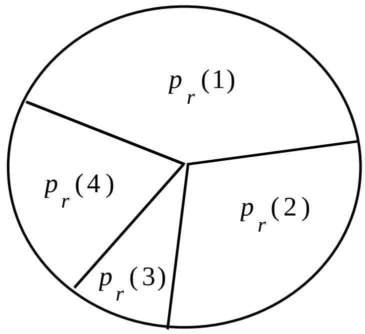

Figure 2 - Reproduction operator.

2.2 Crossover

The second operator, simple crossover, is a process that works as follows: the members reproduced in the new mating pool are mated randomly and afterward each pair of strings undergoes a cross change. In order to do this, an integer position *k* is selected uniformly at random among the string between position 1 and the string length less one [1, *l*-1]. Two new strings are created swapping all characters between selected positions *k*+1 and the end of the string. Figure 3 shows a simple crossover operator.

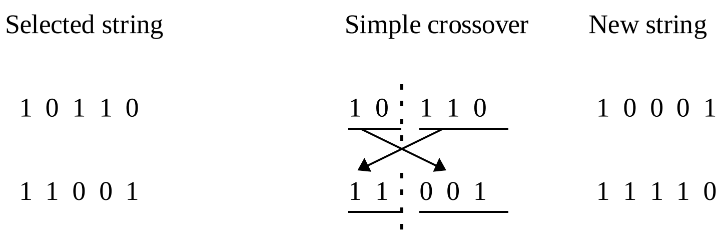

Figure 3 - Simple crossover operator.

An improvement of the simple crossover operator which enhances the exchange of information between the old population member in order to obtain the new one, is to consider two integer positions uniformly selected at random among the string, and swapping the characters between the selected positions. Figure 4 shows the crossover operator improvement.

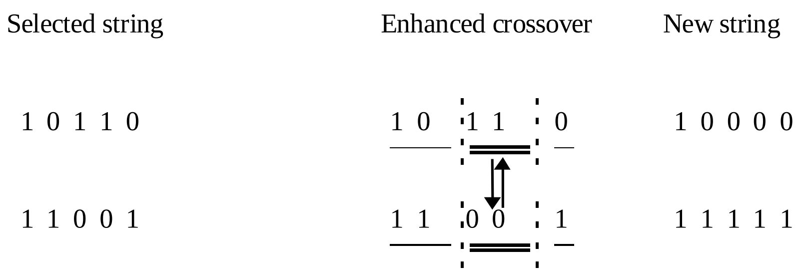

Figure 4 - Crossover operator improvement.

2.3 Mutation

The mutation operator plays a secondary role with respect to reproduction and crossover operators. Nevertheless mutation is needed to prevent an irrecoverable loss of potentially useful information that occasionally reproduction and crossover can cause. This operator is an occasional random alteration, with small probability, of the allelic value of a gene. Mutation is practically a random walk through the string space, thus guaranteeing the possibility of exploring the whole search space independently from the specific initial population and reduces the probability of finding a false peak.

3. GAs ON TRANSPORTATION ANALYSIS REVIEW

3.1 A Multicriteria Analysis for Urban Network Design and Parking Location

Cantarella and Vitetta (1994) shows that GAs can be efficiently used to solve the Urban Network Design Problem. The proposed model is multi-level.

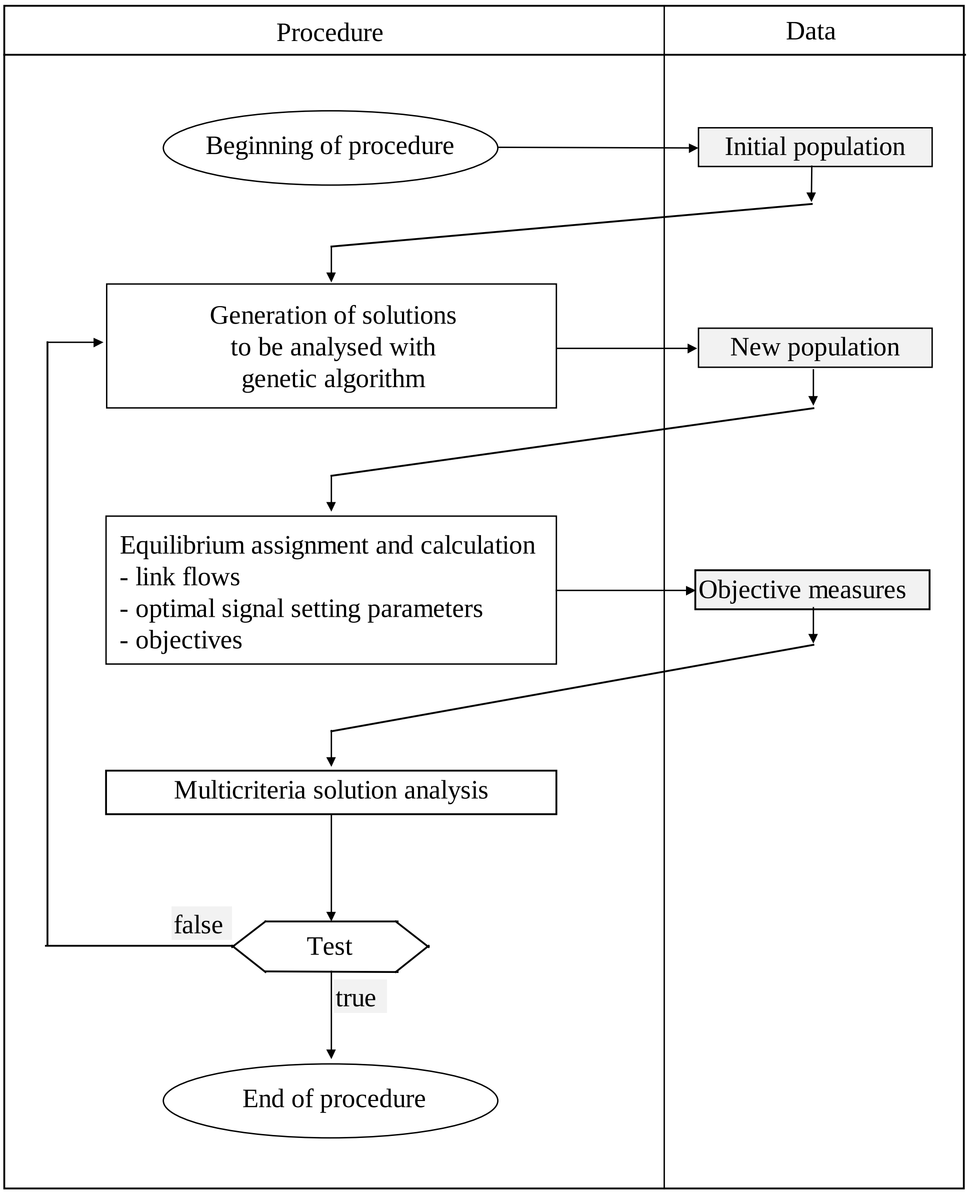

Figure 5 - The global procedure of Urban Network Design Problem - Cantarella and Vitetta (1994).

A multi-level heuristic algorithm is used in order to find nondominated approximate solutions to the Urban Network Design Problem (figure 5). At the outer level a new configuration of network is evaluated through a genetic procedure.

At the inner level the traffic signal setting and link flow assignment are carried out by an iterative method inside the traffic assignment procedure where, the signal setting, the delays on the network and the temporary flows are cyclically computed until two successive flow patterns are closed within a specified tolerance.

3.2 The Continuous Equilibrium Optimal Problem: A Genetic Approach

Cree et al. (1996) has implemented a computers program, GANDES, which is a genetic approach to solution of Network Design Problem. The inputs to GANDES are a network and parameters controlling the frequency and the type of the genetic operators. The program generates a random population of chromosomes which are evaluated and then manipulated by the GA operators to produce better chromosomes.

3.3 Transportation Network Design Using a Cumulative Algorithm and Neural Network

Xiong and Schneider (1993) introduces an improved version of GAs, the Cumulative Genetic Algorithm (CGA) and applies it to the Transportation Network Design problem, where classical GAs do not work well because good solutions can be lost in new population generations. In CGA all the population members with high f.f. are saved and used together with new population members as input for reproduction. A genetic algorithm that uses the trained neural network is represented in figure 6.

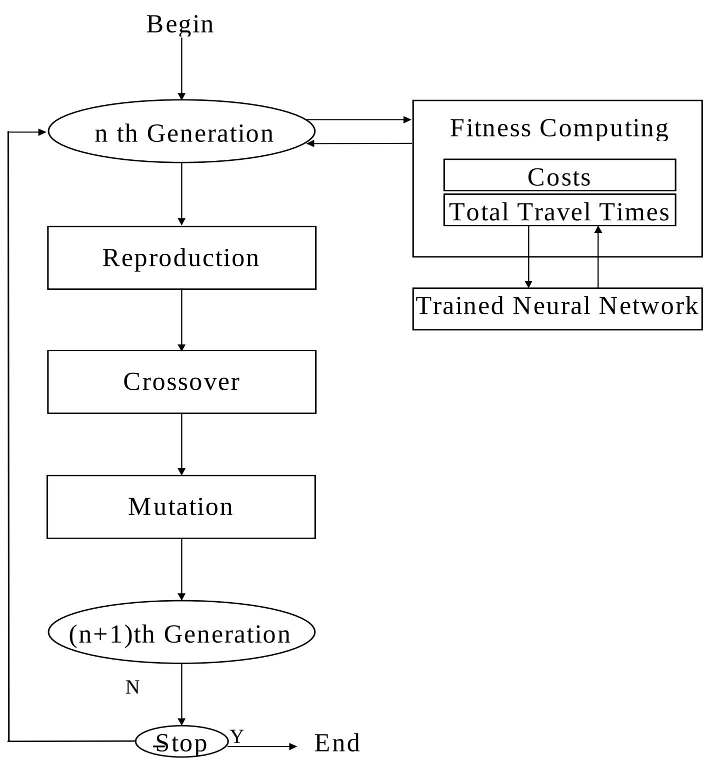

FIGURE 6 - The Simple Genetic Algorithm - Xiong and Schneider (1993).

Each solution is represented as a binary string with a length of 20, which is the same as the input variable format of the neural network. Each solution has a construction cost and total travel time, but in the genetic algorithm each solution can have only one objective (fitness) to represent its performance. To use these two values to define a solution's fitness, the domination comparison method was used. For each solution in a generation, a count is made of the number of times that it is dominated by the other solutions in the same generation. The more times it is dominated, the lower its fitness values will be. Therefore, each solution's fitness value is calculated as: fitness=*C*-number of times dominated, where *C* is the largest "number of times dominated" among all the solutions in that generation. The nondominated solutions will have a fitness *C*.

In CGA (figure 7) the reproduction operation was modified so that it picks up solutions randomly not only from the previous generation (parents), but also from *HNDSS* that is the historical nondominated solution set.

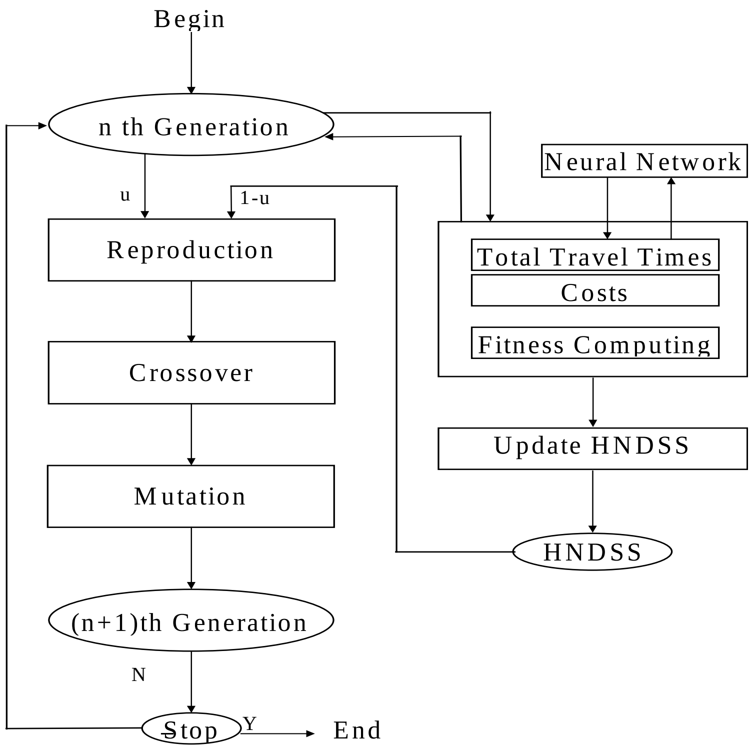

Figure 7 - The Cumulative Genetic Algorithm - Xiong and Schneider (1993).

3.4 Hybrid genetic algorithms for bus driver scheduling

Kwan and Wren (1994) describes a hybrid model for Bus Driver Scheduling Problem which incorporates GAs, a rule-based driver duties Estimator, and an integer programming method called

IMPACS. IMPACS produces near-optimal schedules. GAs are quick for directly producing driver schedules, but so far they tend to converge at sub-optimal solutions. In the hybrid approach, GAs take on an indirect role. They do not form a schedule, but IMPACS is applied using the elite population members produced to yield the optimal schedule almost trivially.

4. GAs FOR BUS NETWORK OPTIMIZATION PROBLEM

Refering to the classification of Ceder and Wilson (1986) of the transit planning process, figure 8, this work tries to unify the planning activities at level A and level B; in fact, starting with predefined Bus Networks with fixed bus lines and frequencies, the proposed algorithm tries to obtain new Bus Networks with better performances and more suitable line frequencies.

Indipendent Inputs	Planning Activity	Output
Demand data Supply data Route performance indicators	Level A Network Design	Route changes New routes Operating strategies
Subsidy available Buses available Service policies Current patronage	Level B Setting frequencies	Service frequencies
Demand by time of day Times for first & last trips Running times	Level C Timetable Development	Trip departure times Trip arrival times
Deadhead times Recovery times Schedule constraints Cost structure	Level D Bus scheduling	Bus schedules
Driver work rules Run cost structure	Level E Driver Scheduling	Driver schedules

Figure 8 - Transit Planning Process - Ceder and Wilson (1986).

Thus, the main interest is to model and optimize Public Bus Network (Baaj 1995; Cascetta 1990) and in so doing considering the previuos ideas with the necessary change we started to use simple genetic algorithm at the first level of the Network Optimization problem as reported in figure 9

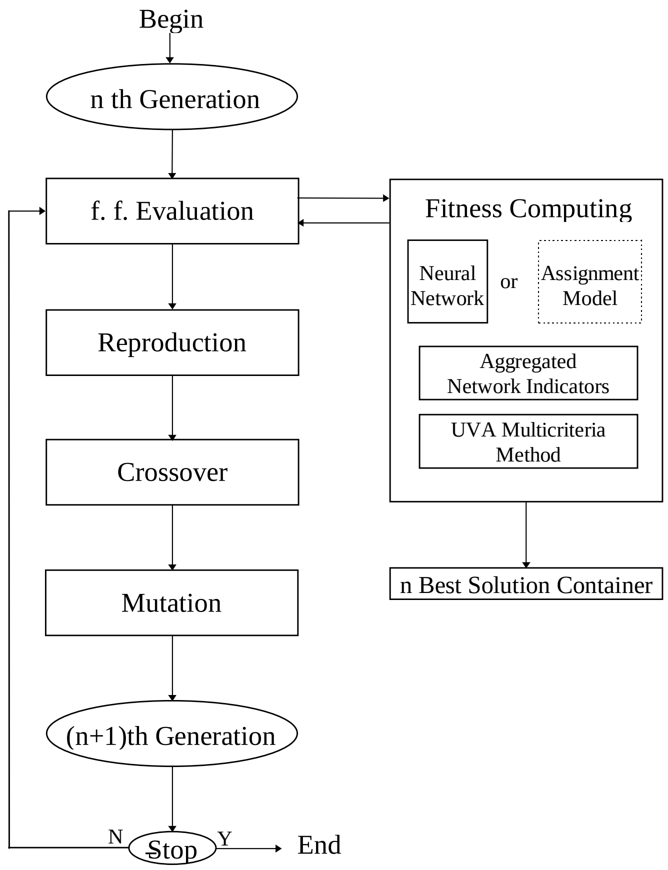

Figure 9 - Simple Genetic Algorithm.

About the genetic representation Figure 10 shows how each chromosome is the hypotetical arrangement of a bus network, and figure 11 that occour two gene to represent each line in a chromosome the first one represent the assigned frequency of the line end the second one represent an on/off switch that able or disable that line. For any specific network some line will be set on and some other will be set off. In any case we have to respect the cut point when apply crossover and mutation operator.

The figures 12 and 13 shoe how works the crossover and mutation operator in this particular implementation.

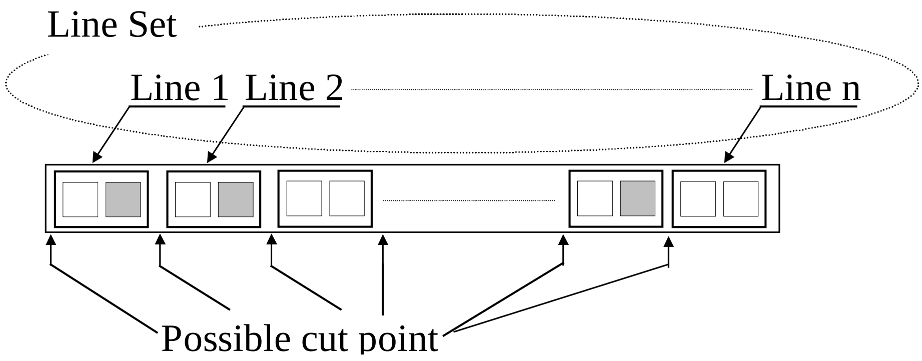

Figure 10 - Genetic Representation: Chromosome and cut point (Network).

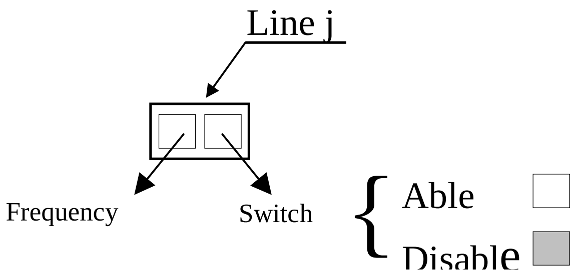

Figure 11 - Genetic Representation: Gene (Line).

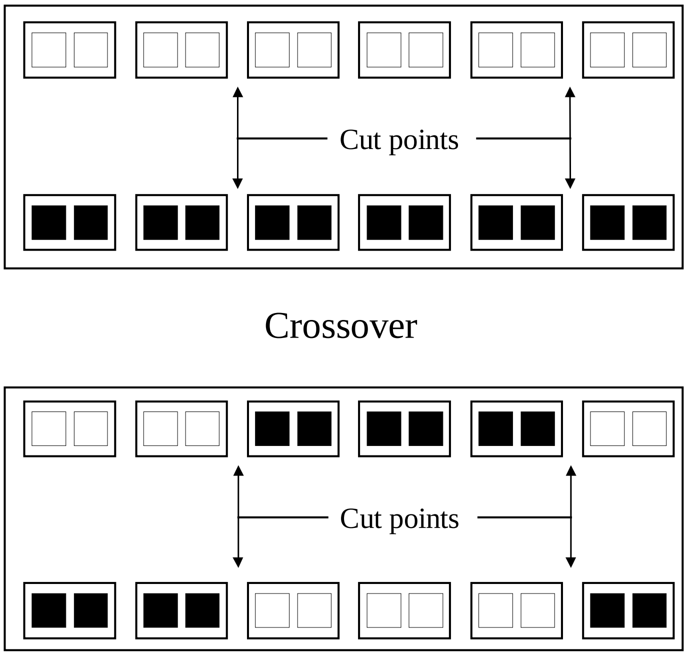

Figure 12 - Crossover Operator Implementation.

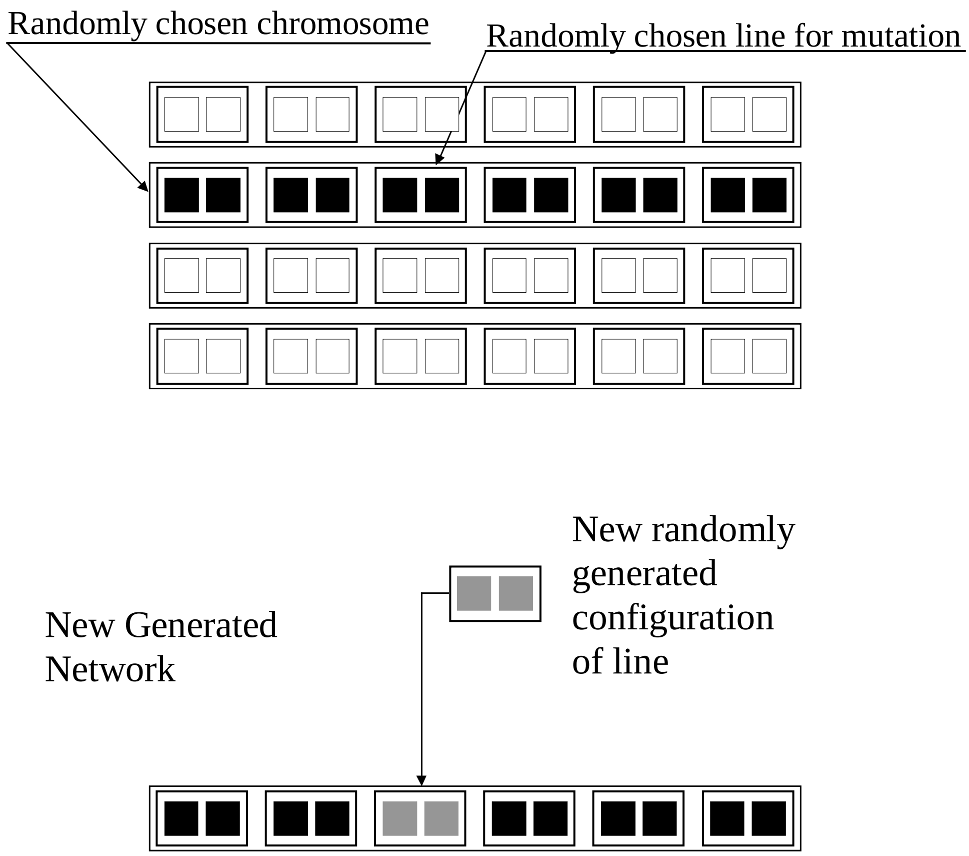

Figure 12 - Mutation Operator Implementation.

The SGA procedure starts loading the initial population provided by the planner that represent the initial network set. At each iteration, the algorithm, has to define the fitness function value for each network initially assigned.

In order to compute them three operations have to be executed.

- The first one is an assignment operation.

  Along with the networks and the line frequencies, an O/D demand has assigned, thus we use an assignment algorithm to spread the demand on the single networks.
- The second one is a routine that provides aggregated network indicator.

  For each network, once the equilibrium values are reached, the second phase starts and these values are aggregrated in order to evaluate some predefined performance indicators.
- Finally a routine that performs a multicriteria analysys.

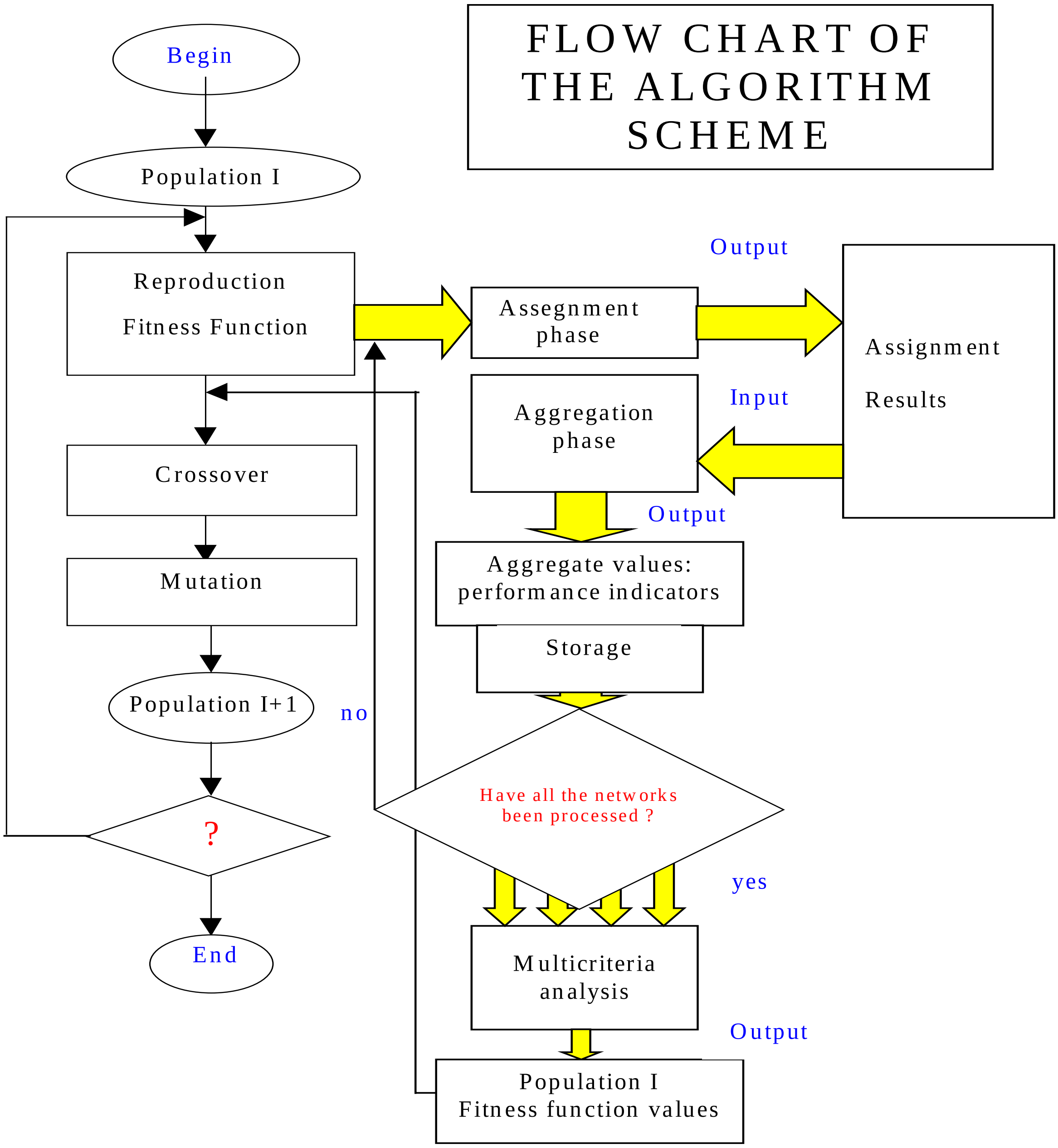

Figure 13 - Detail of Fitness Function evaluation.

When the algorithms have obtained the performance indicators for each network of the current iteration, the third phase starts where, refering to the matrix performance indicators-networks, a multicriteria choice algorithm runs in order to define a classification of the networks themselves; the positions of the single networks in the general classification obtained define the *f.f.* values that are used by the GAs to iterate the process.

When the *f.f.* are evaluated for every chromosomes the genetic operators are applaied to the actual population of network and then a new population is generated. If the end condition is false a new iteration can start,. otherwise the procedure can terminate. The algorithm stops when no improvement arises after a fixed time window or after a certain number of iterations defined previously. The figure 13 shows in detail this process.

The finally consideration is related to the n Best Solution container.This container is very useful to perform an off-line sensitivity analysis on the weight used in the multicriteria analysis process, in order to accept the results obtained or perform a new run with a new set of weigth.

5. CONCLUSIONS

This algorithm scheme seems to be very suitable if compared with the experieces of the authors citated in the algorithm review. We belive that this method can give good results and for this pourpose we have already devoleped some subroutine and have just interfaced them in order to apply the algorithm on real transportation networks.

The first step is to implement the simple genetic algorithm illustrated in figure 9 using a classic assignment algorithm in order to evaluate the CPU time and the effectiveness of the experimental results.

A further improvement is related to the assignment phase: after the generation of experimental results, we will compare those so obtained with ones obtained using a neural network approach for the assignment phase itself.

Finally the cumulative genetic algorithm will be implemeted how we have already defined (figure 14). Also in this case we will consider both the classic and neural network approach for the assignment phase, comparing the results obtained.

Moreover, in this last case, a sentitivity analysis will be effectuated on the results saved in the *n* Best Solutions Container in order to verify the robustness of the solution obtained with respect to the weight vector assigned.

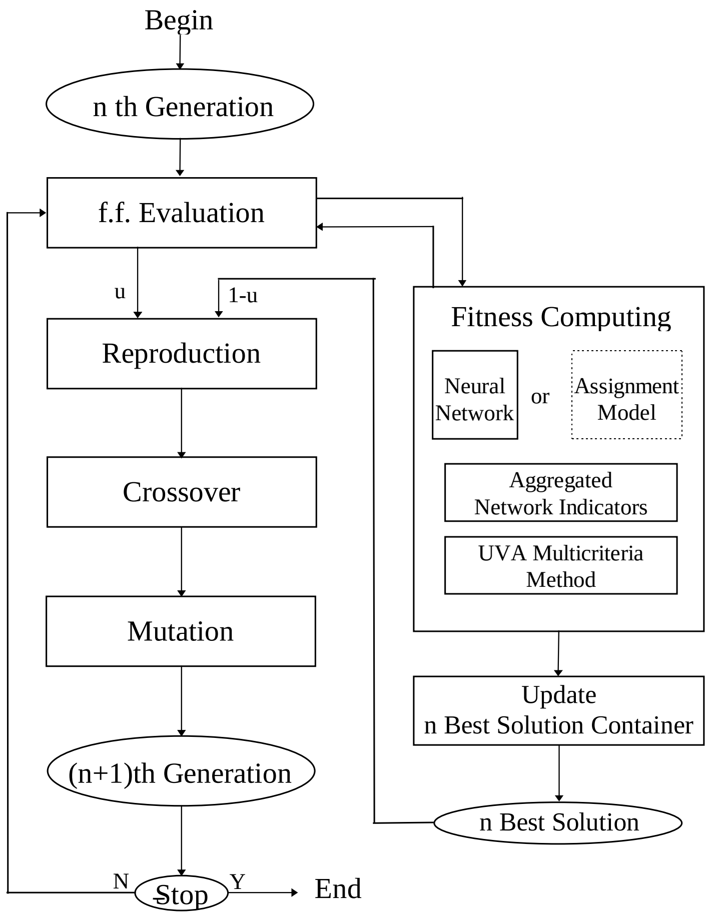

Figure 14 - Cumulative Genetic Algorithm.

6. REFERENCES

M.H. Baaj, H.S. Mahmassani, (1995) Hybrid route generation heuristic algorithm for the design of transit networks. *Transporation Research* - *C* **3**, 31-50.

M. Bielli, P.Carotenuto, M. Gastaldi, (1996) Multicriteria Evaluation Model of Public Transport Networks. *Advanced Methods in Trasportation Analysis* eds. L. Bianco, P. Toth, pp. 135-156.

M.Bielli, P. Carotenuto, (1997) Genetic Algorithms in Transportation Analysis: Review and Perspectives for Bus Network Optimization. *Proceeding of the XV EURO – XXXIV INFORMS Barcelona*.

Cantarella, G.E., Vitetta, A., (1994) A Multicriteria Analysis for Urban Network Design and Parking Location, *TRISTAN II, Conference Proceedings Capri*, pp. 839-852.

Caramia, M., Storchi, G., (1997) Multimodal Shortest Hyperpaths on Transportation Networks. *University of Rome "La Sapienza" Tech. Rep.* **25**.

Cascetta E., (1990) *Metodi quantitativi per la pianificazione dei sistemi di trasporto*. Cedam, Padova.

Ceder, A., Wilson, N.H.M., (1986) Bus Network Design. *Transportation Research - B*, **20B**, 331-344.

Chambers Lance, (1995) *Pratical Handbook of Genetic Algorithms: Applications*. CRC Press, vol 1, Boca Ratoon, Florida.

Cree N.D., Maher J. H., Paechter B., (1996) The Continuous Equilibrium Optimal Problem: A Genetic Approach. *4th Meeting of Euro Working Group on Transportation*, Newcastle (UK).

DRIVE PROJECT V1036, (1991) EVA Manual – Evaluation Process for Road Transport Informatics. eds *EVA Consortitum*.

Fernandez, E., De Cea, J., Florian, M., Cabrera, E., (1994) Network Equilibrium Models with Combined Modes. *Trasportaiton Science* **28**, 182-192.

Goldberg E.D., (1989) *Genetic Algorithms in Search Optimization & Machine Learning*. Addison-Wesley Publishing Company.

Kwan R.S.K., Wren A., (1994) Hybrid genetic algorithms for bus driver scheduling. *Tristan II Conference Proceedings*, Capri.

Michalewiciz Zbigniew, (1992) *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, Berlin Heidelberg.

Vitetta A., (1993) The Urban Network Design Problem Solved through Genetic Algorithms. *2th Euro Working Group on Transportation*, Paris.

Xiong Y., Schneider J.B., (1993) Transportation Network Design Using a Cumulative Algorithm and Neural Network. *Transportation Research Record 1364*.