

GENETIC ALGORITHMS IN OPTIMAL MULTISTAGE DISTRIBUTION NETWORK PLANNING

Vladimiro Miranda

J. V. Ranito

L. M. Proença

INESC - Inst. de Engenharia de Sistemas e Computadores
and FEUP/DEEC - Faculdade de Engenharia da Univ. do Porto
L. Mompilher - 4000 Porto - PORTUGAL

Fax: 351.2.318692 ean: vmiranda@porto.inescn.pt

Abstract - This paper addresses a genetic algorithm approach to the optimal multistage planning of distribution networks, taking in account investment costs, losses, voltage drops and reliability. The paper also presents results of an application example for a real size system. The advantages of adopting this new approach are discussed in the planning context, namely in conjunction with the use of multicriteria decision making methods.

INTRODUCTION

For optimal distribution network planning, many mathematical models have been proposed in the past - for instance, see refs [1 - 9]. Many of them [1, 2, 4, 5, 7] are static models, giving an optimal solution for a fixed set of data and a single time period. Essays in dynamic planning, considering the evolution in power demand through time and consequent topological changes in the networks (new branches, new substations, reinforcements, etc.) have never been a definite success, when applied to real sized networks. The mathematical models behind those proposals either were heavy or neglected several project features that engineers take as important on distribution system design; for example, economies gained from cables in the same trench, reuse of disclassified lines, reshaping the whole network by introducing a new substation, etc.

Furthermore, the vast majority of the model proposals, on the last 15 years, was aimed at a so called "Optimal solution". However, during the last few years the objective of reaching this optimal concept has been challenged more and more, namely within the U.S., with the acceptance of principles of the "least cost planning" approach.

This evolution favours the option for multi-criteria methods and for algorithms giving, as an answer, a large set of possibly good solutions, instead of a single optimum.

Genetic algorithms share precisely this property, and therefore an investigation has been made on their behavior in the distribution network design or distribution planning problem (DPP). This research has been directed into answering the following questions :

- Can genetic algorithms adequately model the DPP problem?
- Is it possible to develop efficient computing algorithms for real size networks?
- Has the genetic approach an easy interface with multicriteria decision making procedures ?
- Do genetic algorithms show any advantages over previous proposed approaches ?

As the following sections report, the answer is positive to all these four questions. In fact, differently from many models previously proposed, this new model is flexible enough so that many realistic features and conditions of practical nature may be taken care of. For instance:

- the inclusion of multiple feeders in the same trench, with the related savings achieved;
- the possibility of disclassifying lines at some stage, and of re-using them at a later stage;
- the possibility of generating solutions with some open loops, which may be valuable from a reliability point of view;
- considering multi-objectives;
- dealing naturally with load diversity factors;
- no need to specify, in advance, at which year should a substation be built or put into service.

REVIEW OF GENETIC ALGORITHMS

Genetic Algorithms (GA) are search and optimization methods based on natural evolution [10]. They consist on a *population of bit strings* transformed by three genetic *operators*: selection, crossover and mutation. Each string (chromosome) represents a possible solution for the problem being optimized and each bit (or group of bits), represents a value for some variable of the problem (gene). These solutions are classified by a *evaluation* function, giving better values, or *fitness*, to better solutions.

Although there are many forms [11] for Genetic Algorithms, we will only refer to the *canonical* algorithm. This means that we will be dealing with three genetic operators (selection, crossover and mutation) and linear, binary, fixed-size chromosomes. Canonical GA use a fixed-size, non-overlapping population scheme and each new generation is created by the selection operator and altered by crossover and mutation. The first population is generated at random.

Genetic Algorithm components

Each chromosome represents a potential solution for the problem to solve and must be expressed in binary form. For instance, if we want to maximize the function $f(x) = x^2$, in the integer interval $I = [0, 31]$, we could simply code x in

binary base, using 5 bits. Each solution must be evaluated by the fitness function to produce a value. In our example, the chromosome 11011 would receive the fitness value $27^2=729$, while the chromosome 00111 would receive the value $7^2=49$. The pair (chromosome, fitness) represents an *individual*.

The *selection* operator creates a new population (or *generation*) by selecting individuals from the old population, biased towards the best. This means that there will be more copies of the best individuals, although there may be some copies of the worst. This operator can be implemented in a variety of ways, although we use here a technique known as Stochastic Tournament [12]. This implementation is suited to a future distributed implementation and is very simple: every time we want to select an individual for reproduction, we choose two, at random, and the best wins with some fixed probability, typically 0.8. This scheme can be enhanced by using more individuals on the competition [13] or even by considering evolving winning probability, eventually leading to *Boltzman Tournament* [12], generalizing the *Simulated Annealing* paradigm [14].

Crossover is the main genetic operator and consists in swapping chromosome parts between individuals. The simplest crossover operator is implemented by selecting a random crossover point in the chromosome, and swapping the genes that reside between the crossover point and the end of the chromosome. For example, if we have two individuals:

A=010 | 00 ; B=010 | 11

and choose a crossover point C=3 (indicated by '|') the resulting individuals after crossover would be:

A'=010 | 11 ; B'=010 | 00

Crossover is not performed on every pair of individuals, its frequency being controlled by a crossover *probability*. This probability should have a large value, typically $P_c=0.8$.

The last genetic operator is *mutation* and consists in toggling a random bit in an individual. This operator should be used with some care, with low probability, typically $P_m=0.001$, for normal populations.

How does a Genetic Algorithm work?

A canonical GA is a very simple process: we first generate a random initial population, evaluate it and start creating new populations by applying genetic operators. This high-level behavior can be depicted on the following piece of pseudo-C:

```
main()
{
    int gen;

    generate(oldpop);
    for(gen = 0; gen < MAXGEN; gen++)
    {
        evaluate(oldpop);
        newpop = select(oldpop);
        crossover(newpop);
        mutation(newpop);
        oldpop = newpop;
    }
}
```

Obviously, there is the need for some bookkeeping functions, for statistics and so on, but they are not central to this explanation.

This very simple behavior hides a powerful processing, done by the GA. In fact, the combination of selection and crossover leads to a proliferation of individuals that possess small, tightly coupled *blocks* of bits leading to good performance. These blocks, usually called *schemata* [15], are replicated through selection and combined or separated by crossover.

And mutation, what is its job? Mutation works as a kind of "life insurance". Some important bit values (genes) may be lost during selection; mutation can bring them back, if necessary. Nevertheless, too much mutation can be harmful: a mutation probability of 0.5 always leads to random search [10], independently of crossover probability.

So, GA tends to select individuals with good performance and recombine some of their building blocks, creating more and more copies of good schemata, simply by the use of selection and crossover. This hidden processing is called *implicit parallelism* because the number of schemata processed in each generation is typically $O(N^3)$, being N the population size [15]. This compares very well with the number of fitness function evaluations, N . This characteristic is distinctive of Genetic Algorithms [11].

Genetic algorithms in Power Systems

There have been some attempts to apply Genetic Algorithms to solve problems in the Power System area, but so far, to our knowledge, not in multi-stage distribution planning. In references [16] to [22] one may find the reports of attempted approaches to problems such as:

- clustering and network reduction [16];
- optimal capacitor placement [17];
- voltage optimization [18];
- harmonics [19];
- system observability [20];
- reactive power control [21];
- load flow analysis [22];

DISTRIBUTION PLANNING MODEL

This section presents a model to solve the problems of the optimal sizing, timing and location of distribution substation and feeder expansion, using genetic algorithms. The model allows the inclusion of constraints related to network radiality, voltage drops and reliability assessment.

The objectives for distribution system planning that will be discussed are related to providing the designs and associated implementation plans necessary for an orderly expansion of facilities, minimizing new facility installation costs and operation costs, as well as achieving an acceptable level of reliability, under the following constraints :

- a) Operation of the networks under radial configuration (although some open loops may exist);

- b) Voltage drop constraints;
- c) Power demand specifications;
- d) Power flow availability, namely constrained by line thermal limits;
- e) Possible site locations for substation and lines.

Facility installation costs will be divided in three elements : substation cost, substation capacity expansion cost and new feeder cost. Power losses in the network will be taken as operation costs. Already existing elements (substations or feeders) are included at no investment cost in the model; however, their power losses are taken in account.

The following assumptions are made:

- a) A peak load is considered for each stage expansion planning (load forecasting is out of the scope of this paper). Some extra information about load curves (such as load factor) is required, in order to evaluate losses and assess reliability indices such as average load disconnected or average annual energy not supplied.
- b) New installation facility candidates are known beforehand, and their location and installation costs estimated.

An expansion strategy will be driven by load growth. The planning period will be divided into several stages (one year, for instance). One aims at having, as a result, a list of investments to be made at each stage.

The genetic algorithm approach to the DPP is drawn under the following general lines:

1. A set of variables is chosen to represent a multi-stage network solution; these variables are encoded in a chromosome.
2. A genetic algorithm is applied to a family of solutions, giving birth to new generations.
3. Each solution in the new generation is evaluated through a fitness function, that includes investment costs, power loss costs, reliability, voltage drop deviations; non compliance with other constraints is dealt with by the fitness function - an unfeasible solution, with a low fitness value, will hardly survive.
4. At the end of the process, a family of well fitted plans is available.

Variables:

For a m-stage planning problem, the following (0-1) integer decision variables could be defined:

$$F_{is} : \begin{cases} F_{is} = 1 & \text{if feeder } i \text{ is used at stage } s \\ F_{is} = 0 & \text{otherwise} \end{cases}$$

$$S_{is} : \begin{cases} S_{is} = 1 & \text{if substation } i \text{ is used at stage } s \\ S_{is} = 0 & \text{otherwise} \end{cases}$$

$$E_{is} : \begin{cases} E_{is} = 1 & \text{if subst. expansion } i \text{ is used at stage } s \\ E_{is} = 0 & \text{otherwise} \end{cases}$$

Existing facilities may be considered by fixing their respective values to 1.

Chromosome coding

The direct coding of the above variables into a chromosome has been tried and tested. Although the results obtained were satisfactory, the process was not very efficient, because the extremely large number of unfeasible solutions appearing at each generation led to a large computing time before reaching an acceptable stability.

Therefore, we devised a new coding process. Its requirements were: it should lead to a minimum amount of unfeasible solutions generated; and it should provide very fast decoding, as this operation is required at every fitness evaluation. Its characteristics are:

- a) at one time stage, some nodes may be assigned with load values, while other may have a forecasted zero load;
- b) a node with a positive load must be connected to, at least, one feeder, while nodes with zero load may either have no connections or have one or more connections in order to allow power flow to other nodes;
- c) each node is represented in the chromosome by a number of bits needed to encode the number of possible connections to it - e.g., if four lines connect to the node, one needs two bits, if it is a positive load node, or a three bit string, if it is a zero load node (to include the case where no line goes through that node);
- d) a substation is assumed as a special type of line, that may be connected to the nodes chosen as possible building sites;

This encoding strategy is such that it gives, as a result, only solutions with a number of lines=(nodes-1), a radiality condition. A few other details were included in the model, which are not relevant to its understanding but helped in gaining computer efficiency.

Fitness Function

The fitness function must reflect both the desired and the unwanted properties of a solution, rewarding the former and strongly penalizing the latter. In the DPP, desired properties are, for instance, low cost and high reliability, while unwanted features are non-radial configurations (open loops are accepted, but not closed loops), violations of thermal cable limits or of voltage drop constraints.

Fitness is evaluated *a posteriori*; therefore it may be non-linear, non-continuous, non-convex, whatever. This is very advantageous over strict mathematical programming approaches.

The general trend is to maximize fitness. Figure 1 presents the general scheme of evaluation of the fitness of a solution, represented by a chromosome, at any generation. The functions $g()$, $h()$ and $v()$, referred to in this figure, must be chosen so that, for no matter what solutions x are evaluated, one always obtains

$$g(x_i) < h(x_j) < v(x_k) < f(x_m), \quad \forall x_{i,j,k,m}$$

In function f , we have included the following:

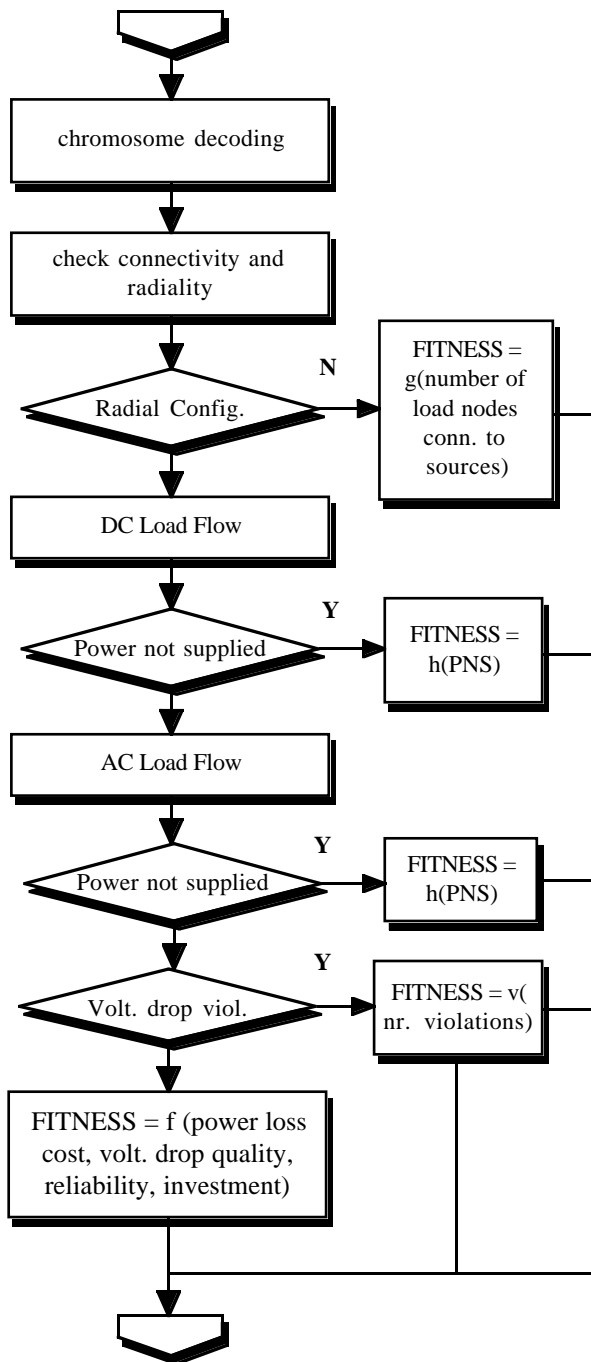


Fig.1 - Fitness function evaluation scheme

- investment costs IC are added for all the planning stages, considering a fixed discount rate;
- power loss costs PL are treated in the same way;
- voltage quality VQ may be assessed in many ways - we have used the following scheme:
 - 1) if the voltage drop at any node exceeds some threshold (e.g., 8%), the solution has been taken as unfeasible and its fitness evaluated by function v ;
 - 2) else, if the voltage drops at some nodes lie within a given interval (e.g., [5%,8%]), the voltage quality index of the solution is assessed by the sum of the square deviations to the lower limit of this interval;

3) else, if at all nodes the voltage drop stays below that lower limit, the voltage quality index receives 0 value.

- the reliability RB of a solution is evaluated through the approximate calculation of the expected annual energy not supplied - this calculation takes in account the existence of open loops in the network; we have used the following scheme:

1) an upper bound U_r in reliability level is calculated assuming that no switching devices are included in the network - therefore, disconnections take place only at the substation;

2) a lower bound L_r is calculated assuming that all branches are equipped with switching devices, allowing the isolation of failed branches and service restoration (taking in account the line capacities);

3) reliability fitness is given by

$$RB = \alpha U_r + (1 - \alpha) L_r$$

where $\alpha \in [0,1]$ is an "improvement coefficient" that aims at simulating the effect of a compromise solution in switching device location policy - as this, in itself, is a very complex problem [23].

The fitness value f of a solution x is given by

$$f(x) = M - c_1(IC+PL) - c_2VQ - c_3RB$$

where M - Large (enough) constant value;
 c_i - constants externally fixed.

Other special features

a) the DC load flow calculations are used as a first filter for unfeasibilities, and are performed at the same time as the structure of the network is recognized - this strategy proved very efficient;

b) for AC load flow calculations, we used the method described in [24], applied to balanced systems, as it was specially developed for radial networks;

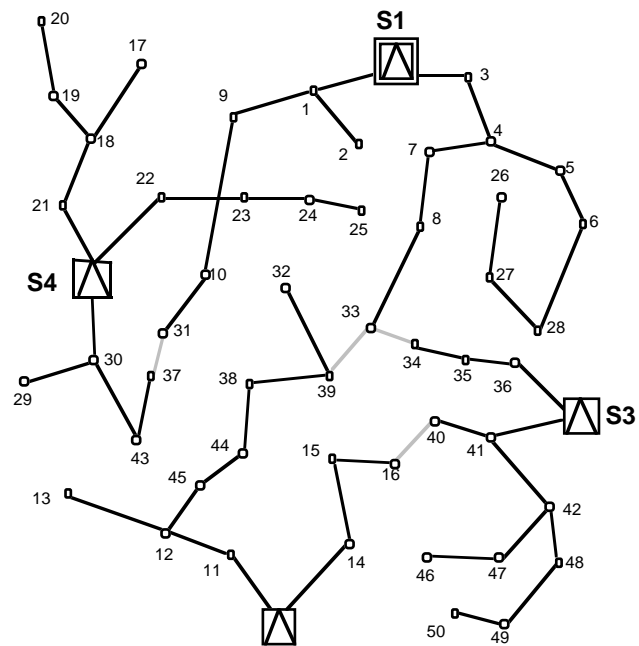
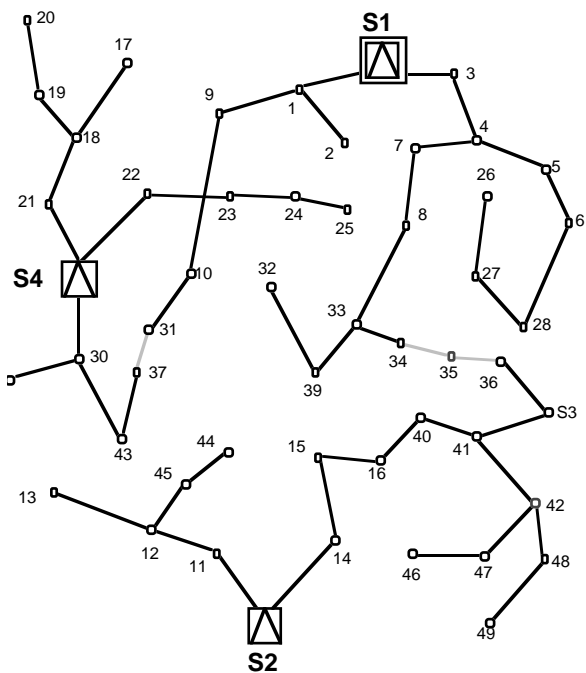
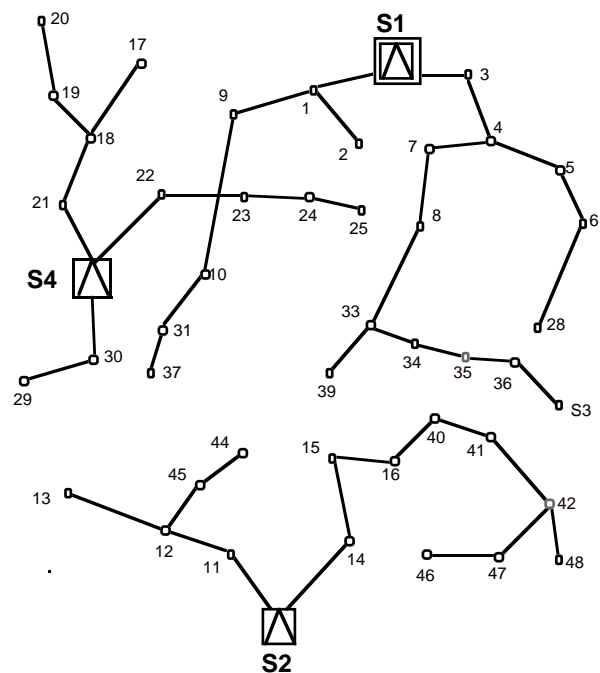
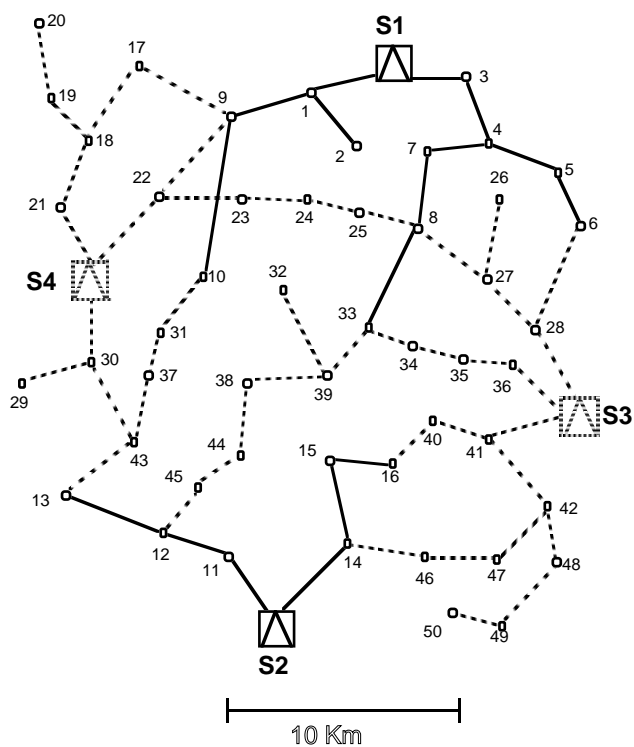
c) before load flow calculations, loads values to be used may be affected by coefficients to take in account the diversity in demand load curves - this feature also distinguishes this GA approach from other models.

APPLICATION EXAMPLE

Data

The GA approach was applied to the system shown in Fig. 2, where solid lines represent existing cables in the initial radial system, and dotted lines represent possible sites for the expansion of the system. A complete data listing may be obtained from the authors, by request. The proposed substation sizes and other complementary system data are shown below

Number of time stages	3 (+ initial)
Discount Rate	10%
Nominal voltage	15 kV
Voltage thresholds	5%, 8%
No. of nodes	3 x 50
No. of branches - total	3 x 64
No. of potential branches	3 x 48
Total load for each time stg. (MVA)	45; 63; 74.5
Feeder cost (PTE)	4*10 ⁶ /km



Substations	S1	S2	S3	S4
Initial cap (MVA)	16.7	16.7		
Poss. expansion (MVA)	16.7	13.3		
-> cost in stage 1 (PTE*10 ⁶)	100	80		
Planned (MVA)			22.2	22.2
-> cost in stage 1 (PTE*10 ⁶)			200	240

Run data - Several runs were made. The results presented were obtained for a population of 40, and for 300 generations (a value for which a robust stability in the "best-so-far" solution has been consistently observed). On average, user run times have been of around 300 sec., in a NeXTStation (33 MHz), but we are now aware that these times may still be largely improved by carefully revising the computer code.

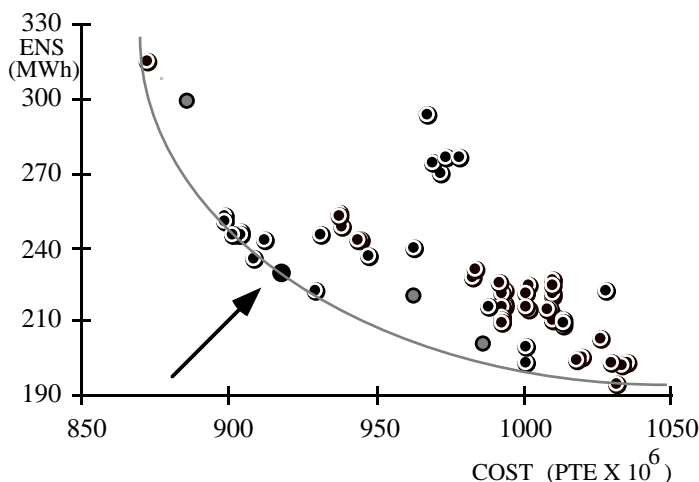


Fig. 6 - Plot of a family of solutions in a bi-attribute space (investment vs. reliability).

Results

In figures 3-4-5, a substation symbol within a square means that there has been a reinforcement in capacity of that substation; shadowed lines represent feeders already installed that are not used in a given stage, remaining, however, as open loops. See, for instance, branches 33-34-35-36, built in stage 1; in stage 2, 34-35-36 are unused, but in stage 3 they are put in active service while 33-34 is now open. Notice that node 35 has no forecasted load until stage 3; nevertheless, a solution was built through it in order to reach node S3, right in stage 1, which had a positive load value attached. Notice also the rearrangement in the network when a new substation gets in.

Figure 6 presents a family of solutions, obtained at the end of the evolution process, in a bi-attribute space (investment, in Portuguese Escudos, vs. reliability, in average annual energy not supplied, both to be minimized). We have drawn a curve representing a convex approximation to the non-dominated border; the black dot, pointed by the arrow, stands for the solution depicted in figs. 2-5, and the three shadowed dots represent some special non-dominated solutions: they would not be found by classical mathematical programming methods such as the weighted addition of the functions representing the objectives - nevertheless, they may be thought of as good compromises, by the planner.

One may clearly see how few the non-dominated solutions (also called Pareto optimal) are in this example which, in principle, are the interesting solutions to look at. This clearly indicates that the combination of genetic algorithms followed by multi-criteria screening is likely to be a powerful tool in decision aid for network planning. We believe that there is field here for further research.

CONCLUSIONS

The research reported in this paper clearly demonstrates that a GA approach to a dynamic multi-stage planning problem is both feasible and advantageous.

It provides the planner with a set of time-ordered investment decisions which is not obtained from static sub-optimizations, but directly from the consideration of the time dimension of the problem. In this respect, it is more complete than many published approaches which claim feasibility under constrained computing environments.

Furthermore, it allows the representation of non-linearities which are hard to include in pure mathematical programming methods; in fact, the existence of non-linearities enhances the advantages of using GA against pure mathematical programming. These non-linearities arise not only from the non-linear character of objective functions and constraints but also from the discrete nature of many aspects of the distribution planning problem. These in some cases could lead to a non-convex domain, perhaps in some cases not even continuous - but GA are able to deal with such environments and can detect local minima or even "islands" of solutions.

The results of a GA are a generation of solutions, filtered through the struggle for survival. Therefore, many interesting and valuable exercises on comparisons and trade offs may be executed, helping the planner to gain insight on the problem he is faced with and allowing field for better decisions to be taken.

The fact that many solutions will be available also enhances the opportunity for multicriteria methods to be explicitly applied, which is, in our point of view, a step towards an adequate direction in distribution planning.

REFERENCES

- [1] Y. Baklund, J.A. Bubenko, "Computer-aided distribution system planning", *Electrical Power & Energy Systems*, vol.1, no.1, Apr 1979
- [2] M.F. Oliveira, V. Miranda, "Etudes d'optimisation dans les réseaux de distribution comprenant des calculs de fiabilité", *Proceedings of CIRED'79*, s.6, Liège, Belgium, Apr 1979
- [3] R.N. Adams et al., "A methodology for distribution system planning", *Proceedings of the 8th PSCC*, Helsinki, Finland, Aug. 1984
- [4] M.A el-Kady. "Computer aided planning of distribution substation and primary feeders", *IEEE Transactions on PAS*, vol. 103, Jun 1984
- [5] T. Burkhardt et al., "Decision making including forecast uncertainties and optimal routing in distribution networks", *Proceedings of CIRED'85*, s.6, Brighton, U.K., Apr 1985
- [6] T. Gönen, I.J. Ramirez-Rosado, "Review of distribution planning models: a model for optimal multistage planning", *IEEE Proceedings*, Vol. 133, Pt. C, No. 7, Nov 1986

- [7] K.Aoki et al., "New approximate optimization method for distribution system planning", *IEEE Transactions on PWRs*, vol 5, no.1, Feb 1990
- [8] N. Kagan, R.N. Adams, "Application of Benders decomposition technique to the distribution planning problem", Proceedings of the 10th. PSCC, Graz, Austria, Aug 1990, ed. Butterworths, London
- [9] K.Nara et al., "Multi-year expansion planning for distribution systems", *IEEE Transactions on PWRs*, vol 6, no.3, Aug 1991
- [10] D. E. Goldberg, "Genetic algorithms in Search, Optimization and Machine Learning", 1989 Addison-Wesley
- [11] J. Grefenstette, "Conditions for Implicit Parallelism", Navy Center for Applied Research in Artificial Intelligence, Internal Report, 1991 Washington
- [12] D. E Goldberg, "A Note on Boltzman Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing", 1991, Complex Systems 3
- [13] D. E Goldberg, K. Deb, "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms", Genetic Algorithms in Search, Optimization and Machine Learning Summer School, 1991 Stanford
- [14] S. Kirkpatrick et al. "Optimization by Simulated Annealing", *Science*, vol. 220, May 1983
- [15] J. H. Holland, "Adaptation in Natural and Artificial Systems", The University of Michigan Press, 1975 Ann Arbor
- [16] H. Ding et al, "Optimal clustering of power networks using genetic algorithms", Proc. 3rd Biennial Symp. Indust. Elect. Applications, Ruston, LA, USA, LA Tech. Univ., 1992
- [17] V. Ajjarapu et al., "Application of genetic based algorithms to optimal capacitor placement", Proc. First International Forum on Applications of Neural Networks to Power Systems, New York, NY, USA, IEEE 1991
- [18] T. Haida et al., "Genetic algorithms approach to voltage optimization", Proc. First International Forum on Applications of Neural Networks to Power Systems, New York, NY, USA, IEEE 1991
- [19] H. Yang, "Worst case analysis of distribution system harmonics using genetic algorithms", Proc. IEEE SOUTHEASTCON '92 New York, NY, USA, IEEE 1992
- [20] H. Mori, "A genetic approach to power system topological observability", Proc. IEEE International Symp. on Circuits and Systems, New York, NY, USA, IEEE 1991
- [21] K. Iba, "Reactive Power Optimization by Genetic Algorithm", PICA'93, Phoenix, AR, USA, May 1993
- [22] X. Yin, N. Germany, "Investigations on Solving the Load Flow Problem by Genetic Algorithms", Electric Power Systems research, 22 (1991), 1991 Elsevier
- [23] V. Miranda, "Using fuzzy reliability indices in a decision aid environment for establishing interconnection and switching location policies", Proc. CIRED'91, s.6, Liège, Belgium, Apr 1991
- [24] V. Miranda, F.M. Barbosa, "Three phase load flow for radial networks", Proc. of MELECON'83, rp. D4.07, Athens, Greece, 1983

Vladimiro Miranda was born in Oporto, Portugal, on March 11, 1955. He received his Licenciado, Ph.D. and Agregado degrees from the Faculty of Eng. of the University of Oporto (FEUP) in 1977, 1982 and 1991, in Electrical Engineering. In 1981 he joined FEUP and currently holds the position of Professor Associado. In 1985 he joined also INESC - Research Engineering Institute for Systems and Computers and holds presently the position of Project Manager - Head of Information and Decision in Energy Systems area.

João Vasco Ranito was born in Oporto, Portugal, on January 23, 1965. He received his Licenciado and M.Sc. degrees from FEUP in 1988 and 1993, in Electrical Engineering and Computers. In 1988 he joined INESC as a researcher in the Information Systems area. He presently holds there the position of Team Leader in the Software Engineering Development Center and is a Ph.D. student.

Luís Miguel Proença was born in Lisbon, Portugal, on May 10, 1966. He received his Licenciado degree from FEUP in 1989 in Electrical Engineering and Computers. He is now a M.Sc. student at INESC, in the Information and Decision in Energy Systems group.