An application of genetic algorithms to reactive power compensation in electrical transmission systems

M. Delfanti(*) G.P. Granelli(**) P. Marannino(**) M. Montagna(**)

(*) *Politecnico di Milano, Dipartimento di Elettrotecnica, Milan, Italy*
(**) *University of Pavia, Dipartimento di Ingegneria Elettrica, Pavia, Italy*

Abstract

The problem of reactive power compensation is considered aiming to determine the location and size of the capacitor banks required to satisfy suitable network reactive constraints with the minimum investment. Due to the discrete nature of capacitor banks, a nonlinear programming problem with mixed variables results. The problem is solved by means of an iterative algorithm based on successive linear programming. The mixed integer linear programming problem to be solved at each iteration is tackled by applying either the branch and bound method or genetic algorithms. Hybrid approaches exploiting the best features of deterministic and genetic algorithms are also considered. The proposed procedures are tested and compared with reference to a small sample system and two actual networks derived from the Italian transmission and distribution system.

1. Introduction

The reactive compensation problem must provide the location and size of proper reactive power compensation devices to ensure a satisfactory voltage profile while keeping reactive generation within permissible limits. The objective function to be minimized is the cost of the investment, possibly taking into account the benefit deriving from the reduction of real transmission losses. The effects of generator voltages, tap changing transformer ratios and network topology must also be considered.

Due to the discrete nature of reactive compensation devices, the model considered leads to a nonlinear programming problem with mixed (discrete and continuous) variables. Such a formulation is presented in [1]–[4]. In former proposals [5]–[8] the sizes of reactive compensation devices are handled as continuous variables.

The solution method adopted in [1]–[3] is based on the generalized Benders decomposition which leads to consider two separate subproblems having integer and continuous variables respectively. In [4] the same problem is handled by a successive linearization method; the mixed integer linear programming problem solved at each iteration of the procedure is approached by means of a heuristic algorithm. More recently, the use of various non-deterministic methods has been described. These new proposals include genetic algorithms [9]–[11], simulated annealing [12] and tabu search [13].

In this paper, a procedure for minimum investment capacitor placement is considered. With reference to a forecasted loading scenario, the amount of capacitor installation and its location are to be determined so as to satisfy reactive operation and security constraints. The mixed integer nonlinear programming problem is solved by adopting an algorithm based on successive linear programming. The mixed integer linear programming problem to be solved at each iteration is handled either by the deterministic branch and bound technique or by genetic algorithms [14], with particular reference to the micro-genetic strategy [15]. Moreover, two hybrid procedures are designed. The former approach employs the branch and bound algorithm to test the optimality of the best current solution provided by the genetic algorithm. In the latter approach, the genetic algorithm surrogates branch and bound in case of failure due to numerical problems.

The proposed procedures are tested with reference to a small sample system and two actual networks derived from the Italian transmission and distribution system. In the tests considered, the number of variables may be over one hundred, with up to seventy discrete variables.

2. Minimum investment capacitor placement

The problem model refers to an electrical power system consisting of generation and load buses interconnected by lines and transformers (both with fixed ratio or with tap changing under-load). Some load buses are identified as candidate locations for capacitor installation.

The minimum investment capacitor placement problem requires the minimization of the following objective function:

$$\sum_i \left(C_{0i} \chi_i + C_{1i} q_{Ci} \right) \tag{1}$$

where each χ_i is a zero-one binary variable whose value depends on whether capacitors are installed ($\chi_i = 1$) or not ($\chi_i = 0$); variable q_{Ci} (either continuous or discrete) represents the size of a capacitor bank. The summation index *i* covers all candidate locations.

The objective function consists, for each candidate bus, of the sum of a fixed installation cost C_{0i} (to be accounted for only when $\chi_i = 1$) and of a cost proportional to q_{Ci} (C_{1i} being the per Mvar cost of capacitor banks).

Capacitor placement must satisfy the following constraints:

$$0 \le q_{Ci} \le \chi_i \overline{q}_{Ci} \tag{2}$$

which force q_{Ci} to zero when no capacitor installation is allowed; otherwise, an upper bound $\overline{q}_{Ci}$ is set on the capacitor size.

The problem must also take into account proper upper/lower limits on the following reactive network variables:

- voltages at all generation buses and at those load buses whose voltage is placed under surveillance ("sentinel buses")
- reactive power generations
- ratios of tap changing transformers.

Moreover, all the variables above are constrained to satisfy the load flow balance equations imposed by the network topology.

Assuming that real power generation and load are held fixed at the forecasted level, problem variables are partitioned into the subset of control and dependent variables.

The control variables are:

- voltages $\mathbf{v}_G$ at some generation buses (PV type buses)
- reactive power generations $\mathbf{q}_G$ at the remaining generation buses (PQ type buses)
- tap changing transformer ratios $\mathbf{t}$
- capacitor bank sizes $\mathbf{q}_C$ with the corresponding binary switches χ.

The dependent variables are:

- voltages relative to the PQ type generation buses and to the sentinel buses
- reactive power generations relative to the PV type generation buses.

According to the adopted non-sparse formulation of the problem, dependent variables are expressed as implicit vector functions of the control variables by making use of the load flow equations.

In conclusion, the minimum investment capacitor placement problem consists in minimizing the objective function (1) subject to constraints (2), to upper and lower bounds on the control variables and to the nonlinear functional constraints on the dependent variables. The model adopted leads to a nonlinear programming problem with mixed (discrete and continuous) variables.

3. Successive linear programming

The mixed integer nonlinear programming problem is solved by a successive linear programming algorithm which consists of the following steps:

a) The optimal value F_{best} of the objective function is initialized to an arbitrarily large number. A load flow calculation is performed using a starting value for the control variables.

b) The problem is linearized in correspondence to the load flow solution. The resulting mixed integer linear programming problem (MILP) is solved and the value of the control variables is updated.

c) A load flow calculation is performed corresponding to the current working solution.

d) Nonlinear constraints on dependent variables are checked; in case of violation, go to Step b); otherwise, go to Step e).

e) The current solution is stored together with the corresponding value F of the objective function.

f) Exiting conditions are checked. Compare F with F_{best}: if a satisfactory decrease in the objective function value is found, set $F_{best} = F$ and go to Step b); otherwise, the algorithm is over.

Problem MILP has the following form:

$$\min\left(\mathbf{C}_c^T\mathbf{x}_c + \mathbf{C}_d^T\mathbf{x}_d\right) \qquad \text{subject to:}$$

$$\mathbf{A}_c\mathbf{x}_c + \mathbf{A}_d\mathbf{x}_d \le \mathbf{b}$$

$$\underline{\mathbf{x}}_c \le \mathbf{x}_c \le \overline{\mathbf{x}}_c$$

$$\underline{\mathbf{x}}_d \le \mathbf{x}_d \le \overline{\mathbf{x}}_d$$

in which the original variables are used instead of their increments.

The control variable vector $\mathbf{x}$ is partitioned into the subsets of continuous $\mathbf{x}_c = \left[\mathbf{q}_G, \mathbf{v}, \mathbf{t}, \mathbf{q}'_C\right]^T$ and discrete $\mathbf{x}_d = \left[\mathbf{q}''_C, \chi\right]^T$ variables. Vector $\mathbf{q}_C$ is partitioned into the subsets of the continuous and discrete capacitor bank sizes $\mathbf{q}'_C$ and $\mathbf{q}''_C$ respectively.

Problem MILP is solved using any of the following procedures:

- Linear branch and bound algorithm of Land and Doig in the implementation of [16].
- Micro-genetic algorithm (μGA) [15].
- Hybrid procedure (H1) in which branch and bound is used as an optimality test for μGA. Branch and bound is fed with the current best value of the objective function obtained by μGA. Branch and bound is capable of quickly ascertaining if this value is the optimum or if further μGA search is required.
- Hybrid procedure (H2) in which μGA surrogates branch and bound when numerical problems occur. In this case, a limited search, starting from the best discrete solution obtained, was found profitable.

4. Micro-genetic algorithm approach

4.1 Generalities

Most genetic algorithms perform poorly with small sized populations because of insufficient information processing and premature convergence towards non-optimal results. The micro-genetic algorithm (μGA) [15] explores the possibility of employing smaller populations with a potential advantage deriving from the reduction of the computation burden compared with classical methods. To prevent μGA from remaining trapped in local optima, it is crucial to adopt a strategy to keep a sufficient genetic diversity in the population. In [15] a periodic refreshment of the population is performed whenever the genetic diversity has fallen below a given threshold.

The μGA implemented in the present work is reported in the following algorithm:

1. Select a population of n randomly generated individuals. Alternatively, $n-1$ individuals may be generated randomly together with one "good" individual obtained from previous search. A population of five individuals was used in all tests.
2. Evaluate fitness and determine the best individual which is always transferred to the next generation. This elitist strategy guarantees against the loss of good information embedded in the best individual produced thus far.
3. Select individuals for reproduction with the tournament selection strategy [17].
4. Apply crossover with probability equal to 1 to favor exchange of genetic material among the population.
5. Check for convergence by measuring the amount of diversity left in the population (by counting the total number of bits which are unlike those possessed by the best individual). If population diversity has fallen under a preselected threshold, go to Step 1; otherwise, go to Step 2.

Mutation is not needed according to μGA since enough genetic diversity is introduced after every convergence, when all the population (with the exception of the best individual) is randomly recreated afresh.

The tournament selection strategy is employed in Step 3 instead of the roulette wheel selection method [14]. Tournament selection is carried out by randomly picking up a number K of individuals; the one selected for reproduction is the best among this reduced population set. A value $K = 2$ was used in all tests.

4.2. Application to the minimum investment capacitor placement problem

In the solution of MILP, only the discrete variable vector $\mathbf{x}_d$ is directly operated on by μGA. The variables $\mathbf{x}_d$ are coded in binary form, with the relative upper and lower bound constraints implicitly accounted for by the "coding" mechanism [14]. Each individual (corresponding to a possible choice $\tilde{\mathbf{x}}_d$ of the discrete variable values) is evaluated by solving the following linear programming problem (LPP) obtained from MILP by setting $\mathbf{x}_d = \tilde{\mathbf{x}}_d$:

$$\min\left[\mathbf{C}_c^T \mathbf{x}_c + \mathbf{C}_d^T \tilde{\mathbf{x}}_d - C_p(\varphi - 1)\right] \qquad \text{subject to:}$$

$$\mathbf{A}_c \mathbf{x}_c - (\mathbf{b} - \mathbf{A}_d \tilde{\mathbf{x}}_d)\varphi \le 0$$

$$\underline{\mathbf{x}}_c \le \mathbf{x}_c \le \overline{\mathbf{x}}_c$$

$$0 \le \varphi \le 1$$

In the above model, C_p is a large positive penalization constant intended to drive the feasibility variable φ towards the value of 1 which corresponds to the optimal feasible solution. The value of C_p was selected at least one order of magnitude greater than the expected value of objective function. When constraints are infeasible, the solution obtained (with $\varphi < 1$) is as close as possible to constraint enforcement. The fitness of feasible individuals ($\varphi = 1$) is equal to the objective function of MILP. Infeasible individuals are provided with a direct measure of their fitness by means of the penalized objective function of LPP. Ranking of infeasible individuals succeeds in avoiding the loss of the genetic material embedded in highly promising nearly feasible individuals.

Regarding the implementation of μGA, it was found profitable to include a "smarter" individual in the initial population. The discrete variables $\tilde{\mathbf{x}}_d$ relative to this individual are obtained by solving a linear programming problem (derived from MILP by relaxing the integer variable restrictions) and successively rounding the solution to the nearest discrete values.

Finally, satisfying constraints $0 \le q_{Ci} \le \chi_i \bar{q}_{Ci}$ may become troublesome for discrete type capacitor banks since both the reactive power q_{Ci} and the corresponding binary switch χ_i belong to the discrete variable vector directly handled by the genetic algorithm. For example, individuals with a nonzero q_{Ci} and the corresponding $\chi_i = 0$ may appear in the population as a result of crossover. A character dominance strategy was devised to enforce feasibility: the χ_i character is declared dominant with respect to q_{Ci} meaning that a nonzero q_{Ci} shows up only if the corresponding χ_i is 1; otherwise it is set to 0. In the latter case, however, the binary coding of q_{Ci} is left unchanged to allow the recessive character to show up again in successive generations.

5. Tests and results

The considered procedures were tried on three electrical systems. Initial tests were performed on a small sample network (with 41 buses) derived from a CIGRE system. More significant applications regarded the Sicilian regional network, including transmission and distribution voltage levels (with about 200 buses) and the Italian transmission system (with about 500 buses). The main features of the capacitor placement problem are summarized in Table 1.

All tests were performed using a 233 MHz Pentium II personal computer. The codes implementing branch and bound (problem MILP) and linear programming (problem LPP) employ double precision arithmetic.

A comparison between the branch and bound and the μGA based procedures is reported in the remaining part of the present Section.

5.1. Tests on the CIGRE and Sicily networks

The test cases relative to the CIGRE and Sicily networks require four iterations of the successive linear programming method.

Table 1 - Characteristics of the capacitor placement problem

	Network		
Number of	CIGRE	Sicily	Italy
problem variables	55	108	147
functional constraints	84	164	315
generators	10	14	77
tap changing transformers	5	10	0
candidate sites	20	42	35
continuous capacitors	16	11	0
discrete capacitors	4	31	35
discrete capacitor sizes	21	400	475

The four MILP problems are solved by the branch and bound algorithm with an overall number of simplex iterations of 136 for the CIGRE network and of 4903 for the Sicily network. The corresponding CPU times are 1 and 15 seconds respectively.

The procedure based on μGA reaches the same optimal solution as branch and bound for each MILP, albeit at the expense of a much greater computational burden. Several tests were performed by using different randomly generated initial populations. A limit of 2000 generations was found adequate to guarantee a sufficiently reliable exploration of the solution space by means μGA.

The same test cases were run using hybrid approach H1. The adoption of branch and bound as the optimality test for μGA allows a remarkable reduction in the number of generations needed to achieve the optimum. The number of LPP iterations and the CPU times are correspondingly reduced with respect to the genuine μGA. Table 2 shows the overall data relative to the number of generations, LPP iterations and CPU times.

5.2. Tests on the Italian network

When working with the Italian network, it was impossible to complete the branch and bound procedure because of numerical problems encountered in computation of all the MILP problems. The branch and bound algorithm performs a periodic check on the accuracy of the inverse of the basis matrix. If the accuracy check is not passed, the basis matrix is re-inverted as many times as needed to regain the desired accuracy. In this case, it was impossible to maintain the desired accuracy of the inverse basis matrix; suppression of the accuracy check only led to unrecoverable numerical errors at a later stage of computation.

A sub-optimal solution of the capacitor placement problem was obtained by taking the best discrete solution achieved before the occurrence of the numerical problems in the solution of each MILP. A value of 3875 ₹ (₹ being a fictitious monetary unit) was obtained for the objective function after three sequential linear programming iterations requiring a total of 3412 simplex iterations. The CPU time needed to carry out the whole computation was 438 s.

The genetic algorithm based approach required carrying out at least 3000 generations to obtain MILP solutions comparable to those obtained by the incomplete branch and bound procedure, leading to very long computation times.

Hybrid approach H1 was not successful because the branch and bound procedure used to test the convergence of μGA still suffered from the above mentioned numerical problems.

To overcome the drawbacks of these procedures, hybrid approach H2 was employed. For each MILP, μGA is carried out for 500 generations including the incomplete branch and bound solution in the initial population.

The procedure based on H2 requires three sequential linear programming iterations and a CPU time of over two hours. A substantial saving in installation cost (3268 instead of 3875 ₹) is achieved as a consequence of the reduced amount of capacitor power installed (565 instead of 670 Mvar). Fig. 1 shows the different installation patterns given by the incomplete branch and bound and by the hybrid procedure. The figure reports only those candidate locations involved in capacitor placement.

The success of H2 mainly depends on the robustness of the genetic algorithm approach: genetic algorithms can tolerate that a few individuals in the population give rise to numerical

Table 2 - Comparison between μGA and hybrid approach H1

	Network					
	CIGRE			Sicily		
	No. of generations	No. of LPP iterations	CPU time (s)	No. of generations	No. of LPP iterations	CPU time (s)
μGA	8000	2 978 187	881	8000	2 809 308	2202
H1	2453	785 269	290	1229	350 628	254

problems in the evaluation phase. These individuals are given a low fitness value so that the relevant genetic material is not lost and contributes to some extent in building the genes of successive generations. In contrast, a deterministic search procedure may be subject to failure whenever even a single step of the computation chain cannot be carried out up to normal completion because of numerical problems.

6. Conclusions

In this paper, a procedure to solve the minimum investment capacitor placement problem is considered. A successive linear programming method is employed, which requires a mixed integer linear programming problem (MILP) to be solved at each iteration. Branch and bound and a micro-genetic algorithm (μGA) were tested in solving each MILP problem.

With the smaller test cases (CIGRE and Sicily systems), branch and bound is more efficient than μGA. Indeed, the branch and bound algorithm is designed to reduce the number of linear programming calculations required to explore the tree of possible solutions. The micro-genetic algorithm obtains the same solution as branch and bound, but at the expense of a much larger number of simplex iterations, leading to very long computation times. A hybrid approach (H1) adopting branch and bound as the optimality test for μGA allows a remarkable reduction both in the number of generations needed to achieve the optimum and in the corresponding computation burden.

In the larger test case relative to the Italian network, the branch and bound procedure was able to provide only a feasible but sub-optimal solution of MILP because of numerical problems. However, the micro-genetic algorithm would require overlong computation times. Therefore, a different hybrid procedure (H2) was employed. This procedure consists in carrying out a limited μGA search including the incomplete branch and bound solution in the initial population. Hybrid procedure H2 achieved a saving in installation cost of about 16% with respect to the incomplete branch and bound solution.

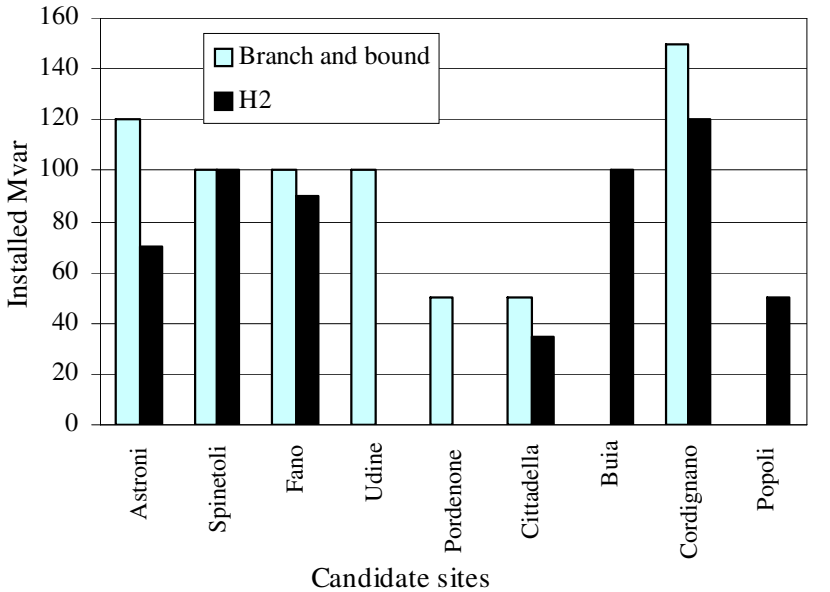

Fig. 1 - Italian system: installed capacitor patterns according to the incomplete branch and bound and hybrid procedure H2

References

[1] W.M. Lebow, R. Rouhani, R. Nadira, D.W. Sobiewski, M.K. Pal, P.B. Usoro, R.K. Mehra and M.P. Bhavaraju, "A hierarchical approach to reactive volt ampere (VAR) optimization in system planning", *IEEE Trans. Power Appar. & Systems*, Vol. PAS-104, No. 8, August 1985, pp. 2051–2057.

[2] S. Rama Iyer, K. Ramachandran and S. Hariharan, "Optimal reactive power allocation for improved system performance", *IEEE Trans. Power Appar. & Systems* , Vol. PAS-103, No. 6, June 1984, pp. 1509–1515.

[3] S. Granville, M.V.F. Pereira and A. Monticelli, "An integrated methodology for VAR sources planning", *IEEE Trans. Power Systems*, Vol. PWRS-3, No. 3, May 1988, pp. 539–557.

[4] K. Aoki, M. Fan and A. Nishikori, "Optimal VAR planning by approximation method for recursive mixed-integer linear programming", *IEEE Trans. Power Systems*, Vol. PWRS-3, No. 4, November 1988, pp. 1741–1747.

[5] R.A. Fernandes, F. Lange, R.C. Burchett, H.H. Happ and K.A. Wirgau, "Large scale reactive power planning", *IEEE Trans. Power Appar. & Systems*, Vol. PAS-102, No. 5, May 1983, pp. 1083–1088.

[6] A. Ardito, B. Cova, M. Innorta, P. Marannino and S. Nevaloro, "Active constraint strategies in optimization algorithms applied to VAR compensation planning of electric power systems", *CIGRE Conf.*, paper 37-301, Paris, February 1984.

[7] M.O. Mansour and T.M. Abdel-Rahman, "Non-linear VAR optimization using decomposition and coordination", *IEEE Trans. Power Appar. & Systems*, Vol. PAS-103, No. 2, February 1984, pp. 246–255.

[8] M. Innorta, P. Marannino and G. Montanino, "Optimal VAR planning procedures in large scale systems", *Proc. 8th PSCC*, Helsinki, August 1984, pp. 97–101.

[9] L.L. Lai and J.T. Ma, "Application of evolutionary programming to reactive power planning - Comparison with nonlinear programming approach", *IEEE Trans. Power Systems*, Vol. 12, No. 1, February 1997, pp. 198–206.

[10] K.N. Miu, H.D. Chiang and G. Darling, "Capacitor placement, replacement and control in large-scale distribution systems by a GA-based two-stage algorithm", *IEEE Trans. Power Systems*, Vol. 12, No. 3, August 1997, pp. 1160–1166.

[11] K.Y. Lee and F.F. Yang, "Optimal reactive power planning using evolutionary algorithms: a comparative study for evolutionary programming, evolutionary strategy, genetic algorithm, and linear programming", *IEEE Trans. Power Systems*, Vol. 13, No. 1, February 1998, pp. 101–108.

[12] Ying-Tung Hsiao, Chun-Chang Liu, Hsiao-Dong Chiang and Yuan-Lin Chen, "A new approach for optimal VAR sources planning in large scale electric power systems", *IEEE Trans. Power Systems*, Vol. 8, No. 3, August 1993, pp. 988–996.

[13] Yann-Chang Huang, Hong-Tzer Yang and Ching-Lien Huang, "Solving the capacitor placement problem in a radial distribution system using Tabu search approach", *IEEE Trans. Power Systems*, Vol. 11, No. 4, November 1996, pp. 1868–1873.

[14] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, 1989.

[15] K. Krishnakumar, "Micro-genetic algorithms for stationary and non-stationary function optimization", *Proc. SPIE Conference on "Intelligent Control and Adaptive Systems"*, (Guillermo Rodriguez Ed.), February 1990, pp. 289–297.

[16] A.H. Land and S. Powell, *Fortran Codes for Mathematical Programming: Linear, Quadratic and Discrete*, John Wiley, London, 1970.

[17] Z. Michalewicz, *Genetic Algortihms + Data Structures = Evolution Programs*, 3-rd edition, Springer-Verlag, Berlin, 1996.