ELSEVIER

Advanced Engineering Informatics 16 (2002) 205–215

ADVANCED ENGINEERING INFORMATICS

www.elsevier.com/locate/aei

An Evolutionary Parallel Tabu Search approach for distribution systems reinforcement planning

A. Augugliaro, L. Dusonchet, E.R. Sanseverino*

Dipartimento di Ingegneria Elettricà de, Università de Palermo, Vialle delle Scienze, Parco d'Orleans, Palermo 90128, Italy

Received 15 January 2002; revised 16 November 2002; accepted 30 November 2002

Abstract

In this paper a new meta-heuristic optimisation technique is proposed. The method is based on the Parallel Tabu Search (PTS) algorithm and the application is the optimal electrical distribution systems reinforcement planning through the installation of photovoltaic plants, parallel cables, capacitor banks and transformers. The issue is a combinatorial optimisation problem; the objective function is a non-linear expression of a large number of variables. In these cases, meta-heuristics have proved to work well and one of the most efficient is the Tabu Search algorithm. For large-scale problems, parallelisation improves Tabu Search computational efficiency as well as its exploration ability. In this paper, an enhanced version of PTS, Evolutionary Parallel Tabu Search (EPTS), is proposed. It performs reproduction operators on sub-neighbourhoods directing the search towards more promising areas of the search space. The problem of distribution systems reinforcement planning has been studied in detail and the results of the application show that the EPTS outperforms the PTS and Particle Swarm Optimisation algorithms.

The algorithm's performance is also tested on mathematical test functions and other properties of the proposed algorithm are examined.

Keywords: Combinatorial optimisation; Evolutionary computation; Parallel Tabu Search; Particle Swarm Optimisation; Optimal planning of distribution systems

1. Introduction

Since their inception in the early 1980s, meta-heuristics have developed dramatically. They have had widespread success in dealing with a variety of practical and difficult combinatorial optimisation problems. Among these approaches there are: greedy random adaptive search procedures, genetic algorithms, problem-space search, neural networks, simulated annealing, Tabu Search, threshold algorithms, and their hybrids. They include concepts based on biological evolution, intelligent problem solving, mathematical and physical sciences, nervous systems and statistical mechanics. Since the 1980s, a great deal of effort has been invested in the field of combinatorial optimisation theory in which heuristic algorithms have become an important area of research and applications.

At low and medium voltage (LV, MV) level, where the energy is actually delivered, designing networks and their reinforcement strategy are difficult problems which are getting more and more complex due to the presence of new private actors in the energy market. In particular, if the load growth in a certain area leads to capacity limits violations, the designers have to project a network reinforcement plan. There are more options available than in the past, since new technologies for dispersed generation (DG) make these systems suitable for the solution of such problems. Moreover, utilities tend to avoid any investment risk in competitive power markets and they may have difficulties in making large investments such as in building new substations or in repowering old substations. For this reason, the network reinforcement planning problem needs to be formulated again so as to take into account the DG installation as a viable alternative to traditional cables and substation reinforcement. The type of DG here considered is made of photovoltaic cells installed at the LV bus-bars of the MV/LV transformers injecting active power in the LV

* Corresponding author. Tel.: +39-091-6615221; fax: +39-91-488452.
E-mail addresses: eriva@diepa.unipa.it (E.R. Sanseverino), augugliaro@diepa.unipa.it (A. Augugliaro), dusonchet@diepa.unipa.it (L. Dusonchet).

1474-0346/02/$ - see front matter
doi:10.1016/S1474-0346(02)00012-5

network and reducing the flows of the same active power in the MV system above. Besides, the loads also require reactive power; this can be locally produced by means of capacitor banks installed at some of the load nodes of the MV system. The set of capacitor banks can be considered as well a DG but with reference to the reactive power generation. As a result, the DG of real and reactive power, reducing the current flows in the lines, reinforces the system in a way that is similar to the traditional one: (1) lines sections increase; (2) rated power of the MV/LV transformers increase. In this paper the DG technology both for real and reactive power generation, is considered as a new option for solving distribution systems capacity problems. Compound solutions consisting of both feeder and substations reinforcement and DG integration are considered in an optimal reinforcement problem formulation for distribution systems. An application to a medium size network is carried out comparing the performance of three heuristic methods: a Parallel Tabu Search approach (PTS), a Particle Swarm Optimisation approach (PSO), and an EPTS approach. The results obtained encourage the use of the latter algorithm, proposed by the authors, which incorporates some evolutionary features into the standard PTS approach.

In Ref. [1] the PTS algorithm is clearly described and the application is that of voltage and reactive power control in power systems. It develops a parallel search in different parts of the neighbourhood. The PSO is one of the modern heuristic algorithms [2,3] and it can be applied to continuous non-linear optimisation problems. It has been developed through simulation of simplified social models. As far as the planning problem is concerned, the following review can be considered. In Ref. [4], the idea of distribution systems reinforcement planning using DG resources is clearly formulated. The authors discuss the possibility to consider DG as a feasible alternative to traditional reinforcement planning. They also give an indication on how to formulate this problem in detail. In many other papers the pros and cons of the installation of DG units are studied and all the technical aspects of this problem are examined [5,6]. Other papers address the formulation and solution of optimisation problems involving both sizing and location of DG units. In Ref. [7], it is proposed a method to identify optimal locations of distributed resources to minimise losses, line loadings, and reactive power requirements. The method used to solve the optimisation problem is a second-order method. In Ref. [8], the problem of optimal sizing and location of DG units is again taken into consideration. The authors use a genetic algorithm approach considering technical constraints as voltage profile and short circuit current at the load nodes. In Ref. [9], the authors develop a method for generating a combination of several construction plans of distribution systems, considering the yearly increase of network loads, but they do not include the installation of DG units.

This paper describes an optimal reinforcement planning method for medium voltage networks, considering DG as well as other conventional options. The issue is a combinatorial optimisation problem, with non-linear objective function and constraints; therefore, a heuristic method has been used. The solution algorithm is a modified version of the PTS algorithm in its standard form, including an evolutionary phase based on a suitable selection mechanism. The PTS is used to improve the performance of TS with a couple of strategies [1,10,11]: one is the decomposition of the neighbourhood into sub-neighbourhoods in order to reduce the computational effort; the other is to introduce multiple Tabu lengths into TS, in order to maintain a diversity of solution candidates and provide better solutions. In Section 2, the problem of distribution systems reinforcement planning is presented and formulated.

2. The problem of distribution systems reinforcement planning

One of the main problems for distribution utilities is to determine the optimal reinforcement strategy to provide reliable and economical service to customers. The design of distribution systems is carried out so that the system can work correctly for a given number of years, with no additional intervention. In this case, the load increase is evaluated on a statistical basis. In the real life operation of distribution systems, unexpected and localised load increases can take place. In this case, the capacity of the system that has been designed for lower load levels is exceeded and the main negative effects are: the worsening of the voltage profile at the load nodes and the increase of the currents in the systems' elements, such as lines and transformers, with overheating effects. In order to back up this situation, utilities perform the reinforcement of the system by means of the traditional installation of new lines and transformers instead of (or in parallel with) the existing ones or by means of the installation of new power sources.

A problem of network reinforcement planning is dealt with through taking into consideration the installation of DG units as a further means to perform network reinforcement. In this way, the power source is installed locally, in the same place where it is required and the system above is relieved from the burden of transmitting and delivering more power.

Both DG installation and traditional network reinforcement through cables and transformers can be adopted. For simplicity reasons, the considered DG units are (i) only photovoltaic (PV) plants to be installed at the LV busbars, as far as real power generation is concerned; (ii) capacitor banks installed on the MV side of MV/LV transformers for the reactive power generation. The issue studied here is a combinatorial optimisation problem, where the main objective is that of minimising the overall cost while meeting technical constraints such as keeping the voltage drop below a certain value and the currents below the elements capacity. The overall cost can be expressed as the summation of different terms, some deriving from the installation of the elements, others from the operating conditions. In the present formulation, for a given load factor increase, $\Delta\alpha$, each node can be equipped with distributed PV systems at the LV level and with capacitor banks at the MV level. The capacity of each branch can be multiplied by installing some other cables in parallel with the existing ones. As a consequence, in this case, the capacity of each substation can be increased by installing other transformers in parallel with the existing ones.

The load courses follow the load profile of the average day of the year and they are considered all to be of residential type and 'conformally varying' (two load duration diagrams, $P_1(t)$ and $P_2(t)$, are conformally varying if $P_2(t) = kP_1(t)$, $\forall t$.

Fig. 1 shows a generic MV/LV load node of the considered system. There are different cost terms that must be considered as follows.

(a) *The PV systems installation cost*

$$C_1 = C_{PV} \times P_{PV} \tag{1}$$

where C_{PV} is the yearly cost per kW, whereas P_{PV} can be expressed as

$$P_{PV} = (P_{OUT}/\eta_{PV})\eta_C \tag{2}$$

where P_{OUT} is the required output active power and η_{PV} and η_C, respectively, are the efficiency of the PV cell and of the converter. η_{PV} is calculated keeping into account for a year the equivalent number of sunny hours and the load peak duration, since the sizing of PV plants can be done considering the equivalence

$$P_{PV} = E_L/h_S$$

where P_{PV} is the rated power of the PV plant; E_L, the energy required by the loads in the year; h_S is the average equivalent number of sunny hours in the same interval of time. The parameter η_C is considered constant, for simplicity reasons, and taken equal to 85%. The PV units are installed in a distributed fashion at the LV bus-bars. These units are assumed to be grid-connected and devoid of any energy storage system.

(b) *The cable installation cost*

$$C_2 = C_{Ca}(I) \times L \tag{3}$$

where C_{Ca} is the yearly cost of the cable for a given capacity I and L is the length. In this application only underground cables are considered.

(c) *The capacitor bank installation cost*

$$C_3 = C_C \times Q \tag{4}$$

where C_C is the yearly cost of the capacitor per kVAR and Q is the reactive power installed.

(d) *The MV/LV substation transformer installation cost*

$$C_4 = C_{Sk}(A_{nk}) \tag{5}$$

where C_{Sk} is the yearly cost of the required transformer having rated power A_{nk}.

(e) *The operating cost*

Finally, the operating costs are essentially the cost of the losses of the network

$$C_5 = C_P \times P_r \times h_x \tag{6}$$

where C_P is the unit cost of losses and P_r are the peak power losses evaluated when the maximum loading condition has taken place and h_x is the P_r duration in the year.

3. The objective function

The objective function represents the overall cost, which can be obtained as the summation of all the cost terms (Eqs. (1) and (3)–(6)):

$$C_{TOT} = C_1 + C_2 + C_3 + C_4 + C_5 \tag{7}$$

The technical constraints are that the voltage drop should be kept within desired limits and that the current flowing in the branches should be kept below their rated capacity. Another constraint is related to the load supply. Namely, all loads with their new increased values have to be entirely supplied.

A single solution is coded into a string having three times the number of branch elements, so that each tuple of elements is related to one branch and therefore, to its ending bus. Indeed, the nodes are given the same number of the branches upstream.

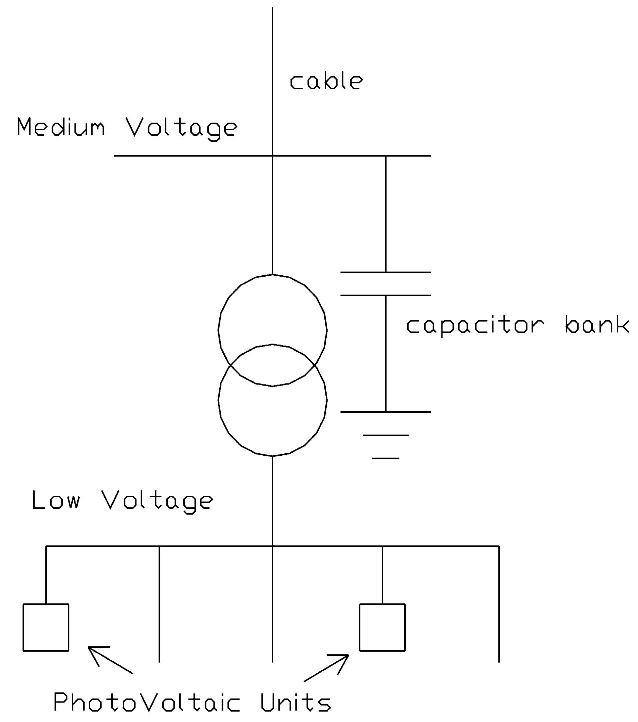

Fig. 1. A generic node of the distribution system.

Each tuple (S_i, P_i, Q_i) composing the solution string takes into account the different means to attain reinforcement, which are not mutually exclusive.

The solution string is therefore, a vector composed as follows

$$[S_1, P_1, Q_1, S_2, P_2, Q_2, \ldots, S_{\mathrm{br}}, P_{\mathrm{br}}, Q_{\mathrm{br}}] \tag{8}$$

where S_i is the section of the reinforcement cable at the ith branch; P_i, the sizing power of the PV plants below the ith node; Q_i is the rated power of the capacitor bank installed at the MV level of the ith node and br is the total number of branches in the network.

Each of these elements can vary in a discrete fashion among a set of possible available sizes.

In particular, there are three possible rated sizes available for cables, 35 for PV units and three for capacitor banks. The optimisation problem here dealt with is then a combinatorial minimisation problem; so that the fitness can be expressed as the inverse of the cost function, Eq. (7).

The overall optimisation problem should also consider, for all of the installations, land availability and cost, permit costs, plants operating and maintenance costs.

Such costs increases depend on local factors such as the national energy saving policy, the town-planning scheme, the installation mode of the underground cables, etc. Nevertheless, their consideration within the problem formulation would not affect the problem complexity, even though the determination of the single cost coefficients is quite hard. Finally, it is not difficult to include all these factors within this mathematical model. However, the final results which depend on the values assigned to these local factors, cannot be generalised to support the validity of one solution as compared to another. For these reasons, such costs increases have not been addressed in this paper.

4. The meta-heuristic algorithms

The three meta-heuristic methods used for the solution of the distribution system reinforcement planning problem, PTS, PSO and EPTS, are described. The problem is a combinatorial optimisation problem with a wide search space and, in the adopted formulation, the number of variables is three times the number of branches of the network and each variable can take on many discrete values.

This makes the problem quite complex, since the search space is quite big.

The proposed EPTS is derived from the parallelised version of the TS.

The TS is intrinsically a feasible search algorithm for this kind of problem. Indeed, it does not need many iterations to obtain better solutions; it is effective for optimising discrete combinatorial problems and it is straightforward and deterministic. On the other hand it does not have the potential to handle efficiently problems with large search spaces.

4.1. Basic Tabu Search approach

The Tabu Search [12,13] algorithm is basically different from the hill-climbing (or descent) algorithm because the latter stops at a local minimum whereas TS includes a mechanism for escaping local minima. Unlike other heuristic methods, like genetic algorithms or simulated annealing, TS does not converge to the best solution at the final iteration. Rather, it provides better solutions during the solution search process.

The Tabu Search algorithm is based on a local search and thus, a suitable neighbourhood should be defined beforehand. Nearby points can then be identified and a set of 'Tabu rules' guiding the move can be derived. These rules are often based on memory structures. The approach followed in the present application takes as Tabu rules a recency-based criterion and some practical rules derived from the problem knowledge. That means that recent moves will not be repeated for a number of times (Tabu tenure) and that some moves will be more easily allowed than others.

Generally the basic tasks of local heuristics involve directing the search towards high quality areas of the search space and avoiding being trapped into local minima.

The TS algorithm tries to achieve these goals by using the Tabu rules to guide the local search process. If a recency-based Tabu rule is adopted, the search proceeds as follows:

At iteration t, the most cost-effective 'Allowed' move in the neighbourhood of the current solution is chosen and if no Tabu rule is violated, the move is made. The move then becomes a Tabu move and it will be forbidden (more or less strictly) at the next steps. At iteration t between the sets of the Allowed and Tabu moves the following relations are verified:

$$A(t) \cup T(t) = M(t)$$

$$A(t) \cap T(t) = \varnothing$$

The starting point is randomly generated and at the first iteration all the moves are allowed.

The introduction of a recency-based Tabu rule has proved to be sensible, as the current point can be a local minimum. Indeed, escaping this situation is possible only if a certain number of moves backwards are forbidden, moving along the neighbourhood's minima for each current solution and not considering, at each iteration, the quality of the current solution itself.

The efficiency of the TS algorithm depends essentially on the type of restriction selected and on the Tabu tenure. As a meta-heuristic method TS is in many cases better than simulated annealing and genetic algorithm for solving combinatorial optimisation problems in terms of computational effort and solution accuracy. However, TS is inclined to deteriorate the performance of the solution search for a large scale problem.

4.2. Parallel Tabu Search

TS is efficient for solving combinatorial optimisation problems. However, as the problem size gets larger, TS has some drawbacks:

1. TS needs to compute the cost function for solution candidates in the neighbourhood around a solution at each iteration. The calculation is very time consuming in large-scale problems. The large size problem often gives large neighbourhood even though the neighbourhood is defined as a set of solution candidate with the Hamming distance equal to 1.
2. The complicated non-linear optimisation problem has many local minima in large scale problems. That implies that one-point search does not give satisfactory solutions due to the huge search space. Complicated optimisation problems require the solution diversity.

In this paper, the decomposition of the neighbourhood is accommodates drawback (1). The neighbourhood is decomposed into several sub-neighbourhoods. A processor may be assigned to each sub-neighbourhood so that the best solution candidate is selected independently in each sub-neighbourhood. After selecting the best solution in each sub-neighbourhood, the best solution is eventually selected from the best solutions in the sub-neighbourhoods. Also, the multiple Tabu lengths is proposed to deal with problem (2). TS itself has only one Tabu length.

Moreover, it is important to find out better solutions from different directions rather than from only one direction for a longer period. Namely it is effective to make the solution search process more diverse.

4.3. The Particle Swarm Optimisation algorithm

PSO [2,3] is an optimisation technique and belongs to evolutionary computation techniques. The method has been developed through a simulation of simplified social models. The features of the method are: (1) the method is based on research on swarms such as fish schooling and bird flocking. (2) PSO is similar to a genetic algorithm in that the system is initialised with a population of random solutions. It is unlike a genetic algorithm, however, in that each potential solution is also assigned a random velocity and the potential solutions, called particles, are the flown into the problem space. PSO is basically developed through simulation of bird flocking in two-dimension space. The position of each individual (agent) is represented by *XY* axis position and also the velocity is expressed by v_X (the velocity of X axis) and v_Y (the velocity of Y axis). Modification of the agent position is realized by the position and velocity information. An optimisation technique based on the above concept can be described as follows: namely, bird flocking optimises a certain objective function. Each agent knows its best value so far (pbest) and its *XY* position. Moreover, each agent knows the best value so far in the group (gbest) among pbests. Each agent tries to modify its position using the following information:

- the current positions (X, Y),
- the current velocities (v_X, v_Y),
- the distance between the current position and pbest,
- the distance between the current position and gbest,

This modification can be represented by the concept of velocity. Velocity of each agent can be modified by the following equation

$$v_i^{k+1} = wv_i^k + c_1\text{rand}(\text{pbest}_i - s_i^k) + c_2\text{rand}(\text{gbest} - s_i^k) \quad (9)$$

where v_i^k is the velocity of agent i at iteration k; w, the weighting function; c_j, the weighting factor; rand, the random number between 0 and 1; s_i^k, the current position of agent i at iteration k; pbest_i, the pbest of agent i; gbest is the gbest of the group.

Using the above equation, a certain velocity which gradually gets close to pbest and gbest can be calculated. The current position (searching point in the solution space) can be modified by the following equation:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (10)$$

The process for implementing the PSO algorithm for a problem space having d dimensions, is as follows:

1. Initialise the population of particles with random positions and velocities on d dimensions in the problem space.
2. For each particle evaluate the desired optimisation fitness function in d variables.
3. Compare particle's fitness evaluation with the particle's pbest. If current value is better than pbest then set pbest value equal to the current value, and the pbest location equal to the current location in the d-dimensional space.
4. Compare fitness evaluation with population's overall previous best. If current value is better than gbest, then reset gbest to the current's particle array index and value.
5. Change the velocity and position of the particle according to Eqs. (9) and (10), respectively.
6. Loop to step 2 until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations.

Particles velocities on each dimension are clamped to a maximum velocity, $V_{\max}$. This parameter determines the resolution with which regions between the present position and the target position (overall best so far) are searched. The acceleration constants c_j in Eq. (9) represent the weighting of the stochastic acceleration terms that pull each particle towards the gbest and pbest positions. Early experience with PSO led us to set the acceleration constants c_1 and c_2 equal to 2.0 for almost all applications. $V_{\max}$ was thus the only parameter routinely adjusted and we often set it about

10–20% of the dynamic range of the variable on each dimension. The population size selected is problem dependent. The maximum velocity $V_{\max}$ serves as a constraint to control the global exploration ability of a particle swarm. A larger $V_{\max}$, facilitates global exploration, while a smaller $V_{\max}$ encourages local exploitation. The inclusion of an inertia weight, w, (Eq. (9)) has provided improved performance in a number of applications. As originally developed, w often decreased linearly from about 0.9 to 0.4 during a run. Suitable selection of the inertia weight provides a balance between global and local exploration and exploitation, and results in a lower number of iteration on average to find sufficiently optimal solutions.

4.3.1. PSO for discrete problems

As for the engineering problem treated here, many other engineering problems are formulated as combinatorial optimisation problems. In such cases, each agent should be modelled in a way that it can easily make one decision or another (boolean decision) depending on 'personal' or 'social' factors contained in the quantity v_i^k. One of the functions that could accomplish this feature is the sigmoid function: $\mathrm{sig}(v_i^{k+1}) = 1/(1+\exp(-v_i^{k+1}))$.

In this way, the agent's position at each iteration $k+1$ is modified according to the velocity, whose value changes to the following expression (9).

Therefore, it turns that: if $\rho_i^{k+1} < \mathrm{sig}(v_i^{k+1})$ then $s_i^{k+1} = 1$ else $s_i^{k+1} = 0$, where ρ_i^{k+1} is a positive random number between 0 and 1.

4.4. The Evolutionary Parallel Tabu Search

The PTS efficiency is improved by including features deriving from the evolutionary and co-operative algorithms. Each solution of a population is provided with a different sub-neighbourhood, this being treated as a part of the solution string. The evolution indeed involves the solutions as well as their sub-neighbourhoods. After a few iterations in the evolutionary loop, these represent indeed the most interesting search directions. In particular, selection is applied to solutions and sub-neighbourhoods, crossover and mutation are applied to sub-neighbourhoods.

Each solution stores historical information about its own evolution, namely the best solution found during its own local search. In what follows, the local best solution will be called pbest. The algorithm works as follows.

1. A sufficiently large and diversified population of random parents solutions is created and for each of them a random sub-neighbourhood is chosen. Each solution is evaluated using the cost function in Eq. (7).
2. The pbest of each solution is updated.
3. A Tabu Search algorithm is run for each of the members of the parents population into the relevant sub-neighbourhood. This implementation of TS has a decreasing Tabu tenure and a stopping criterion which is the flattening on one best solution. At each cycle, namely when the parents population is again created, the information about the relevant Tabu lists is lost, since it is strictly related to the couple (solution, sub-neighbourhood). Indeed this link may be broken after the recombination operators.
4. A Roulette wheel selection mechanism [12] is applied to select the elements of the parent population that will be subjected to recombination. This operator selects more frequently those solutions having a higher quality as compared to that of the other solutions of the current population.
5. The recombination operators, crossover and mutation, are applied to solutions sub-neighbourhoods. The crossover operator exchanges the sub-neighbourhoods of two solutions, the mutation operator performs a random variation of the sub-neighbourhoods.
6. The offspring population is filled in using the outputs of the recombination operators. Then the parents population is created using the pbests of the offspring population and the process goes on to step 2 until a maximum number of cycles is reached.

In Fig. 2 the pseudo-code of the EPTS is shown. The basic feature of the algorithm is the evolution of 'promising' areas of the search space (sub-neighbourhoods) on the basis of the overall cost function. Fig. 3 shows the crossover and mutation operators for this algorithm. This algorithm introduces new crossover and mutation operators. In genetic algorithms these operators are applied to solution strings. In particular, the crossover operator takes two parent solution-strings and generates two offspring strings made of the combination of the parents chromosomes. The mutation operator instead randomly changes one chromosome. In the evolutionary algorithms the crossover and mutation parameters (i.e. mutation step size) evolve during the search, though influencing the search direction and convergence.

The proposed EPTS algorithm performs crossover and mutations on the solutions sub-neighbourhoods. The algorithm borrows concepts both from other meta-heuristic search strategies and from the evolutionary computation methods. Storing information about the search history of each solution (pbest) is typical of the PSO algorithm, whereas the evolution of features of different solutions by means of reproduction operators comes from the genetic algorithms literature. Finally the idea to deterministically consider sub-neighbourhoods is borrowed from PTS. The idea introduced by this method is in combining and evolving sub-neighbourhood information rather than solution sub-strings.

5. Application

The application is devoted to the identification of the minimum cost for the reinforcement planning of a 35 buses

```
Procedure EPTS
Begin
 Generate a population of different starting solutions each having a randomly chosen sub-neighbourhood.
 Initialize the tabu memories.
 Initialise the pbests of the starting population
 igen=0 (igen: cycles counter)
 Repeat
  Parent:=pbest
  For j:=1 To dimpop Do
    If flip(pcross) { flip(pcross) is an operator that implements the
                     tossing of a coin with probability to get true, pcross.}
      Select two individuals
      Cross their neighbourhoods
  For j:=1 To dimpop Do
    Select the current solution
      If flip(pmut)
        Mutate its sub-neighbourhood
   Apply Tabu Search
   Update pbest
   Update the overall best
   Inc(igen)
Until igen=ngen
```

Fig. 2. Pseudo-code of the EPTS.

radial electrical distribution system. The network is depicted in Fig. 4, which also shows the obtained design solutions using the proposed EPTS approach for a 40% load increase.

The load flow solution of the distribution system has been carried out using the algorithm set up in Ref. [14]. The reinforcement is here carried out by means of the installation of: PV units, considered as active power injections; capacitor banks, considered as reactive power injections; new cables; substation transformers.

In this application, the authors have used a value of C_{PV} of 2065.82€/kW as suggested in Ref. [15]. The three values of $C_{Ca}(I)$ range from 13.95€ (for $I = 109$ A) to 18.60€ (for $I = 305$ A). In this study, C_c is 36.13€/kVAR. $C_{Sk}(A_{nk})$ ranges from 8007.40€ (for $A_n = 400$ kV A) to 8279.45€ (for $A_n = 630$ kV A). C_P is taken to be 0.05€/kW h. The 35 values of P_{PV} range from 2 to 400 kW.

The proposed EPTS algorithm has been compared to the standard PTS and PSO algorithms and results show the particular efficiency of the algorithm in terms of exploration and diversification of the search.

Fig. 5 shows the convergence of EPTS as compared to traditional PTS and PSO for a load increase of 40%. As it can be noted from the same figure, the proposed algorithm works well from the beginning and improves upon a lot the best solution until the end. That means that diversity of solutions along the iterations remains and that the used recombination operator works efficiently. Of course the computational effort is larger than that employed in the other two cases, especially as compared to the PSO. The Authors believe that the PTS algorithm has a poor performance because the starting population is made up of

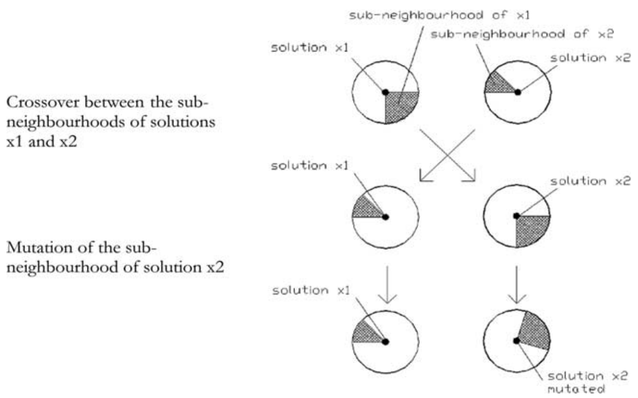

Fig. 3. Sub-neighbourhood crossover and mutation operators applied to the best solution found so far, pbest, during the TS of each solution.

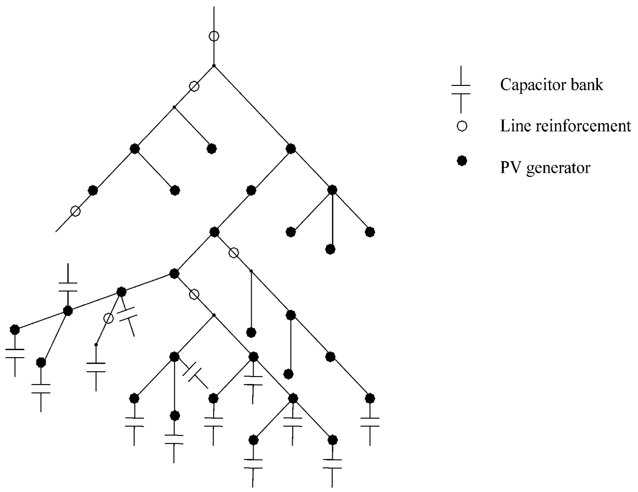

Fig. 4. Design solution obtained using the proposed EPTS strategy for a 40% load increase.

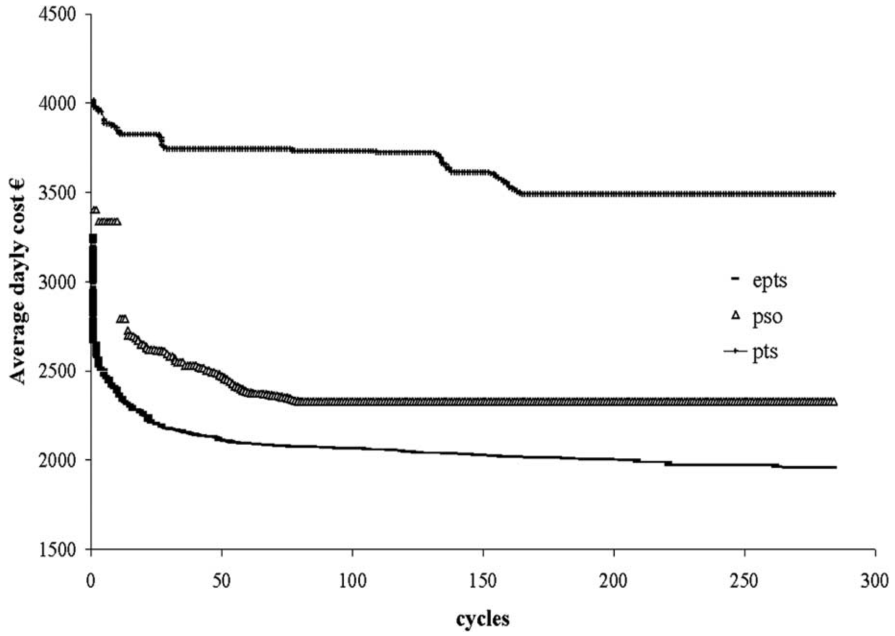

Fig. 5. Convergence of the EPTS as compared to PTS and PSO algorithms for a 40% load increase. In the Y axis the average daily cost in euro is reported, in the X axis the number of cycles.

copies of the same solution, each having simply a different sub-neighbourhood and different Tabu tenures. Diversity can be therefore, attained only by means of a different search direction identified by the relevant Tabu list. A considerable number of runs of the EPTS algorithm have proved its robustness in attaining almost the same solution or design solutions that are nearby. Fig. 5 also shows that applying an efficient algorithm would produce considerable costs savings.

Fig. 6(a)–(c) shows one EPTS run with a population size of 150. The white pixels on a grey background represent the activated search directions for each population member. On the abscissa the search directions are represented (each individual solution is a vector with 105 integer elements). On the Y axis the population members are reported. Fig. 6(a) shows the first cycle, in which there is a random assignation of sub-neighbourhoods to the population members, Fig. 6(b) shows that, at the 100th cycle, the algorithm actually succeeds in directing the search towards certain 'promising' directions, indeed, it seems that almost all the population members search in the directions 94–96. Fig. 6(c) shows that, at the 200th cycle, namely at the end of the run, for all the population members the Tabu Search investigation is performed in the direction of 38–40.

The parameters for each algorithm are the following.

PTS runs 400 cycles each including a Tabu Search running 500 iterations maximum. The population size is of 150 elements and the Tabu tenure is variable along the search.

The Tabu Search for each solution is run with a stopping condition that is the flattening for a given number of iterations of solutions on one best solution found with no improvement.

The PSO has 150 agents for 500 cycles, the parameters c_1 and c_2 are set, respectively, to 1.5 and 2.0, whereas the inertia weight w is set to 0.9 and linearly decreases to 0.4. $V_{\max}$ is set to 2.0.

EPTS has a population size of 150, for 400 cycles each of which runs 500 iterations of TS maximum. The crossover probability is 0.7 and the mutation probability is 0.1. The selection operator is the Roulette wheel selection. The Tabu Search for each solution is run with a stopping condition that is the flattening for a given number of iterations of solutions on one best solution found with no improvement. The stopping criterion for the three algorithms is the flattening on the best solution found for 100 cycles. The crossover and mutation operators borrowed by the EC literature are applied not to solution strings, but to sub-neighbourhood. This pushes the search potential of the algorithm as compared to traditional PTS. In this way evolution involves the promising search directions as well as solutions themselves. The parameters used in the applications have been both heuristically determined and taken from the literature.

5.1. Mathematical function optimisation problems

This section presents selected results from the application of the EPTS algorithm to three well known function optimisation problems, namely the Rosenbrok function, the Gaussian Quartic and the Sphere function.

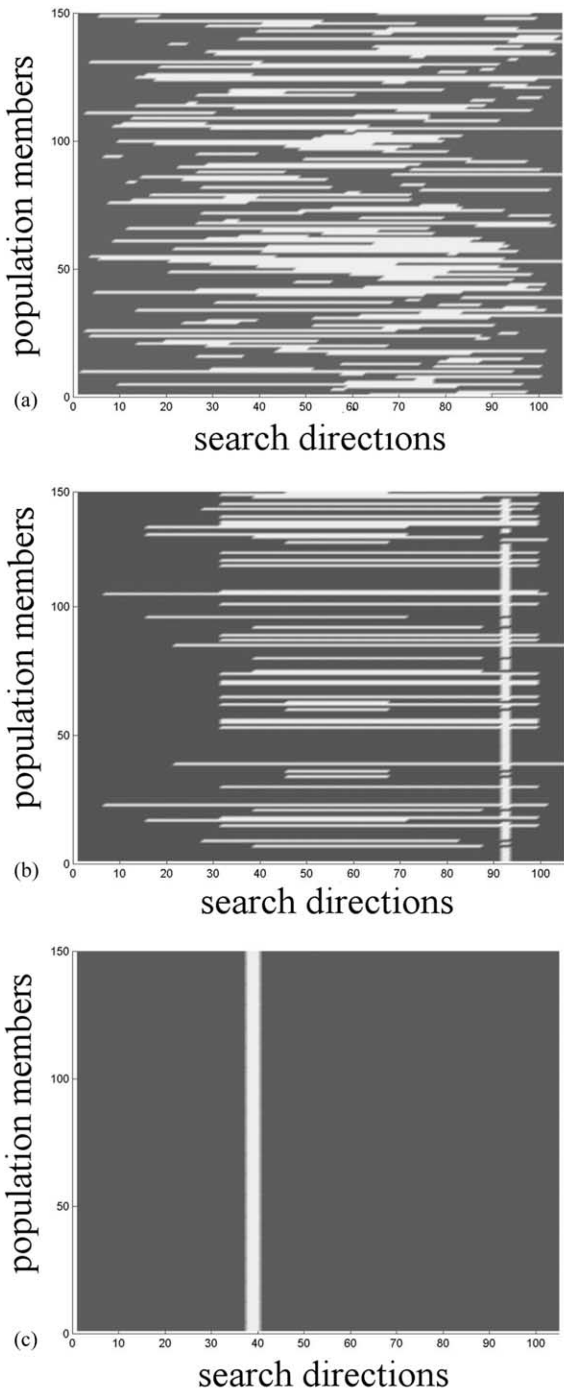

Fig. 6. Neighbourhoods evolutions during one run of the EPTS algorithm. In the Y axis the population members are reported, in the X axis the search directions are reported. (a) first cycle; (b) 100th cycle; (c) 200th cycle.

Problem 1. Find the minimum of the Rosenbrok function with two or four variables ranging from -2.048 to 2.048

$$\mathrm{Ros}(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n-1} (100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

This is rather a complex optimisation problem because the function has a very narrow and sharp ridge and runs around a parabola, so the variables are interrelated (see Fig. 7).

Problem 2. Find the minimum of the Gaussian Quartic function with 30 variables ranging from -1.28 to 1.28

$$\mathrm{gauss}(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} i x_i^4 + \mathrm{gauss}(0, 1)$$

This is a simple unimodal function with padded noise. The Gaussian noise ensures that the algorithm never gets the same value at the same point. Algorithms that do not well on this test function will perform poorly on noisy data.

Problem 3. Find the minimum of the Sphere function with 3 and 30 variables ranging from -5.12 to 5.12

$$\mathrm{sphere}(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i^2$$

Results in terms of number of functions evaluations for the problems outlined above are reported in Table 1. From Table 1, it can be noted that the results found for the Rosenbrok function model with four variables are better than those found by Schlierkamp-Voosen and Muhlenbein [16], since their breeder GA required about 240 000 evaluations to achieve a result with a precision of 0.1. Again good results have been found for the sphere model and $n = 3$, for which the number of evaluation seems low for the attained accuracy as compared to other results found in literature [17]. The algorithm has a poor performance on the Gaussian Quartic.

As it can be noted, in many of the above reported cases, EPTS is still efficient, even though the variables have been coded into binary strings and therefore, the computational performances have been affected. The concept of neighbourhood is indeed strongly different than in the case of the reinforcement planning problem.

On the other hand, for many engineering problems and especially in design problems the ‘computational time’ dimension is irrelevant as compared to the quality of the obtained results. This condition is related to the ability of the algorithm not to get stuck into local minima and to improve the results till the very end. This is evident from Fig. 8 where a run on the Rosenbrok function with four variables has been reported. Other runs of the three considered algorithms have been carried out with the same test function; the stopping condition was that of the flattening on one solution for 100 iterations. It is important to notice that EPTS was improving till the 4000th iteration attaining a much higher quality than the other two which got stuck at the 192nd iteration (PTS) and at the 105th iteration (PSO). This is maybe the most interesting property of the algorithm for engineering design problems.

For all the tests carried out the variables have been coded into binary strings of 10–15 bits. All the results have been averaged on a sample of 10 runs.

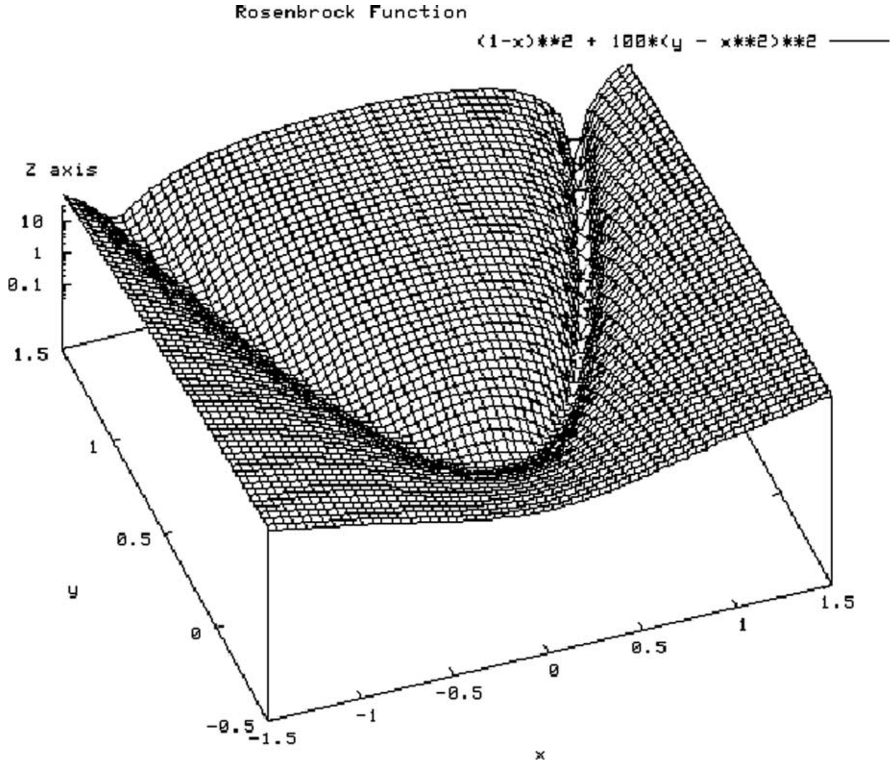

Fig. 7. An inverted graph of a two-dimensional projection of the Rosenbrok function.

Table 1
Performance of EPTS with some function optimisation problems

Test function	Rosenbrok $n = 2$	Rosenbrok $n = 4$		Sphere $n = 3$	Sphere $n = 30$		Gauss $n = 30$
Number of function evaluations	1383	1 000 000	17 000	754	23 000	57 000	35 000
Error	0.00	0.00	0.0	0.00	0.0	0.00	0.00

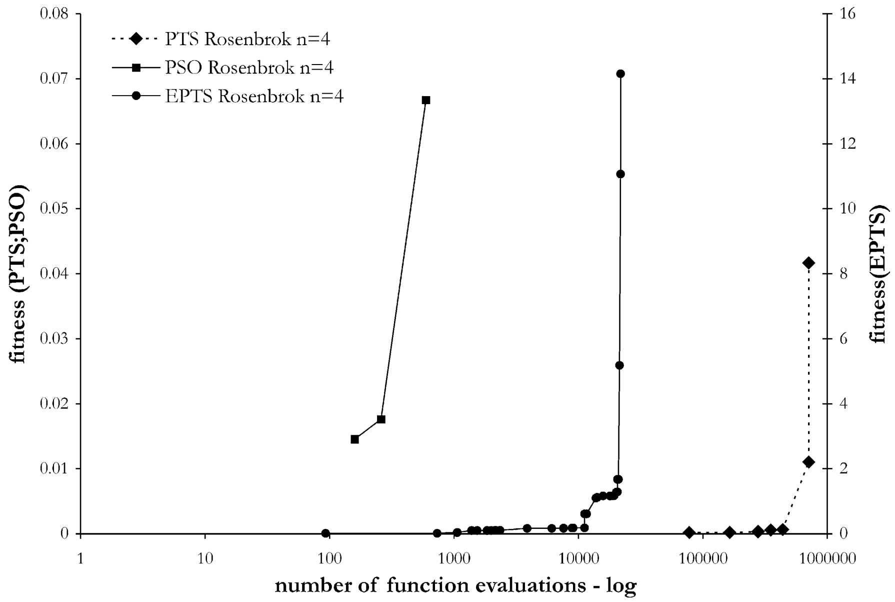

Fig. 8. Convergence of the EPTS as compared to PTS and PSO algorithms for the Rosenbrok Saddle with $n = 4$; the stopping condition is attaining a solution with a precision 0.1.

6. Conclusions

Three heuristic search algorithms have been compared on a difficult test problem of radial distribution systems reinforcement planning. The problem formulation takes into account the possibility to introduce some DG units at the load nodes, using them to replace feeders and substations reinforcement partially. The total cost is made up of the installation cost of feeders and substations as well as of DG units and a of distributed compensation system in order to back up the reactive power production. The cost of losses is also part of the total cost and it can be reduced by a correct sizing and placement of DG units and capacitor banks. The issue is a combinatorial optimisation problem and it can be therefore, efficiently solved by means of a heuristic or meta-heuristic technique. The authors have set up a modified version of the PTS which works quite efficiently when compared to standard PTS and to PSO. It borrows concepts from both algorithms and merges them in an efficient and robust implementation. The convergence properties of the proposed EPTS have been tested on a medium size radial system successfully. Other properties, such as the ability to accurately explore the search space, improving the results till the very end, have been evidenced testing the algorithm on some mathematical test functions.

References

[1] Mori H, Hayashi T. New Parallel Tabu Search for voltage and reactive power control in power systems. Proc IEEE Int Symp Circuits Syst, Monterey, CA, USA 1998;3:431–4.

[2] Eberhart RC, Shi Y. Particle Swarm Optimization: developments, applications and resources. Proc IEEE Int Conf Evol Comput 2001; (I):81–6.

[3] Kennedy J, Eberhart R. Particle Swarm Optimization. Proc IEEE Int Conf Neural Networks (ICNN), Perth, Australia 1995;IV: 1942–8.

[4] Dugan RC, McDermott TE, Ball GJ. Planning for distributed generation. IEEE Ind Appl Mag 2001;80–8.

[5] Sweet W. Networking assets. IEEE Spectrum 2001;84–8.

[6] Philipson L. Distributed and dispersed generation: addressing the spectrum of consumer needs. Power Engng Soc Summer Meet 2000; 3:1663–5.

[7] Rau NS, Wan Y-h. Optimum location of resources in distributed planning. IEEE Trans Power Syst 1994;9(4):2014–20.

[8] Carpinelli G, Celli G, Pilo F, Russo A. Distributed generation siting and sizing under uncertainty. Proc IEEE Porto Power Tech Porto, Portugal 2001;4.

[9] Asakura T, Yura T, Hayashi N, Fukuyama Y. Long-term distribution network reinforcement planning considering multiple construction plans. Proc Power Syst Technol, PowerCon 2000;2: 1101–6.

[10] Mori H, Matzuzaki O. A Parallel Tabu Search approach to Unit Commitment problem. Proc IEEE Int Conf Syst Man Cybernet, Tokyo, Japan 1999;6:521–6.

[11] Matsamura T, Nakamura M, Tamaki S, Onaga K. A parallel tabu search and its hybridization with genetic algorithms. Proc Int Symp Parallel Architect, Algorithms Networks, 2000 I-SPAN 2000; 18–22.

[12] Reeves CR, Soderberg B, Peterson C, Laguna M, Glover F, Dowsland K, Beasley J. Modern heuristic techniques for combinatorial problems. New York: McGraw-Hill; 1995.

[13] Reeves CR. Modern heuristic techniques for combinatorial problems. Oxford: Blackwell Scientific Publications; 1993.

[14] Augugliaro A, Dusonchet L, Ippolito MG, Riva Sanseverino E. An efficient iterative method for load-flow solution in radial distribution networks. Proc IEEE Porto Power Tech Porto, Portugal 2001;3.

[15] Italian Ministry for the environment, Ministerial Decree, March 16th 2001—Programma Tetti fotovoltaici. Gazzetta Ufficiale Italiana n. 74 il 29 marzo 2001.

[16] Schlierkamp-Voosen D, Muhlenbein H. Strategy adaptation by competing subpopulations. In: Davidor Y, Schwefel H-P, Manner R, editors. Parallel problem solving from nature, Third Workshop, PPSN III, Jerusalem, Israel. Berlin: Springer; 1994.

[17] Eshelman LJ, Schaffer JD. Preventing premature convergence in genetic algorithms by preventing incest. In: Belew R, Booker L, editors. Proceedings of the Fourth International Conference on Genetic Algorithms and their Applications. Los Altos, CA: Morgan Kaufmann; 1991.

Antonino Augugliaro (1949) received a Doctoral degree in Electrical Engineering from the University of Palermo, Italy, in 1975. From 1978 to 1994 he has been Associate Professor and now he is Full Professor of Electrical Power Generation Plants at the Faculty of Engineering of the University of Palermo. His main research interests are in the following fields: simulation of electrical power system; transmission over long distances; mixed three-phase/six-phase power system analysis; optimisation methods in electrical distribution system's design and operation; distribution automation.

Luigi Dusonchet (1948) received a Doctoral degree in Electrical Engineering from the University of Palermo, Italy, in 1975. From 1978 to 1990 he has been Associate Professor and now he is Full Professor of Industrial Electrical Systems at the Faculty of Engineering of the University of Palermo. His main research interests are in the following fields: simulation of electrical power system; transmission over long distances; mixed three-phase/six-phase power system analysis; optimisation methods in electrical distribution system's design and operation; distribution automation.

Eleonora Riva Sanseverino (1971) received a Doctoral degree in Electrical Engineering from the University of Palermo, Italy, in 1995. Since 1995 she has been working in the Research Group of Electrical Power Systems. She successfully finished her Phd in Electrical Engineering at the same University in February 2000. Then, she had a temporary researcher position at the University of Palermo in the Department of Electrical Engineering till November 2001. From December 2001 to October 2002 she has been a researcher in a Computer Science department of the Italian National Council of Research. Now she is Associate Professor of Power Systems at the Faculty of Engineering of the University of Palermo. Her main research interest is in the field of optimisation methods for electrical distribution systems design, operation and planning.