Abstract

This paper presents a new algorithm based on integrating the use of genetic algorithms and simulated annealing methods to optimal allocation of distributed generation resources in distribution networks. Through this algorithm a significant improvement in the optimization goal is achieved. With a numerical example the superiority of the proposed algorithm is demonstrated in comparison with the simple genetic algorithm.

Keywords: distributed generation, distribution networks, genetic algorithm, simulated annealing.

1. Introduction

The utilities requirements regarding flexible electric systems, energy saving, loss reduction and environmental impact are providing impetus to the development of Distributed Generation (DG). DG includes the application of small generators, scattered throughout a power system, to provide the electricity service required by customers. The DG allocation can be obtained by a complete enumeration of all feasible combinations of sites and sizes of DGs in the network. Since the number of alternatives could be very large, the load flow analysis should be performed for every feasible network combination. Of course the optimized solution among these alternatives should be selected [1-3].

The artificial intelligence techniques are the most widely employed tool for solving most of the optimization problems. These methods (e.g. genetic algorithm, simulated annealing and tabu search) seem to be promising and are still evolving. The published are on the DG allocation by application of genetic algorithm (GA)[4,5]. Tabu search (TS) algorithm is used for DG allocation in distribution systems[6]. A new hybrid algorithm for evaluation of the DG site and size in MV networks is proposed. The GA and SA are employed for DG allocation. The results showed that the proposed combined GA-SA method is better than the simple GA in terms of solution quality and number of iteration.

2. Distributed Generation

DG can be defined as a small-scale generating unit located close to the load being served. A wide variety of DG technologies and types exists: renewable energy source such as wind turbines, photovoltaic, micro-turbines, fuel cells, and storage energy devices such as batteries. Both the distribution company and/or the customer can, in principle, invest in and operate units. Due to the availability of such a flexible option of DG as an energy source at the distribution voltage level, the distribution network is now being transformed from a passive network to an active one. The DG as an energy source in the distribution network will play a significant role in operation, structure, design and upgradation issues. DG technologies, their benefits and concepts, and their valuable effect on the electricity market make it a credible alternative in the distribution system planning. The importance of DG is now being increasingly accepted and realize by power engineers. From distribution system planning point of view, DG is a feasible alternative for new capacity especially in the competitive electricity market environment and has immense benefit such as [2-3]:
- Short lead-time and low investment risk since it is built in modules.
- Small-capacity modules that can track load variation more closely.
- Small physical size that can be installed at load centers and does not need government approval or search for utility territory and land availability.
- Existence of a vast range of DG technologies

For these reasons, the first signs of a possible technological change are beginning to arise on the international scene, which could involve in the future the presence of a consistently generation produced with small and medium size plants directly connected to the distribution network (LV and MV) and characterized by good efficiencies and low emissions. This will create new problems and probably the need of new tools and managing these systems.

3. Problem Formulation

The problem is to determine allocation and size of DGs which minimizes the distribution power losses for a fixed number of DGs and an specific total capacity of DGs. Therefore the following assumptions are employed in this formulation [6, 7]:
- The maximum number of installable DGs is given \(D\).
- The total installation capacity of DGs is given \(Q\).
- The possible locations for DG installation are given for each feeder.
- The upper and lower limits of node voltages are given.
- The current capacities of conductors are given.
The objective function in this optimization problem is:

\[ f = \sum_{i=1}^{n} P_i \quad \text{(1)} \]

Where \( P_i \) is the nodal injected power at bus \( i \), and \( n \) is the total number of buses.

If the total injected power of distributed generation was constant as \( C \, \text{MW} \), this equality constraint should be expressed in form of a penalty function as shown [7]:

\[ f = \sum_{i=1}^{n} P_i + \alpha \left( \sum_{k=1}^{L} P_k - C \right) \quad \text{(2)} \]

Constraints:

- Maximum number of DGs:
  \[ \sum_{i=1}^{M} \sum_{g=1}^{N} n_{gi} \leq D \quad \text{(3)} \]

- Total capacity of DGs:
  \[ \sum_{i=1}^{M} \sum_{g=1}^{N} G_g n_{gi} \leq Q \quad \text{(4)} \]

- One DG per installation position:
  \[ \sum_{g=1}^{N} n_{gi} \leq 1 \quad \text{(5)} \]

- Upper and lower voltage limits:
  \[ V_i \leq V_n \pm \Delta V \quad \text{(6)} \]

- Current capacity limits:
  \[ I_i \leq I_{i,\text{max}} \quad \text{(7)} \]

\( k = 1, 2, ..., L, l = 1, 2, ..., M, g = 1, 2, ..., N \)

Where,

- \( P_i \): nodal injection of power at bus \( i \),
- \( P_k \): load power of bus \( k \),
- \( V_i \): magnitude of voltage of bus \( i \),
- \( V_n \): nominal magnitude of voltage in the network,
- \( G_g \): capacity of \( g^{th} \) DG,
- \( n_{gi} \): 0-1 variable for determining whether one DG with \( g^{th} \) capacity is allocated at \( l^{th} \) location (1: allocated, 0: not allocated),
- \( L \): total number of load buses,
- \( M \): total number of DG location candidates,
- \( N \): total number of capacity types of DGs,
- \( Q \): total installation capacity of DGs,
- \( D \): maximum number of installable DGs,
- \( \alpha \): penalty weight of equality constraint,
- \( C \): total injected dispersed generation for network,
- \( \Delta V \): maximum permissible voltage deviation,
- \( I_i \): current of section \( i \),
- \( I_{i,\text{max}} \): maximum current capacity of section \( i \).

4. Genetic Algorithm

Genetic Algorithm is a general-purpose search techniques based on principles inspired from the genetic and evolution mechanisms observed in natural systems and populations of living beings. Their basic principle is the maintenance of a population of solutions to a problem (genotypes) as encoded information individuals that evolve in time [8].

Generally, GA comprises three different phases of search: phase 1: creating an initial population; phase 2: evaluating a fitness function; phase 3: producing a new population.

A genetic search starts with a randomly generated initial population within which each individual is evaluated by means of a fitness function. Individual in this and subsequent generations are duplicated or eliminated according to their fitness values. Further generations are created by applying GA operators. This eventually leads to a generation of high performing individuals [9].

There are usually three operators in a typical genetic algorithm [8]: the first is the production operator (elitism) which makes one or more copies of any individual that posses a high fitness value; otherwise, the individual is eliminated from the solution pool; the second operator is the recombination (also known as the 'crossover' ) operator. This operator selects two individuals within the generation and a crossover site and carries out a swapping operation of the string bits to the right hand side of the crossover site of both individuals. Crossover operations synthesize bits of knowledge gained from both parents exhibiting better than average performance. Thus, the probability of a better offspring is greatly enhanced; the third operator is the 'mutation' operator. This operator acts as a background operator and is used to explore some of the invested points in the search space by randomly flipping a 'bit' in a population of strings. Since frequent application of this operator would lead to a completely random search, a very low probability is usually assigned to its activation.

5. Simulated Annealing

Simulated Annealing (SA) is a metaheuristic which has been successfully applied to solve a variety of difficult optimization problems [10]. It is based on an analogy between statistical mechanics systems and combinatorial optimization problems. The term annealing refers to the process of cooling after heating, in order to toughen and temper the material. SA is a Monte Carlo approach that simulates this process. A solution to a combinatorial optimization problem is seen as a possible state of a thermo dynamical system which is submitted to slow down the cooling, starting from an initial high temperature.
In SA the cost of a solution is equivalent to the energy of the physical state, and the temperature, although it has no physical meaning, can be seen as controlling the entropy of the system. At high temperature, all solutions to the optimization problem are equally likely while, at low temperature only the minimal cost solutions are accepted. Following an appropriate cooling schedule, SA has the potential to avoid local minima and converges to the global optimal solutions within a reasonable computing time [10].

In the implementation of SA, starting from an initial configuration, new ones are proposed through local changes, called moves, and accepted within the probability

$$P = \begin{cases} 
1 & \text{if } f^* \leq f \\
\exp \left( \frac{f - f^*}{T} \right) & \text{if } f^* > f
\end{cases} \quad (8)$$

where $f$ and $f^*$ are the costs of the current and proposed configurations, respectively, and $T$ is the temperature. The algorithm is run until a stopping condition is reached, typically a minimum temperature value, specified as part of the annealing schedule.

### 6. The Proposed Algorithm

The major steps of the proposed algorithm are:

- **Step1**: Initialize the variables of GA and SA
- **Step2**: Creating an initial population by randomly generating a set of feasible solutions (chromosomes);
- **Step3**: Evaluating each chromosome by running the load flow program;
- **Step4**: Determining the fitness function of each chromosome in the population (1/Power Losses);
- **Step5**: Apply the crossover operator, the child chromosomes are accepted according to eq. 8, where $f$ and $f^*$ are the fitness values of parent and child chromosomes;
- **Step6**: Apply the mutation operator to the new population;
- **Step7**: Let the current population be the new population;
- **Step8**: If the convergence criterion is satisfied, stop. Otherwise go to step 3.

Figure 1 shows flowchart of the proposed algorithm.

The solution of the problem is represented by a binary coding. As in the example, 4 types of DGs are used for each MV candidate bus, 3 bits are considered for coding: one for presenting the DG on bus and 2 bits for type of DG.

The fitness value of each chromosome is the reverse of power loss for the related network. So the proper load flow program [11] is prepared and it is run for each network. Output of the program is the fitness value of chromosome (1/Power Loss).

### 7. Numerical Example

In order to test the proposed algorithm, the 34-node IEEE distribution test feeder has been considered [12]. A number of tests on the performance of the proposed algorithm have been carried out on the example to determine the most suitable GA and SA parameters setting. Table 1 shows the control parameters which have been chosen after running a number of simulations.
Table 1. GA& SA setting parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>50</td>
</tr>
<tr>
<td>Crossover Probability</td>
<td>0.82</td>
</tr>
<tr>
<td>Mutation Probability</td>
<td>0.04</td>
</tr>
<tr>
<td>Initial Temperature</td>
<td>10000</td>
</tr>
</tbody>
</table>

we assumed DG can be installed on all of the nodes in the example network. Furthermore, maximum total capacity \(Q\) and maximum number \(D\) of DGs assumed about 0.1 power demand and one-third of nodes number for each network. Therefore, \(Q, D\) are considered 200 kW and, respectively.

Different experiments with different random number seeds were carried out to investigate the performance of the proposed algorithm. It was found that the proposed algorithm performs in a better form, in terms of both solution quality and number of iterations, than the individuals SGA. Table 2 shows the comparison of the results for GA-SA and SGA for the test network. According to [12] the test network has 273 kW power losses without DG.

Figure 2 shows the convergence process of the GA-SA and SGA when employed to solve the optimization problem of this test network.

8. Conclusion

In this paper the results of application of GA-SA algorithm to the optimal allocation of DGs in distribution network is presented. The effectiveness of the proposed algorithm to solve the DG allocation problem is demonstrated through a numerical example. The IEEE 34-node distribution test feeders have been solved with the proposed algorithm and, the simple genetic algorithm. The results demonstrated the better characteristics of the GA-SA algorithm in comparison with the SGA specially in terms of; solution quality and number of iterations.

Table 2. Comparison of SGA &GA-SA results

<table>
<thead>
<tr>
<th>Result</th>
<th>SGA</th>
<th>GA-SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Capacity of Installed DGs (kW)</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Power Losses (kW) with DG</td>
<td>213.2</td>
<td>201.8</td>
</tr>
<tr>
<td>Percent of Reduced Power Losses</td>
<td>21.9</td>
<td>26.1</td>
</tr>
</tbody>
</table>

References