Author(s): Dahal, K.P., Burt, G.M., McDonald, J.R. and Galloway, S.J.

Conference Paper: GA/SA-based hybrid techniques for the scheduling of generator maintenance in power systems

Conference: Congress on Evolutionary Computation (CEC). La Jolla, CA, USA. 16-19 July 2000.

Publication year: 2000

Publication title: Proceedings of the Congress on Evolutionary Computation

ISBN: 0-7803-6375-2

Publisher: IEEE

Original online publication is available at:
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=870347&isnumber=18852

GA/SA-based hybrid techniques

for the scheduling of generator maintenance in power systems

K.P. Dahal **G.M. Burt** **J.R. McDonald** **S.J. Galloway**

k.p.dahal@strath.ac.uk g.burt@eee.strath.ac.uk j.mcdonald@eee.strath.ac.uk stuart.galloway@strath.ac.uk

Rolls-Royce University Technology Centre,
Department of Electrical and Electronic Engineering,
University of Strathclyde, 204 George Street, Glasgow G1 1XW, UK.

Abstract- This paper proposes the application of a genetic algorithm and simulated annealing based hybrid approach for the scheduling of generator maintenance in power systems using an integer representation. The adapted approach uses the probabilistic acceptance criterion of simulated annealing within the genetic algorithm framework. A case study is formulated in this paper as an integer programming problem using a reliability based objective function and typical problem constraints. The implementation and performance of the proposed solution technique are discussed. Results contained in this paper will demonstrate that the technique is more effective than approaches based solely on genetic algorithms or solely on simulated annealing. It therefore proves to be a valid approach for the solution of generator maintenance scheduling problems.

1 Introduction

1.1 Generator maintenance scheduling

It is vital for a utility to determine when its generators should be taken off-line for preventive maintenance. This is primarily because other short-term and long-term planning activities such as unit commitment, generation dispatch, import/export of power and generation expansion planning are directly affected by such decisions. In modern power systems the demand for electricity has greatly increased with related expansions in power system size, which has resulted in higher numbers of generators and lower reserve margins, making the generator maintenance scheduling (GMS) problem more complicated. The goal of GMS is to allocate a maintenance timetable for generators in order to maintain a high system reliability, reduce total operating costs, and extend generator life time, whilst still satisfying constraints on the individual generators and the power system itself.

Previous studies of GMS have considered the objectives of maximizing system reliability [1-5] and minimizing economic cost [2,3,6-9]. The most common reliability criterion is the leveling of the reserve generation, which is the difference between the total capacity of the units not undergoing maintenance and the demand over the planning period. The most common economic objective is to minimize the total operating costs, which includes the costs of energy production and maintenance. However, this is an insensitive objective and as such it requires many approximations [2,3].

The GMS problem has a series of constraints related to the generating units and the power system. Maintenance window constraints define the possible times and duration of maintenance for each unit. The relative timetabling of maintenance of certain units may be restricted. The available power must exceed the load, and the manpower and resources available for maintenance work are limited. Further constraints may be posed involving the reliability, transmission capacity and maintenance in local areas of the power system. In general GMS is a multi-criterion constrained combinatorial optimization problem, with nonlinear objective and constraint functions.

1.2 Solution techniques for GMS problems

Conventional solution methods for GMS problems are generally based on heuristic techniques or mathematical methods including integer programming, branch-and-bound techniques and dynamic programming [1-3,6]. The heuristic approach uses a trial-and-error method to evaluate the maintenance objective function, usually by considering each unit separately. This requires significant operator input and in some situations it fails to produce even feasible solutions [1,2]. In contrast, the above mathematical approaches are severely limited by the 'curse of dimensionality' and are poor in handling the nonlinear objective and constraint functions that characterize the GMS problem.

In an attempt to overcome some of the above limitations, genetic algorithms (GAs) and simulated annealing (SA) have been implemented for solving complex scheduling problems [10]. These techniques are completely distinct from classical mathematical programming and trial-and-error heuristic methods. The GA method replicates the principles of the population genetics, i.e. selection and inheritance. GAs are based on natural genetic and evolutionary mechanisms that operate on populations of solutions. The SA method is based on the analogy between the physical annealing process of a solid and the problem of finding the minimum or maximum of a given function depending on many parameters, as encountered in combinatorial optimization problems [10].

The GA and SA approaches have been applied to solve a range of optimization problems in electrical power systems with encouraging results [11,12]. GAs have recently been applied to GMS in [9]. Generator maintenance scheduling problems were considered by using a SA method in [9,13]. In all these applications, the problems were formulated using the economic objective and typical problem constraints. These papers used a binary string representation to encode a trial solution and penalty functions were used in the formulation of the evaluation function to take account of violations of problem constraints. The results reported for a number of test problems were promising.

In the authors' previous works [4,5], the application of GAs for GMS was demonstrated. In [5], a SA approach was also employed to solve the GMS problem. These applications investigated different architectures of the GA and SA methods using an integer representation to encode candidate solutions, to a GMS test problem.

1.3 Hybrid solution techniques

It has been demostrated that the performance of a GA approach can be improved by combining it with other techniques [10]. A GA approach with an initial population seeded by heuristics was applied to solve GMS problems in [5].

This paper proposes the application of a hybrid approach that combines a GA with a feature of SA. This hybrid algorithm employs the probabilistic acceptance criterion of SA for selecting new solutions, and utilizes it within the GA framework. This permits some control over the acceptance of newly created solution.

Recently the GA/SA hybrid approach has been employed to solve many optimization problems arising in power systems [7,8,11,12] and has been demostrated to improve performance over the simple SA and GA methods.

A generator maintenance scheduling problem is considered using a binary GA combined with SA in [7,8]. The use of the acceptance probability of the SA method for the survival of candidate solutions during the GA evolution process improved the convergence of the simple GA. In [7] the hybrid approach was applied to a test GMS system to obtain a solution whose cost value was around 0.1% less than the best solution obtained using a simple GA. In [8] a Tabu Search (TS) technique was coupled with a GA/SA hybrid method. In each generation, the best solution was selected as the new trial solution for the TS. The TS searches in the neighborhood of this solution in order to locate any local improvement. The hybridization improved the convergence of the algorithms.

In the GMS applications discussed above the GA/SA hybrid approaches were developed using populations of binary strings to represent the maintenance state of a generating unit over the scheduling period. With this encoding type, the length of a chromosome becomes very long for genuine problems, increasing the size of the search space. In [4] it was observed that a GA with an integer encoding to represent the maintenance start period for a generating unit gives a better performance than a GA with a binary encoding.

The hybrid approach presented in this paper uses the integer encoding for solving the GMS problem. The effect of varying the parameters of the method on its performance is analyzed and the results are compared with those obtained using the GA and SA approaches alone. Whereas earlier GA/SA hybrids used initial population pools with randomly generated candidate solutions, the research reported herein includes the inoculation of the GA/SA approach by seeding the initial population pool.

The paper is organized as follows. The following section describes the GMS test problem and the mathematical model. Section 3 introduces the proposed GA/SA hybrid solution technique and details its implementation to the test problem. It also summarizes the results obtained for the test GMS problem using a simple GA and SA method. The performance and the results obtained from the GA/SA and inoculated GA/SA approaches are discussed in section 4, and conclusions follow in section 5.

2 GMS test problem formulation

The test problem consists of scheduling the maintenance of 21 generating units over a planning period of 52 weeks. This test problem is loosely derived from the example presented in [2] with some simplifications and additional constraints, and has been previously studied in [4,5]. Table 1 gives the capacities, allowed periods and duration of maintenance and the manpower required for each unit. The power system peak load is 4739 MW, and there are 20 technical staff available for maintenance work in each week. The problem involves the reliability criterion of minimizing the sum of squares of the reserves in each weekly time period. Each unit must be maintained (without interruption) for a given duration within an allowed period. The allowed period for each generator is the result of a technical assessment and the experience of the maintenance personnel, which ensures adequate maintenance frequency. Due to its complexity the exact optimum solution for this problem is unknown.

The GMS problem can be formulated as an integer programming problem by using integer variables to represent the period in which the maintenance of each unit starts. The variables are bounded by the maintenance window constraints. However, for clarity the problem is first formulated using binary variables which indicate the start of maintenance of each unit at each time.

Unit	Capacity (MW)	Allowed period	Outage (weeks)	Manpower required for each week
1	555	1-26	7	10+10+5+5+5+5+3
2	555	27-52	5	10+10+10+5+5
3	180	1-26	2	15+15
4	180	1-26	1	20
5	640	27-52	5	10+10+10+10+10
6	640	1-26	3	15+15+15
7	640	1-26	3	15+15+15
8	555	27-52	6	10+10+10+5+5+5
9	276	1-26	10	3+2+2+2+2+2+2+2+2+3
10	140	1-26	4	10+10+5+5
11	90	1-26	1	20
12	76	27-52	3	10+15+15
13	76	1-26	2	15+15
14	94	1-26	4	10+10+10+10
15	39	1-26	2	15+15
16	188	1-26	2	15+15
17	58	27-52	1	20
18	48	27-52	2	15+15
19	137	27-52	1	15
20	469	27-52	4	10+10+10+10
21	52	1-26	3	10+10+10

Table 1: Data for the test system.

Notation:

i	index of generating units
I	set of generating unit indices
N	total number of generating units
t	index of periods
T	set of indices of periods in planning horizon
ep_i	earliest period for maintenance of unit i to begin
lp_i	latest period for maintenance of unit i to end
d_i	duration of maintenance for unit i
P_{it}	generating capacity of unit i in period t
L_t	anticipated load demand for period t
M_{it}	manpower needed by unit i at period t
AM_t	available manpower at period t

Suppose $T_i \subset T$ is the set of periods when maintenance of unit i may start, so $T_i=\{t\in T:\ ep_i \le t \le lp_i - d_i + 1\}$ for each i. We define,

$$X_{it}=\begin{cases}1 & \text{if unit i starts maintenance in period t}\\ 0 & \text{otherwise}\end{cases},$$

to be the maintenance start indicator for unit $i \in I$ in period $t \in T_i$. It is convenient to introduce two further sets. Firstly let S_{it} be the set of start time periods such that if the maintenance of unit i starts at period k that unit will be in maintenance at period t, so $S_{it}=\{k\in T_i:\ t-d_i+1 \le k \le t\}$. Secondly, let I_t be the set of units which are allowed to be in maintenance in period t, so $I_t=\{i:\ t\in T_i\}$. Then the problem can be formulated as a quadratic 0-1 programming problem as below.

The objective is to minimize the sum of squares of the reserve generation,

$$\underset{X_{it}}{\text{Min}} \left\{ \sum_{t\in T} \left[\left(\sum_{i\in I} P_{it} - \sum_{i\in I_t} \left(\sum_{k\in S_{it}} X_{ik} P_{ik} \right) - L_t \right)^2 \right] \right\}, \quad (1)$$

subject to the maintenance window constraint,

$$\sum_{t\in T_i} X_{it} = 1 \qquad \text{for all } i\in I, \quad (2)$$

the manpower constraint,

$$\sum_{i\in I_t} \sum_{k\in S_{it}} X_{ik} M_{ik} \le AM_t \qquad \text{for all } t\in T, \quad (3)$$

and the load constraint,

$$\sum_{i\in I} P_{it} - \sum_{i\in I_t} \sum_{k\in S_{it}} X_{ik} P_{ik} \ge L_t \qquad \text{for all } t\in T. \quad (4)$$

3 GA/SA hybrid technique

3.1 Introduction

A GA approach maintains a population of candidate solutions throughout the solution process. In a simple GA, an initial population of candidate solutions is generated randomly or by other means. During each iteration step, a new population is formed by applying selection, crossover and mutation operators to solutions in the current population based on their individual goodness.

A SA approach maintains a single solution in the search space throughout the solution process. First an initial solution and an initial 'temperature' are selected. As the algorithm progresses a new trial solution is generated by making a move from the current solution and the temperature is reduced according to a specified cooling schedule. If the new solution is an improvement, it is accepted unconditionally, otherwise it is accepted with a probability defined by the current temperature and quality of the new solution. Progression through successive iterations leads to a gradual reduction in the probability of accepting non-improved trial solutions.

The proposed hybrid GA/SA in this paper combines these GA and SA approaches. The mechanism of the proposed GA/SA approach for a minimization problem is shown diagrammatically in Figure 1.

The proposed GA/SA approach maintains a population of candidate solutions throughout the solution process using a steady state approach. The steady state approach directly inserts a new solution into the population pool replacing a less fit solution. First an initial population of candidate solutions is generated randomly or by other means and an initial temperature is selected. The initial temperature should be large enough to allow the free movement of a trial solution in the search space in the early stages of the search process.

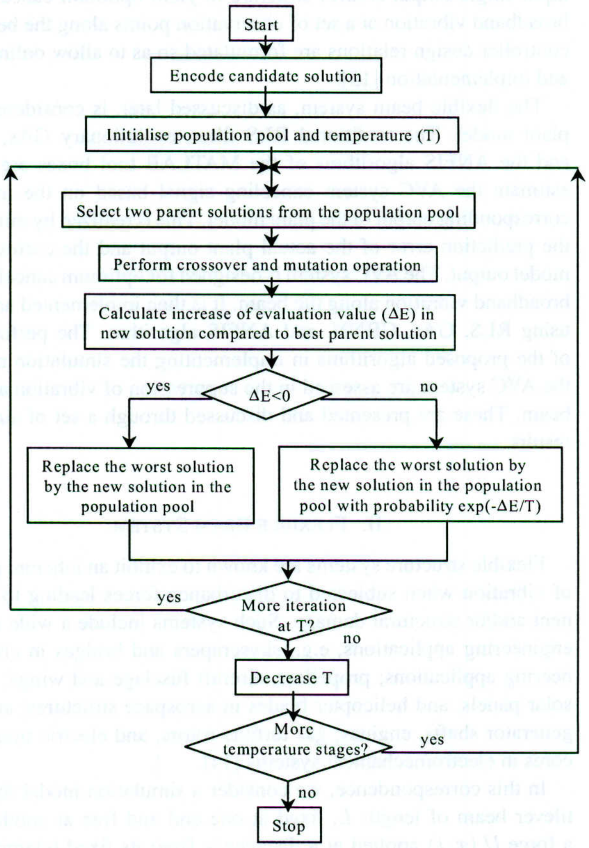

Start
Encode candidate solution
Initialise population pool and temperature (T)
Select two parent solutions from the population pool
Perform crossover and mutation operation
Calculate increase of evaluation value (ΔE) in new solution compared to best parent solution
yes
ΔE<0
no
Replace the worst solution by the new solution in the population pool
Replace the worst solution by the new solution in the population pool with probability exp(-ΔE/T)
yes
More iteration at T?
no
Decrease T
More temperature stages?
yes
no
Stop

Figure 1: The algorithm of the proposed GA/SA hybrid method.

In each iteration the GA/SA hybrid approach selects two solutions from the population pool and applies a crossover operator. One of the "crossovered" solutions is randomly selected to undergo mutation. The resulting solution replaces an existing member of the population pool. The solution is inserted in a controlled manner, by taking account of its evaluation value and the stage reached in the search process. To implement this, the probabilistic acceptance approach of the simple SA is incorporated into the GA algorithm to decide whether the new solution should be included in the population. This is expressed by,

$$\mathrm{Prob}(\Delta E)=\exp(-\Delta E/T), \qquad (5)$$

where ΔE is the increase of the evaluation value in the new solution (as described in section 3.2.2) and T is the temperature which defines the stage in the process.

If the new solution is an improvement, it is accepted, otherwise it is accepted with a defined probability given by equation (5). As in a simple SA algorithm, the initial temperature is fixed to a large value and is reduced gradually according to a cooling schedule as the algorithm progresses. As the temperature is reduced from high to low during the GA/SA process, the probability of accepting non-improved newly born solutions is reduced. At the beginning of the search process new solutions are accepted with a high probability. In the latter stages however, the GA/SA approach is constrained to a local search space due to the reduction in the probability of accepting non-improved solutions.

3.2 Implementation

A number of decisions must be made in order to implement the proposed GA/SA method for solving the GMS problem. Firstly, there are problem specific decisions which are concerned with the search space (and thus the representation) of feasible solutions and the form of the evaluation function. The second class of decisions is generic, and involves the operators and the parameters of the technique itself.

3.2.1 Solution encoding

The encoding of the problem using an appropriate representation is a crucial aspect of the implementation of the GA/SA hybrid technique. The encoding used to represent solutions of the problem defines the size and the structure of the search space.

In the previously reported work [4], GAs were applied to GMS using binary and integer strings to represent solutions. The integer representation was found to be more effective for GMS problems as this respects the maintenance window constraint (2). This representation allows the GA to focus in the area of the solution space where constraint (2) is not violated - thereby greatly reducing the size of the search space by ignoring unfeasible solutions. Therefore, integer strings are used here to represent candidate solutions of the problem. The string is given by $t_1,t_2,...,t_i,...,t_N$, where t_i is an integer which indicates the maintenance start period for unit i, $ep_i \le t_i \le lp_i-d_i+1$.

3.2.2 Evaluation function

The goodness (evaluation value) of every trial solution is calculated by using an evaluation function. The evaluation function formulated for the test problem is a weighted sum of the objective function and the penalty function for violations of the constraints. The penalty value for each constraint violation is proportional to the amount by which the constraint is violated. Hence,

$$\text{evaluation} = \omega_O \times SSR + \omega_M \times TMV + \omega_L \times TLV, \qquad (6)$$

where SSR is the sum of squares of reserves as in (1), TMV is the total manpower violation of (3), and TLV is the total load violation of (4). The weighting coefficients ω_O, ω_M and ω_L are set such that the penalty values for the constraint violations dominate over the objective function, and to ensure that the violation of the relatively hard load constraint (4) gives a greater penalty value than for the relatively soft crew constraint (3). This balance is because a solution with a high reliability but requiring more manpower may well be accepted by the power utility as the unavailable manpower may be hired. In fact, there is a trade-off between the level of reliability (i.e. sufficient reserve margin) and the required extra manpower. Feasible solutions with low evaluation measures have high fitness values while unfeasible solutions with high evaluation measures have low fitness measures.

3.2.3 SA and GA applications

For the purpose of comparison, the applications of an SA and a simple GA to the same test GMS problem are discussed in this section. Both methods use the integer encoding and the same evaluation function. The discussion about these applications is relatively limited in this paper; a fuller description is given in previous works [4,5].

The total number of iterations (i.e. fitness evaluations) for each run of the SA and GA methods has been set to 30,000, which was determined by an empirical analysis of the convergence of these methods. Both of the methods have been implemented on a Sun Sparcstation 1000 using the Reproductive Plan Language, RPL2 [14]. The design of these approaches to give the best performance in terms of finding good solutions to the test GMS problem has been established after extensive experimentation. The adopted experimentation approach involved conducting ten runs with particular selection of parameters and identifying the best solution (lowest evaluation value) over these runs, and the average of the best solutions from each of the ten experiments.

SA results

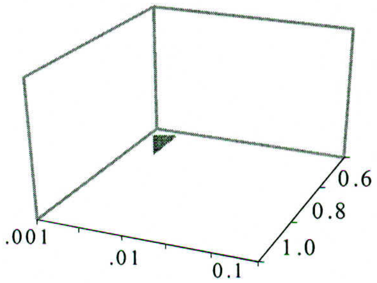
For the SA method, the initial temperature has been set to a value of 10,000 following an earlier period of experimentation. A stage-wise cooling schedule, which executes a number of iterations at each temperature before reducing the temperature, has also been found to give better results. Ten SA runs were made with the identified design. The average evaluation value of the best solutions obtained over ten SA runs was 146.06 and the best solution had evaluation value of 140.49.

GA results

For the simple GA method, the genetic operators used were tournament selection, standard two-point crossover and standard random mutation again based on extensive experimentation. The tournament selection method picks a subset of solutions at random from the population to form a tournament selection pool, from which one solution is selected with probability based upon the evaluation values of the solutions. Two solutions selected are then subjected to crossover with a defined crossover probability (CP). The two-point crossover operator splits the selected solutions at two randomly chosen positions and exchanges the center sections with probability CP. One of the resulting solutions is then chosen to undergo mutation, which changes the integer at each position in the solution within the allowed range with a defined mutation probability (MP). The elitist approach, which ensures that the best solution in the population pool is always retained, has been applied. The population size and the tournament pool size have been taken to be 100 and 10 respectively following a period of experimentation.

The best performance in terms of finding good solutions to the problem was obtained with a steady state population updating approach.

The sensitivity of the steady state GA with variation of crossover probability (CP) and mutation probability (MP) is shown in Figure 2. The GA gives the best result when CP=1.0 and MP=0.05. The average evaluation value of the best solutions obtained over ten GA runs with these values for CP and MP was 146.71 and the best solution had an evaluation value of 137.91.



and the manpower constraint were violated in one time period. The evaluation value for this solution is 483.70.

The average evaluation value of the best solution over ten experiments for the inoculated GA was 142.67 and the best solution had evaluation value of 139.95. Although the best solution found by the GA approach is slightly better than that found by the inoculated GA, the average performance of the inoculated GA over ten runs was found to be significantly better than that of the GA approach.

For comparison's sake the results obtained using these SA, GA, heuristic and inoculated GA methods are summarized in Table 2. The designs of the individual GA and SA approaches that have been found to be effective have been incorporated into the proposed hybrid approach.

	Average of best solutions	Best solution
SA	146.06	140.49
GA	146.71	137.91
Heuristic	-	483.70
Inoculated GA	142.67	139.95
GA/SA	145.78	138.12
Inoculated GA/SA	141.71	139.10

Table 2: Comparison of results obtained using different methods.

3.2.4 GA/SA Architecture

The features for the GA and SA adapted in the hybrid approach have been borrowed from the results described in the previous section. In summary these are: a steady state approach, tournament selection, two-point crossover, random mutation, a population size=100, a tournament pool size=10, an initial temperature=10,000 and a stage-wise cooling schedule.

A temperature defines a stage of the GA/SA process. The stage-wise cooling schedule executes a number of genetic operations (iterations) at a temperature (i.e. at one stage) before reducing the temperature according to equation,

$$T_s=\alpha T_{s-1}, \qquad (7)$$

where T_s is the temperature at stage s and α is the cooling parameter. In the reported experimentation the genetic operations have been performed 100 times for each temperature. The number of temperature alterations (or stages) was fixed to 300, giving 30,000 fitness evaluations per run of the algorithm.

In each iteration the evaluation value of a newly created solution (E_{new}) is compared with the evaluation value of the best amongst its parents ($E_{current}$) to calculate the increase in evaluation value ($\Delta E= E_{new} - E_{current}$). The new solution is then accepted with probability given by (5). An acceptance of a new solution replaces the worst solution of the population pool.

The GA/SA approach has also been implemented using the Reproductive Plan Language, RPL2 [14].

4 GA/SA test results and discussion

4.1 Sensitivity analysis

The particular design that gives the best performance of the GA/SA is typically identified after a process of experimentation. The general approach adopted during experimentation takes the same format as that for the GA and SA methods. That is, over a series of ten GA/SA runs the average evaluation measure of the best solutions and the evaluation value of the best solution found are identified. These averaged and best evaluation values are used to compare the performance of the various approaches.

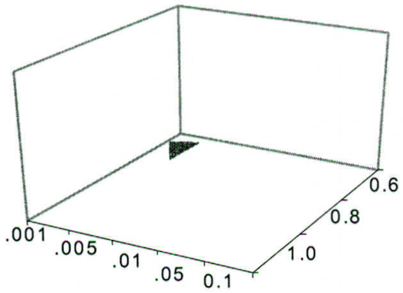
In order to determine the best value of the cooling parameter (α) for the proposed hybrid method, a number of experiments have been performed and the results obtained are summarized in Table 3. Three values of α covering a relatively wide range have been selected for the experiments based on previous experience with the SA method. With α=0.92, the temperature decrease is very rapid and the algorithm lacks in exploration, concentrating more on exploitation in the neighborhood of a solution in the population pool. With α=0.98, the temperature does not drop sufficiently far within 30,000 iterations and the method works as a simple GA technique. The cooling schedule with α=0.95 provides a good compromise between the exploitation and exploration during the search process and this is supported by observing the best performance of the algorithm for 30,000 iterations. This cooling parameter value is used for further investigation of the GA/SA method.

α	Average evaluation value
0.92	148.03
0.95	145.78
0.98	352.24

Table 3: Average performance of the GA/SA method with different values of cooling parameter and 30,000 iterations.

The sensitivities of the method to variation of crossover probability (CP) and mutation probability (MP) have also been established. Results were obtained for varying CP in the range [0.6, 1.0] and MP in the range [0.001, 0.1]. For each value of CP and MP ten independent experiments were performed using the same collection of ten random initial populations. The sensitivity of the GA/SA approach to variation of CP and MP is depicted in Figure 3.

Comparing Figure 2 and Figure 3 directly, it can be observed that the performance of the GA/SA is generally less sensitive than that of the simple GA for the given range of crossover probability. The performance of the GA/SA and the simple GA method does not differ much for MP=0.001. However, for higher MP values the GA/SA method is more robust than the simple GA method alone in terms of consistently finding better results. Although mutation can introduce new information to solutions, it can also destroy useful information. In the simple GA the mutation operator becomes disruptive as MP increases as seen in the climbing evaluation value of the graph. In the GA/SA hybrid method, the SA probabilistic acceptance test tends to preserve the positive effects and counter the adverse effects of the mutation operator. That is, new solutions, even those whose evaluation function values are lower than those of current solutions, are fully accepted at the beginning of the search, thus introducing more diversity amongst the candidate solutions. However, at later stages of the search process, the chance of mutated solutions of lesser fitness being accepted will be low.



demonstrated for a test problem of generator maintenance scheduling. The sensitivity of the approach to the variation of cooling parameter, crossover probability and mutation probability has been studied. An inoculated GA/SA using a seeded initial population pool has also been employed to the test GMS. The performance and results obtained from these GA/SA approaches have been compared with those of other techniques.

The test results show that the GA/SA approach is sensitive to the cooling parameter; this should be selected to make a good compromise between exploration and exploitation of the search space for the given number of iterations (computational time).

The best crossover and mutation probabilities of a GA approach are generally decided upon after a number of experiments. The results presented in this paper show that the GA/SA approach is more robust and stable for solving GMS problems in a wide range of crossover and mutation probabilities than a GA approach. Hence, the parameter selection process in the GA/SA method involves fewer experiments than that in the GA method. Furthermore, the hybrid method also improved the convergence of the simple GA.

The study of the inoculated GA/SA using a heuristically derived solution in the initial population shows that inoculation can enhance the performance of the GA/SA approach. Comparing the individual average results of different approaches considered, the inoculated GA/SA approach gives the best average performance.

Acknowledgements

This work was carried out in the Rolls-Royce University Technology Centre in Power Engineering at the University of Strathclyde in Glasgow. The authors acknowledge the use of the Reproductive Plan Language, RPL2, produced by Quadstone Limited, in the conduct of this work.

References

[1] G.T. Egan, T.S. Dillon and K. Morsztyn, “An experimental method of determination of optimal maintenance schedules in power systems using branch-and-bound technique”, IEEE Transactions on Systems, Man and Cybernetics. SMC-6, 538-547, 1976.

[2] Z. Yamayee and K. Sidenblad, “A computationally efficient optimal maintenance scheduling method”, IEEE Transactions on Power Apparatus and Systems PAS-102, 330-338, 1983.

[3] S. Yang, “Maintenance scheduling of generating units in a power system”, in X. Wang and J.R. McDonald (Eds.) Modern Power System Planning, McGraw-Hill, London, 247-307, 1994.

[4] K.P. Dahal and J.R. McDonald “Generator maintenance scheduling of electric power systems using genetic algorithms with integer representation”, Proceedings of Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA’97), IEE Conference Publication No. 446, 456-461, 1997.

[5] K.P. Dahal, J.R. McDonald and G.M. Burt, “Modern heuristic techniques for scheduling generator maintenance in power systems”, to appear in the Transactions of Institute of Measurement and Control 22, 2000.

[6] J.F. Dopazo and H.M. Merrill, “Optimal generator maintenance scheduling using integer programming”, IEEE Transactions on Power Apparatus and Systems. PAS-94, 1537-1545, 1975.

[7] H. Kim, K. Nara and M. Gen, “A method for maintenance scheduling using GA combined with SA”, Computers and Industrial Engineering 27, 477-480, 1994.

[8] H. Kim, Y. Hayashi and K. Nara, “An algorithm for thermal unit maintenance scheduling through combined use of GA, SA and TS”, IEEE Transactions on Power Systems 12, 329-335, 1997.

[9] K.B. Burke, J.A. Clarke and A.J. Smith, “Four methods for maintenance scheduling”, in G.D. Smith, N.C. Steele and R. Albrecht (Eds.) Proceedings of Third International Conference Artificial Neural Nets and Genetic Algorithms (ICANNGA’97), Springer-Verlag, Vienna, 265-270, 1998.

[10] C.R. Reeves (Ed.), “Modern heuristic techniques for combinatorial problems”, Blackwell Scientific Publications, Oxford, 1993.

[11] V. Miranda, D. Srinivasan and L.M. Proença, “Evolutionary computation in power systems”, Electrical Power & Energy Systems. 20, 89-98, 1998.

[12] K.P. Wong, “Solving power system optimization problems using simulated annealing”, Engineering Applications of Artificial Intelligence. 8, 665-670, 1995.

[13] T. Satoh and K. Nara, “Maintenance scheduling by using simulated annealing method”, IEEE Transactions on Power Systems. 6, 850-857, 1991.

[14] “Reproductive Plan Language (RPL2) - User manual”, Quadstone Ltd, 1997.