A REVIEW OF GENERATOR MAINTENANCE SCHEDULING USING ARTIFICIAL INTELLIGENCE TECHNIQUES

K.P. Dahal, J.R. McDonald

Centre for Electrical Power Engineering, University of Strathclyde, Glasgow, UK.

ABSTRACT

New Artificial Intelligence (AI) approaches such as simulated annealing, genetic algorithms, simulated evolution, neural networks, tabu search, fuzzy logic and their hybrid techniques have been applied in recent years to solving Generator Maintenance Scheduling (GMS) problems. This paper presents a review of these AI approaches for the GMS problem. The formulation of problems and the methodologies of solution are discussed and analysed. A case study is also included which presents the application of a genetic algorithm to a test system based on a practical power system scenario.

1. INTRODUCTION

In recent years researchers have focused much attention on new theoretical and methodical approaches for Generator Maintenance Scheduling (GMS) from the power system planning, design and operational points of view. GMS for a power utility is a complex combinatorial constrained optimisation problem. A practical problem considering economic, security, reliability and operational requirements must be formulated, and an efficient methodology must be developed to solve the formulated problem.

Conventional solution methods are generally based on heuristic techniques or mathematical methods including integer programming, branch-and-bound techniques and dynamic programming [1]. 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]. 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 characterise the GMS problem.

In order to overcome the above limitations, a number of Artificial Intelligence (AI) techniques have been recently implemented for GMS problems. The general objectives and constraints of the GMS problem are described in Section 2. Section 3 reviews the application of simulated annealing, genetic algorithms, simulated evolution, neural networks, tabu search, fuzzy logic and their hybrid methods to the formulation and solution of GMS problems. Section 4 presents results from a case study of applying a Genetic Algorithm (GA) to a test system based on a practical power system scenario. Conclusions follow in Section 5.

2. GMS PROBLEM DESCRIPTION

There are generally two categories of objective functions in GMS, those based on reliability [1,4,6-9] and those based on economic cost [2,3,5,9-11]. The levelling of the reserve generation over the entire operational planning period is the most common reliability criterion. This levelling can be realised by maximising the minimum net reserve of the system during any time period. In the case of a large variation of reserve, minimising the sum of squares of the reserves can also be an effective approach. Alternatively, the quality of reserve is considered, whereby the risk of exceeding the available capacity is levelled over the entire period by using the load carrying capacity (ELCC) for each unit and an equivalent load (EL) for each interval. Minimising the sum of the individual loss of load probabilities (LOLP) for each interval can also be an objective for the reliability criterion under the condition of load uncertainty and random forced outage of units.

The most common objective based on economic cost criteria is to minimise the total operating cost, which includes the costs of energy production and maintenance. If outage duration is allowed to vary in the minimisation of the generation operating cost, this results in a trade-off solution between the energy production and the maintenance cost. Higher maintenance costs lead to shorter outage durations, thus reducing the load of expensive generation and possible energy purchases, consequently resulting in a lower energy production. The production cost alone could also be chosen as the objective function by minimising the total energy replacement cost due to preventive maintenance scheduling. However, this is an insensitive objective as it requires many approximations [1].

The GMS problem has a series of constraints related to the generating units and the power system:

- Maintenance window constraints - define the earliest and latest time and the duration of maintenance for each unit.
- Crew constraints - consider the manpower availability for maintenance work.
- Resource constraints - specify the limits on the resources needed for maintenance at each period.

- Exclusion constraints - prevent the simultaneous maintenance of a set of units.
- Sequence constraints - restrict the initiation of maintenance of some units after a period of maintenance of some other units.
- Load constraints - consider the demand on the power system during the scheduling period.
- Reliability constraints - consider the risk level on the selected maintenance schedule.
- Transmission capacity constraints - specify the limit of transmission capacity in the interconnected power system.
- Geographical constraints - limit the number of generators under maintenance in each region.

3. AI SOLUTION TECHNIQUES

3.1 Simulated Annealing

Satoh & Nara [2] solved a thermal GMS problem using Simulated Annealing (SA) to find the start period of maintenance for each unit. The SA method is based on the analogy between the physical annealing process of a solid and the problem of finding the minimum of a given function depending on many parameters, as encountered in combinatorial optimisation problems.

The authors formulated the GMS problem as a mixed-integer programming problem, and three systems of different sizes were tested. At each stage (cf. temperature) a trial solution in the neighbourhood of the current solution is generated by selecting a generating unit (cf. molecule) with uniform probability and randomly changing its maintenance period. If the trial solution is an improvement, it is accepted, otherwise it is accepted with a defined probability. The process is repeated within each stage until the number of generated solutions in this stage is sufficiently large (equilibrium condition reached). Successive stages then lead to a gradual reduction in the probability of accepting trial solutions. When the number of acceptable solutions is small enough, the "freezing point" is reached and the algorithm is terminated.

The authors showed that SA was between 12 and 70 times quicker than integer programming (IP) in finding the same solution for their small and medium sized problems respectively. For the large system studied, IP could not be adopted from the computational point of view but the SA approach was able to find a solution despite requiring a large computational time (about 21 hours). However, the authors do not see this as a defect as the maintenance scheduling is carried out over a year or more, and it is important to find a near optimal solution for a real size problem.

3.2 Genetic Algorithms

Genetic Algorithms (GAs) are based on natural genetic and evolution mechanisms and work on populations of solutions. GAs are iterative procedures which maintain a population of candidate solutions to an optimisation problem. First, 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 and recombination (crossover and mutation) operators to solutions in the current population based on their individual goodness. The crossover operator exchanges information between candidate solutions and the mutation operator introduces a random change in the solution to reach new parts of the search space.

For GMS problems, a binary [3-5] or integer [6] string can be used to represent each solution in the population, this represents a vector of parameters which is analysed by a fitness function in order to determine its goodness. GAs were initially developed using populations of binary strings, which for GMS may be used to represent the maintenance state [3] or maintenance start period [4,5] of a unit within the scheduling period. However, the GMS problem variables are numeric and representing them directly as integers rather than bit strings reduces the search space greatly making the GA very effective [6]. Section 4 describes a case study which uses this integer encoding to represent maintenance start periods. Generally penalty functions are introduced in the formulation of the fitness function for GMS problems [3-6], which take care of the various constraints imposed on the system.
Kim et al. [3] presented an application of the GA approach for GMS using the acceptance probability of the SA method for the survival of individuals during the evolution process. Here the crossover was performed not only between solutions but within a solution as well (effectively implementing a mutation-like operator). This hybrid approach was applied to a test system with 15 units over 260 periods to obtain a solution whose cost value was around 0.1% less than the best solution obtained using a simple GA.

3.3 Simulated Evolution

Sutoh et al. [7] proposed a new method for solving a large scale GMS problem using Simulated Evolution (SE), which is also based on an analogy of the natural selection process in biological environments. Unlike GAs, each individual represents one variable and the whole population of individuals represents one solution. The SE algorithm allows the selection of many variables (individuals) simultaneously. The probability that a variable is selected depends on its suitability (goodness) to the environment.

In this approach, each individual corresponds to a generating unit to be maintained. The goodness is effectively given by the variance of the ratio of reserve to load during the maintenance periods of the units. The method needs initialisation with a feasible initial solution, which was found using a branch-and-bound based depth-first search approach. The subsequent iterative process involves evaluation, selection and rescheduling steps until a termination criterion is satisfied. A unit that is maintained in the low reserve margin periods is evaluated as having low goodness and is highly likely to be selected to be reset and rescheduled in higher reserve margin periods. A depth-first search is applied to find a better schedule for the units in the rescheduling stage.

The authors presented the results for systems with 10 units over 36 periods and 40 units over 52 weeks, and compared the SE technique's effectiveness with the simple depth-first search approach. The SE method was about 32 times faster in finding the optimal solution for the small system, and found a significantly better solution for the large system than the depth-first search for a fixed number of iterations.

3.4 Neural Networks

Yoshimoto et al. [8] presented a Neural Network (NN) based approach for a large-scale GMS problem. The design of a neural network is motivated by an analogy with the human brain for solving problems more complex than those based on conventional hard-wired design techniques. The authors used a Dynamic Canonical Network, which is an extension of the standard Hopfield Neural Network, in order to handle 0-1 integer programming problems with nonlinear objective and inequality constraints. This network has variable neurons which represent the value of the variables in the programming problem and constraint neurons which represent the satisfaction of inequality constraints. If a constraint is not satisfied the corresponding neuron makes no output, otherwise it makes a non-zero output. Provided that the response time of the constraint neurons is much faster than that of the variable neurons, the energy function of the network decreases monotonically along with the action of the neural network, and converges to a stable equilibrium point. This convergence is however dependent on the selection of the initial state. The stable equilibrium point is one of the local minima of the objective function.

The approach decomposes the scheduling period into several sub-periods and the NN is applied to the respective sub-problems. As the number of target generators is small and the duration of the targeted periods is short within the sub-problems, the volume of computation can be considerably reduced.

The authors also adopted some heuristic ideas during the neural network procedure. The algorithm first checks the feasibility of the given conditions and constraints of the GMS problem. Then the maintenance of large-scale units is performed during the periods which have enough supply. After decomposing the problem by grouping generating units that require maintenance in the same sub-period, the neural network is applied to each sub-problem. The obtained schedule is evaluated and the part which is unsatisfactory is modified. When such an approach has been applied to all groups of units, the computation is completed.

This approach was applied to two test systems using different initial conditions. The NN approach found a solution within 10% of the optimal solution obtained by using implicit enumeration in 1/15th of the time. The effectiveness of the decomposition technique was demonstrated for the large system, for which a solution was found in 10 minutes whose cost was within 3% of the solution found by solving the whole system collectively in over 4 hours.

3.5 Tabu Search

Burke et al. [5] compared Tabu Search (TS) methods with a hybrid of SA/TS, SA and binary GA techniques for GMS problems. The TS method considers the neighbourhood of a current solution to create the next solution during the search process. In each iteration, only one variable (maintenance start period) is moved by changing the start period either to each possible starting period or just to the periods adjoining it.

Two different approaches, known as 'solution tabu' and 'move tabu', have been applied. The first method compares the state of a trial solution to the solutions from previous iterations, so it cannot revisit a solution from that number of iterations. The move tabu compares the move from the current solution to the trial solution with a number of previous moves and prevents the algorithm from performing the same move more than once during these iterations. The authors also presented a hybrid approach using TS within an SA technique in which a candidate solution in a neighbourhood is accepted only if it is not in the list of recently accepted solutions.

These methods were compared for three systems as given in [2]. The solution tabu method found solutions with 0.2-2% lower cost but took 4 to 11 times more computational time than other methods. The move tabu found a solution whose cost was 0.5% of that found by the solution tabu method, but was 5 times faster.

3.6 Fuzzy Logic

The GMS problem involves multiple objectives. The general expression of the objective function for GMS may be a combination of several individual objectives that may conflict with each other. Furthermore, a real GMS problem includes many uncertainties. The maintenance window, manpower and resource constraints of a GMS problem are not as rigid as conventional deterministic techniques treat them. In fact, utility requirements to schedule maintenance work with minimum cost and maximum reliability are not as crisp as is commonly believed.

An approach based on fuzzy sets can deal with multiple conflicting objectives [9] and take account of the imprecise and flexible environment of the GMS problem [9-11]. In the formulation of maintenance scheduling problems under a fuzzy environment the objective function (multiple or single) and constraints, which include uncertain variables, are all expressed in fuzzy set notation. Huang et al. [9] implemented fuzzy dynamic programming to determine the optimal GMS decision. Imprecise constraints and two objectives, the reserve margin and the additional production cost at the maintenance time are fuzzified by using linear membership functions, and a recursive algorithm similar to conventional dynamic programming is applied to calculate the highest membership value at each stage of the search path.

Gibson et al. [10] formulated the GMS problem with a fuzzy objective for levelling reserve which considered the uncertainty of the load. The branch-and-bound technique was used to achieve the optimal solution.

Noor and McDonald [11] demonstrated fuzzy 0-1 linear programming for a small GMS problem. The linear fuzzy objective is converted into a fuzzy constraint by using an aspiration level and the flexible/imprecise constraints are represented by fuzzy functions. The problem is then converted into a crisp linear mixed integer programming problem to maximise membership in the decision set. These fuzzy methods provide a greater solution flexibility, but are based on mathematical methods and suffer from the corresponding limitations as explained in section 1.

4. A CASE STUDY USING A GA TECHNIQUE

In this section we demonstrate the application of a GA to solve a GMS test problem comprising 21 units over a planning period of 52 weeks. The data for the test problem is given in [4,6]. The objective is to schedule the maintenance outages of generators to minimise the sum of the squares of the reserve generation. The problem includes many features which characterise real systems, such as maintenance window, crew and load constraints. The GA was implemented using the publicly available GENITOR package. The following discussion is relatively limited, whereas the results are discussed in more detail in [6].

Integer strings are used to represent candidate solutions of the problem in the population. Each integer of the string indicates the period when maintenance for a unit starts. The evaluation function is the weighted sum of penalty values for each constraint violation and the objective function itself. Feasible solutions with low evaluation measures have high fitness values while unfeasible solutions with high evaluation measures have low fitness measures.

The crossover operator used here is a simple two-point crossover. The crossover is applied in each iteration when the exchanged information is unique to each parent. The mutation operator takes each integer in a solution string and with the given mutation probability (MP) changes it within the allowed integer interval. The distribution of the new integer value within the interval is approximately uniform during mutation. The population size (PS) specifies the number of individuals in the solution population. The selection bias (SB) value specifies the amount of preference to be given to the superior individuals in the population.

A number of GA runs have been carried out to observe the sensitivity of the GA to the variation of MP, SB and PS. Table 1 presents the test results for different values of MP and SB taking other GA parameters as constant. Each case presents the minimum, average and maximum evaluation measures of the best solutions obtained for 5 GA runs. The total number of trials for each run was fixed at 30000.

The top portion of Table 1 shows that the average evaluation measure of the best solutions varies from 200 to 144 with the variation of MP from 0.001 to 0.1. For this particular problem, higher than generally expected values of MP have been found to give better results. The second portion of Table 1 demonstrates that a trade-off needs to be applied in the choice of the SB value. If SB is too high, then a superior solution strongly dominates the less fit solutions and this may lead the GA to converge prematurely to a local minimum. There is however little difference between average evaluation measures of the best solutions for the studied problem.

Table 1 Effect of GA parameters for the GMS problem.

	MP Value	min	avg	max
Selection	0.001	191	200	227
Bias=2.5	0.005	144	176	194
	0.01	141	160	198
Population	**0.05**	**138**	**144**	**157**
Size=50	0.1	147	157	170
	SB Value	min	avg	max
MP=0.05	1.5	148	156	174
	2.0	147	153	165
Population	**2.5**	**138**	**144**	**157**
Size=50	3.0	143	151	162

The GA was also tested with different population sizes between 10 and 500 keeping other GA parameters fixed: SB=2.5 and MP=0.05. It was found that the best solution was achieved with population size 175, though the performance of the GA did not vary greatly over the different cases.

Unlike the binary GA [4], it can be seen from the above test results that the integer GA is very stable for a wide range of variation in the GA parameters. The best solution found by the GA, whose cost is 138, is feasible and better than a heuristic solution (cost 222) calculated by ranking the generator units in order of decreasing capacity to level the reserve generation. Due to its complexity the optimal solution for this problem is unknown. The GA took 16.81s computational time to find the best solution on a DEC Ultrix 5000/260 workstation.

5. CONCLUSIONS

To overcome the limitations of conventional methods, a variety of Artificial Intelligence techniques have been applied to tackle GMS problems with different degrees of success. It is difficult to say which method is most appropriate for GMS problems, as the reviewed studies applied different techniques for different problems with different assumptions. The success of a method depends on many factors such as size and composition of the studied power system, the objective to be optimised, constraints to be considered, and the particular implementation of the methodology. AI techniques are not guaranteed to find optimum solutions but it can readily achieve good solutions to complex problems like GMS.

A case study of the application of a GA to a test GMS problem has been demonstrated. Good solutions to the problem can be found if appropriate problem encoding, evaluation function and GA parameters are selected. The use of integer encoding to represent GMS problem variables in a genetic structure can implicitly consider some of the problem constraints and greatly reduces the GMS search space. The results presented above show the integer GA is a robust technique for GMS problems and can find good solutions with a wide range of variations in GA parameters. The use of problem specific knowledge in the solution representation, the formulation of the evaluation function and the design of the GA operators can improve the GA technique for use in solving genuine large-scale GMS problems. Research on these particular issues is on-going and will be reported elsewhere.

REFERENCES

1. Wang, X. & McDonald, J.R., Modern power system planning, McGraw-Hill, London, 247-307, 1994.
2. Satoh, T. & Nara, K., Maintenance scheduling by using simulated annealing method, IEEE Transactions on Power Systems, 6:850-857, 1991.
3. Kim, H., Nara, K. & Gen, M., A method for maintenance scheduling using GA combined with SA, Computers Industrial Engineering, 27(1-4):477-480, 1994.
4. Dahal, K.P., McDonald, J.R., Generational and steady state genetic algorithms for generator maintenance scheduling problems, to appear in Proc. of ICANNGA'97, Norwich, 1997.
5. Burke, K.B., Clarke, J.A., Smith, A.J., Four methods for maintenance scheduling, to appear in Proc. of ICANNGA'97, Norwich, 1997.
6. Dahal, K.P., McDonald, J.R., Generator maintenance scheduling of electric power systems using genetic algorithms with integer representation, submitted to GALESIA'97, Glasgow, 1997.
7. Sutoh, T., Suzuki, H., & Nagai, N. Large-scale generator maintenance scheduling using simulated evolution, Proc. of ISAP'94, 567-573, Montpellier (France), 1994.
8. Yoshimoto, K., Yasuda, K., Yokoyama, R., Tanaka, H. & Akimoto, Y., Extended Hopfield network applied to maintenance scheduling of generator units based on decomposition, Proc. of ESAP'93, 62-67, 1993.
9. Huang, C.J., Lin, C.E. & Huang, C.L., Fuzzy approach for generator maintenance scheduling, Electric Power Systems Research, 24, 31-38, 1992.
10. Gibson, G., Kita, H., Nishiya, K. & Hasegawa, J., Thermal power station maintenance scheduling based on fuzzy theory, Proc. of ESAP'93, 50-55, 1993.
11. Noor, S. F. & McDonald, J.R., Application of fuzzy linear programming to maintenance scheduling, Proc. of American Power Conference, 57(2): 889-894, 1995.

Keshav Prasad Dahal
Centre for Electrical Power Engineering
University of Strathclyde, 204 George Street
Glasgow G1 1XW, UK.
email: k.p.dahal@strath.ac.uk
tel. +44 141 548 2168, fax +44 141 552 2487.