

# An investigation into the effectiveness of simulated annealing as a solution approach for the generator maintenance scheduling problem



E.B. Schlünz, J.H. van Vuuren \*

Department of Logistics, Stellenbosch University, Private Bag X1, Matieland 7602, South Africa

## ARTICLE INFO

### Article history:

Received 20 November 2012

Received in revised form 18 April 2013

Accepted 20 April 2013

### Keywords:

Generator maintenance scheduling

Simulated annealing

Power system

Reliability

## ABSTRACT

The *generator maintenance scheduling* (GMS) problem is the difficult combinatorial optimisation problem of finding a schedule for the planned maintenance outages of generating units in a power system. The GMS model considered in this paper is formulated as a mixed integer program, with a reliability optimality criterion, subject to a number of constraints. A new version of the *simulated annealing* (SA) method for solving the GMS problem is presented. Four cooling schedules (the geometric and three adaptive schedules), two neighbourhood move operators (an elementary move and an ejection chain move operator), and a hybrid local search heuristic/SA algorithm are compared. To our knowledge, this is the first study considering a different SA cooling schedule and move operator in a GMS context. A new 32-unit GMS test system is established and used in conjunction with a benchmark test system from the literature in this investigation. It is found that choosing a different cooling schedule and an ejection chain move operator yield improved results to that of the SA algorithm currently employed in the GMS literature. The hybrid SA algorithm performs very well compared to other methods on the benchmark test system from the literature, and an improved lower bound on the objective function value is presented for this test system.

© 2013 Elsevier Ltd. All rights reserved.

## 1. Introduction

A key focus area for an electricity utility is the planned preventative maintenance of the power generating units in its generation system. Regular preventative maintenance of generating units is required in order to prolong the life-expectancy of the generating units so as to ensure safe operating conditions, and most importantly to reduce the risk of unplanned outages caused by generating unit failures. In this paper, the problem of finding a schedule for the planned maintenance outages of generating units in a power system, known as the *generator maintenance scheduling* (GMS) problem, is considered. As power systems become larger and the demand for electricity increases continually, the difficulty of finding maintenance schedules increases in complexity, especially in highly constrained systems.

Due to the large combinatorial nature of the problem, exact solution approaches are not very effective within a reasonable computational time-scale, resulting in an increasing prevalence of approximate solution methodologies, such as heuristic and metaheuristic techniques. An exact solution approach guarantees a globally optimal solution to a problem, given sufficient computation time, whereas a solution produced by an approximate solution approach may or may not be (globally or even locally) optimal, but requires significantly less computation time.

In this paper, we consider the metaheuristic technique of *simulated annealing* (SA) for solving the GMS problem. Although SA has been adopted a number of times in the GMS literature, we could find no reference to experimentation with respect to improving the SA algorithm's effectiveness in the GMS context – a single cooling schedule (geometric) and a single elementary neighbourhood move operator (a random unit's maintenance starting time is changed to a random new time) are utilised throughout the literature [1–7]. In this paper, three adaptive cooling schedules and a new compound neighbourhood move operator are introduced in the context of GMS and compared to the schedules currently suggested in the literature. Additionally, a local search heuristic is introduced into the SA algorithm and the effectiveness of this hybrid solution technique is investigated. A new GMS test system is established and is used for comparison purposes along with a benchmark test system from the literature. It was found that choosing a different cooling schedule and the compound neighbourhood move operator yield improved results to that of the SA algorithm currently employed in the GMS literature. The hybrid SA algorithm performs very well compared to other methods on the benchmark test system from the literature, and an improved lower bound on the objective function value is presented for this system.

### 1.1. GMS model considerations

The optimality criteria for the GMS problem most often found in the literature may be grouped into three categories, namely

\* Corresponding author. Tel.: +27 (0) 21 808 2879; fax: +27 (0) 21 808 3406.

E-mail addresses: [schlunzeb@gmail.com](mailto:schlunzeb@gmail.com) (E.B. Schlünz), [vuuren@sun.ac.za](mailto:vuuren@sun.ac.za) (J.H. van Vuuren).

economic criteria, reliability criteria and convenience criteria [8]. These categories present conflicting requirements, ultimately making the GMS problem multiobjective in nature. However, both single and multiobjective approaches have been pursued in the literature.

The objective most commonly chosen within the category of economic criteria, is the minimisation of operating cost – typically comprising production cost and maintenance cost. Single objective formulations containing such an objective are wide-spread [9,10]. Another economic objective that has recently surfaced due to the emergence of competitive market environments, is the maximisation of profit [11]. In the category of reliability criteria, the objective is usually chosen as the levelling of the reserve load over the planning period. This is typically achieved by minimising the sum of the squares of the reserve loads; an approach successfully adopted in the single objective formulations found in [4,5,12]. The category of convenience criteria is the least used and we could not find any reference in the literature to single objective formulations in this category. However, objectives from this category appear in multiobjective formulations [13,14]. Objectives within this category include minimising the degree of constraint violations or minimising possible disruptions to the existing schedule. Finally, objectives from any of these categories may be combined in a multiobjective modelling approach [13–16].

The GMS problem may be subjected to various constraints, depending on the complexity of the model, assumptions and the requirements of the utility. In its simplest form, the GMS problem incorporates maintenance window constraints and load constraints in order to ensure, respectively, that each unit is scheduled for maintenance between an earliest and latest date, and that the system load demand is met for each time period. Additional constraints may be added as required. An alternative approach is to include constraints from the unit commitment and economic dispatch problems into the GMS problem [17]. The unit commitment and economic dispatch problems are short term scheduling problems (e.g. day-to-day or week-to-week) that determine which generating units should be in service during each time period, and the allocation of the load demand among those generating units during each time period, respectively [18–20].

## 1.2. Typical solution techniques

A wide variety of solution techniques for the GMS problem have been employed in the literature. Heuristic techniques are typically simple to understand and require very little computation time. Usually, generating units are scheduled for maintenance in a unit-by-unit manner with possible corrections made according to some externally defined scheduling order. Modern exact software suites capable of solving mathematical programs generally use branch-and-bound methods for solving integer problems. A decomposition method, known as Benders' decomposition, has also recently been employed in [9,10] to solve the typically large-scale GMS problem. A considerable amount of research has gone into the application of metaheuristic techniques for solving the GMS problem approximately. Different metaheuristics, including genetic algorithms in [1,4,21], simulated annealing in [1,4,6,7], tabu searches in [1,22], ant colony optimisation in [12,23] and particle swarm optimisation in [24], have successfully been applied to the GMS problem. Hybrid metaheuristic techniques have been employed in [1,5,25] to solve the GMS problem, achieving improved results in some cases compared to those obtained by the separate metaheuristic approaches. A relatively new modelling and solution approach to the GMS problem is the application of fuzzy set theory in order to address multiple objectives and uncertainties in the constraints. A fuzzy dynamic programming technique is employed in [15] and fuzzy metaheuristic techniques are employed in [26].

Finally, expert systems incorporate the many years of experience of field experts into a solution methodology [27].

## 2. Mathematical problem formulation

The structure of the GMS problem naturally calls for a mathematical programming modelling approach: a schedule must be obtained that optimises some objective, subject to restrictions on the schedule. Therefore, the mathematical model for the GMS problem considered in this paper takes the form of an integer program according to [28]. Reliability is chosen as the optimality criterion, with the goal of levelling the reserve load over the planning horizon, where the reserve load is defined as the available generating capacity less the system load demand. The objective function chosen to achieve this goal is to minimise the sum of the squares of the reserve loads. The constraints present in the model consist of the specification of maintenance windows for each unit, the system meeting the load demand together with a safety margin, adherence to the availability of maintenance crew and respecting general exclusion constraints.

Suppose there are  $n$  generating units in the power system and  $m$  time periods in the planning horizon. Let  $\mathcal{I} = \{1, \dots, n\}$  index the set of generating units and let  $\mathcal{J} = \{1, \dots, m\}$  index the set of time periods in the planning horizon. Let the binary decision variable  $x_{ij}$  take the value 1 if maintenance of generating unit  $i \in \mathcal{I}$  commences during time period  $j \in \mathcal{J}$ , or zero otherwise. Furthermore, define  $y_{ij}$  as a binary auxiliary variable taking the value 1 if generating unit  $i \in \mathcal{I}$  is in maintenance during time period  $j \in \mathcal{J}$ , or zero otherwise.

Let  $e_i$  and  $\ell_i$  denote the earliest and latest time periods, respectively, during which maintenance of generating unit  $i \in \mathcal{I}$  may start. Since maintenance is allowed only once during a time window, the maintenance window constraint set may be formulated as

$$\sum_{j=e_i}^{\ell_i} x_{ij} = 1, \quad i \in \mathcal{I}. \quad (1)$$

It is known that a unit will not be in maintenance outside its maintenance window. Therefore, the explicit constraints

$$x_{ij} = 0, \quad j < e_i \text{ or } j > \ell_i, \quad i \in \mathcal{I}, \quad (2)$$

$$y_{ij} = 0, \quad j < e_i \text{ or } j > \ell_i + d_i - 1, \quad i \in \mathcal{I}, \quad (3)$$

may be included in the model to reduce the number of free decision variables, where  $d_i$  denotes the maintenance duration of generating unit  $i \in \mathcal{I}$ . Since the maintenance of each unit must last for a given duration, the maintenance duration constraint set

$$\sum_{j=e_i}^{\ell_i+d_i-1} y_{ij} = d_i, \quad i \in \mathcal{I}, \quad (4)$$

is included. Since the maintenance of a generating unit must occur over consecutive time periods, a non-stop maintenance constraint set of the form

$$\begin{aligned} y_{ij} - y_{i,j-1} &\leq x_{ij}, \quad i \in \mathcal{I}, \quad j \in \mathcal{J} \setminus \{1\}, \\ y_{i,1} &\leq x_{i,1}, \quad i \in \mathcal{I}, \end{aligned} \quad (5)$$

is also included.

The load demand constraints restrict the maintenance schedule so that the total demand for electricity is at least met during every time period. Let  $g_{ij}$  denote the power generating capacity of unit  $i \in \mathcal{I}$  during time period  $j \in \mathcal{J}$  and let  $D_j$  denote the load demand during time period  $j \in \mathcal{J}$ . A safety margin, denoted by  $S$ , and measured as a proportion of the demand for the power system, is also

introduced. The load demand constraint set may then be formulated as

$$\sum_{i=1}^n g_{ij}(1 - y_{ij}) = D_j(1 + S) + r_j, \quad j \in \mathcal{J}, \quad (6)$$

where  $r_j$  is the reserve level variable, defined as the unused power during time period  $j \in \mathcal{J}$ , excluding the safety margin.

The maintenance crew constraints deal with the availability of manpower for maintenance work. Let  $m'_{p,i,j}$  denote the required manpower for unit  $i \in \mathcal{I}$  when in maintenance during time period  $j \in \mathcal{J}$  if maintenance of this unit were to commence during time period  $p$ . If  $m_i^k$  denotes the required manpower for unit  $i \in \mathcal{I}$  in its  $k$ th period of maintenance, the parameters  $m'_{p,i,j}$  are calculated as

$$m'_{p,i,j} = \begin{cases} m_i^{j-p+1} & \text{if } j - p < d_i, \\ 0 & \text{otherwise.} \end{cases}$$

The maintenance crew constraint set may be formulated as

$$\sum_{i=1}^n \sum_{p=1}^j m'_{p,i,j} x_{i,p} \leq M_j, \quad j \in \mathcal{J}, \quad (7)$$

where  $M_j$  denotes the available manpower during time period  $j \in \mathcal{J}$ .

Exclusion constraints prevent certain units from being in a state of simultaneous maintenance. Consider a more general exclusion constraint where at most some specified number of units, within some subset of units, are allowed to be in a state of simultaneous maintenance. Let  $\mathcal{K}$  denote the set of indices of generating unit exclusion subsets. If there are  $K$  subsets, then  $\mathcal{K} = \{1, \dots, K\}$ . Define  $\mathcal{I}_k \subseteq \mathcal{I}$  as the  $k$ th subset of generating units that form an exclusion set with  $k \in \mathcal{K}$ . The exclusion constraint set may be formulated as

$$\sum_{i \in \mathcal{I}_k} y_{ij} \leq K_k, \quad j \in \mathcal{J}, \quad k \in \mathcal{K}, \quad (8)$$

where  $K_k$  denotes the maximum number of units within subset  $\mathcal{I}_k$  that are allowed to be in simultaneous maintenance during any time period.

Finally, the constraint sets that specify the nature of the variables are

$$x_{ij} \in \{0, 1\}, \quad i \in \mathcal{I}, \quad j \in \mathcal{J}, \quad (9)$$

$$y_{ij} \in \{0, 1\}, \quad i \in \mathcal{I}, \quad j \in \mathcal{J}, \quad (10)$$

$$r_j \geq 0, \quad j \in \mathcal{J}. \quad (11)$$

The objective, namely to minimise the sum of the squares of the reserve loads, may be written as

$$\text{minimise } \sum_{j=1}^m (D_j S + r_j)^2, \quad (12)$$

subject to the constraint sets in Eqs. (1)–(11). The model Eqs. (1)–(12) is a mixed integer quadratic program formulation of the GMS problem.

### 3. Simulated annealing solution approach

An SA solution approach is adopted due to its ease of implementation, its observed capability of producing good quality solutions for a wide variety of combinatorial optimisation problems [29], the proven existence of a theoretical cooling schedule that guarantees convergence to a global optimum [30], and its previously successful application to the GMS problem in the literature, as mentioned in Section 1. However, as mentioned before, we could

find no reference to experimentation with respect to improving the basic SA algorithm's effectiveness in the GMS context and thus identified an opportunity for the research reported in this paper. In this section, some information on the basic SA algorithm is presented, followed by the specific SA implementation adopted in this paper.

#### 3.1. Introduction to simulated annealing

The SA method is a metaheuristic technique for solving combinatorial optimisation problems and is based on the physical phenomenon of annealing. It was first proposed by Kirkpatrick et al. [31] in 1983. The method solves a combinatorial optimisation problem in a manner that is analogous to the process of annealing and is based on two results from statistical physics, namely the probability of a system having a given energy  $E$  at thermodynamic balance, and the so-called Metropolis algorithm which may be used to simulate the evolution of a system towards thermodynamic balance at a given temperature. A control parameter is introduced to mimic the temperature of a system. The temperature controls the number of accessible energy states and should lead to a locally/globally optimal state when lowered gradually. The energy in the system corresponds to the objective function value in a minimisation problem, while a feasible solution corresponds to a certain state of the system. The final solution corresponds to the system being frozen in its ground state.

The SA method starts with an initial solution at an initial temperature  $T$ . A small modification is applied to the solution (i.e. a neighbouring solution is selected according to some neighbourhood move operator). If the modification results in a decrease in objective function value (energy), the modified solution is accepted as the new solution with probability 1 (i.e. with certainty). However, a modification causing an increase  $\Delta E$  in objective function value (energy) is only accepted with a probability of  $\exp(-\Delta E/T)$ . By allowing an occasional increase in objective function value, the system may avoid becoming trapped in local minima. Repeated iterations of this modification process (the Metropolis algorithm) leads to the system approaching thermodynamic balance at a given temperature. The temperature determines how many worsening solutions are accepted: at high temperatures, the factor  $\exp(-\Delta E/T)$  is close to 1, causing an acceptance of the majority of solutions, whereas lower temperatures result in the factor  $\exp(-\Delta E/T)$  being close to 0, causing a rejection of the majority of worsening solutions. Therefore, the SA method should start at a high temperature in order to consider as many solutions as possible in a bid to explore the solution space, after which the temperature is gradually lowered according to a cooling schedule in order to converge to a solution achieving a locally (possibly globally) minimum objective function value (frozen energy state). There are different approaches in SA with respect to choosing an initial temperature, cooling schedule, neighbourhood move operator and termination criteria.

#### 3.2. Implementation of the simulated annealing algorithm

In the implementation of the SA algorithm adopted in this paper, a solution to the GMS problem is denoted by a vector  $\mathbf{x} = (x_1, \dots, x_n)$  of length  $n$  where the element  $x_i$  is an integer value representing the time period during which the maintenance of unit  $i \in \mathcal{I}$  commences. If a candidate solution violates any of the constraints, a corresponding penalty is incurred and the total penalty value is added to the objective function value associated with the candidate solution. The total penalty value  $P$  is calculated as the weighted sum

$$P = w_w P_w + w_\ell P_\ell + w_c P_c + w_e P_e, \quad (13)$$

where the values  $P_w$ ,  $P_\ell$ ,  $P_c$  and  $P_e$  are the constraint violations associated with the maintenance window, load demand, maintenance

crew and exclusion constraint sets, respectively, and the values  $w_w$ ,  $w_e$ ,  $w_c$  and  $w_e$  the corresponding weights.

### 3.2.1. Initialisation

The initial solution is determined as follows. For each unit  $i$ , a random maintenance starting time period  $x_i$  is chosen between its earliest ( $e_i$ ) and latest ( $l_i$ ) starting time periods, according to a uniform distribution. The feasibility of this random solution is determined by calculating the constraint violations from the vector  $\mathbf{x}$ . If it is feasible, the penalty is set to zero, otherwise, the penalty is calculated according to (13).

The initial temperature  $T_0$  is calculated according to a method presented in [32] as  $T_0 = -\overline{\Delta E}^{(+)} / \ln(\chi_0)$ , where  $\chi_0$  is the initial acceptance ratio and  $\overline{\Delta E}^{(+)}$  is the average increase in energy (worsening of the objective function value). The ratio  $\chi_0$  is defined as the number of accepted worsening solutions divided by the number of attempted worsening solutions and may typically be set to a value of 0.5, whereas the value of  $\overline{\Delta E}^{(+)}$  is estimated by executing a random walk over the solution space, using the initial solution as starting point.

### 3.2.2. The cooling schedule

The only cooling schedule found in the literature that has been used in the GMS context, is the well-known geometric cooling schedule. The updating rule for this schedule is

$$T_{s+1} = \alpha T_s, \quad (14)$$

where  $T_s$  is the temperature at stage  $s$  of the search process and  $\alpha \in (0, 1)$  is a constant called the cooling parameter, typically taken between 0.8 and 0.99. This cooling schedule is implemented here along with three additional adaptive cooling schedules. The first of these three schedules was proposed by Huang et al. [33] and the updating rule is given by

$$T_{s+1} = T_s \exp\left(-\frac{\lambda T_s}{\sigma_s}\right), \quad (15)$$

where  $\lambda \in (0, 1]$  is a constant with a typical value of 0.7 and  $\sigma_s$  is the standard deviation observed in the changing values of the objective function when reaching stage  $s$ . The second schedule was proposed by Van Laarhoven and Aarts [34] with an updating rule of

$$T_{s+1} = T_s \frac{1}{1 + \frac{\ln(1+\delta)}{3\sigma_s} T_s}, \quad (16)$$

where  $\delta$  is a “small” real number. Finally, the third schedule was proposed by Triki et al. [32] and the updating rule is given by

$$T_{s+1} = T_s \left(1 - T_s \frac{\Delta}{\sigma_s^2}\right), \quad (17)$$

where  $\Delta$  is the expected decrease in the average objective function value when reaching the next temperature stage of the search process. For details on the workings of and motivations behind these cooling schedules, the reader is referred to [32].

During each temperature stage in the progression of the SA algorithm, the number of iterations of the Metropolis algorithm determines the time spent at that temperature. The suggested scheme presented in [29] is implemented here and states that the inner Metropolis loop should terminate when one of the following two conditions is satisfied: a maximum of  $12N$  solutions are accepted, or a maximum of  $100N$  solutions are attempted, where  $N$  denotes the number of degrees of freedom of the problem. In this case  $N = n$ .

### 3.2.3. The neighbourhood move operator

Only two neighbourhood move operators were found in the GMS literature, the one being a simplification of the other. Accord-

ing to the first of these move operators, hereafter referred to as the *classical operator*, one unit is randomly selected according to a uniform distribution and its maintenance starting time is then randomly changed to a new value within the allowed maintenance window according to a uniform distribution. The classical operator is implemented in the SA algorithm along with a new neighbourhood move operator in the GMS context, known as an *ejection chain* neighbourhood move operator. This operator includes more global information on the entire maintenance schedule in order to explore the solution space more effectively.

The ejection chain operator generates a list of units whose maintenance starting times are randomly altered with the property that adjacent units in the list are connected in such a way that the preceding unit's new maintenance starting time is the same as the succeeding unit's old maintenance starting time. The list is created as follows and where any reference to a random selection is made, it is assumed to be performed according to a uniform distribution. An initial unit is selected at random and its maintenance starting time is randomly changed to a new starting time within its allowed maintenance window. Now, a unit whose maintenance starts during this newly selected time is chosen at random, and its maintenance starting time is randomly changed to a new starting time within its allowed maintenance window. This process is repeated until the newly selected starting time corresponds to the initial maintenance starting time of the initial unit that was selected, or the process is repeated until no unit is found for which maintenance starts during the newly selected time.

### 3.2.4. Termination criteria

The SA algorithm terminates when the temperature loop terminates according to pre-specified criteria. The following two termination criteria are implemented here: the temperature at the current stage reaches a pre-specified minimum temperature  $T_{min}$ , or a pre-specified number,  $\Omega_{frozen}$ , of successive temperature stages occur without the occurrence of any acceptance. A modification to the standard SA algorithm is implemented whereby the best solution found so far, called the *incumbent* solution, is stored. On completion of the SA algorithm, this incumbent solution is returned as an approximate solution to the GMS problem instance.

### 3.2.5. Hybridisation by means of a local search heuristic

A hybridisation of the SA algorithm is achieved by introducing a local search heuristic into the algorithm. The implementation of the local search heuristic adopts the classical neighbourhood move operator. The heuristic receives some solution as an initial solution and its full neighbourhood (with a maximum size of  $nm$ ) is searched in order to find the best neighbour. If the best neighbour improves the current solution, it is set as the new current solution and the process is repeated. The search terminates if no further improvement can be made (i.e. the search follows a steepest descent hill climbing approach).

In the hybridisation, the heuristic is applied to the incumbent solution each time a new incumbent solution is encountered during the SA algorithm's execution. Only the incumbent solution is updated by means of the heuristic, the current solution remains unaffected in order to prevent premature convergence. The hybrid SA algorithm is compared to the standard SA algorithm in order to investigate its effectiveness in improving the solution quality.

## 4. Experimental results

Two GMS test systems were used as benchmarks in order to investigate the effectiveness of the different cooling schedules, the new ejection chain neighbourhood move operator and the hybridisation. An extensive parameter optimisation process was



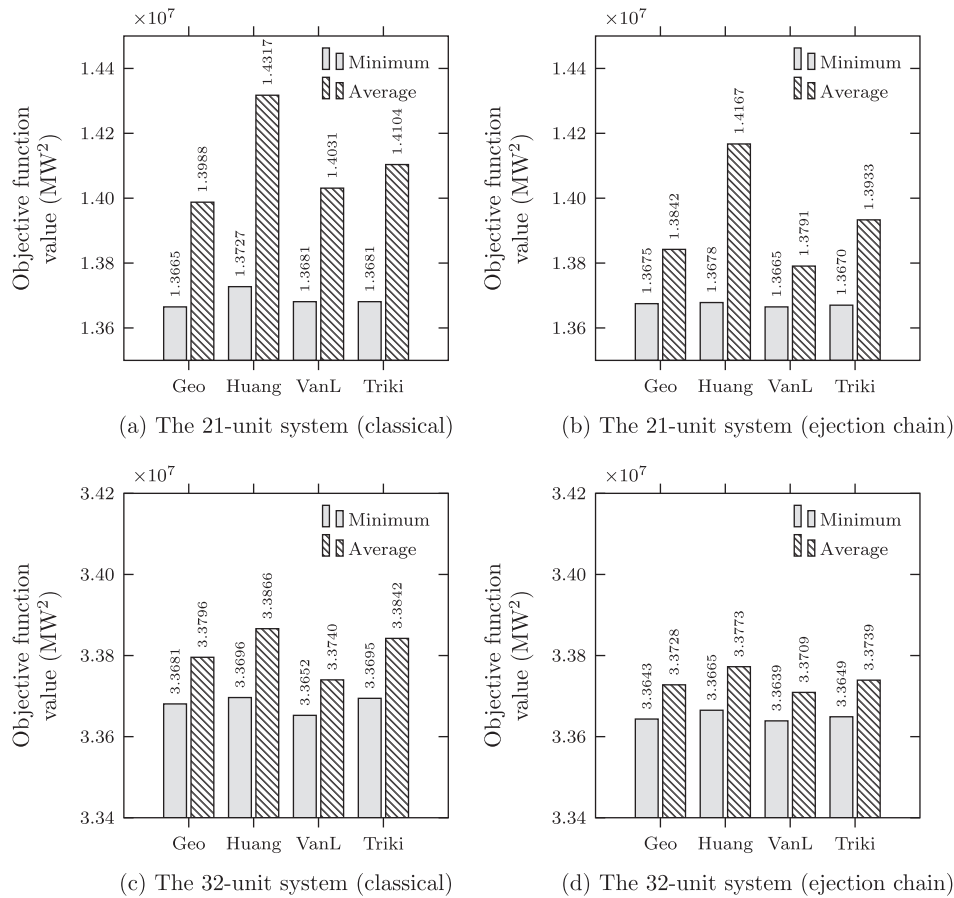


Fig. 1. Comparison of the cooling schedules and neighbourhood operators via incumbent objective function values.

performed in order to find the most suitable values for each of the different cooling schedule parameters for each test system [28]. The penalty weight values of the constraints for each of the test systems were calculated empirically, prior to the comparative study, in order to have equally scaled penalty values in the objective function [28]. A total of 50 computational runs of each test system was solved by each variation of the SA algorithm (i.e. using different random initial solutions). Note that the same set of 50 random initial solutions was used for each SA variation and the set is available from the authors, on request.

All the computational evaluations were performed on a personal computer with a 3.0 GHz Intel® Core™2 Duo E8400 processor and 3.25 GB RAM, running on Microsoft Windows XP Professional. The SA algorithm was implemented in the MathWorks software suite MATLAB, version R2009a.

#### 4.1. The test systems

The first test system is the 21-unit GMS benchmark established by Dahal and McDonald [35] in which reliability is assumed as the optimality criterion. The objective is to minimise the sum of the squares of the reserve loads over the planning period, while the constraints of the problem are restricted to the adherence of maintenance windows for each unit, the system meeting the load demand and the availability of maintenance crew. The test system is highly constrained by its maintenance crew constraint set. A theoretical lower bound for the objective function value is 11 861 100 MW<sup>2</sup>, as calculated from the average reserve level. This test system has previously been studied in the literature [4,5,23,26] and its system specifications may be found in [5].

In order to further investigate the efficiency and effectiveness of the SA algorithm and its variations, a new 32-unit GMS test system containing all the constraint sets in our model formulation was created. This test system may serve as a new GMS benchmark for use by researchers and was derived from the load model and generation system in the 1979 IEEE Reliability Test System [36] with additional constraints and parameter values. The objective is also to minimise the sum of the squares of the reserve loads over the planning period, while the constraints of the system comprise the specification of maintenance windows, the meeting of the load demand together with a safety margin, adhering to the availability of maintenance crew and respecting exclusion constraints. The test system is highly constrained by both its load demand and maintenance crew constraint sets. A theoretical lower bound for the objective function value is 33 363 252 MW<sup>2</sup>. The test system specifications may be found in the appendix at the end of the paper.

#### 4.2. The cooling schedules

The four cooling schedules described in Section 3.2.2, were compared in order to determine which schedule has the potential of performing best in the GMS context. The cooling schedules may be compared over four experimental cases – the two test systems mentioned above, each with two neighbourhood structures (the classical and ejection chain operators). The results of the experiments are shown in Fig. 1. The graphs in the figure represent the minimum and average incumbent objective function values obtained for each test system over all four cooling schedules, for each neighbourhood structure. Table 1 contains the average solution

**Table 1**

Comparison of the cooling schedules and neighbourhood operators via average solution times.

System	Schedule	Classical		Ejection chain	
		Average time (s)	Standard deviation	Average time (s)	Standard deviation
21-unit	Geo	48.35	0.19	59.1	7.13
	Huang	2.98	0.33	7.19	0.76
	VanL	23.84	1.52	66.74	4.66
	Triki	17.61	24.93	59.79	119.97
32-unit	Geo	78.95	3.49	131.9	9.45
	Huang	13.78	1.19	37.3	2.72
	VanL	119.89	6.74	151.25	7.3
	Triki	286.01	1008.9	170.2	187.35

**Table 2**

Performance analysis of the hybridisation with respect to the standard SA algorithm.

System	Schedule	Classical neighbourhood			Ejection chain neighbourhood		
		# Solutions improved	Average improvement (%)	Maximum improvement (%)	# Solutions improved	Average improvement (%)	Maximum improvement (%)
21-unit	Geo	19/50	0.49	1.27	9/50	0.50	1.48
	Huang	23/50	1.25	3.94	31/50	0.82	3.49
	VanL	27/50	0.73	2.59	16/50	0.43	1.39
	Triki	21/50	0.80	2.86	22/50	0.87	2.90
32-unit	Geo	27/50	0.03	0.22	50/50	0.02	0.08
	Huang	28/50	0.05	0.42	50/50	0.05	0.34
	VanL	41/50	0.03	0.18	50/50	0.03	0.16
	Triki	29/50	0.09	0.75	49/50	0.03	0.23

times (and standard deviations) required to achieve the objective function values in Fig. 1.

The cooling schedule proposed by Huang et al. [33] performs the worst in each test system for the minimum, as well as for the average incumbent objective function values. However, in each experimental case, the minimum objective function value is very close to those of the other schedules (within 1% for both test systems). The schedule achieves by far the best (fastest) average solution time in all cases at speeds approximately 10 times faster than that of the other schedules. Therefore, the schedule proposed by Huang et al. [33] may be favourable when a quick, relatively good solution is required for a GMS problem instance. However, in this case it is advisable to solve the instance a number of times. The cooling schedule proposed by Triki et al. [32] results in an average solution time with a very large standard deviation, thereby having the potential of highly fluctuating solutions times – a phenomenon best avoided. Furthermore, its minimum and average incumbent objective function value levels are second to worst. Due to its unpredictable (and potentially long) solution times and not achieving any real solution quality advantage above the other cooling schedules, the schedule proposed by Triki et al. [32] is not recommended as a cooling schedule in the GMS context.

The cooling schedule proposed by Van Laarhoven and Aarts [34] achieves the best solution quality over all the test systems – in three of the four cases it attains the lowest minimum and average incumbent objective function values, and it achieves very consistent solution times (according to the standard deviations). However, a drawback is its longer average solution time, requiring slightly more time than the geometric cooling schedule. The geometric schedule achieves the second to best solution quality – it attains the lowest minimum and average incumbent objective function values in one of the cases. Its solution times are also very consistent. Therefore, the schedule proposed by Van Laarhoven and Aarts [34] is concluded to be the most favourable schedule when solving a GMS problem instance that is not limited by stringent time constraints. It produces superior solution quality to that of

the geometric schedule within a computational time being of the same order of magnitude.

#### 4.3. The new neighbourhood move operator

A similar analysis to the one above was performed in order to compare the new ejection chain neighbourhood move operator to the classical operator. The minimum and average incumbent objective function values, as well as average solution times obtained using the four coolings schedules, for each of the two test systems, are considered for the comparison – thus resulting in eight experimental cases. The results of the experiments are also contained in Fig. 1 and Table 1.

The ejection chain operator performs superior to the classical operator in seven of the eight cases, when considering the minimum incumbent objective function value, and in all eight cases when considering the average incumbent objective function value. The solution time of the ejection chain operator will necessarily be longer than that of the classical operator, as reflected by the average times in Table 1 when ignoring the fluctuating behaviour of the cooling schedule proposed by Triki et al. [32]. The difference between the average solution times of the two neighbourhood structures range between 22% and 180%. These results clearly illustrate the superiority in solution quality of the ejection chain neighbourhood move operator over that of the classical neighbourhood operator; however, at the cost of potentially requiring significantly more solution time.

#### 4.4. The hybridisation

The results obtained by the hybridisation were compared to the results obtained by the unmodified SA algorithm for each cooling schedule, within each test system, using both neighbourhood structures, thus yielding 16 experimental cases. Recall that 50 computational runs were performed on each test system using each solution variation. The results of the experiments are

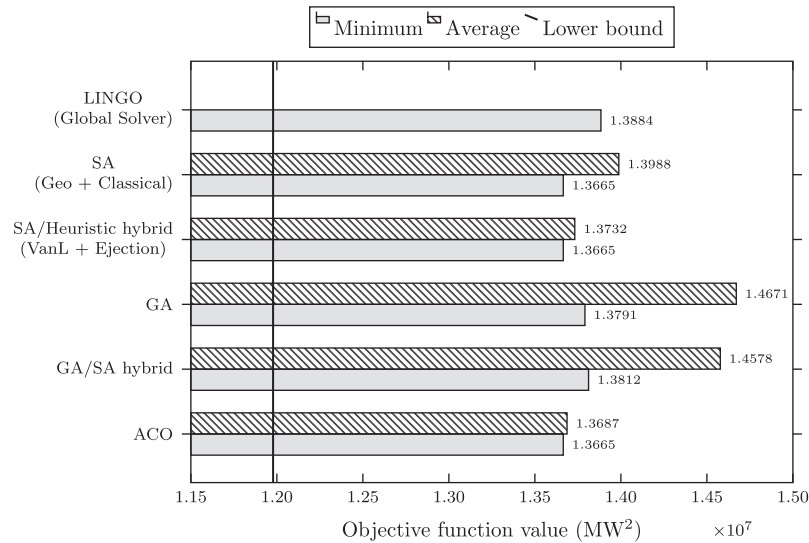


Fig. 2. Comparison of results.

presented in Table 2 and indicate that the hybridisation has the potential of achieving a significant improvement over the SA algorithm. This may be seen in the significant number of final incumbent solutions, uncovered by both neighbourhood operators in both test systems, that were improved upon by the hybridisation. In 10 of the 16 cases more than half of the incumbent solutions were improved upon. While the average improvement may not seem significant, individual improvements are up to 3.94% better. Since the hybridisation achieves such a frequent improvement in incumbent solutions, it may be included in the SA algorithm as an efficient method for improving solution quality at a negligible cost of increased computational time.

#### 4.5. Comparison with other methods

Given the positive results obtained by the investigation into different cooling schedules, a different neighbourhood move operator, and a hybridisation, we used the hybrid SA algorithm utilising the ejection chain neighbourhood move operator and the cooling schedule proposed by Van Laarhoven and Aarts [34] in order to compare our results with those of other methods in the literature. Furthermore, we may also compare our (and the literature's) approximate solution approaches with an exact solution approach. We implemented the GMS model in LINDO Systems' optimisation software suite, *LINGO*, version 9.0 with a cut-off time set to 12 h.

The objective function values corresponding to the best solutions that we found for the 21-unit system via LINGO after 12 h, via the standard SA algorithm (geometric cooling and classical neighbourhood operator) and via the hybrid SA algorithm described above are presented in Fig. 2. These values are compared to those of the best solutions found in the literature via a genetic algorithm (GA) and a genetic algorithm/simulated annealing hybrid (GA/SA hybrid) by Dahal and Chakpitak [5], and via an ant colony optimisation (ACO) algorithm by Foong [37]. Note that the results obtained by Foong [37] were the best known results for the 21-unit system to date. Clearly, our hybrid SA algorithm outperforms the GA and GA/SA hybrid algorithms and we were able to match the best known objective function value of 13 665 000 MW<sup>2</sup>; however, we were unable to match/improve the best known average incumbent solution quality. All the approximate solution approach methods were able to obtain a better solution than the local optimum obtained by LINGO after 12 h,

thereby validating the use of approximate solution approaches to solve the GMS problem. It also demonstrates the difficulty of solving a GMS problem exactly within a reasonable computational time-scale. Furthermore, an improved lower bound (as calculated by LINGO after 12 h) on the objective function value for the 21-unit system is presented here for the first time. The value of this improved lower bound is 11 977 600 MW<sup>2</sup> and indicated in the graph by the vertical line.

Since the 32-unit system is new (presented for the first time in this paper), the best results that we obtained are stated here. LINGO was able to obtain a best objective function value of 33 904 230 MW<sup>2</sup>, while a lower bound was calculated as 33 479 440 MW<sup>2</sup>. Our hybrid SA algorithm obtained a best objective function value of 33 627 292 MW<sup>2</sup> (only 0.44% away from the lower bound) and an average objective function value of 33 699 566 MW<sup>2</sup>.

#### 5. Conclusion

A general model for the GMS problem was presented in this paper in the form of a mixed integer program with reliability as objective, subject to the constraints of maintenance windows, the meeting of the load demand together with a safety margin, the adherence to the availability of maintenance crew, and respecting exclusion constraints. The effectiveness of a simulated annealing solution approach towards solving the GMS problem was investigated. A comparative study was performed, comprising four different cooling schedules (three were newly introduced in the GMS context), two neighbourhood move operators (one was newly introduced in the GMS context), and a hybrid local search heuristic/SA algorithm. It was found that the standard implementation of the SA algorithm in the GMS context, comprising the geometric cooling schedule and an elementary neighbourhood move operator, may be improved by adopting the schedule proposed by Van Laarhoven and Aarts [34] and/or adopting an ejection chain neighbourhood move operator. Furthermore, in situations where computational time may be limited, the schedule proposed by Huang et al. [33] may be adopted to obtain a runtime improvement of approximately 10 times with similar minimum incumbent solution quality. Very large GMS problems would particularly benefit from this variation. An additional benefit in solution quality may also be achieved by introducing a local search heuristic to form the SA hybrid algorithm described in this paper. The SA hybrid

with improved cooling schedule and neighbourhood move operator performed very well when compared to other techniques with respect to the 21-unit GMS benchmark system in the literature. It outperformed a genetic algorithm and genetic algorithm/simulated annealing hybrid, and matched the best known solution present in the literature, obtained via an ant colony optimisation algorithm. An improved lower bound for the benchmark was also established via the LINGO software suite. Finally, a new 32-unit GMS test system was established which may aid GMS research as a future benchmark.

## Appendix A. The new 32-unit test system

A total of  $n = 32$  generating units have to be in maintenance over a planning period of  $m = 52$  weeks. The objective of the problem is to minimise the sum of the squares of the reserve loads over the planning period. The specifications of the generation system are presented in Tables A.1 and A.2. The generating capacity of

each unit remains constant over the planning period, thus  $g_{ij} = g_i$  and a maximum of  $M_j = 25$  maintenance personnel are available for maintenance work during each week. The weekly peak load demands of the power system are presented in Table A.3 and a safety margin of  $S = 15\%$  has to be maintained throughout the planning period. The daily, as well as hourly, peak load demands of the power system may be found in the original 1979 IEEE RTS system [36], if required.

## References

- [1] Burke EK, Clark JA, Smith AJ. Four methods for maintenance scheduling. In: Int conf artif neural netw and gen algorithms. New York (NY): Springer; 1998. p. 264–9.
- [2] Burke EK, Smith AJ. Hybrid evolutionary techniques for the maintenance scheduling problem. IEEE Trans Power Syst 2000;15(1):122–8.
- [3] Chen S, Huang K, Huang Y. A new approach for generator maintenance scheduling in deregulated power systems. In: Int conf innov comput inf and control (ICICIC '08). Dalian: IEEE; 2008.
- [4] Dahal KP, McDonald JR, Burt GM. Modern heuristic techniques for scheduling generator maintenance in power systems. Trans Inst Meas Control 2000;22(2):179–94.
- [5] Dahal KP, Chakpitak N. Generator maintenance scheduling in power systems using metaheuristic-based hybrid approaches. Electr Power Syst Res 2007;77:771–9.
- [6] Saraiva JT, Pereira ML, Mendes VT, Sousa JC. A simulated annealing based approach to solve the generator maintenance scheduling problem. Electr Power Syst Res 2011;81:1283–91.
- [7] Satoh T, Nara K. Maintenance scheduling by using simulated annealing method. IEEE Trans Power Syst 1991;6:850–7.
- [8] Kralj BL, Petrović R. Optimal preventive maintenance scheduling of thermal generating units in power systems – a survey of problem formulations and solution methods. Eur J Oper Res 1988;35:1–15.
- [9] Canto SP. Application of Benders' decomposition to power plant preventive maintenance scheduling. Eur J Oper Res 2008;184:759–77.
- [10] Kuzle I, Pandzic H, Brezovec M. Implementation of the benders decomposition in hydro generating units maintenance scheduling. In: Hydro 2007 conf: new approaches for a new era. Cairns; 2007.
- [11] Kim JH, Park JB, Park JK, Chun YH. Generating unit maintenance scheduling under competitive market environments. Electr Power Energy Syst 2005;27:189–94.
- [12] Foong WK, Simpson AR, Maier HR, Stolp S. Ant colony optimization for power plant maintenance scheduling optimization – a five-station hydropower system. Ann Oper Res 2008;159:433–50.
- [13] Kralj B, Petrović R. A multiobjective optimization approach to thermal generating units maintenance scheduling. Eur J Oper Res 1995;84:481–93.
- [14] Leou RC. A new method for unit maintenance scheduling considering reliability and operation expense. Electr Power Energy Syst 2006;28:471–81.
- [15] Huang CJ, Lin CE, Huang CL. Fuzzy approach for generator maintenance scheduling. Electr Power Syst Res 1992;24:31–8.
- [16] Moro LM, Ramos A. Goal programming approach to maintenance scheduling of generating units in large scale power syst. IEEE Trans Power Syst 1999;14(3):1021–8.
- [17] Ekpenyong UE, Zhang J, Xia X. An improved robust model for generator maintenance scheduling. Electr Power Syst Res 2012;92:29–36.
- [18] Chakraborty S, Ito T, Senjyu T, Saber AY. Unit commitment strategy of thermal generators by using advanced fuzzy controlled binary particle swarm optimization algorithm. Electr Power Energy Syst 2012;43:1072–80.
- [19] Saber AY. Economic dispatch using particle swarm optimization with bacterial foraging effect. Electr Power Energy Syst 2012;34:38–46.
- [20] Tehzeeb-ul-Hassan H, Ahmad A. Profit based unit commitment and economic dispatch of IPPs with new technique. Electr Power Energy Syst 2013;44:880–8.
- [21] Volkanovski A, Mavko B, Boševski T, Čauševski A, Čepin M. Genetic algorithm optimisation of the maintenance scheduling of generating units in a power system. Reliab Eng Syst Saf 2008;93:757–67.
- [22] El-Amin I, Duffuaa S, Abbas M. A tabu search algorithm for maintenance scheduling of generating units. Electr Power Syst Res 2000;54:91–9.
- [23] Foong WK, Maier HR, Simpson AR. Ant colony optimization for power plant maintenance scheduling optimization. In: Int conf gen and evol comput (GECCO '05). New York (NY): ACM; 2005. p. 249–56.
- [24] Yare Y, Venayagamoorthy GK, Aliyu UO. Optimal generator maintenance scheduling using a modified discrete PSO. IET Gener Transm Distrib 2008;2(6):834–46.
- [25] Reihani E, Sarikhani A, Davodi M, Davodi M. Reliability based generator maintenance scheduling using hybrid evolutionary approach. Electr Power Energy Syst 2012;42:434–9.
- [26] Dahal KP, Aldridge CJ, McDonald JR. Generator maintenance scheduling using a genetic algorithm with a fuzzy evaluation function. Fuzzy Sets Syst 1999;102:21–9.
- [27] Lin CE, Huang CJ, Huang CL, Liang CC, Lee SY. An expert system for generator maintenance scheduling using operation index. IEEE Trans Power Syst 1992;7(3):1141–8.

**Table A.1**

Data for the new 32-unit test system.

$i$	$g_i$	$e_i$	$\ell_i$	$d_i$	$m_i^k$	$i$	$g_i$	$e_i$	$\ell_i$	$d_i$	$m_i^k$
1	20	1	25	2	7, 7	17	12	1	51	2	4, 4
2	20	1	25	2	7, 7	18	12	1	51	2	4, 4
3	76	1	24	3	12, 10, 10	19	12	1	51	2	4, 4
4	76	27	50	3	12, 10, 10	20	155	1	23	4	5, 15, 10, 10
5	20	1	25	2	7, 7	21	155	27	49	4	5, 15, 10, 10
6	20	27	51	2	7, 7	22	400	1	21	6	15, 10, 10, 10, 5
7	76	1	24	3	12, 10, 10	23	400	27	47	6	15, 10, 10, 10, 5
8	76	27	50	3	12, 10, 10	24	50	1	51	2	6, 6
9	100	1	50	3	10, 10, 15	25	50	1	51	2	6, 6
10	100	1	50	3	10, 10, 15	26	50	1	51	2	6, 6
11	100	1	50	3	15, 10, 10	27	50	1	51	2	6, 6
12	197	1	23	4	8, 10, 10, 8	28	50	1	51	2	6, 6
13	197	1	23	4	8, 10, 10, 8	29	50	1	51	2	6, 6
14	197	27	49	4	8, 10, 10, 8	30	155	1	23	4	12, 12, 8, 8
15	12	1	51	2	4, 4	31	155	1	49	4	12, 12, 8, 8
16	12	1	51	2	4, 4	32	350	1	48	5	5, 10, 15, 15, 5

**Table A.2**

Exclusion data for the new 32-unit test system.

Exclusion set $k$	Units $i$ within $\mathcal{I}_k$	$K_k$
1	1, 2, 3, 4	2
2	5, 6, 7, 8	2
3	9, 10, 11	1
4	12, 13, 14	1
5	15, 16, 17, 18, 19, 20	3
6	24, 25, 26, 27, 28, 29	3
7	30, 31, 32	1

**Table A.3**

The weekly peak load demands (MW) for the new 32-unit test system.

$j$	$D_j$	$j$	$D_j$	$j$	$D_j$	$j$	$D_j$
1	2457	14	2138	27	2152	40	2063
2	2565	15	2055	28	2326	41	2118
3	2502	16	2280	29	2283	42	2120
4	2377	17	2149	30	2508	43	2280
5	2508	18	2385	31	2058	44	2511
6	2397	19	2480	32	2212	45	2522
7	2371	20	2508	33	2280	46	2591
8	2297	21	2440	34	2078	47	2679
9	2109	22	2311	35	2069	48	2537
10	2100	23	2565	36	2009	49	2685
11	2038	24	2528	37	2223	50	2765
12	2072	25	2554	38	1981	51	2850
13	2006	26	2454	39	2063	52	2713



- [28] Schlünz EB. Decision support for generator maintenance scheduling in the energy sector. Master's thesis, Stellenbosch University, Stellenbosch; 2011.
- [29] Dréo J, Pétrowski A, Siarry P, Taillard E. *Metaheuristics for hard optimization – methods and case studies*. Berlin: Springer-Verlag; 2006.
- [30] Hajek B. Cooling schedules for optimal annealing. *Math Oper Res* 1988;13(2):311–29.
- [31] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *Science* 1983;220(4598):671–80.
- [32] Triki E, Collette Y, Siarry P. A theoretical study on the behaviour of simulated annealing leading to a new cooling schedule. *Eur J Oper Res* 2005;166:77–92.
- [33] Huang MD, Romeo F, Sangiovanni-Vincentelli AL. An efficient general cooling schedule for simulated annealing. In: *IEEE int conf computer-aided design*. Santa Clara (CA): IEEE; 1986. p. 381–4.
- [34] Van Laarhoven PJM, Aarts EHL. *Simulated annealing: theory and applications*. Dordrecht: Reidel; 1987.
- [35] Dahal KP, McDonald JR. A review of generator maintenance scheduling using artificial intelligence techniques. In: *32nd Univ power eng conf (UPEC '97)*. Manchester: University of Manchester; 1997.
- [36] Albrecht PF, Bhavaraju MP, Biggerstaff BE, Billington R, Jorgensen GE, Reppen ND, et al. IEEE reliability test system. *IEEE Trans Power App Syst* 1979;PAS-98(6):2047–54.
- [37] Foong WK. Ant colony optimisation for power plant maintenance scheduling. Ph.D. thesis, The University of Adelaide; Adelaide; 2007.