Ant Colony Optimization for Power Plant Maintenance Scheduling Optimization

Wai Kuan Foong
Centre for Applied Modeling in Water Engineering (CAMWE), School of Civil & Environmental Engineering, The University of Adelaide, Australia 5005.
+61 40 3162457
wfoong@civeng.adelaide.edu.au

Holger R. Maier
Centre for Applied Modeling in Water Engineering (CAMWE), School of Civil & Environmental Engineering, The University of Adelaide, Australia 5005.
+61 8 83034139
hmaier@civeng.adelaide.edu.au

Angus R. Simpson
Centre for Applied Modeling in Water Engineering (CAMWE), School of Civil & Environmental Engineering, The University of Adelaide, Australia 5005
+61 8 83035135
asimpson@civeng.adelaide.edu.au

ABSTRACT
In order to maintain a reliable and economic electric power supply, the maintenance of power plants is becoming increasingly important. In this paper, a formulation that enables ant colony optimization (ACO) algorithms to be applied to the power plant maintenance scheduling optimization (PPMSO) problem is developed and tested on a 21-unit case study. A heuristic formulation is introduced and its effectiveness in solving the problem is investigated. The performance of two different ACO algorithms is compared, including Best Ant System (BAS) and Max-Min Ant System (MMAS), and a detailed sensitivity analysis is conducted on the parameters controlling the searching behavior of ACO algorithms. The results obtained indicate that the performance of the two ACO algorithms investigated is significantly better than that of a number of other metaheuristics, such as genetic algorithms and simulated annealing, which have been applied to the same case study previously. In addition, use of the heuristics significantly improves algorithm performance. Also, ACO is found to have similar performance for the case study considered across an identified range of parameter values.

Categories and Subject Descriptors
1.2.8 [**Artificial Intelligence**]: Problem Solving, Control methods, and Search – *heuristics methods, scheduling.*

1.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *intelligent agents, multiagent systems.*

General Terms
Algorithms; Management; Performance; Experimentation.

Keywords
Ant Colony Optimization; power plant maintenance scheduling; heuristics; BAS; MMAS; GA; SA; sensitivity analysis; optimum parameter.

1. INTRODUCTION
As a result of rapid development, global power demand has increased dramatically over the past decade. In order to achieve a reliable and economic electric power supply, adequate maintenance of power plants is vital. Maintenance is aimed at extending the lifetime of power generating facilities, or at least extending the mean time to the next failure. In addition, an effective maintenance policy can reduce the frequency of service interruptions and their consequences [16].

Over the past two decades, many studies have focused on the development of methods for optimizing maintenance schedules for power plants. Traditionally, mathematical programming approaches have been used, including dynamic programming [22], integer programming [9] and mixed-integer programming [1]. More recently, evolutionary algorithms (EAs) and other metaheuristics have been favored, including genetic algorithms (GAs) [2], simulated annealing (SA) [19] and tabu search (TS) [15]. These methods have been shown to outperform mathematical programming methods and other conventional approaches in terms of the quality of the solutions found, as well as computational efficiency [2, 19].

Ant Colony Optimization is a relatively new metaheuristic that is based on the foraging behaviors of ant colonies [12]. Compared to other optimization methods, such as GA, ACO has been found to produce better solutions in terms of computational efficiency and quality when applied to a number of combinatorial optimization problems, such as the Traveling Salesman Problem (TSP) [10] and De Jong's test function [21]. Recently, ACO has also been successfully applied to scheduling problems such as the resource-constrained project scheduling (RCPSP), single machine tardiness, job-shop and flow-shop problems [3, 6, 18, 20]. Interested readers are referred to [4] for the state-of-the-art categorization of scheduling problems.

The objective of this study is to introduce a formulation that enables ACO to be applied to the power plant maintenance scheduling optimization (PPMSO) problem, including the development of a formulation for the incorporation of heuristic information, which is used as part of the decision policy at each decision point. The proposed formulation is tested on a modified version of the 21-unit problem introduced by Aldridge et al. [2] and the results obtained using ACO are compared with those obtained using Genetic Algorithms (GAs) and Simulated Annealing (SA) in previous studies [2, 7, 8]. Sensitivity analysis is also carried out to gain a better understanding of the role each parameter plays in the optimization process and to identify the 'optimum' parameter set for the problem considered.

2. APPLICATION OF ACO TO POWER PLANT MAINTENANCE SCHEDULING OPTIMIZATION

In the PPMSO problem, decisions have to be made with regard to the timing of the maintenance periods of each of the machines (units) used for power generation. Generally, the duration of the maintenance period for each machine is fixed, and the decision variable is the maintenance start time. The aim of the optimization procedure is to obtain a maintenance schedule that minimizes the objective function subject to a number of constraints. The objectives generally include cost minimization, system reliability maximization, or both [22]. The most commonly used constraints are load constraints, resources constraints and the window of time during which maintenance can be carried out.

It should be noted that the PPMSO problem is different from RCPSPs to which ACO has been applied to recently [18]. This is because in the PPMSO problem, precedence relations between maintenance activities are emphasized much less than the absolute periods within a planning horizon where maintenance task are carried out, which are strongly dependent upon factors such as seasonal inflows (in the case of hydropower), daily demands and the maintenance time frame of individual generating units. Therefore, instead of using the generic schedule generation scheme as outlined in [18], an activity-oriented schedule generation procedure, coupled with a finite number of optional decision paths, is proposed in this paper for the PPMSO problem. In addition, the PPMSO is not necessarily resource-constrained. In many instances, the dominant constraint is load, which cannot be accounted for explicitly, and has to be checked once a complete trial maintenance schedule has been generated with the aid of a simulation model.

Before the PPMSO problem can be optimized using ACO, it has to be expressed in terms of a set of points at which decisions have to be made (**D** = {d_n, where n=1,2,…N}) and the set of options that is available at each decision point (**F** = {$l_{n,j}$, where $d_n \in$ **D**, j=1,2,…, k_n}) [18]. The decision points consist of the N units at which maintenance needs to be carried out and the corresponding decisions are the k_n potential commencement times for maintenance (Figure 2.1).

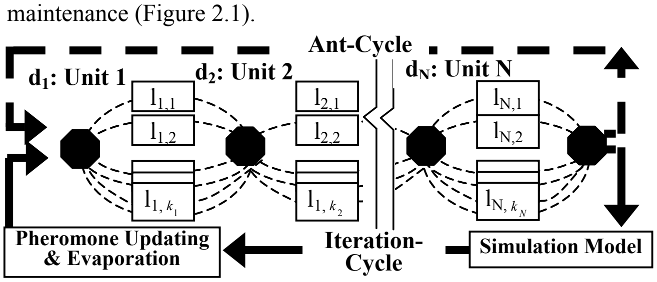

Figure 2.1: ACO algorithm applied to the PPMSO problem

As part of the ACO algorithm, ants generate trial maintenance schedules by choosing a maintenance commencement date for each of the units to be maintained. The probability that a particular commencement date will be chosen from the set of available options at a particular decision point is a function of the pheromone and the local desirability of that option based on heuristic information (generally referred to as the heuristic), as shown in Eq. 2.1.

$$p_{n,j}(t) = \frac{[\tau_{n,j}(t)]^{\alpha} \cdot [\eta_{n,j}]^{\beta}}{\sum_{j=1}^{k_n} [\tau_{n,j}(t)]^{\alpha} \cdot [\eta_{n,j}]^{\beta}} \qquad \text{(Eq. 2.1)}$$

where $p_{n,j}(t)$ is the probability that start time $l_{n,j}$ is chosen for maintenance of unit d_n in iteration t; $\tau_{n,j}(t)$ is the pheromone intensity deposited on start time $l_{n,j}$ for unit d_n in iteration t; $\eta_{n,j}$ is the heuristic for start time $l_{n,j}$ for unit d_n; k_n is the total number of start time periods available for unit d_n; α is the relative importance of pheromone intensity; β is the relative importance of the heuristic.

The pheromone level associated with a particular option (i.e. maintenance commencement date for a particular unit) is a reflection of the quality of the maintenance schedules that have been generated that contain this particular option. The heuristic associated with a particular option is related to the likely quality of a solution that contains this option based on some heuristic information. It can be seen from Eq. 2.1 that during the early stages of an ACO run, before pheromone trails are significantly distinct, heuristic information is the dominant factor affecting the selection of decision paths. In other words, the heuristic plays a crucial role in defining the region in a solution search space in which the ACO algorithm commences its search. As the way in which heuristic information is represented mathematically is problem specific [14], the transformation of any heuristic information into a formulation to be used in the ACO algorithm is an important task.

As ACO has not been previously applied to the PPMSO problem, a heuristic formulation (Eq. 2.2) is introduced for a typical PPMSO problem in this paper. Furthermore, the following variables are defined:

- $J_{n,j} = \{ l_{n,j} \le k \le l_{n,j} + dur_n - 1\}$ is the set of time periods k such that if the maintenance of unit d_n starts at period $l_{n,j}$, that unit will be in maintenance during period k.
- $Y_{ManV(k)=0}$ is switched to 1 if there is no personpower violation in time period k. Otherwise it is switched to 0.
- $Y_{LoadV(k)=0}$ is switched to 1 if there is no load violation in time period k. Otherwise it is switched to 0.

$$\eta_{n,j} = (\eta^{M}_{n,j})^{w1} \cdot (\eta^{C}_{n,j})^{w2} \qquad \text{(Eq. 2.2)}$$

$$\eta^{M}_{n,j} = \frac{\sum_{k \in J_{n,j}} Y_{ManV(k)=0} \cdot M_{n,j}(k)}{\sum_{k \in J_{n,j}} (1 - Y_{ManV(k)=0}) \cdot M_{n,j}(k)}$$

$$\eta^{C}_{n,j} = \frac{\sum_{k \in J_{n,j}} Y_{LoadV(k)=0} \cdot C_{n,j}(k)}{\sum_{k \in J_{n,j}} (1 - Y_{LoadV(k)=0}) \cdot C_{n,j}(k)}$$

where $\eta_{n,j}$ is the heuristic value of unit d_n to start maintenance at time period $l_{n,j}$; dur_n is the outage duration required for unit d_n; $M_{n,j}(k)$ is the prospective personpower available in reserve in time period k if unit d_n is maintained starting at period $l_{n,j}$; $C_{n,j}(k)$ is the prospective generation capacity available in reserve in time period k if unit d_n is maintained starting at period $l_{n,j}$.

It can be seen from Eq. 2.2 that the heuristic formulation comprises personpower-related heuristics, $\eta_{n,j}^M$ and load-related heuristics, $\eta_{n,j}^C$. $\eta_{n,j}^M$ is designed to direct the optimization algorithm to regions in the search space where there are fewer personpower constraint violations. This is achieved mathematically by making the probability of a start time being chosen for any machine unit directly proportional to the prospective personpower available in reserve and inversely proportional to the amount of personpower shortfall. The same applies to $\eta_{n,j}^C$, where start times at which no tasks are scheduled are preferred to avoid violation of load constraints. It should be noted that where personpower and load constraints are easily satisfied inherently in a problem, the two heuristics are expected to evenly distribute maintenance tasks over the entire planning horizon, which potentially maximizes the overall reliability of a power system. In order to implement the heuristic, each ant is provided with a memory matrix on personpower reserve and another matrix on generation capacity reserve prior to construction of a trial solution, which is updated every time a unit maintenance commencement time is added to the partially completed schedule.

Once a trial maintenance schedule has been constructed by choosing a maintenance commencement time at each decision point (i.e. for each machine to be maintained), taking into account pheromone levels and the heuristic information introduced above, one ant-cycle has been completed (Figure 2.1). After r ant-cycles, where r equals the number of ants used, the ACO algorithm enters the iteration-cycle (Figure 2.1). During this stage, the quality of the r trial solutions is evaluated using a simulation model, as part of which the objective function values, such as maintenance cost and power system reliability, are calculated and violations of any constraints are identified. The objective function values (OFVs) of these trial solutions are then determined by an evaluation function, which is the weighted sum of the objective function values and penalty costs associated with constraint violations. It should be noted that some constraint violations can only be identified once a complete trial solution has been constructed, and hence these constraints cannot be accounted for explicitly, necessitating the use of penalty functions.

Next, pheromone is updated in a way that reinforces good solutions. The general form of the pheromone update equation is given by:

$$\tau_{n,j}(t+1) = \rho \cdot \tau_{n,j}(t) + \Delta\tau_{n,j}(t) \qquad \text{(Eq. 2.3)}$$

where $\tau_{n,j}(t+1)$ is the pheromone intensity of decision path $l_{n,j}$ in iteration (t+1); (1-ρ) is the pheromone evaporation rate; $\Delta\tau_{n,j}(t)$ is the pheromone awarded to decision path $l_{n,j}$ in iteration t.

The way the change in pheromone, $\Delta\tau_{n,j}(t)$, is calculated can vary depending on the particular ACO algorithm used. In the Ant System [13], the pheromone associated with all of the decision paths chosen during the r ant cycles is updated upon completion of an iteration. More recently, alternative pheromone updating schemes have been proposed, including the Best-Ant System (BAS) [17], Max-Min Ant System (MMAS) [20], Ant Colony system (ACS) [11] and Elitist-Rank Ant System [5]. In this study, the BAS and MMAS algorithms are adopted due to their superior performance in previous studies [17, 20]. The BAS represents a limit state of the Elitist-Rank Ant System in which only the paths that are chosen by the top-ranking ant are reinforced. This places greater emphasis on exploitation of the search space, resulting in faster convergence. Using the BAS, the change in pheromone from one iteration to the next is given by:

$$\Delta\tau^*_{n,j}(t) = \begin{cases} \dfrac{Q}{OFV_{n,j}(t)} & \text{if } n,j \in \text{best ant} \\ 0 & \text{otherwise} \end{cases} \qquad \text{(Eq. 2.4)}$$

where Q is the reward factor.

MMAS also uses information from the best performing ant in the pheromone updating process (Eq. 2.4), but imposes upper and lower bounds (τ_{max} and τ_{min}) on the pheromone intensities in order to prevent premature convergence and greater exploration of the solution surface. The τ_{max} and τ_{min} values are given by:

$$\tau_{max}(t+1) = \frac{1}{1-\rho} \cdot \frac{Q}{OFV_{best\,ant}(t)} \qquad \text{(Eq. 2.5)}$$

$$\tau_{min}(t+1) = \frac{\tau_{max}(t+1)(1-\sqrt[n]{p_{best}})}{(avg-1)\sqrt[n]{p_{best}}} \qquad \text{(Eq. 2.6)}$$

where p_{best} is the probability that the paths of the current iteration-best-solution, $OFV_{best\ ant}(t)$, will be selected, given that non-iteration best-options have a pheromone level of $\tau_{min}(t)$ and all iteration-best options have a pheromone level of $\tau_{max}(t)$.

The algorithm terminates when either the number of iterations specified is met or a stagnation of the evaluation function value is encountered.

3. CASE STUDY

3.1 Description

The case study considered in this research is the 21-unit power plant maintenance problem investigated by [2], [7] and [8] using a number of metaheuristics. This case study is a modified version of the 21-unit problem introduced by [22], and consists of 21 generating facilities, of which 20 units are thermal and one is hydropower. The system details are listed in Table 3.1. All of the machines are to be scheduled for maintenance in the first or second half of a year's planning horizon, which results in a combinatorial optimization problem with approximately 5.18 x 10^{28} total possible solutions. The objective of the problem is to even out reserve generation capacity over the planning horizon, which can be achieved by minimizing the sum of squares of the reserve (SSR) generation capacity in each week. A single peak load, 4739 MW, and a limit of 20 maintenance staff are used as demand and personpower constraints, respectively.

As mentioned in Section 1, a number of metaheuristics have been applied to this problem. Aldridge et. al [2] used generational (GN) and steady state (SS) Genetic Algorithms (GAs) and found that the GAs outperformed a heuristic method, which schedules maintenance outages in order of decreasing capacity. By coupling GAs with fuzzy logic, which utilizes knowledge-based experience in the problem formulation, Dahal et. al [7] obtained a maintenance schedule that resulted in a better objective value than the best solution given by [2], although this required slight violations of personpower constraints. In another study, Dahal et. al [8] applied Simulated Annealing (SA), a Simple GA and an Inoculated GA to this problem, further highlighting the ability of metaheuristics to outperform more traditional methods used for optimizing power plant maintenance scheduling. The best results obtained by the studies mentioned above are summarized in Section 3.4.3.

3.2 Mathematical Formulation

The specification of this maintenance scheduling optimization problem can be represented by mathematical equations using binary (1-0) variables, which indicate the state of a unit in a given time period. In the case study under consideration, a time period of one week has been adopted. $X_{n,t}$ can be switched to 1 to indicate that unit d_n is scheduled to be maintained during period t. Otherwise, $X_{n,t}$ is switched to a value of 0. Furthermore, the following sets of variables need to be defined:

- $T_n = \{t \in T: ear_n \leq t \leq lat_n - dur_n + 1\}$ for each unit d_n, which is the set of periods when maintenance of unit d_n may start.
- $S_{n,t} = \{k \in T: t - dur_n + 1 \leq k \leq t\}$ is the set of start time periods k, such that if the maintenance of unit d_n starts at period k, that unit will be in maintenance during period t.
- $D_t = \{n: t \in T_n\}$ is the set of units which is considered for maintenance in period t.

where t is the index of periods; T is the set of indices of periods in the planning horizon; d_n is the index of generating units; ear_n is the earliest period for maintenance of unit d_n to begin; lat_n is the latest period for maintenance of unit d_n to end; dur_n is the duration of maintenance for unit d_n.

In the case study considered, the number of units to be maintained, N is 21. Consequently, the set of decision points is given by $\mathbf{D} = \{d_1, d_2, \ldots, d_{21}\}$. In addition, set **F** can be defined such that $\mathbf{F} = \{d_n \in \mathbf{D}, j \in T_n: l_{n,j}\}$. For example, unit 8 is allowed to undergo maintenance within the second half of the year, which must be completed by Week 52 (Table 3.1). Since a maintenance job for this unit takes 6 days, the earliest and latest date for Unit 8 to start its maintenance are Weeks 26 and 47, respectively. Hence, the decision paths associated with decision point d_8 are $\{l_{8,1}=26, l_{8,2}=27, \ldots, l_{8,22}=47\}$.

Mathematically, this optimization problem can be defined as the determination of maintenance schedule(s) such that SSR, which is defined as the sum of square of reserve generation capacity within the planning horizon, is minimized (Eq. 3.1) without violating the personpower and load constraints (Eqs. 3.2 & 3.3).

$$\mathrm{Min}\left\{ SSR = \sum_{t \in T} \left(\sum_{n=1}^{N} P_n - \sum_{n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n - L_t \right)^2 \right\} \quad \text{(Eq. 3.1)}$$

$$\sum_{n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} M_{n,k} \leq 20 \quad \text{(Eq. 3.2)}$$

$$\sum_{n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \leq 4739 \quad \text{(Eq. 3.3)}$$

where L_t is the anticipated load demand for period t; P_n is the generating capacity of unit d_n.; $M_{n,k}$ is the personpower needed by unit d_n at period k.

Upon completion of an ant-cycle, the maintenance schedule generated is assessed by a simulation model (Figure 2.1) that returns an overall quality of the schedule. The quality of a maintenance schedule in this problem is given by an objective function value (OFV), which is a function of the value of SSR and the total violation of both constraints (Eq. 3.4). The calculations of constraint violations are given in Eq. 3.5 to 3.8.

$$OFV = c_R \cdot SSR + c_M \cdot ManVio_{tot} + c_L \cdot LoadVio_{tot} \quad \text{(Eq. 3.4)}$$

where SSR is the sum of squares of reserve generation capacity; c_R is the relative weight of SSR; $ManVio_{tot}$ is the total personpower violation; c_M is the relative weight of personpower violation; $LoadVio_{tot}$ is the total load violation; c_L is the relative weight of the load violation.

Table 3.1: Details of 21-unit system [7]

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

For a proposed maintenance schedule, the total personpower violation, $ManVio_{tot}$, is given by summation of the personpower shortage in all periods within the planning horizon, such that

$$ManVio_{tot} = \sum_{t \in T_{MV}} \left(\sum_{n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} M_{n,k} - AM_t \right) \quad \text{(Eq. 3.5)}$$

where T_{MV} is the periods where personpower constraints are violated (Eq. 3.6).

$$T_{MV} = \left(t: \sum_{n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} M_{n,k} > AM_t \right) \quad \text{(Eq. 3.6)}$$

where AM_t is the available personpower at period t.

The total load violation, $LoadVio_{tot}$ is the summation of load shortfall in all periods within the planning horizon. The calculation of this value may be represented by the following equation.

$$LoadVio_{tot} = \sum_{t \in T_{LV}} \left(\sum_{n} P_n - \sum_{n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \right) \quad \text{(Eq. 3.7)}$$

where T_{LV} is the periods where load constraints are violated, and is given by:

$$T_{LV} = \left(t : \sum_{n} P_n - \sum_{n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n < L_t \right) \quad \text{(Eq. 3.8)}$$

3.3 Analysis Conducted

3.3.1 Assessment of heuristic information and ACO algorithms

The ACO formulation introduced in Section 2 was used to solve the 21-unit case problem. For the heuristic information in Eq. 2.1, Eq. 2.2 was used and the performance obtained when the heuristic information was used was compared with that obtained when no heuristic information was used. For both scenarios, the BAS and MMAS algorithms were applied to the problem. Due to the probabilistic nature of ACO methods, 14 runs with different random starting conditions were conducted for each of the scenarios. A total of 150 iterations, which is equivalent to the construction of 30,000 trial solutions in each run, was chosen as the termination criterion of all runs to provide a basis for direct comparison between the performance of ACO and that of other metaheuristics used for the same case study. It should be noted that the values of the parameters controlling the behaviour of the ACO algorithms used (Table 3.2) were chosen based on preliminary sensitivity analysis. The same parameters were used for all optimisation runs.

Table 3.2 Chosen values of ACO parameters

r	ρ	α	β	c_R	w1
200	0.7	1	1	1	7
p_{best}(MMAS only)	Q	τ_0	c_L	c_M	w2
0.7	5×10^5	1,000	200	1×10^5	1

3.3.2 Sensitivity Analysis

As ACO algorithms rely on a number of user-defined parameters that control their behavior, extensive sensitivity analyses were conducted in 3 stages to study the importance of each parameter and to determine the optimum parameter set for the problem considered. The ACO algorithm that performed best in the optimization runs specified in Section 3.3.1 was used in the sensitivity analysis.

The purpose of stage 1 of the sensitivity analyses was to obtain a detailed understanding of the impact of each of the user-defined ACO parameters on the algorithms' searching behavior. In order to achieve this, each parameter was varied over a specified range (Table 3.3), while all other parameters were kept at their 'standard' values (Table 3.2). The parameters investigated as part of the sensitivity analyses are summarized in Table 3.3. The relative weight of SSR, c_R in Eq. 3.4 was set to 1 as a control.

In Stage 2, a best parameter set for the case study considered was derived based on the understanding of the parameters obtained as part of Stage 1 of the sensitivity analysis. Initially, the parameter that was found to have the biggest impact on algorithm performance in Stage 1 was varied over its prescribed range, while all other parameters were kept at their 'standard' values. The value of the parameter investigated that gave the best performance was chosen, and then sensitivity analysis was conducted on the parameter that had the second biggest impact during Stage 1 and so on. This process was repeated until the best set of all parameters was obtained.

In Stage 3, 20 different random number seeds (different from the 14 random number seeds used in previous runs) were used to solve the problem considered with the best parameter set derived. The results were compared with those obtained using the 'standard' parameter values shown in Table 3.2.

Table 3.3: ACO parameters investigated as part of sensitivity analysis

Parameter	Description	Range investigated
r	Number of ants	100 to 1,000
τ_0	Initial pheromone level	0 to 1×10^4
c_M	Relative weight of personpower constraint violation in OFV (Eq. 3.4)	0 to 1×10^6
c_L	Relative weight of load constraint violation in OFV (Eq. 3.4)	0 to 1,000
p_{best}	Only applicable to MMAS (Eq. 2.6)	0.05 to 1
Q	Reward factor	1 to 1×10^8
ρ	1 - pheromone evaporation rate	0.1 to 1
w1	Relative importance of personpower-related heuristics (Eq. 2.2)	0 to 10
w2	Relative importance of load-related heuristics (Eq. 2.2)	0 to 7
α	Relative importance of pheromone	0 to 3
β	Relative importance of heuristic information (Eq. 2.1)	0 to 2

3.4 Results & Discussion

3.4.1 Heuristic information and ACO algorithms

The results obtained for the heuristic information formulation and ACO algorithms investigated are presented in Tables 3.4 and 3.5 in terms of best, average, worst objective function values (referred to as OFVs hereafter) and the standard deviation of the OFVs. It should be noted that the various statistics were calculated for results with the same ('standard') ACO parameters, but with 14 different random starting positions in objective function space, as described previously.

In order to gain a better understanding of the impact of heuristic information on the searching behavior of the two ACO algorithms, the iteration-best objective function value curve (referred to as IB-OFV curve hereafter) and the iteration-best SSR curve (referred to as IB-SSR curve hereafter) were examined (Figures 3.1 to 3.3). It should be noted that these curves were extracted from the results given by one of the random number seeds used. Theoretically, the gap between the IB-OFV (thicker) and IB-SSR (thinner) curves is the penalty cost incurred due to the violation of load plus personpower constraints. However, due to the nature of the case study considered, the load constraint is always satisfied (refer to Section 3.4.2 for detailed explanation). Hence, the difference between the IB-OFV and IB-SSR curves can be treated as the violation of the personpower constraint only.

Tables 3.4 and 3.5 indicate that, in general, the results obtained using MMAS are better than those obtained using BAS. It is clearly shown in Figure 3.1 that when no heuristic information was used, both ACO algorithms have searched in the regions of the solution space where constraints are severely violated. Despite its convergence later in the optimization to a solution without constraint violations, the solution found by the MMAS algorithm is 'sub-optimal', whereas no feasible solution was found using the BAS algorithm once convergence had occurred. When the heuristic information heuristic introduced in this paper was used, a new best-known solution was found by MMAS at about Iteration 135 (Figure 3.3), where convergence had occurred for BAS at this stage (Figure 3.2). This is due to a mechanism in MMAS that allows exploration of new solutions when convergence has occurred. In addition, the lower and upper bounds on pheromone trails used in MMAS are also designed to avoid premature convergence. On the basis of an unmatched, two-tailed t-test, MMAS is shown to be significantly better, at a significance level of 5%, than BAS with and without heuristic information. Tables 3.4 and 3.5 show that not only better OFVs were found for both ACO algorithms when heuristic information was incorporated, higher consistency (smaller standard deviation) of optimization outcome was also achieved. An unmatched, two-tailed t-test provided statistical proof that the heuristic has a significant influence on finding the 'good' solutions in optimization runs.

Table 3.4: Results given by Best Ant System (BAS)
[% deviation from best-found OFV]

	Best OFV ($x10^5$)	Average OFV ($x10^5$)	Worst OFV ($x10^5$)	Std dev. ($x10^5$)
No heuristics	154.58 [13.12]	163.69 [19.79]	175.40 [28.36]	6.27
With heuristics	136.81 [0.12]	137.60 [0.70]	140.12 [2.54]	0.91

Table 3.5: Results given by Max-Min Ant System (MMAS)
[% deviation from best-found OFV]

	Best OFV ($x10^5$)	Average OFV ($x10^5$)	Worst OFV ($x10^5$)	Std dev. ($x10^5$)
No heuristics	142.46 [4.25]	146.56 [7.25]	151.34 [10.75]	3.1
With heuristics	136.65 [0]	136.87 [0.16]	137.22 [0.42]	0.15

Referring to Table 3.4, the high SSR-value given by the BAS algorithm when no heuristic was used is attributed to penalties due to constraint violation (gap between curves when converged in Figure 3.1). When heuristic information was used, the BAS algorithm converged to a 'feasible' best-found solution (Figure 3.2), where constraints are not violated. This is because in the early stage of optimization, heuristic information has directed the algorithm to search in the regions where fewer constraints are violated (Figures 3.2 & 3.3). Therefore, the effect of heuristic information on the BAS algorithm is more pronounced than that on the MMAS algorithm, as concluded from the results in Tables

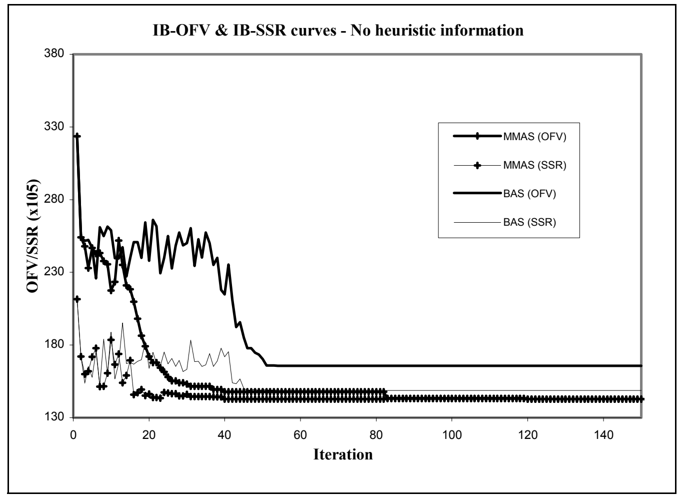

Figure 3.1: Results given by BAS & MMAS without heuristics

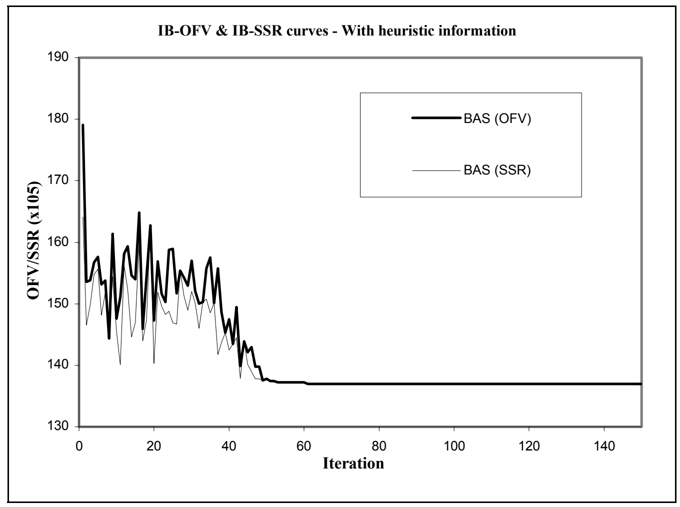

Figure 3.2: Results given by BAS with heuristics

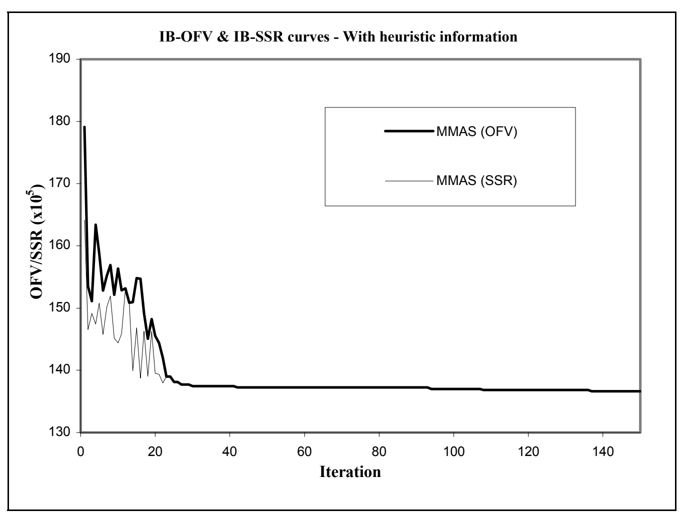

Figure 3.3: Results given by MMAS with heuristics

3.4 & 3.5. The results of the case study clearly demonstrate that heuristic information is vital in finding the good solutions during ACO runs for the PPMSO problem.

Also, Fig 3.1 to 3.3 has clearly demonstrated the ability of pheromone in continuous searching for solutions of low OFVs and hence, determining the optimum/near-optimal solutions for the case study.

3.4.2 Sensitivity Analysis

Stage 1: The MMAS algorithm was used throughout the sensitivity analysis, as it was found to have performed best. Overall, it was found that initial pheromone level, τ_0, reward factor, Q, and relative weight of load violation, c_L in Eq. 3.1 have no impact on the outcome of this optimisation problem.

In the MMAS algorithm, τ_0 has to be set high enough so that all pheromone trails are equivalent to τ_{max} at the end of the first iteration. Hence, the optimization outcome is the same, as long as τ_0 is set to an arbitrarily high value.

The amount of pheromone rewarded upon completion of an iteration is directly proportional to Q (Eq. 2.4). Despite being arbitrary, in BAS, the given rewards must retain a certain ratio in relation to τ_0 on the decision paths so that neither 'exploration' nor 'exploitation' will be overly emphasized. However, this is not necessary in MMAS, as τ_{max}, which acts as the 'effective' initial pheromone, and τ_{min}, are adjusted according to the given reward factor, Q (Eq. 2.5) (τ_{max} and τ_{min} values corresponding to several arbitrary Q-values are given in Table 3.6). In this way, τ_0 and Q are always kept at the same ratio, which resulted in exactly the same solution regardless of the Q-value.

Table 3.6: Distances to best-found best-TMV SSRs' given by different Q-values

Reward factor, Q	$\tau_{max} \cdot OFV_{best\ ant}$	$\tau_{min} \cdot OFV_{best\ ant}$
100	333.333	0.228394
1000	3333.333	2.28394
100000	333333.333	228.394

It was realized in the optimization runs that as load constraints are not critical in this problem, similar results were achieved regardless of the c_L-values used. The total generating capacity of this system is 5,688 MW, with a constant demand of 4,739 MW. Hence, when no machine is scheduled for maintenance, the system reserve capacity is 949 MW. Although 13 out of 21 of the maintenance tasks are required to be scheduled in the first outage window (weeks 1 to 26), most of them represent only a small amount of generating capacity that can be met easily in parallel, provided personpower constraints are satisfied. For the second outage window (weeks 26 to 52), despite the average generating capacity of the units being higher compared to the first outage window, the average maintenance duration of the units is shorter. Therefore, the maintenance tasks can be easily distributed over the optional start weeks without violating demand constraints. It can therefore be deduced that the demand constraint in this problem can be easily satisfied, which is reconfirmed by the identical optimization results regardless of c_L-value. Moreover, the heuristic used favors an even-distribution of maintenance tasks over the planning horizon, which again helped in reducing the possibility of violating demand constraints.

The remaining MMAS parameters investigated were found to have little influence on the optimization outcome. Nevertheless, r seemed to have the largest impact on the results, followed by ρ, w1, w2, α and β, p_{best} and c_M.

Stage 2: The best parameter set identified is identical to the standard parameter set except when w1=8 is used which resulted in a slightly improved standard deviation.

Stage 3: Even though the best parameter set was only slightly different from the 'standard' parameter set, the best parameter set was used to solve the optimization problem using the MMAS algorithm with a new set of 20 random number seeds. The average OFV obtained was 136.85×10^5, which is very similar to the average OFV found with the 'standard' parameter values listed in Table 3.2. This confirms that optimization outcome has been very consistent across a range of random number seeds used. Also, it was found that within the range of parameter values specified in Table 3.7, the optimization outcome is fairly consistent with an average OFV of less than 0.3% higher than the best-found OFV.

Table 3.7 Range of parameter values

r	ρ	w1	w2	pbest	c_M	α	β
200 to 300	0.6 to 0.8	3 to 9	1	0.2 to 0.9	100,000 to 120,000	1	1 to 1.5

3.4.3 Comparison of results of ACO and other metaheuristics

The best results obtained when the ACO algorithms were used are given in Figure 3.4. It should be noted that, as discussed previously, the number of evaluations (trial solutions) allowed in the ACO and other metaheuristics runs are identical. It can be seen that the BAS and MMAS algorithms have outperformed the algorithms that have been applied to this case study previously.

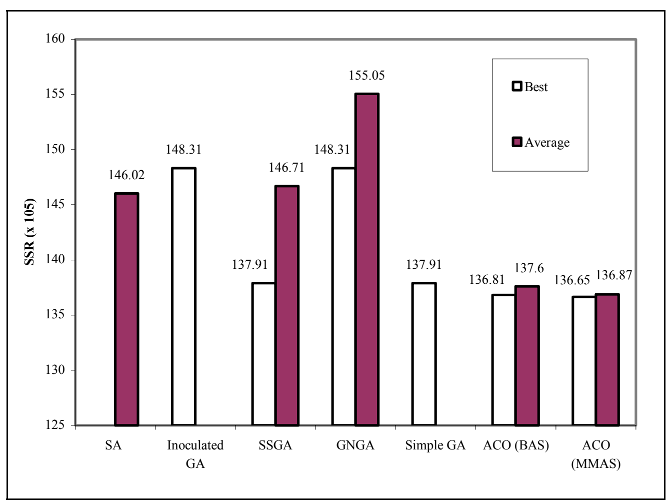

Figure 3.4 Comparison between the best result given by other optimization methods and ACO algorithms

4. SUMMARY & CONCLUSION

In this paper, a formulation for applying Ant Colony Optimization (ACO) to power plant maintenance scheduling optimization (PPMSO) has been developed and successfully tested on a 21-unit power plant case study to which other metaheuristics had been applied previously. A formulation for heuristics to be used at each decision point was introduced for the PPMSO problem and tested on the 21-unit problem using two ACO algorithms, including the BAS (Best-Ant System) and MMAS (Max-Min Ant System). Detailed sensitivity analyses on ACO parameters were also conducted in order to obtain a better understanding of the impact of the parameters on algorithm behavior and to obtain the best parameter set for the case study considered.

The results obtained indicate that ACO has performed better than any of the other metaheuristics that had been applied to the case study considered previously. Of the two ACO algorithms investigated, MMAS performed significantly better when tested statistically.

The heuristic information introduced for the PPMSO problem in this paper resulted in a marked increase in algorithm performance, both in terms of the best solution found and convergence speed. The impact of the heuristic was more pronounced on the BAS algorithm, particularly in terms of finding a near-optimal solution from different starting positions in objective function space. Sensitivity analysis suggested that MMAS has similar performance for the case study across a range of parameter values identified. However, more instances will be investigated to understand the influence of ACO parameters on a general PPMSO problem, and to test the effectiveness of ACO algorithms in solving the general PPMSO problem, including a real hydropower system with a high degree of complexity.

5. REFERENCES

[1] Ahmad, A. and D.P. Kothari, *A Practical Model for Generator Maintenance Scheduling With Transmission Constraints.* Electric Machines and Power Systems, 2000. **28**: p. 501-513.

[2] Aldridge, C.J., K.P. Dahal, and J.R. McDonald, *Genetic Algorithms For Scheduling Generation And Maintenance In Power Systems*, in *Modern Optimisation Techniques in Power Systems*, Y.-H. Song, Editor. 1999, Kluwer Academic Publishers: Dordrecht ; Boston. p. 63-89.

[3] Besten, M.d., T. Stützle, and M. Dorigo. *Ant Colony Optimization for the total weighted tardiness problem*. in *Proceedings of Parallel Problem Solving from Nature (PPSN-VI)*. 2000: Springer Verlag.

[4] Blum, C., *Theoritical and Practical Aspects of Ant Colony Optimization*, in *Akademische Verlagsgesellschaft Aka GmbH*. 2004: Berlin, Germany. p. Volume 282.

[5] Bullnheimer, B., R.F. Hartl, and C. Strauß, *A new rank based version of the ant system - a computational study*. 1997, Vienna University of Economics: Vienna. p. 14 pages.

[6] Colorni, A., et al., *Ant system for job-shop scheduling.* Belg. J. oper. Res. Stat. Comp. Sci., 1994. **34**(1): p. 39-53.

[7] Dahal, K.P., C.J. Aldridge, and J.R. McDonald, *Generator Maintenance Scheduling Using A Genetic Algorithm With A Fuzzy Evaluation Function.* Fuzzy Sets and Systems, 1999. **102**: p. 21-29.

[8] Dahal, K.P., J.R. McDonald, and G.M. Burt, *Modern Heuristic Techniques For Scheduling Generator Maintenance In Power Systems.* Transactions of the Institute of Measurement and Control, 2000. **22**(2): p. 179-194.

[9] Dopazo, J.F. and H.M. Merrill, *Optimal Generator Maintenance Scheduling Using Integer Programming.* IEEE Transactions on Power Apparatus and Systems, 1975. **PAS-94**(5): p. 1537-1545.

[10] Dorigo, M. and L.M. Gambardella, *Ant colonies for the travelling salesman problem.* BioSystems, 1997. **43**: p. 73-81.

[11] Dorigo, M. and L.M. Gambardella, *Ant colony system: a cooperative learning approach to the traveling salesman problem.* IEEE Transactions on Evolutionary Computation, 1997. **1**: p. 53-66.

[12] Dorigo, M. and T. Stützle, *Ant Colony Optimization*. 2004, Cambridge, MA: MIT Press.

[13] Dorigo, M., V. Maniezzo, and A. Colorni, *The ant system: optimization by a colony of cooperating ants.* IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics, 1996. **26**: p. 29-42.

[14] Dorigo, M., G. Di Caro, and L.M. Gambardella, *Ant algorithms for distributed discrete optimization.* Artificial Life, 1999. **5**: p. 137-172.

[15] El-Amin, I., S. Duffuaa, and M. Abbas, *A Tabu Search Algorithm for Maintenance Scheduling of Generating Units.* Electric Power Systems Research, 2000. **54**: p. 91-99.

[16] Endrenyi, J., et al., *The Present Status of Maintenance Strategies and the Impact of Maintenance on Reliability.* IEEE Transactions on Power Systems, 2001. **16**(4): p. 636-646.

[17] Maier, H.R., et al., *Ant Colony Optimisation for Design of Water Distribution Systems.* Journal of Water Resources Planning and Management, 2003. **129**(3): p. 200-209.

[18] Merkle, D., M. Middendorf, and H. Schmeck, *Ant Colony Optimisation for Resource-Constrained Project Scheduling.* IEEE Transactions on Evolutionary Computation, 2002. **6**(4): p. 333-346.

[19] Satoh, T. and K. Nara, *Maintenance Scheduling By Using Simulated Annealing Method.* IEEE Transactions on Power Systems, 1991. **6**(2): p. 850-857.

[20] Stützle, T. and H.H. Hoos, *MAX-MIN Ant System.* Future Generation Computer Systems, 2000. **16**: p. 889-914.

[21] Wodrich, M. and G. Bilchev, *Cooperative Distributed Search: The Ant's Way.* Journal of Control and Cybernetics, 1996. **26**(3).

[22] Yamayee, Z., K. Sidenblad, and M. Yoshimura, *A Computational Efficient Optimal Maintenance Scheduling Method.* IEEE Transactions on Power Apparatus and Systems, 1983. **PAS-102**(2): p. 330-338.