Chapter 1
Introduction

1.1 Research background

Under the pressure of rapid development around the globe, power demand has increased drastically during the past decade. To meet this demand, the development of power system technology has become increasingly important in order to maintain a reliable and economic electric power supply. One major concern of such development is the optimisation of power plant maintenance scheduling. Maintenance is aimed at extending the lifetime of power generating facilities, or at least extending the mean time to the next failure for which repair costs may be significant. In addition, an effective maintenance policy can reduce the frequency of service interruptions and the consequences of these interruptions. In other words, having an effective maintenance schedule is very important for a power system to operate economically and with high reliability.

Determination of an optimum maintenance schedule is not an easy process. The difficulty lies in the high degree of interaction between several subsystems, such as commitment of generating units, economical planning and asset management. Often, an iterative negotiation is carried out between asset managers, production managers and schedule planners until a satisfactory maintenance schedule is obtained. In addition, power plant maintenance scheduling is required to be optimized with regard to a number of uncertainties, including power demand, forced outage of generating units, hydrological considerations in the case of hydropower systems and trading value forecasts in a deregulated electricity market. Consequently, the number of potential maintenance schedules is generally extremely large, requiring a systematic approach in order to ensure that optimal or near-optimal maintenance schedules are obtained within an acceptable timeframe.

Ant Colony Optimisation (ACO) is a relatively new metaheuristic for combinatorial optimisation problems that is based on the foraging behavior of ant colonies. Compared to other optimisation methods, such as genetic algorithms (GA), ACO has been found to produce better solutions in terms of computational efficiency and quality when applied to a number of benchmark combinatorial optimisation problems. Recently, ACO has also been successfully applied to scheduling, including the job-shop, flow-shop, machine tardiness and resource-constrained project scheduling problems. ACO is highly suitable for scheduling optimisation problems, especially in handling various constraints, such as the precedence and sequential constraints, which can be attributed to the decision-tree based structure adopted by the ACO metaheuristic. In addition, multiple alternative schedules of similar quality can be produced in an ACO run, which is extremely useful in real-world power plant maintenance scheduling for negotiation with the asset manager, for example. A major drawback when using metaheuristics is not being able to incorporate non-quantifiable criteria, such as the operational or trading protocols adopted by a power system organization, in the optimisation process. This drawback can be overcome by having alternative maintenance schedules of similar quality that can be critically assessed using criteria not specified as part of the formal optimisation. In addition, the ability of ACO to utilize heuristic information in the optimisation process can effectively reduce the search space of a problem.

1.2 Research objectives

The major goal of this research is given as follows:

To develop, test and apply an ACO-based formulation to real power plant maintenance scheduling optimisation problems.

In order to meet the goal, a number of objectives are addressed, including:

Objective 1: To develop a generalized formulation for the power plant maintenance scheduling problem. Various issues, such as objectives and constraints commonly encountered in real-world power plant maintenance scheduling problems, are examined.

Objective 2: To develop a framework for utilizing ACO for the generalized PPMSO problem.

Objective 3: To test the ACO-PPMSO formulation with two benchmark case studies.

Objective 4: To apply the ACO-PPMSO formulation to real-world maintenance scheduling problems, including a simplified version and a full version of the Hydro Tasmania system.

1.3 Thesis layout

In Chapter 2, the research background related to power plant maintenance scheduling optimisation is reviewed. The objectives and constraints commonly used in past studies are discussed (Section 2.1). Optimisation methods previously adopted for power plant maintenance scheduling, namely heuristic approaches, mathematical programming, expert systems and metaheuristics, are reviewed in terms of the strengths and drawbacks of each method (Section 2.2). The motivation for considering metaheuristics in solving the problem are discussed.

In Chapter 3, various aspects of the Ant Colony Optimisation metaheuristic are presented, including the derivation of a metaheuristic from the foraging behaviour of real ant colonies (Section 3.1), the general framework for ACO to solve a combinatorial optimisation problem (Section 3.2), the two major categories of ACO algorithms (Section 3.3) and the previous applications of ACO to benchmark and real-world scheduling problems (Section 3.4). The chapter is concluded by the motivation for adopting the ACO metaheuristic for power plant maintenance scheduling in this research.

The proposed approach developed in this research for power plant maintenance scheduling problems is presented in detail in Chapter 4. A generalized formulation for the power plant maintenance scheduling problem is detailed (Section 4.1). The new ACO formulation proposed for the maintenance scheduling problem, including a new heuristic formulation and a proposed local search strategy, is introduced (Section 4.2). In Section 4.3, the mechanisms of the ACO algorithm implemented utilizing the proposed ACO formulation are detailed. Lastly, the two categories of constraints commonly encountered in power plant maintenance scheduling problem, as well as the techniques proposed to address these constraints in the ACO formulation, are discussed in Section 4.4.

In Chapter 5, an experiment is carried out to test the effectiveness of the new heuristic formulation and the local search strategy, as well as the overall performance of the proposed ACO formulation for power plant maintenance scheduling problems using four benchmark case studies, namely the 21-unit case study, the 22-unit case study and the modified version of the two case studies (Sections 5.1 to 5.3). The results and analysis derived from the experiment are detailed in Section 5.4.

In Chapter 6, the proposed ACO-PPMSO formulation is applied to real-world maintenance scheduling problems, including a five-station hydropower system (Section 6.2) and a full Hydro Tasmania system (Section 6.3).

Chapter 2
Literature Review

In this chapter, the background of the research work presented in this thesis is reviewed. In particular, the definition of power plant maintenance scheduling optimisation adopted in past studies and the methods previously applied to this problem are discussed.

2.1 Power plant maintenance scheduling optimisation

Power plant maintenance scheduling optimisation (PPMSO) has been described as a "*multi-criterion constrained combinatorial optimisation problem, with non-linear objective and constraint functions*" (Aldridge *et al.*, 1999). The definition of a combinatorial optimisation problem $P = (S, f)$ has been given by Blum *et al.* (2003) as:

- a set of variables $R = \{r_1, \ldots, r_n\}$;
- variable domains $D_1, \ldots, D_n$;
- a set of constraints; and
- an objective function f to be minimized (for a minimization problem).

The search space of a problem, S, can thus be defined as:

$S = \{s = \{(\mathrm{r}_1, v_1), \ldots, (\mathrm{r}_n, v_n)\} \mid v_i \in D_i, s \text{ satisfies all the constraints}\}$

The aim of an optimisation problem is to find a set of globally optimum solutions $S^* \subseteq S$ for (S, f) such that $f(s^*) \leq f(s)\ \forall\ s^* \in S^*, s \in S$.

In relation to PPMSO, the aim has been specified as the determination of the ***timing*** and ***sequence*** of the maintenance periods of each of the generating machines (units) used for power generation, assuming maintenance durations are fixed (Dopazo *et al.*, 1975; Yamayee *et al.*, 1983;

Mukerji *et al.*, 1991; Satoh *et al.*, 1991; Kim *et al.*, 1997; Aldridge *et al.*, 1999; Dahal *et al.*, 1999; Dahal *et al.*, 2000; El-Amin *et al.*, 2000; Foong *et al.*, 2005a; Foong *et al.*, 2005b). The set of variables X in a PPMSO problem is therefore implicitly represented by the maintenance commencement time for all generating units considered, with the optional commencement times given by the variable domains D. The objectives and constraints of PPMSO on the other hand, are less well defined and were a research area of their own at earlier stages of PPMSO research. Generally, the objectives and constraints being employed for maintenance scheduling in the past have been quite different, depending on the concerns of individual power utilities. In this section, different objectives and constraints being adopted in previous studies are reviewed.

2.1.1 Objectives

The objectives commonly utilized for PPMSO are generally reliability or cost based (Figure 2.1):

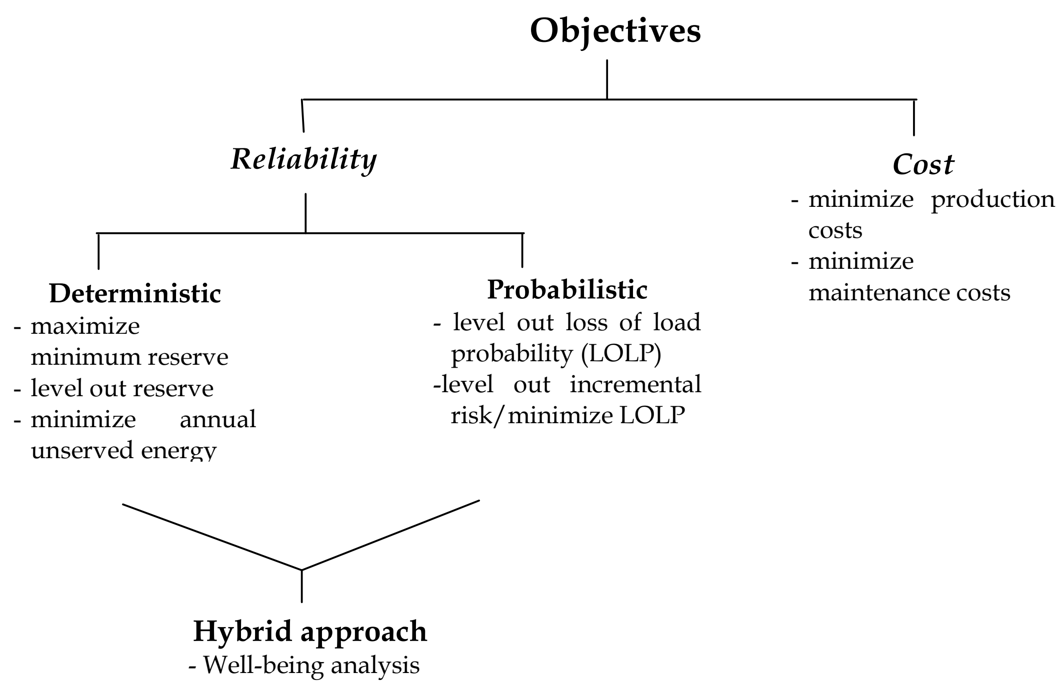


Figure 2.1: Objectives of power plant maintenance scheduling optimisation

2.1.1.1 Reliability-based criteria

Apart from meeting demands, a power system needs to provide a reserve generation capacity to secure the provision of electricity to customers in the event of a sudden breakdown of generating units or unexpectedly high peak demands. Reliability-based criteria previously used can be roughly divided into two categories: deterministic and probabilistic approaches (Figure 2.1). In addition, hybrid approaches can also be used.

Deterministic approaches

Deterministic approaches usually utilize historical data for the assessment of maintenance schedules. An example of such data are daily peak demands averaged over the past 5 years. Some deterministic reliability-based criteria aim to maximize the minimum reserve in the planning horizon (Christiaanse *et al.*, 1971; Mukerji *et al.*, 1991; El-Amin *et al.*, 2000), level the reserve throughout the planning horizon (Escudero *et al.*, 1980; Kim *et al.*, 1997; Moro *et al.*, 1999; Dahal *et al.*, 2000; El-Amin *et al.*, 2000; Wang *et al.*, 2000) or minimise the annual expected unserved energy (Ahmad *et al.*, 2000).

Probabilistic approaches

Some elements of a power system are naturally stochastic, including system demands, the forced outage rates of generating units and the system inflows in the case of a power system with hydropower plants. If one or more of these elements are modelled probabilistically during the assessment of the reliability of a trial maintenance schedule, a probabilistic reliability-based approach is employed. A number of surveys revealed that from 1964 to 1987, all Canadian utilities changed their reliability assessment approach from deterministic to probabilistic (Billinton, 1991). By taking into account the uncertainties associated with the forced outage of generating units by incorporating their effective load carrying capabilities, Garver (1972) was able to achieve uniform loss of load probability (LOLP) for all time periods in a year. The method was extended by Stremel *et al.* (1981) to account for load forecast uncertainty. In the proposed method, equivalent loads were calculated for the three time periods where peak loads and their corresponding probabilities were specified. Maintenance scheduling was then carried out such that the overall LOLP was minimized, based on the calculated equivalent loads. The maintenance schedules obtained in this way were claimed to be much more representative of actual planning operations (Stremel *et al.*, 1981). In another study, Garver (1972) method was modified by Chen *et al.* (1990) to level incremental risks, which is equivalent to the minimization of annual LOLP.

Hybrid approach

Well-being analysis that combined deterministic and probabilistic approaches in a single framework was developed by Billinton *et al.* (1996). As part of the framework, the reserve capacity of a system is analysed using a probabilistic formulation and compared with an accepted deterministic criterion, such as the loss of the largest unit, in order to measure overall system comfort (Billinton *et al.*, 1996). A probability of health (PH) index used as part of the well-being analysis, which represents the probability that the available reserve is equal to or greater than the required capacity reserve, was later used as a fitness function for the genetic algorithm optimisation formulation proposed by Abdulwhab *et al.* (2004).

Despite the existence of many different formulations for reliability-based criteria, it has been shown that the final optimisation outcome (optimized schedule(s)) obtained based on a reliability criterion is usually acceptable in terms of other reliability-based criteria (Zürn *et al.*, 1977).

2.1.1.2 Cost-based criteria

For planned maintenance scheduling, the major costs involved are energy production cost and maintenance cost. The latter is only important if outage durations are allowed to vary within a given limit (Yamayee *et al.*, 1983). A survey carried out by Mukerji *et al.* (1991) on 25 major power plants in the US found that 16 had chosen production cost minimisation as the only objective in determining an optimum maintenance schedule. The author addressed two major modelling problems in such an approach, the first being production cost as a complex non-linear function of the maintenance schedule; the second that the cost function is dependent on load shapes and forced outage rates, which generally required extensive simulations for the cost calculations. To overcome the first problem, Egan *et al.* (1976) suggested that reasonable production cost could be achieved by maximizing system reliability under uncertainties (loads and random forced outages) and minimizing the capital plant needed to achieve a given reliability in the future. With regard to the second problem, load uncertainty and probability of forced outage can be modelled using a fuzzy logic approach that incorporates economic and technical knowledge of the problem domain (Dahal *et al.*, 1999).

A study by Chattopadhyay *et al.* (1995) that investigated the performance of different objectives for maintenance scheduling optimisation of two interconnected power utilities in India found that using annual operating cost was an ineffective objective when used alone. In the study, two of the three reliability-based objectives tested produced maintenance schedules associated with reasonable annual operating costs, but the reliability criterion was found to be unsatisfactory when cost was used as the only objective function (Chattopadhyay *et al.*, 1995). In contrast, a study conducted by Ahmad *et al.* (2000) revealed that optimisation based on a cost criterion can produce schedules that result in significant savings with an associated reliability level that is almost as good as that produced when only the reliability criterion is used. The contradictory conclusions of the two studies might be attributed to the differences in the search space characteristics exhibited by the two case study systems.

Potential problems with local minima in the search space have been reported by Arzamascev *et al*. (1970), who also used only production cost as the objective function in their optimisation algorithm. In such situations, difficulties in finding near globally optimal solutions could be overcome by using evolutionary algorithm optimisation methods, which work with a set of solutions, and not on a single solution, thus reducing the chance of convergence to local optima (Ekwue, 1999).

In other studies, production cost was found to be an insensitive objective, i.e. production costs were almost constant in the vicinity of the optimum region of the search space (Zürn, 1975; Hoover *et al.*, 1976; Yamayee *et al.*, 1983). However, during the discussion on the study carried out by Yamayee *et al*. (1983), Stremel (1983) pointed out that an appropriate objective function of PPMSO should comprise of production cost and the value of unserved energy.

In previous studies, maintenance scheduling of power plants has been treated at its lowest level of complexity, without consideration of a number of complicating factors. For example, the cost of hydropower plant maintenance is influenced by loss-of-revenue due to spill at storages, which is caused by machines being taken off-line for maintenance. Since spill is the major cause of energy loss, it also affects the reliability of power supply systems. In order to cater for such issues, a simulation model is often utilized to assist in planning activities such as generation dispatch and unit commitment, given a proposed maintenance schedule. Consequently, there is a need to develop an optimisation model capable of incorporating such simulation models.

2.1.1.3 Other criteria

Other objectives addressed in the literature include the earliest possible schedule and the minimum change from an existing schedule (Dopazo *et al.*, 1975). In an earliest possible schedule, maintenance tasks are scheduled to commence as early as possible within individual timeframe windows without violating system constraints. A criterion can also be specified such that a new maintenance schedule that minimizes disruption to an existing schedule is desired. Assuming the event of a sudden breakdown of a major generating unit, the existing optimum maintenance schedule must be reviewed. A new optimum schedule is determined such that the least disruption is introduced to the original schedule (minimum change from existing schedule) while the machine broken down unexpectedly could be taken offline.

2.1.1.4 Multiple criteria

Although maintenance scheduling optimisation has been defined as a multi-objective problem, only few researchers have successfully included more than one criterion in their optimisation model. In the expert system developed by Lin *et al*. (1992), whether a reliability (maximization of the minimum reserve margin) or cost (minimization of production cost) criterion is used depends on an operation index, which is defined by the amount of reserve generation capacity for maintenance activities. Mukerji *et al*. (1991), Yamayee *et al*. (1983), El-Amin *et al*. (2000) and Moro *et al*. (1999) considered more than one criterion in their studies by carrying out separate optimisation runs using each of the cost and reliability criteria, which resulted in two different sets of optimised schedules for each criterion. However, the optimized maintenance schedules produced by these studies only represent a subset of the Pareto-optimal solutions of the case studies being solved. In contrast, a complete set of Pareto-optimal solutions of a multi-objective PPMSO problem should consist of maintenance schedules that cannot be improved in any one objective without degrading one other objective. A decision maker can then choose from these schedules either based on other non-quantifiable criteria or during negotiations with an asset manager, for example.

2.1.2 Constraints

A feasible maintenance schedule must not only achieve its objectives, but must also be practical in terms of implementation. Therefore, constraints must be specified in an optimisation model to ensure solutions are feasible. The following are the most commonly used constraints:

(A) Demand Constraints

In providing power supply, energy demand has to be met. In addition to the actual expected demand, a certain level of energy reserve is generally provided to cover accidental loss of generating plants.

(B) Maintenance window constraints

Generating units in power plants should be inspected and maintained on a regular basis. This is to ensure that they are performing at reasonable efficiency, to reduce the likelihood of forced outages and to extend the lifetime of the machines (Egan *et al.*, 1976). Normally, the earliest and latest possible start time for the maintenance activities of generating units are specified. In addition, if more than one regular maintenance task for a generating unit is required to be scheduled, it is important that the duration between these tasks is longer than a prescribed time period.

(C) Resource constraints

Experienced personnel should be involved in maintenance to avoid possible major damage (Lin *et al.*, 1992). Therefore, the number of machines that can be maintained at one time is usually limited by the availability of manpower. Also, other resources, such as specialist tools, might be required during a maintenance session and their availability must be taken into account when a maintenance schedule is proposed. Failure to take into account the availability of appropriate tools may cause unnecessary delays in machinery maintenance, making the machinery unavailable and thus further delaying the overall plant maintenance schedule (Egan *et al.*, 1976).

(D) Precedence and sequence constraints

Some maintenance tasks can only commence when other tasks have been completed. In some cases, minimum and maximum gaps between consecutive outages of a particular unit need to be specified. For example, investigative maintenance is carried out prior to the major overhaul of some large generating units. Also, the major overhaul might not be able to start earlier than 2 weeks (to organize for maintenance resources), or later than 6 weeks, for example (for the validity of investigation results), after completion of the investigative task. In such a scenario, it has to be ensured that the optimised maintenance schedule(s) is/are feasible with regard to these constraints. Other constraints might include the specification of a minimum gap between outages of two units operating in the same plant (Mukerji *et al.*, 1991).

(E) Exclusion constraints

Constraints can also be used to ensure that two machines of high capacity are not taken off-line for maintenance activities at the same time.

It should be noted that individual power plant utilities generally have unique sets of restrictions that influence maintenance scheduling, which are related to power system operation characteristics, seasonal variations, geographical conditions and usual practice (Lin *et al.*, 1992).

In view of the number of constraints that may need to be imposed in a PPMSO problem, it is desirable that a method proposed for PPMSO can effectively handle some, if not all, of these constraints. In addition, the degree of rigidity with which certain constraints have to be satisfied should be able to be specified in the formulation of the optimisation problem. For instance, demand constraints generally have to be satisfied at all times to ensure an adequate supply of electricity. On the other hand, additional personnel can sometimes be brought in if the resulting increase in reliability achieved outweighs the cost imposed. Hence, flexibility in manpower constraints might be desirable so that the search for schedules with better objective values that result in the violation of certain constraints is allowed.

It should be noted that much of the debate on the number and type of objectives for PPMSO took place before 2000, when power systems were mostly independent utilities that sell electricity to their customers at tariffs regulated by governments. Since the spread of electricity market deregulation around the globe, the context in which PPMSO research is carried out, in particular the objectives and constraints used for optimisation, has changed dramatically.

2.1.3 Deregulation of electricity market

A deregulated electricity system is a system for effecting the purchase and sale of electricity using supply and demand to set the price. Competing generators trade their electricity to retailers in a wholesale electricity market. Among the countries that have developed wholesale electricity markets and the corresponding management bodies are (Wikipedia, 2006a):

- Australia – National Electricity Market Management Company (NEMMCO)
- Canada - Independent Electricity System Operator (IESO) Ontario Market and Alberta Electric System Operator (AESO)
- New Zealand – M-co
- Denmark, Finland, Sweden and Norway – Nordic Power Exchange

Following the increasing popularity of electricity market deregulation around the globe, the fundamental objective of a power utility (competing generator) when scheduling for maintenance has become the maximization of benefits derived from the electricity wholesale market. This is far more complicated than the reliability- and cost-based criteria discussed in Section 2.1.1. For instance, a power utility may no longer treat system demands as a hard constraint, especially when market clearing price is forecasted to be low, as cheap electricity can be purchased from the wholesale market. As a result, large generating units may be taken offline for maintenance during such periods so that they are available for generation when the market prices are forecasted to be high.

Changes in maintenance scheduling practices as a result of electricity market deregulation also has a significant impact on the applicability of maintenance schedulers' accumulated experience/engineering judgement. Power utilities relying heavily on schedulers' experience for maintenance scheduling could face a difficult situation due to the change of context in which scheduling is carried out. Therefore, a desirable method for PPMSO must be able to adapt to changes of optimisation objectives and system constraints with relative ease.

2.2 Optimisation methods previously adopted for PPMSO

2.2.1 Design requirements for a maintenance scheduling tool

When developing an optimisation method for PPMSO problems, the follow characteristics of a method are desired:

Criterion 1: Simple to implement

The proposed method is preferably a generalised algorithmic framework that can be readily applied to PPMSO with only little modifications.

Criterion 2: Easily incorporate a simulation model

As mentioned previously, simulation models are used extensively due to the complexity of operations involved in a power system. Therefore, the proposed optimisation method must be able to incorporate a simulation model as part of its algorithm.

Criterion 3: A priori information on case study systems is not required

The proposed method should not require a large number of inputs from users of the case study system to be solved. Furthermore, objectives and constraints must be able to be addressed easily.

Criterion 4: Effective handling of constraints

In view of the large number and complexity of constraints involved in a real-world PPMSO problem, the proposed method must be able to handle practical constraints effectively. In this way, a problem search space can be reduced and computational run-time can be cut down significantly.

Criterion 5: Manage to adapt to changes in case problem easily

Although maintenance scheduling is a long-term planning activity, unexpected changes in a power system, such as the purchase of new generating units or the permanent loss of a power station due to a new environmental policy, for example, are not uncommon. An ideal maintenance-scheduling tool must be able to be modified with relative ease in response to the changes.

Criterion 6: Able to find more than one desired schedule

In a PPMSO problem where the global optimum is often unknown, maintenance schedule(s) associated with the lowest objective function cost found in an optimisation run is/are desired. It should be noted that there may be different schedules associated with the lowest objective function cost, or the objective functions for a number of schedules might be similar. It is desirable to identify a number of these alternatives schedules as part of this optimisation, as this leaves room for negotiation with assets managers, for example, in order to identify the optimal maintenance schedule from a practical point of view.

Criterion 7: Able to find good solutions in reasonable computational time

An optimisation method should ideally find the globally optimum solution for a given problem. However, the size of a real-world case study system maybe too large so that determination of the true optimum maintenance schedule of a case study system is almost impractical. Therefore, a desirable optimisation method for PPMSO is one that is able to identify good or near-optimal maintenance schedule(s) for a case system with reasonable computational effort.

Over the past two decades, many studies have been conducted on developing methods for the maintenance scheduling optimisation of power plants. These methods, when differentiated based on searching mechanism, can be categorised into heuristic approaches, mathematical programming, expert systems and metaheuristics (Figure 2.2).

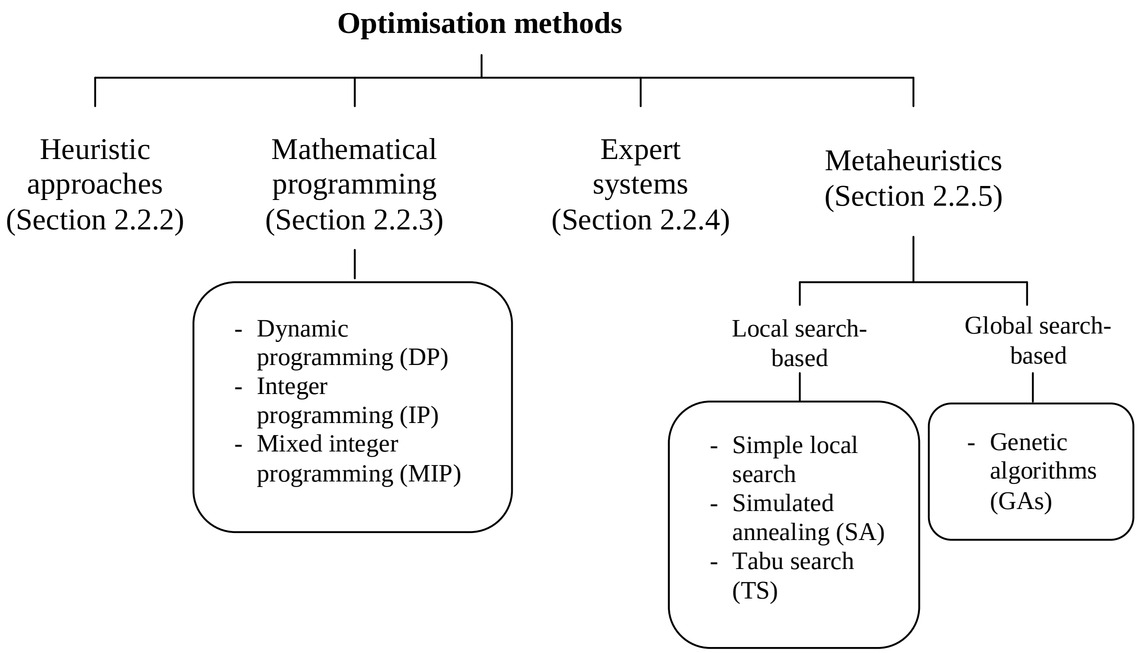


Figure 2.2: Optimisation methods adopted previously for PPMSO

2.2.2 Heuristic approaches

Heuristic approaches were developed to solve PPMSO mostly during the early stages of maintenance scheduling research. In general, heuristic approaches involve allocating maintenance unit outages sequentially by utilizing a set of rules, such as the biggest capacity generating units first, the generating units that require most maintenance resources first etc. A heuristic approach employing a branch-and-bound technique was proposed by Christiaanse *et al*. (1972) to maximize the system's lowest net reserve over the planning horizon. In the proposed method, a maintenance schedule is constructed sequentially by scheduling for the personpower category that is required by the largest number of maintenance tasks. In addition, the period during whom a maintenance task is scheduled to begin depends on the current level of minimum reserve capacity. If any of the system constraints were violated by the allocation of a maintenance task to a timeslot within the planning horizon, the procedure would be reversed and other arrangements would be made such that a feasible maintenance schedule is obtained. A similar heuristic has also been used by Garver (1972), which is aimed at equalizing the system's loss-of-load-probability (LOLP) throughout the planning horizon.

Despite being able to incorporate constraints during optimisation, heuristic approaches perform an exhaustive search and usually suffer from the possibility of not being able to find a feasible schedule, even when one exists. Therefore, the likelihood of the optimal solution(s) being found by using a heuristic method is relatively small. In addition, a heuristic method is developed based on the characteristics, in particular the objectives and constraints, of a specific case study system. Hence, it has limited applicability to other PPMSO case study systems. Furthermore, a slight change in the original objective functions or constraints might affect the utility of a heuristic method. Another shortcoming of purely heuristic approaches is the need for the objective function value associated with a partially built schedule to be calculated every time a maintenance task is added, which cannot be done for complex power systems. For example, in a hydropower system, storages are interconnected and the dispatch of generating units (i.e. the decision about which generating units should be used for meeting a demand) must utilize a simulation model. For this purpose, the maintenance schedule used must be complete.

2.2.3 Mathematical programming

Since the 1960s, mathematical programming methods have been investigated for their application to generator maintenance scheduling optimisation. The most commonly used methods in this category are dynamic programming (DP), integer programming (IP), mixed integer programming (MIP) and linear programming (LP) (Figure 2.2).

2.2.3.1 Dynamic programming (DP)

Dynamic programming (DP) was considered to be suitable for solving PPMSO problems due to the following reasons (Yamayee *et al.*, 1983): (1) It is suitable for solving optimisation problems where a sequence of decisions is involved; (2) The objective function used in DP does not need to be a continuous function of decision and state variables; and (3) Neither the objective function or constraint functions are required to be represented in analytic forms, provided these function values can be obtained by other means (eg. through a simulation model) when required. However, application of pure dynamic programming to complex combinatorial problems has been limited due to the "curse of dimensionality', which states that a problem that is complex enough to be interesting is too large to be solved within practical computational time and storage. In addition, the constraint representation in the DP formulation is not stringent enough to limit the number of feasible solution (Christiaanse *et al.*, 1972). However, this problem was later resolved by Zürn (1975) and Zürn *et al*. (1977) using dynamic programming successive approximations (DPSA). The DPSA method was also used by Yamayee *et al*. (1983) to solve a PPMSO case study that considered a reliability and a cost criterion in separate optimisation runs.

2.2.3.2 Integer programming (IP)

Integer programming (IP), coupled with the branch-and-bound technique, has been applied to maintenance optimisation problems previously (Dopazo *et al.*, 1975; Egan *et al.*, 1976; Mukerji *et al.*, 1991). However, IP was considered to be unable to model stochastic uncertainties efficiently (Ahmad *et al.*, 2000). Also, the computational time required for IP for solving optimisation problems tends to grow prohibitively with problem size (Satoh *et al.*, 1991).

2.2.3.3 Mixed integer programming (MIP)

A mixed integer programming model has been proposed by Chattopadhyay *et al*. (1995) to obtain least-cost maintenance schedules (in monthly time blocks) for two large interconnected Indian power utilities. The mixed integer programming model developed by Chattopadhyay *et al*. (1995) was later deemed incapable of handling the large number of decision variables introduced when the exact start dates, rather than the month/week of the maintenance tasks, are considered (Chattopadhyay, 1998). Also, the computational time taken to run such a model, when using the computing power at that time, would be impractically long when uncertainties are to be taken into account by repetitive Monte Carlo simulations (Chattopadhyay, 1998). In order to overcome the shortcomings associated with using the mixed integer programming model, a similar model employing linear programming (LP) was proposed by Chattopadhyay *et al*. (1998). However, this approach results in the selection of real-numbered values for the decision variables, which might produce invalid maintenance schedules. In order to rectify this problem, a heuristic algorithm was implemented.

Moro *et al*. (1999) developed a two-stage mixed integer programming formulation to solve the maintenance scheduling problem of a Spanish electric power system. In their formulation, optimisation based on cost criteria was carried out and the best-cost schedule obtained was used as an input for the second-stage optimisation, where reliability is maximized without exceeding a prescribed level of the cost associated with the best-cost schedule (Moro *et al.*, 1999). However, the impact of optimizing the two criteria in a different order has not been discussed. In addition, the search space of the stage-2 optimisation could be restricted by the results from stage 1, which may result in finding only the local optima of the problem search space. Therefore, depending on the characteristics of the fitness landscape of the case study system being investigated, the 'true' optimum solution might not be identified by the formulation.

Mixed integer programming models accounting for transmission constraints have been developed by Ahmad *et al*. (2000) and Moro *et al*. (1999), and applied to an existing Indian power utility. As neither of the case studies investigated by Chattopadhyay *et al*. (1995), Moro *et al*. (1999) and Ahmad *et al*. (2000) have previously been solved by other optimisation methods, the relative performance of mixed integer programming in solving PPMSO problems remains unknown.

In general, the performance of mathematical programming methods for solving power plant maintenance scheduling optimisation is unsatisfactory due to the need to specify mathematical equations to represent the power system as part of the problem formulation. These equations are difficult, if not impossible, to derive for real life applications. Very often, simplified equations that do not fully reflect the power system at hand were used. Secondly, difficulties arise when changes made to the power plant system have to be reflected in the problem formulation, as this requires modification of the equations mentioned above. Thirdly, the relative importance of constraints cannot be specified. For example, slight violation of constraints would not be permitted, even though the resulting objective values might be much better. Furthermore, the computational time needed to implement this approach increases prohibitively with problem size.

2.2.4 Expert systems

An expert system formulation was developed by Lin *et al*. (1992) to schedule maintenance tasks for the Taiwan Power Company. As part of the expert system, whether branch-and-bound or dynamic programming is used during the optimisation process depends on the objective criterion used, which in turn is governed by the satisfaction of system demands throughout the planning horizon. The drawback of the proposed formulation is that the heuristics and rules embedded in the expert system need to be updated when there is a slight change in system inputs (eg. constraints, objectives or decision maker's preference). For the same reason, it is difficult to apply the same expert system to other PPMSO case studies.

2.2.5 Metaheuristics

Due to the shortcomings of heuristic and mathematical programming approaches, the possibility of applying metaheuristics to solving PPMSO problems has intrigued researchers over the last 10 years. Metaheuristics, as defined in the literature:

- Are high-level algorithmic frameworks which utilize algorithms ranging from simple local search to complex learning processes (Blum *et al.*, 2003).

- Are approximate and usually non-deterministic, and therefore may avoid being trapped in local minima in a search space (Blum *et al.*, 2003).

- Are not problem-specific, and can be applied to different combinatorial optimisation problems with relatively little modification (Dorigo *et al.*, 2004a).

Metaheuristics can be categorised in different ways depending on the characteristics considered for differentiating them. For instance, 'nature-inspired' vs. 'non-nature inspired' categorisation traces the origin of metaheuristics, whereas the 'memory usage vs. memory-less methods' categorisation differentiates metaheuristics that use long term and short term memories (Blum *et al.*, 2003). In their review of metaheuristics, Blum *et al*. (2003) categorised metaheuristics based on the number of solutions used at the same time. Based on this characteristic, algorithms operating on a single solution are called trajectory methods, whereas population-based methods perform the search process via the evolution of a set of trial solutions (Blum *et al.*, 2003).

In this thesis, rather than presenting a thorough discussion on all metaheuristics, the focus is on metaheuristics that have been used previously for PPMSO problems, including Simulated Annealing (Satoh *et al.*, 1991), Tabu Search (El-Amin *et al.*, 2000) and genetic algorithms (Aldridge *et al.*, 1999). These methods are categorised based on whether a local search procedure or a global search procedure is adopted in the metaheuristics (Figure 2.2).

2.2.5.1 Local search-based metaheuristics

A local search-based metaheuristic is essentially a higher-level algorithmic framework that consists of a simple local search algorithm and additional features designed to enhance the performance of the algorithm. A simple local search method is firstly described, followed by the discussion of two local search-based metaheuristics.

Simple local search

Given a combinatorial optimisation problem with a search space S, the following formal definitions are given by (Blum *et al.*, 2003):

- A neighborhood structure is a function $N:S \rightarrow 2^S$ that assigns to every $s \in S$ a set of neighbours $N(s) \subseteq S$. $N(s)$ is called the neighborhood of s.

- A locally minimal solution (or local minimum) with respect to a neighborhood structure N is a solution $\hat{s}$ such that $\forall s \in N(s): f(\hat{s}) \leq f(s)$. We call $\hat{s}$ a strict locally minimal solution if $f(\hat{s}) < f(s) \ \forall s \in N(\hat{s})$.

Starting from a single solution, *s*, generated either at random or by using some heuristics, local search scans the neighborhood, *N*(*s*), of the current solution for better neighbour solutions. Either a first-improvement or a best-improvement technique is usually used to determine the move to be performed (Blum *et al.*, 2003). Using the first-improvement technique, the first improving neighbour found in *N*(*s*) is used to replace the current solution. On the other hand, exhaustive search is performed to find the best-improving neighbour among *k* neighbours being assessed. It should be noted that a move is only performed when a neighbour solution is found such that a better objective function cost is achieved. The local search is stopped when a local minimum is reached.

Defining a neighborhood structure is essential before utilizing local search in solving a combinatorial optimisation problem. Some examples of neighborhood structures previously used for PPMSO problems are:

- In the studies conducted by Satoh *et al*. (1991) and El-Amin *et al*. (2000), a neighbour trial maintenance schedule is generated by randomly modifying the maintenance commencement time of a randomly selected generating unit from the current trial maintenance schedule.

- In the local search procedure used by Kim *et al*. (1997), a neighbour trial maintenance schedule is generated by adding 1 to or subtracting 1 from the maintenance start day of a randomly chosen generating unit contained in the current trial maintenance schedule.

The performance of a simple local search algorithm alone in solving combinatorial optimisation problems is unsatisfactory (Blum *et al.*, 2003), the main shortcoming being the inability to escape local minima, once trapped. In order to overcome this problem, various features have been proposed to be added to simple local search algorithms, which result in different local search-based metaheuristics. Examples of such metaheuristics include Simulated Annealing (SA), Tabu Search (TS), the Greedy Randomized Adaptive Search Procedure (GRASP), Variable Neighborhood Search (VNS), Guided Local Search (GLS) and Iterated Local Search (ILS). As Simulated Annealing (SA) and Tabu Search (TS) have been proposed for PPMSO previously (Satoh *et al.*, 1991; Kim *et al.*, 1995; Kim *et al.*, 1997; Burke *et al.*, 2000; Dahal *et al.*, 2000; El-Amin *et al.*, 2000), a more detailed discussion of these methods is included in this thesis.

Simulated Annealing (SA)

Annealing is a process in metallurgy that involves heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. Starting with a high temperature, the particles of a material escape from their initial positions and randomly wander while the temperature is progressively lowered until a highly structured lattice associated with a minimal internal energy is formed. By analogy with this process, Simulated Annealing (SA) is a metaheuristic that uses an enhanced local search procedure and was first applied to combinatorial optimisation problems by Kirkpatrick *et al*. (1983). In order to utilize SA for a combinatorial optimisation problem, the following equivalences are assumed between the annealing process and an optimisation problem (Satoh *et al.*, 1991):

1. The solutions in a combinatorial optimisation problem are equivalent to the states of a physical system, and

2. The cost of a solution is equivalent to the energy of a state.

As shown in Figure 2.3 SA starts with an initial temperature T and an initial trial solution is generated as with a simple local search algorithm. Then, at each iteration, the defined neighborhood $N(s)$, of a trial solution, s, is scanned for first-improving or best-improving neighbours (depending on the user's preference), $\hat{s}$. In a case where no improving neighbour is identified, $\hat{s}$ can be represented by the best-objective function cost neighbour. When deciding whether a move is performed i.e. the current solution s is replaced by the newly found $\hat{s}$, the following rules are employed:

1. If $\hat{s}$ is better than the current solution, s, the probability of replacing s with $\hat{s}$ is 1.

2. If $\hat{s}$ is worse than the current solution, s, the probability of replacing s with $\hat{s}$ is $e^{-\Delta/T}$, where $\Delta = f(\hat{s}) - f(s)$, and T is the current temperature of the annealing process.

The rules presented are a unique feature that distinguishes SA from a simple local search, where non-improving moves are performed probabilistically for the sake of finding an even better solution later during the search. With a high initial temperature, the probability of accepting a non-improving move is higher and therefore, the optimisation search at this stage resembles a simple randomised local search. As the temperature is progressively reduced in accordance with a cooling schedule, $g(T)$, non-improving trial solutions are more likely to be rejected and eventually only improving solutions are accepted if a minimum temperature is set to a sufficiently low value. A SA algorithm stops when the temperature reaches a predefined minimum value, T_{min}.

A cooling schedule, which contains the initial temperature, the cooling rate, the minimum temperature and the size of the neighborhood at each temperature, must be defined beforehand. As pointed out by Van Laarhoven *et al*. (1987), a cooling schedule must be carefully defined for successful applications of SA to combinatorial optimisation problems. Numerical tests conducted by Satoh *et al*. (1991) concluded that a lower cooling rate should be chosen for problems of larger search space to allow for sufficient exploration.

For the sake of brevity, the SA algorithm has only been introduced in its simplest form in this thesis. Readers are referred to Downsland *et al*. (1993) for a more theoretical and mathematical description of SA. Areas to which SA has recently been applied include scheduling (Suresh *et al.*, 2006), chemical process studies (Agostini *et al.*, 2006) and transportation (Zhao *et al.*, 2005), to name a few. Examples of applications to power systems and related optimisation problems are the economic emission load dispatch of fixed head hydrothermal power systems (Basu, 2005), power-system load forecasting (Liao *et al.*, 2006) and dynamic economic dispatch (Panigrahi *et al.*, 2006).

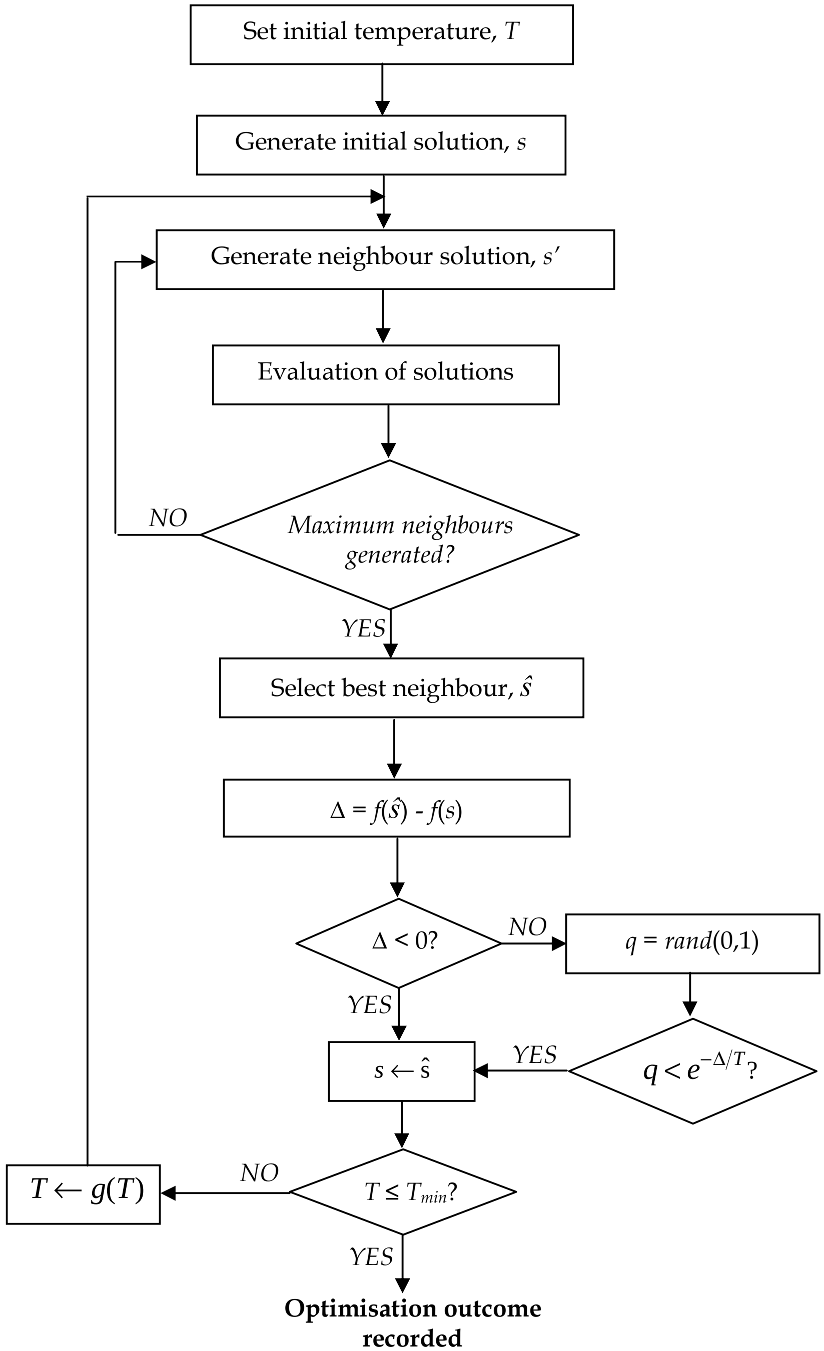


Figure 2.3: Typical Simulated Annealing (SA) algorithm

Tabu Search (TS)

Proposed by Glover (1989), Tabu Search (TS) is a metaheuristic used to manage a local search procedure with the utilization of adaptive memory. A flowchart of a simple Tabu Search is shown in Figure 2.4

Tabu Search begins by initialisation of a Tabu List. A local search is then used to scan the neighborhood, $N(s)$, and an initial trial solution, s is chosen randomly. Among the k neighbours of s, the best neighbour, $\hat{s}$, is selected to replace the current solution, s. Upon execution of a move, selected attributes of the move are stored in a Tabu List, and are declared 'tabu-active' for a predefined number of iterations. An example of a move attribute is the exchange of the cities at positions 4 and 5 in the case of a Travelling Salesman Problem (TSP). For the remainder of the TS run, a move to the best neighbour found at an iteration is banned if one or more of the attributes involved in the move are flagged as 'tabu-active' in the Tabu List. However, an aspiration criterion can be specified such that a prohibited move can still be admissible if this criterion is satisfied. The iterative process of the memory-enhanced local search is repeated until a termination criterion is met. The best solution found during a TS run is regarded as the optimized solution.

It can be envisaged that by prohibiting repetition of previously performed moves, the likelihood of reversal of moves and cycling of solutions is reduced. More importantly, the utilization of adaptive memory in TS helps a local search procedure to escape local optima by allowing non-improving moves.

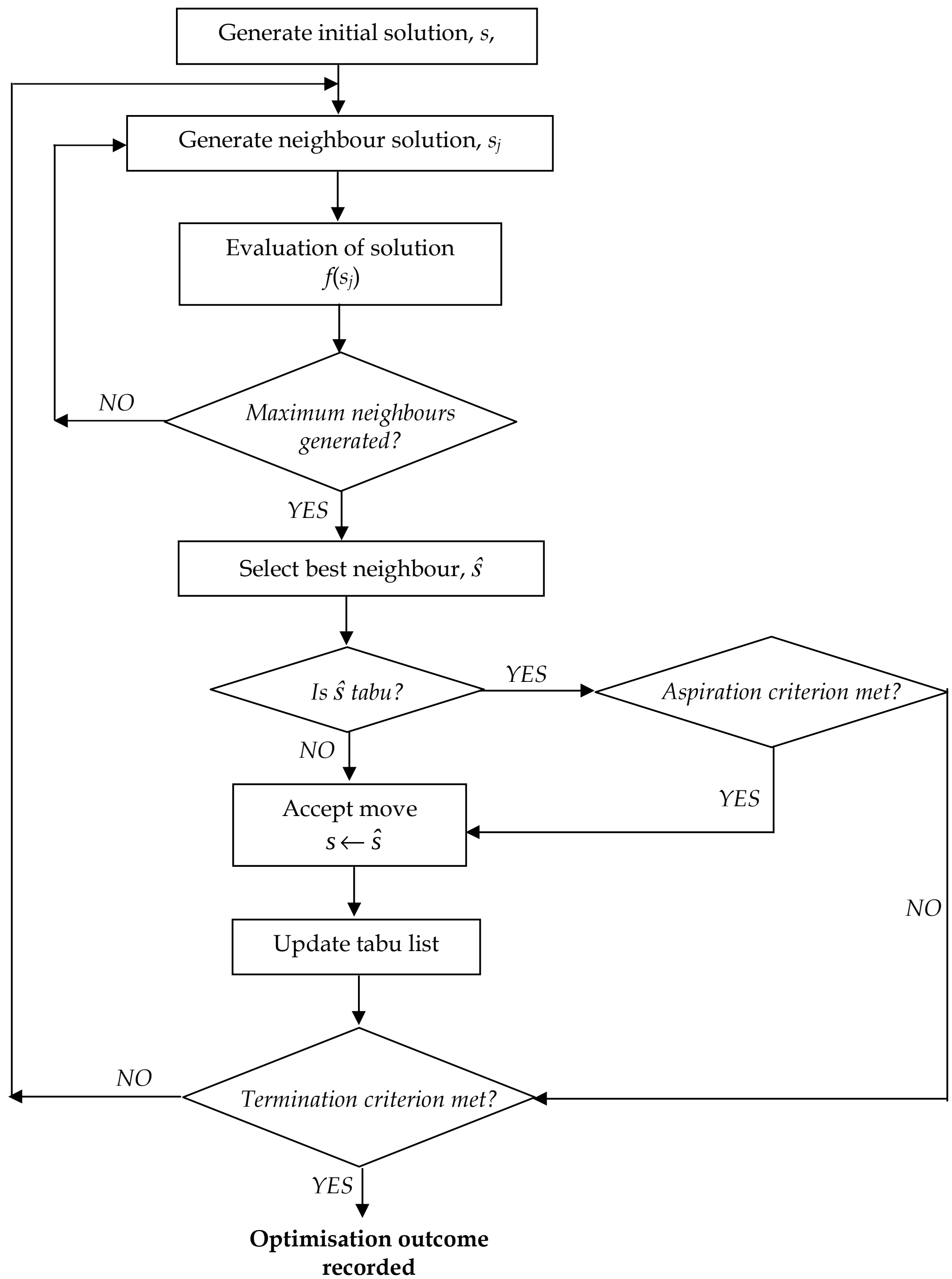


Figure 2.4: Typical Tabu Search (TS) algorithm

Apart from the definition of a neighborhood structure, as required for any simple local search algorithm, the following parameters need to be defined in the application of TS to a combinatorial optimisation problem:

(1) The memory structure:

Short-term memory is the unique feature of TS that is mainly used to avoid reversal of local search moves and cycling of neighbour solutions. The memory used in TS is both explicit and attributive (Glover *et al.*, 1997). Explicit memory records complete trial solutions previously visited. For example, the 10 best found trial solutions are stored. On the other hand, attributive memory is used to record information on changes made by moving from one solution to another. These memories are stored in a Tabu List, which is updated after every move. As pointed out by Glover *et al*. (1997), the effect of memory utilization in TS may be viewed as modifying the neighborhood $N(s)$ of the current solution s. The modified neighborhood, denoted by $N^*(s)$, is essentially a selective record of the history of a search. As part of the memory structure of TS, the length of tabu lists (tabu tenure) must be specified. Tabu tenure can be fixed or dynamic throughout a TS run. The choice of an appropriate memory structure is deemed crucial for the success of TS when applied to any combinatorial optimisation problem (Glover *et al.*, 1997).

(2) The aspiration criterion:

The aspiration criterion is formulated to override the 'tabu-active' status of a move in a case where the move is thought to be beneficial to the search. A common aspiration criterion used is that any move that results in a solution that is better than any solution generated so far is permissible (Kim *et al.*, 1997).

(3) The termination criterion:

As with other optimisation methods, a termination criterion needs to be specified. Examples include the maximum number of iterations, the maximum CPU time and the maximum number of iterations during which solution quality does not improve significantly.

TS has been used extensively for a wide range of benchmark optimisation problems. More recent applications include manufacturing (Lei *et al.*, 2006), production planning (Baykasoglu *et al.*, 2006), electromagnetic design problems (Hajji *et al.*, 2005), real-world scheduling (Xu *et al.*, 2006) and chemistry studies (Blazewicz *et al.*, 2005). In power system and related research areas, TS, or the hybridized version of TS, have been applied to the optimal planning of power distribution systems (Ramirez-Rosado *et al.*, 2006), the unit commitment problem (Victoire *et al.*, 2005) and a long-term hydro-scheduling problem (Mantawy *et al.*, 2003). More importantly, TS has previously been applied to PPMSO as a stand-alone algorithm (El-Amin *et al.*, 2000) and as part of a hybridized algorithm (Kim *et al.*, 1997; Burke *et al.*, 2000).

2.2.5.2 Global search-based metaheuristics

Genetic algorithms

Genetic algorithms (GAs) are optimisation methods inspired by evolutionary adaptation in nature. They were introduced by Holland (1975) in the early 1970s and implemented for optimisation problems by Goldberg (1989) in the late 1980s. In terms of searching behaviour, simple GAs fall into the category of global optimisation methods, as trial solutions of a GA run are generated based on global information accrued throughout the search process. The optimisation mechanism of GAs can be briefly described as follows (Figure 2.5): GAs operate on a population of chromosomes, each representing a trial solution to the problem being solved. The fitness of a chromosome, which is normally defined to correspond to the criterion of the optimisation problem being solved, is evaluated. In each iteration (or generation), relatively fit chromosomes are selected to undergo a series of genetic operations to produce a population of offspring. In this way, better chromosomes (trial solutions) are evolved throughout the optimisation process, and the fittest chromosome(s) found during a GA run is/are regarded as the optimized solution.

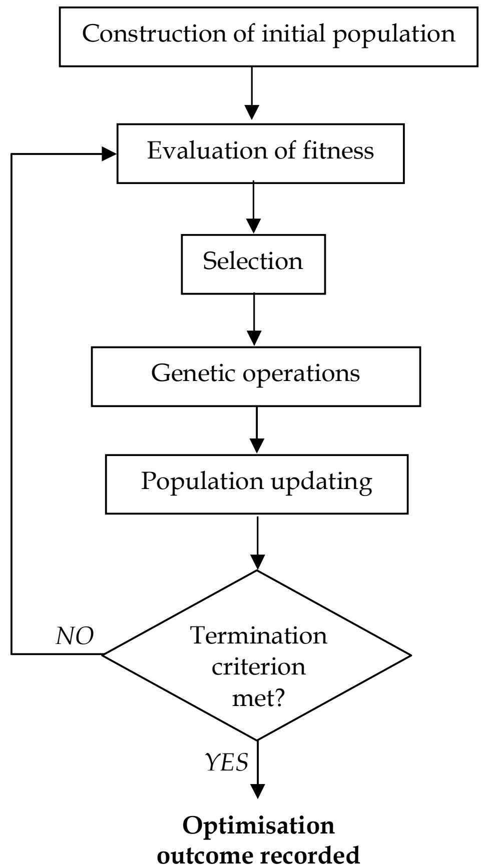


Figure 2.5: Typical simple genetic algorithm

Implementation of GAs for PPMSO problems requires the following issues to be resolved:

(1) Solution representation:

When adopting GAs for combinatorial optimisation problems, trial solutions to the problem are represented by strings of chromosomes, in which the solution parameters are encoded and stored (Aldridge *et al.*, 1999). The encoding scheme used to represent the solutions of a problem defines the fitness landscape of the problem search space (Dahal *et al.*, 2000). Binary encoding (strings of 0s and 1s) was adopted when GAs were originally developed, but this might not be suitable for all types of optimisation problems. Generally, an appropriate solution representation must be developed for an optimisation problem such that (a) the encoding and decoding of chromosome strings does not significantly increase the optimisation computational overhead, and (b) the new offspring generated by the genetic operators (eg. crossover and mutation) are feasible/near feasible.

The selection of an appropriate solution representation has been investigated in order to increase the performance of GAs in solving PPMSO problems. Binary coding was used by Kim *et al*. (1994), Kim *et al*. (1997), Burke *et al*. (1998) and Dahal *et al*. (1998). However, an integer representation was later found to be more appropriate for PPMSO (Dahal *et al*., 1997), as it respects maintenance window constraints and greatly reduces the size of the problem search space, when compared with binary coding. When an integer representation is used, a trial maintenance schedule is represented as a string of integers representing the maintenance start time of all generating units considered. A code-specific and constraint-transparent integral coding method that explicitly specifies the order in which maintenance tasks are carried out was proposed by Wang *et al*. (2000). However, despite an improvement in computational efficiency, a large number of the 'offspring' solutions produced were still found to be infeasible with the new coding scheme (Wang *et al*., 2000). Burke *et al*. (2000) considered using bit-string encoding, where the start period of each maintenance task is grey-coded, but due to a heavy computational requirement for encoding and decoding, an integer representation was adopted instead.

(2) Fitness function:

In GAs, whether or not a chromosome is selected for reproduction depends on its fitness function. Therefore, a fitness function that evaluates the quality of individual trial maintenance schedules must be specified beforehand. As PPMSO is a constrained problem, the overall fitness function comprises the objective and constraint violation terms. The merit of a trial solution (a trial maintenance schedule in PPMSO) is therefore evaluated based on the value of the calculated objective function value, which affects the probability that this trial solution is chosen to participate in subsequent genetic operations.

(3) Genetic operators:

In an attempt to explore the decision space of an optimisation problem, GAs operate on a population of trial solutions by iteratively modifying the components of chromosomes contained in the population. In particular, a number of chromosomes are selected to produce offspring chromosomes, which undergo a series of genetic operations, generally known as recombination.

Selection

In order for a population of chromosomes to evolve towards better solutions, 'parent trial solutions' are stochastically chosen, based on relative fitness, from the current iteration for the reproduction of 'offspring trial solutions'. Although trial solutions of higher fitness should be chosen by higher probability, selection pressure should not be too high to avoid premature convergence.

Crossover & mutation

'Parent trial solutions' selected are recombined to produce a new generation of 'offspring' trial solutions. Recombination methods commonly used include crossover and mutation. Crossover is performed by exchanging chromosome elements (genes) between selected parents, governed by a crossover probability. The objective of performing crossover is to obtain a better chromosome by exploiting partial information contained in two relatively good chromosomes. On the other hand, mutation is essentially a random change made, governed by a mutation probability, to part of a parent chromosome, and is therefore a means for further exploration of the problem search space to maintain solution diversity.

(4) Population updating method

Recombined chromosomes and parent chromosomes of the current generation are combined to form the next generation using the operations described above. Normally, the best chromosome(s) identified in the current generation is/are retained. More importantly, sufficient diversity should be maintained in any new generations to increase the likelihood of finding the global optimum.

(5) Termination criterion

A GA run is stopped when a prescribed maximum number of generations has been reached. Alternatively, termination criteria such as stagnation of the best-found objective function value (cost), can be adopted.

It can be seen that in order to implement GAs, a number of parameters are required to be defined beforehand, including the size of the population, a crossover probability, a mutation probability, a selection method, a population updating method and a termination criterion.

In contrast to SA, GAs generate a population of trial solutions every generation and perform their search from multiple starting points in the problem search space. As a result, the probability of being trapped in local optima is lower and multiple optimal/near-optimal solutions can be found. However, a shortcoming of GAs is that the search is rather coarse-grained and very often, only promising regions, but not the optimum of a search space, are identified. A detailed discussion regarding the implementation of GAs to combinatorial optimisation problems is presented in Reeves (1993). GAs have been applied extensively, either as a stand-alone algorithm or as part of a hybridized algorithm, to many research areas. Some research areas to which GAs have been applied recently include water distribution system design (Goldberg *et al.*, 1987; Simpson *et al.*, 1994; Halhal *et al.*, 1997), transportation (Caputo *et al.*, 2006; Gen *et al.*, 2006), steel-related research (Hodge *et al.*, 2006), remote sensing (Jubai *et al.*, 2006), manufacturing (Li *et al.*, 2006), magnetics (Lovat *et al.*, 2006), chemical process studies (Chang *et al.*, 2006; Sarkar *et al.*, 2006), web communications (Tug *et al.*, 2006), and power systems.

In power systems, various genetic algorithms have recently been proposed for power distribution planning (Pregej *et al.*, 2006), evaluation of power flow (Ting *et al.*, 2006; Todorovski *et al.*, 2006; Yan *et al.*, 2006), short-term load forecasting (Liao *et al.*, 2006), service restoration studies (Kumar *et al.*, 2006) and optimal meter placement problems (El-Zonkoly, 2006).

2.2.5.3 Previous applications of metaheuristics to PPMSO problems

In the study conducted by Satoh *et al*. (1991), a small, medium and large case study system was solved separately using a Simulated Annealing (SA) formulation and Integer Programming (IP). When IP is used, the results obtained are guaranteed to be globally optimal. However, IP is generally only applicable to relatively small problems for computational reasons. The study of Satoh *et al*. (1991) indicated that, for the small system investigated, the global optimum was found by using both SA and IP. For the medium-sized system, the solution obtained by SA was better than that given by IP, which was not optimal, as termination was executed due to long run-times. Finally, by using SA, a solution to the large-sized system was found, which could not be solved by IP from a computational point of view (Satoh *et al*., 1991).

Aldridge *et al*. (1999) applied a genetic algorithm to a case study that involves maintaining 21 generating units over a planning horizon of 52 weeks. Results showed that the GA formulation was able to outperform simple heuristic methods tested in the study. The same case study system was later investigated by Dahal *et al*. (1999) by examining the performance of a GA-fuzzy logic hybrid algorithm for PPMSO. The fuzzy logic approach, which is able to include knowledge-based experience in the problem formulation, resulted in a better objective value (in terms of cost and reliability), although there were slight violations of manpower constraints. A SA formulation was compared to the GA's in relation to solving the 21-unit case study by Dahal *et al*. (2000). It was found that while the performance of SA is mainly affected by the cooling schedule, the GA requires many more parameters to be defined empirically. Overall, both the GA and SA outperformed the two simple heuristic methods tested in the study. Apart from being used as a stand-alone algorithm for PPMSO problems, the GA algorithm was also modified by the fuzzy system formulation proposed by Huang (1998) in order to optimise the parameters required for the construction of membership functions of objectives and constraints.

A Tabu Search (TS) formulation was applied to both a 4-unit and a 22-unit case study systems by El-Amin *et al*. (2000). The objective function costs associated with the best-found maintenance schedules for these case studies were not reported, but were calculated based on the information provided as part of this thesis. Interestingly, the results given by the TS formulation (LVL = 256.93MW) are worse than those obtained by Escudero *et al*. (1980) using implicit enumeration (LVL = 118.81MW) for the 22-unit case study.

SA, GA and TS are based on different search philosophies and are therefore differentiated by unique optimisation mechanisms. While the acceptance of non-improving solutions in SA and the tabu lists in TS are used to avoid becoming trapped in local optima, GAs perform a coarse-grained search for promising regions of a problem search space. In view of these characteristics, hybridization of these metaheuristics has been proposed and claimed to successfully overcome drawbacks and utilize the positive features of individual methods (Song, 1999).

A study comparing the impacts of incorporating a SA, a TS and a hill-climbing algorithm into a GA was carried out by Burke *et al*. (2000). It was concluded that a GA employing a TS operator is the most effective method. In the hybridised algorithm, the GA was responsible for identifying a trial solution that is not too far from the optimum and TS was used to locate the optimum by searching the neighbourhoods of the solution given by the GA. The concept of using a local search algorithm to refine the solutions given by a global optimisation method is similar to what was termed 'memetic algorithm' by Moscato (1989) later in 1989.

Kim *et al*. (1995) used the acceptance probability of SA to improve the convergence speed of GAs, resulting in a GA+SA algorithm. However, it was found that the genetic operators in GAs have difficulties in finding the optimum solutions. In order to improve the optimisation ability of the hybrid algorithm, Kim *et al*. (1997) hybridized TS with the GA+SA algorithm to include the features of global and local search in one algorithm. The hybridized algorithm was tested on a 23-unit test system and found to improve upon the results obtained by a simple GA, a simple SA, as well as the GA+SA algorithm. However, the performance of the GA+SA+TS algorithm could not be verified by applying mathematical programming, as the size of the case study would be too large for the latter method, again highlighting the shortcomings of using mathematical programming for PPMSO.

2.2.6 Comparison of optimisation methods for PPMSO

Table 2.1 assesses the four optimisation methods for PPMSO presented in this section against the seven performance criteria outlined in Section 2.2.1.

Table 2.1: Summary of optimisation methods for PPMSO

	Heuristic	Mathematical programming	Expert system	Metaheuristic
Simple to implement?	NO	NO	NO	YES
Easily incorporate a simulation model?	NO	NO	NO	YES
A priori information required?	YES	YES	YES	NO
Effective handling of constraints?	YES	YES	YES	YES
Easily adapt to changes in a problem?	NO	NO	NO	YES
Obtain more than one desired schedule?	NO	NO	NO	YES
Find good solutions in reasonable computational time?	NO	NO	NO	YES

Among the four categories of optimisation method categories presented, metaheuristics satisfy all the criteria outlined for an ideal maintenance-scheduling tool (Table 2.1). In particular, they present the following advantages and therefore appear to be the most promising approach for PPMSO:

- Metaheuristics are not problem-specific. They can be applied to a wide range of optimisation problems with only little modification.

- Metaheuristics are approximate algorithms that sacrifice the guarantee of finding the exact solution(s) in exchange for the ability to find near-optimal solutions within a reasonable computational time. This is especially important when solving real-world PPMSO problems, which are mostly large in size and contain a high degree of complexity.

- Metaheuristics can be linked easily with a simulation model. Therefore there is no need for representing a power system by simplified mathematical equations in the optimisation algorithm.

- Global search-based metaheuristics perform extensive, coarse-grained search and are therefore able to find multiple promising regions of a search space simultaneously.

- Local search-based metaheuristics can identify optimum points of a problem search space by performing small moves within different solution neighbourhoods.

- Different metaheuristic methods can be easily hybridised to take advantage of the positive features of individual methods.

- Global search-based metaheuristics, such as GAs, work on a population of trial solutions and may therefore obtain more than one schedule associated with the best-found objective function cost for a problem. A decision maker can then choose between these schedules based on some non-quantitative objectives (e.g. political), or as part of negotiations with asset managers, for example.

Despite their strengths for solving PPMSO problems, the metaheuristics that have previously been used for PPMSO have the following shortcomings:

(1) Depending on the nature of individual metaheuristics, some constraints cannot be taken into account explicitly, necessitating the use of other constraint-handling methods such as penalty functions. Penalty functions often require more parameters to be specified in addition to those contained in the metaheuristics. In addition, the inability to avoid the construction of some infeasible trial solutions results in computational inefficiencies.

(2) Many realistic PPMSO problems have very large search spaces, which results in high computational loads and makes it difficult to find globally optimal solutions. However, in most instances, heuristic information exists that would enable the search to be directed

towards promising regions of the search space, thereby increasing computational efficiency and the chances of finding near-globally optimal solutions. Although heuristic information has been used to generate initial populations of trial solutions (Dahal *et al.*, 2000), intrinsically, commonly used metaheuristics such as SA, TS and GAs, are unable to incorporate heuristic information into their search.

(3) The best parameter sets used in an algorithm have to be determined for individual optimisation problems. Most metaheuristics require repetitive tuning of parameter settings before being used and hence can be computationally inefficient.

Ant colony optimisation (ACO) is a relatively new global search-based metaheuristic that has been gaining increasing popularity for combinatorial optimisation problems since 1990s. Despite being driven by similar "evolutionary forces" as the GAs, ACO is deemed more suitable for PPMSO due to its ability to overcome some of the drawbacks of other metaheuristics discussed above, including:

(1) The decision tree-based solution construction mechanism of ACO allows some constraints to be addressed explicitly during the construction of trial solutions. The advantages of this are two-fold: (1) Some of infeasible trial solutions are avoided, thereby reducing the problem search space that needs to be assessed during thr optimisation process; and (2) There is a decreased need to use penalty functions, as some constraints are dealt with explicitly. This feature of ACO is particularly advantageous for solving optimisation problems that involve sequential decision making, such as PPMSO.

(2) The use of heuristic information is imbedded in the ACO algorithm as an optional mechanism. In this way, the preference of a decision maker, based on past experience, can be reflected throughout the optimisation process in order to find better solutions within reduced computational runtime.

Due to the advantages mentioned above, the potential utilisation of ACO for PPMSO is deemed worthwhile to be further investigated in this research.

2.3 Summary and conclusions

In this chapter, two aspects of PPMSO have been reviewed: (1) The definition of PPMSO adopted in the literature and (2) The optimisation methods applied to PPMSO previously.

Various objectives adopted in past studies on PPMSO have been categorised as reliability-based and cost-based criteria. Commonly encountered constraints in PPMSO have also been presented. Following the increasing popularity of electricity market deregulation, its impacts on the practice of many power utilities, especially in relation to the objectives and constraints used for PPMSO, have been discussed.

Optimisation methods previously used for PPMSO have been divided into four categories. Heuristic approaches, mathematical programming and expert systems played an important role in solving PPMSO problems when the optimisation problem was first investigated more than a decade ago. These methods usually suffer from shortcomings such as the inability to handle non-linearity objectives and constraints, requiring impractical computational overhead and having difficulties in adapting to changes made to a power system. In order to overcome these drawbacks, metaheuristics have been proposed and appear promising for solving PPMSO. However, despite their advantages over more traditional optimisation methods, commonly used metaheuristics, such as SA and GAs, have a number of shortcomings in relation to their application to PPMSO. These include the inability to account for heuristic information and constraints explicitly. Ant colony optimisation overcomes some of these shortcomings of more commonly used metaheuristics, and will therefore provide the focus of the remainder of this thesis.

Chapter 3
Ant Colony Optimisation Metaheuristic

"A metaheuristic is a general algorithmic framework which can be applied to different optimisation problems with relatively few modifications to make them adapted to a specific problem." - (Dorigo et al., 2004a)

"A metaheuristic refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality." - (Glover et al., 1997)

Proposed by Dorigo in 1992 (Blum *et al.*, 2003), the Ant Colony Optimisation (ACO) metaheuristic can be seen as a higher-level optimisation strategy that adopts the basic mechanisms underlying the foraging behaviour of ant colonies, which are enhanced by artificial intelligence techniques.

The objective of this chapter is to introduce the Ant Colony Optimisation metaheuristic. Section 3.1 reviews the historical background of ACO, including its origin based on the behaviour of real ants and the additional features given to artificial ants in order to solve complex optimisation problems. Subsequently, various issues regarding the implementation of ACO are addressed, including the representation of a combinatorial problem, a general framework of the ACO metaheuristic and the prerequisites of an ACO application. In Section 3.3, five different ACO algorithms that grew out of the ACO metaheuristics are presented. Various ACO applications in the literature are reviewed in Section 3.4, mainly focusing on benchmark scheduling problems and real-world optimisation problems. Lastly, the characteristics that contribute to the choice of ACO for solving the power plant maintenance scheduling problem in this research are discussed.

3.1 From Real Ants to Artificial Ants

3.1.1 Foraging behavior of real ants

Ant Colony Optimisation (ACO) was inspired by the behaviour of ant colonies searching for the shortest route to a food source. Although ants are almost blind (Deneubourg *et al.*, 1983) and thus a single ant has limited capabilities, ants in colonies exhibit *foraging behaviour*[1] to find the shortest distance between their nest and the food (Dorigo *et al.*, 2004b).

When an ant encounters an intersection (e.g. an obstacle) that has two possible routes (Figure 3.1), it locates the shortest possible route via *pheromone* laid by previous ants, as ants following a path will deposit some pheromone on that path. Ants detect the concentration of pheromone on each path and tend to choose, by probability, the path with the higher intensity of pheromone (Dorigo *et al.*, 1991).

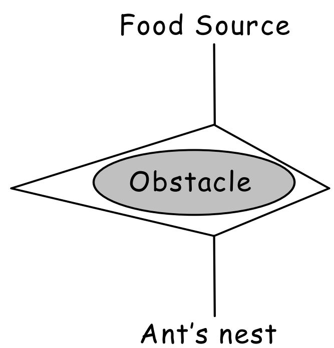
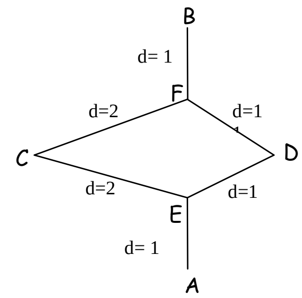


Figure 3.1: Path from nest to food source

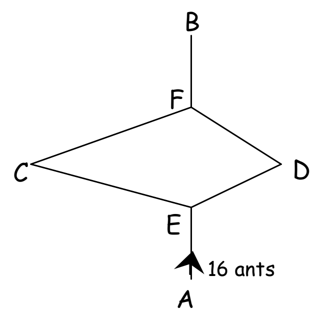
For a better understanding, the following diagrams (Figures 3.2a to 3.2h) are shown to illustrate the ants' behaviour when searching for the shortest route (Foong *et al.*, 2000). The symbol τ is the pheromone trail intensity in unit concentration and d is the unit distance. To simplify the illustration, the ants are assumed to move one unit distance, d, for each unit time, t, and to deposit one unit of τ after reaching the next node. An ant's complete trip consists of reaching the food source from the nest and returning to the nest from the food source. The numbers of ants are shown in brackets in the diagrams.

[1] Foraging behaviour is the behaviour of ants exploring a large area.



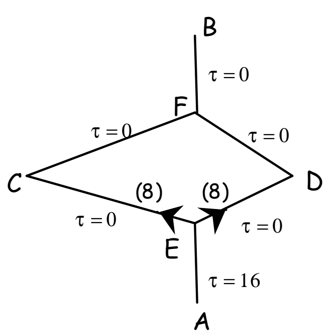
(a)

The length of each path is shown in unit distance. Note that the two possible routes from the nest to the food, namely AEDFB and AECFB, have lengths *d* of 4 and 6 units, respectively.



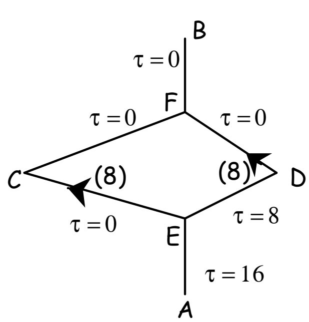
(b) *t* = 0

At t = 0, 16 ants are assumed to depart from A and each ant moves 1 unit distance per unit time, depositing 1 unit of pheromone per unit distance along the paths they are following. In other words, the intensity of pheromone on a path is equal to the number of ants that have traversed the path.



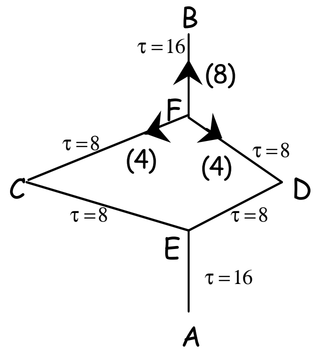
(c) *t* = 1

At t = 1, when the ants arrive at E, the probability of ants choosing the left or right path is the same, as there is no previous pheromone deposited on the trail. As a result, it is assumed that 8 ants follow path EC and 8 ants follow path ED.



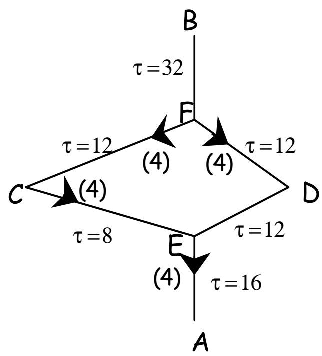
(d) *t* = 2

At t = 2, the 8 ants following path ED have already reached D and have deposited 8 units of pheromone on ED. Since path EC is twice as long as path ED, the ants following path EC have not reached C at t = 2, thus, pheromone has not been laid on the entire EC trail.



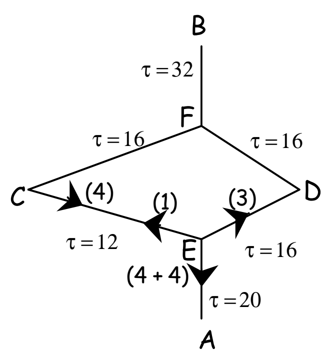
(e) ***t*** **= 5**

As path EDF is shorter than path ECF, the ants following path EDF will reach the food at node B faster at t = 4. The 8 ants with food return to F at t = 5, meeting the ants following path ECF at the same node. At this stage, the pheromone intensity on FD and FC are both 8, thus, by equal probability, 4 returning ants choose path FC and the other 4 choose path FD.



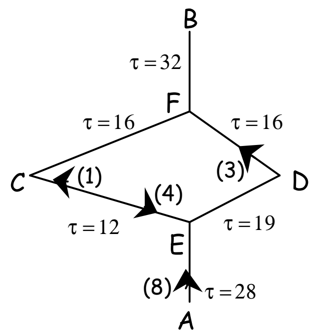
(f) ***t*** **= 7**

At t = 7, the 8 ants that have taken the longer way (ECF) have reached the food and are making their way back to nest, when they encounter paths FC and FD at F with the same amount of pheromone intensity (τ = 12). Again, by equal probability, 4 ants will choose FC and another 4 ants will choose FD. At the same time, the 4 ants that have chosen the shorter path have reached node E and are ready to return to the nest to complete their trip.



(g) ***t*** **= 9**

At t = 9, the first 4 returning ants have reached their nest and start a new trip back to E where they have to make a decision again. Due to the different path lengths, the pheromone concentration on ED (τ = 16) is higher than that on EC (τ = 12), corresponding to probabilities that these paths will be chosen of 57% and 43%, respectively. Stimulated by the stronger pheromone intensity, more ants (say 3, by probability) select path ED, laying more pheromone on this trail than on EC, again reinforcing the shorter route.



(h) *t* = 10

At t = 10, the pheromone intensity on ED (τ = 19) is higher than the intensity on EC (τ = 12). In this way, the action of an ant choosing a particular path and laying pheromone on that path increases the probability of the same path being chosen by future ants. This is referred to as an *autocatalytic* process (Dorigo *et al.*, 1991). Details of the pheromone intensity updated on each path are given in Table 3.1.

Figure 3.2: Illustration of the ants' foraging behaviour

Table 3.1: Pheromone intensity updated on paths

t (unit time)	**Pheromone trail intensity, τ (unit concentration)**					
	AE	**ED**	**DF**	**EC**	**CF**	**FB**
0	0	0	0	0	0	0
1	16	0	0	0	0	0
2	16	8	0	0	0	0
3	16	8	8	8	0	0
4	16	8	8	8	0	8
5	16	8	8	8	8	16
6	16	8	12	8	8	24
7	16	12	12	8	12	32
8	20	12	16	8	12	32
9	20	16	16	12	16	32
10	28	19	16	12	16	32

Shorter route (AE, ED) — Longer route (EC, CF)

As the process illustrated above continues, the difference in pheromone concentration between the shorter and longer route will increase, and eventually most ants will follow the shorter path. As a summary, a shorter route allows more ants to travel on it than a longer path does in a limited time span. Figures 3.2a to 3.2h demonstrate a simplified version of the ants' foraging behaviour. In reality, there will be more than two readily provided possible routes. Consequently, ants will continuously explore the search space to find the shortest possible route. In this process, *evaporation* of pheromone plays an important role. Pheromone will be gradually evaporated and the trail not utilised much will eventually disappear. Hence, this eliminates the possibility of ants following the longer and less favourable routes. In addition, evaporation avoids premature convergence to a frequently-travelled path in the early stages of the search process and therefore allows a continuous exploration for new routes that may be shorter than the ones explored previously.

3.1.2 Artificial ants

The inspiration derived from the foraging behaviour of real ants, after the undertaking of extensive experimentation, has been transformed into a strategy that can be used to solve complex optimisation problems. While the readers are referred to the first chapter of Dorigo *et al*. (2004b) for a detailed coverage of this topic, the final outcome of the transformation process is described in this section.

The ant agents used in the ACO metaheuristic (referred to as ACO hereafter) are generally known as 'artificial ants'. In contrast to their natural counterparts, artificial ants are given the following additional abilities to solve more complex real-world optimisation problems:

(1) Visibility: Artificial ants are given 'visibility' when they encounter an intersection. With this given artificial intelligence, ants are able to judge the distances of different paths at the intersection so that shorter paths are more favourable.

(2) Memory: Real ants are assumed have no memory and make decisions based only on the pheromone intensities of decision paths. In contrast, memory is given to artificial ants for storing records of previously visited paths.

(3) Higher pheromone evaporation rate: Pheromone evaporation reduces the intensity of all pheromone trails by an amount directly proportional to the intensity. Consequently, it can be seen as a means of encouraging exploration of unvisited paths by reducing the overall gap between pheromone trail intensities. Pheromone evaporation also takes place

during the foraging process of real ants, but at a much slower rate. In contrast, higher evaporation rates are suggested for artificial ants, especially when solving more complex problems (Dorigo *et al.*, 2004b).

(4) Daemon actions: Daemon actions are those actions that cannot be performed by an individual ant, for example, additional pheromone being laid on the shortest route found so far, and is optional in the ACO metaheuristic.

3.2 ACO for Combinatorial Problems

Before any optimisation metaheuristic is applied to solve a combinatorial optimisation problem, it is essential that the problem can be represented in a form that is recognizable by the metaheuristic. The objectives of this section are to introduce the representation of a general combinatorial optimisation problem, to introduce the main mechanisms of the ACO metaheuristic and the adaptations that need to be made prior to the application of ACO to a combinatorial problem.

3.2.1 Problem representation

Consider a combinatorial minimization problem (S, f, Ω) where S is the set of trial solutions, f is the objective function that assigns an objective function cost f(s), $s \in S$, and Ω is a set of constraints. The aim of the problem is to find a globally optimal set, S^*, of solutions such that $f(s^*) \leq f(s)$, where $s^* \in S^*$ and $S^* \subseteq S$. The optimal solutions must also satisfy all constraints contained in set Ω. In order to apply ACO to the optimisation problem, a link between the two must be established. In general, a problem representation with the following characteristics is adopted (Dorigo *et al.*, 2002):

- A finite set of N_c components $C = \{c_1, c_2, ..., c_{N_c}\}$ and a set J of arcs fully connecting the components contained in C.

- The states of the problem are defined in terms of sequences $x = \langle c_i, c_j, ..., c_k, ... \rangle$ over the components contained in C, $x \in X$, where X is the set of all possible sequences. The length of a sequence i.e. the number of components contained in a sequence x, is termed $|x|$. A sequence x is equivalent to a complete trial solution s if $|x|$ = D, where D is the total number of decision variables and $s \in S$. A complete trial solution is called 'trial solution' for short, and the sequences contained in the set $X \setminus S$ are incomplete trial solutions, or 'partial solutions' for short. It should be noted that a trial solution s is not necessarily feasible with respect to constraint set Ω.

- A finite set $\tilde{S}$ of feasible trial solutions is defined by the set of constraints Ω, where $\tilde{S} \subseteq S$.

- A cost $f(s)$ is associated with each trial solution s. In some problems, it is possible to calculate the partial cost $f_p(x)$ associated with the state x (partial solution x) of a problem.

Having the problem representation established, artificial ants can then incrementally construct trial solutions by exploiting the construction graph $G(C, L)$ (Dorigo *et al.*, 2004b), as part of the procedures contained in the ACO metaheuristic.

3.2.2 The ACO metaheuristic: a general framework

The basic form of the ACO metaheuristic can be described as the interplay among the following procedures (Dorigo *et al.*, 2004b):

(i) Ant activities: In this procedure, artificial ants incrementally construct trial solutions to the problem being solved. Starting from an empty sequence, $|x| = 0$, an artificial ant progressively adds components to the sequence by moving on the construction graph $G(C, L)$. An ant currently at component c_i chooses which component from set C to visit next (that is, a component to be added to its sequence) by utilizing a *random proportional rule*. In general, the probability of an ant currently at c_i travelling to c_j (the next component in its sequence) is directly proportional to the pheromone trail intensity and heuristic information associated with the move. The pheromone concentration on the arc connecting c_i and c_j is a reflection of the ant colony's acquired experience about this connection based on trial solutions previously generated in the current optimisation search. On the other hand, the heuristic or 'visibility' is the estimated quality of the individual arc, which incorporates the user's knowledge about the problem at hand. A trial solution s to the problem is obtained when the length of the sequence reaches the total number of variables D. An objective function cost $f(s)$ associated with the trial solution is then calculated.

(ii) Pheromone updating: Pheromone updating involves two different mechanisms, pheromone deposition and pheromone evaporation. The general idea behind pheromone deposition is to reward the arcs that connect the components contained in a trial solution based on the objective function cost of the solution. Pheromone evaporation is a process where all pheromone trail intensities contained in a problem search space are decreased by a factor, hence reducing the difference in pheromone intensities among arcs. In this way, a scenario where certain arcs are travelled much more frequently than others, can be avoided, hence increasing the probabilities of unvisited arcs being visited.

(iii) Daemon actions

As an optional procedure in the ACO metaheuristic, daemon actions implement centralized actions that cannot possibly be performed by single ants (Dorigo *et al.*, 2004b). Daemon actions can take the form of (a) a local search procedure, which searches for the local minima of the neighbourhood of solutions given by ACO or/and (b) global information that can be used to further bias the optimisation search. For example, the components of the best solution found so far can be rewarded an additional amount of pheromone.

It should be noted that the scheduling and synchronization of the three procedures to be executed are not specified in the metaheuristic, allowing them to be tailored to the problems at hand. In summary, the ACO metaheuristic is an optimisation process whereby a population of artificial ants generates trial solutions by exploiting information distributed over a search space, and at the same time, iteratively modifying the search space environment to reflect the artificial ants' search experience. In this way, the search is gradually biased towards promising areas of the search space.

3.2.3 Prerequisites of ACO implementation

The ACO metaheuristic outlined in the previous section is a high-level algorithmic framework that needs to be customized to solve a specific optimisation problem. In this thesis, the adaptations made in order to apply the ACO metaheuristic (Section 3.2.2) to a problem P (Section 3.2.1) is referred to as 'an ACO formulation for problem P'. In this section, the issues required to be resolved as part of an ACO formulation, as suggested by Dorigo *et al.* (2004b), are discussed:

(1) Construction of a trial solution

In the search for optimal solution(s) to a specific problem using any metaheuristic, a number of 'candidate solutions' are constructed and evaluated during the optimisation process before one or more 'best-found solution(s)' is/are obtained. These candidate solutions are called 'trial solutions', while the latter is/are called the 'lowest-cost solution(s)'.

Given a problem representation (see Section 3.2.1), a construction graph G = (C, L) is utilized by artificial ants in building trial solutions to an optimisation problem. Starting from scratch, a trial solution is constructed by adding solution components, one at a time, to a partially completed trial solution. For example, the Traveling Salesman Problem (TSP) is a combinatorial optimisation problem in which a salesman is given k cities and he has to visit each city once and finally return to the starting city. In previous studies of TSP using ACO, component set C is defined as the set of cities c_i, c_j, …, c_k given to the salesman, and connection set L is a set of all arcs connecting any two cities, which include $\tilde{l}_{ij} \subseteq L$ as the set of optional paths connecting city c_i and c_j. During the construction of a trial solution, an ant currently at city c_i chooses the next city to visit by implementing a *random proportional rule*, which takes into account the pheromone trails and heuristic values associated with all arcs connecting c_i and other unvisited cities. The city chosen is then added to the partial solution of the ant.

(2) Definition of pheromone trails

During an optimisation process using ACO, artificial ants utilise the information about the decision variable space[2] of a problem captured by pheromone trails and at the same time, iteratively modify the pheromone trails to reflect their search experience. In other words, the ant search process is mainly driven by the distribution of pheromone trails over the problem decision space. Therefore, an appropriate definition of pheromone trail, which is normally problem-specific, must be given. Studies comparing the effectiveness of different pheromone trail definitions concluded that a bad choice of such a definition has an adverse effect on the optimisation outcome. Two different pheromone trail definitions for the TSP have been investigated by Dorigo *et al*. (2004b): (i) a pheromone trail τ_{ij} is interpreted as the desirability of visiting city *j* directly after a city *i* and (ii) τ_{ij} is interpreted as the desirability of visiting city *i* as the *j*th stop during the salesman's tour. It has been shown that since the relative order of the city being visited is more significant in solving TSP, pheromone trail definition (i) is more effective.

(3) Heuristic formulation

Prior knowledge about a problem can be incorporated into an ACO formulation by means of heuristic information, which is taken into account during the construction of trial solutions. During the early stages of an ant's search, before pheromone trails are significantly distinct, heuristic information is the dominant factor affecting the selection of decision paths. In other words, heuristic information provides the optimisation search with a prediction of regions within the search space in which promising solutions are located. Without heuristic information, the initial search would almost be random until dominant paths are established during the latter stages of the search. On the other hand, if heuristic information is heavily emphasized, the behaviour of ACO would be similar to that of a greedy algorithm. As the way in which heuristic information is represented mathematically is problem specific (Dorigo *et al*., 1999), transformation of any heuristic information into a formulation to be used in the ACO algorithm is an important task. A heuristic value

[2] The decision variable space is the n-dimensional space associated with values of the decision variable, x. This is different from the objective function space, which is defined as the m-dimensional space associated with the m objective functions. Two different points in the decision variable space may be mapped to the same point in the objective function space.

given to a solution component can be static or dynamic. In the static case, the heuristic value is defined *a posteriori* and remains fixed throughout the ACO run. On the other hand, dynamic heuristic values of a solution component are calculated based on the current partially built trial solution.

(4) Local search (optional)

Local search can be used optionally as a form of daemon action in the ACO metaheuristic. Local search has been found to result in significant improvements when coupled with ACO for a number of ACO applications (Dorigo *et al.*, 1997b; den Besten *et al.*, 2000) and little in others (Merkle *et al.*, 2002). In general, local search and ACO are conjectured to complement each other in the following way: The ACO metaheuristic is a global-search based metaheuristic that identifies promising regions of a problem search space, whereas local search can perform a detailed search within these promising regions to determine the optimum solution(s) to the problem. Local search has also been used to enhance the performance of the ACO metaheuristic in solving an optimisation problem when heuristic information about the problem at hand is not easily obtained (Dorigo *et al.*, 2004b).

The definition of each of these issues in the development of an ACO formulation is clearly demonstrated using previous ACO applications in Section 3.4.

3.3 Variants of Ant Colony Optimisation algorithms

Despite its original inspiration from the foraging behaviour of ant colonies, various ACO algorithms have evolved. It should be noted that 'ACO metaheuritsic' refers to the higher-level algorithmic framework, which is customized and refined by algorithm designers to form various 'ACO algorithms'. An ACO algorithm consists of the details of the optimisation mechanism that can be executed to solve an optimisation problem.

The three earliest ACO algorithms, namely ant-cycle, ant-density and ant-quantity, were proposed by Dorigo in 1992 as part of his doctoral dissertation (Dorigo, 1992). As ant-cycle was found to outperform its two counterparts, it was regarded as the first ACO algorithm in existence and called the Ant System (AS). It is interesting to note that the ACO metaheuristic introduced in Section 3.2.2 was defined *a posteriori*, rather than before the existence of AS, as a result of extensive studies conducted by Dorigo *et al*. (1999; 2002). The ACO metaheuristic has since provided a general framework for the design of new ACO algorithms.

In this section, the classification technique of ACO algorithms adopted by Dorigo *et al*. (2004b) is used. The first category comprises AS and its direct successors, including Rank-Based Ant System (AS_{rank}) and Max-Min Ant System (MMAS). AS_{rank} and MMAS are largely similar to AS and can be described using the algorithmic framework shown in Figure 3.3. ACO algorithms that cannot be completely described by the framework in Figure 3.3, such as the Ant Colony System (ACS) and the Hyper-Cube Framework for ACO (HCF), belong to the second category. While inspired by AS, these algorithms utilize additional mechanisms that aim to improve the exploitation and exploration features of simple AS-based algorithms. In particular, the synchronisation of the ACO procedures (Section 3.2.2) of ACS and HCF are quite different to those of AS, AS_{rank} and MMAS.

3.3.1 Ant System (AS) and its direct successors

The Ant System (AS) is the first of all ACO algorithms and more importantly, the one that leads to the definition of the ACO metaheuristic outlined in Section 3.2.2. Various ACO algorithms have since been developed by slightly modifying AS, with the goal of improving its optimisation ability. As mentioned previously, ACO algorithms of this kind can be generally described by the framework shown in Figure 3.3. Two major ACO algorithms that belong to this group are the Rank-Based Ant System (AS_{rank}) and the Max-Min Ant System (MMAS).

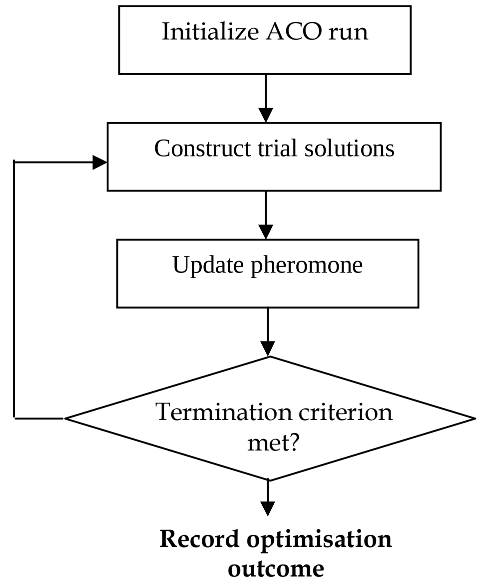


Figure 3.3: Ant-System-based algorithmic framework

Ant System (AS)

Construction of trial solutions: In AS, *m* ants construct trial solutions in parallel, utilizing the *random proportional rule* (Eq. 3.1). It should be noted that the ants are given memory to store the partial solutions constructed. Given a partial solution sequence $x = \langle \ldots, c_i \rangle$, the probability that an ant *k* adds c_j as the next component in *x* given by:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in L_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \tag{3.1}$$

where τ_{ij} is the pheromone trail of arc(*i*, *j*); η_{ij} is the heuristic information of arc(*i*, *j*); α and β are the parameters that control the relative importance of pheromone and heuristic, respectively; L_i^k is the set of optional components considered by ant *k* given a partial solution $x = \langle \ldots, c_i \rangle$.

Pheromone updating: In AS, pheromone evaporation reduces all existing pheromone trails by a factor, given by:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} \text{ for all } i, j \tag{3.2}$$

where τ_{ij} is the pheromone trail of arc(i, j) and $0 < \rho < 1$ is the pheromone evaporation rate.

After pheromone evaporation, *all* ants deposit pheromone on the arcs they have followed, the value of which is given by:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \text{ for all } i, j \tag{3.3}$$

where $\Delta\tau_{ij}^{k}$ is the pheromone deposited by ant k on arc(i, j), the value of which is given by:

$$\Delta\tau_{ij}^{k} = \begin{cases} 1/OFC^{k}, & \text{if arc}(i, j) \text{ belongs to } s^{k} \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

where OFC^k is the objective function cost of the trial solution constructed by ant k, s^k. In other words, the pheromone rewarded/deposited on the arcs constituting a trial solution is higher if the objective function cost of the trial solution is lower.

AS was first proposed in the context of solving the Travelling Salesman Problem (TSP). In spite of the encouraging results obtained by AS in its application to TSP, it was found to be inferior to other state-of-the-art optimisation algorithms that had been applied to the problem (Dorigo *et al.*, 2002). Many ACO algorithms have since been proposed with the goal of improving the performance of AS, such as Rank-Based Ant System (AS_{rank}) and Max-Min Ant System (MMAS), for example, which differ slightly from AS and generally can be described by the algorithmic framework shown in Figure 3.3. In particular, AS_{rank} and MMAS are different from AS mainly in the way pheromone deposition is carried out. The particulars of the two algorithms with respect to pheromone deposition are given below:

Rank-Based Ant System (AS_{rank})

Rank-Based Ant System (AS_{rank}) was proposed by Bullnheimer *et al.* (1999). In contrast to AS, where pheromone deposition applies to *all* ants, only g ants are rewarded in each iteration in AS_{rank}, where g is a user-defined parameter. It should be noted that g includes the best-so-far trial solution (s^{bsf}). Consequently, in each iteration, (g - 1) trial solutions are ranked by increasing objective function values. Eq. 3.3 is thus changed to:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{r=1}^{g} (g-r)\Delta\tau_{ij}^{r} + g\Delta\tau_{ij}^{bsf} \tag{3.5}$$

where the values of $\Delta\tau_{ij}^{r}$ and $\Delta\tau_{ij}^{bsf}$ are given by:

$$\Delta\tau_{ij}^{r} = \begin{cases} 1/OFC^{r}, & \text{if arc}(i,j) \text{ belongs to } s^{r} \\ 0, & \text{otherwise.} \end{cases} \tag{3.6}$$

$$\Delta\tau_{ij}^{bsf} = \begin{cases} 1/OFC^{bsf}, & \text{if arc}(i,j) \text{ belongs to } s^{bsf} \\ 0, & \text{otherwise.} \end{cases} \tag{3.7}$$

It can be seen that the best-so-far ant always deposits the most pheromone with weight g, while the other (g - 1) ants in an iteration deposit a quantity of pheromone in proportion to the objective function costs and ranks of tours.

Elitist-Ant System (EAS) is the first ACO algorithm introduced by Dorigo (1992) and Dorigo *et al*. (1996) as an improvement to AS, as part of which an additional quantity of pheromone is deposited on the arcs contained in the best-so-far trial solution (s^{bsf}) in each iteration. This can be viewed as a special case of AS_{rank}, whereby:

- g is the size of the ant population used;
- the best-so-far trial solution (s^{bsf}) is ranked 1 while all trial solutions are ranked 2.

Experimental analysis carried out by Bullnheimer *et al*. (1999) showed that AS_{rank} performs slightly better than the Elitist-Ant System and significantly better than AS.

<u>Max-Min Ant System (MMAS)</u>

A significant improvement over AS was achieved by the introduction of the Max-Min Ant System (MMAS) (Stützle *et al.*, 1997; Stützle *et al.*, 2000), which was proposed to enhance both the exploitation and exploration features of AS. Two important features of MMAS that contribute to the robustness of the algorithm are:

(1) Updating only the best-so-far (s^{bsf}) or the iteration-best (s^{ib}) trial solution:

As a measure to better exploit the artificial ants' search experience, pheromone is deposited only on the arcs belonging to either the best-so-far trial solution (s^{bsf}) or the best trial solution of the current iteration (s^{ib}). The corresponding pheromone deposition equation is:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^{best} \tag{3.8}$$

where

$$\Delta\tau_{ij}^{best} = \begin{cases} \Delta\tau_{ij}^{bsf} = 1/OFC^{bs} & \text{if only the best-so-far solution } (s^{bsf}) \text{ is updated} \\ \Delta\tau_{ij}^{ib} = 1/OFC^{ib} & \text{if only the iteration-best solution } (s^{ib}) \text{ is updated} \end{cases} \tag{3.9}$$

where OFC^{bsf} is the objective function cost of the best-so-far trial solution, s^{bsf}; OFC^{ib} is the objective function cost of the iteration-best trial solution, s^{ib}.

(2) Pheromone trails are bounded by an interval $[\tau_{min}, \tau_{max}]$:

If only either s^{bsf} or s^{ib} are rewarded with pheromone, convergence to a solution during an early stage of the optimisation search is likely to occur. This is undesirable, as many regions of the problem search space are likely to have been left unexplored. In order to overcome this problem, all pheromone trails within a problem search space are bounded by upper and lower trail limits (τ_{max} and τ_{min}), the values of which are given by:

$$\tau_{max}(t+1) = \frac{1}{1-\rho} \cdot \frac{Q}{OFC^{best}(t)}. \tag{3.10}$$

$$\tau_{min}(t+1) = \frac{\tau_{max}(t+1)(1-\sqrt[n]{p_{best}})}{(avg-1)\sqrt[n]{p_{best}}}. \tag{3.11}$$

where t is an iteration index; $0 < \rho < 1$ is the pheromone evaporation rate; Q is the reward factor, $OFC^{best} = OFC^{bsf}$ and $OFC^{best} = OFC^{ib}(t)$ for the update of the best-so-far and iteration t's best trial solutions, respectively; n is the number of decisions an ant has to make (number of cities in the case of a TSP); when MMAS has converged, p_{best} is the probability that the s^{best} trial solution has been constructed once the algorithm has converged.

For further details of the derivation of Eqs. 3.10 and 3.11, readers are referred to Stützle *et al*. (2000). It can be envisaged that the narrower the bound interval, the smaller the difference of the pheromone levels between arcs and hence the higher the exploration level. In fact, the desired exploration and exploitation levels of a MMAS run can be defined by the user through a number of parameters. A detailed discussion of the impact each of the parameters in Eqs. 3.10 and 3.11 has on the searching behaviour of MMAS is given by Stützle *et al*. (2000). Another important note on the implementation of MMAS is that all pheromone trails must be initialized to a sufficiently high value such that in the second iteration, they are reset to τ_{max}.

MMAS is one of the most studied ACO algorithms and has been used as a tool for the development of new ACO algorithms (Socha, 2003; Al-Shihabi, 2004; de Franca *et al.*, 2004), as well as a comparison benchmark for other ACO algorithms (Socha *et al.*, 2003; Rajendran *et al.*, 2004; Solimanpur *et al.*, 2004; Zecchin *et al.*, 2006).

3.3.2 Non-AS-based ACO algorithms

Some ACO algorithms, while inspired by AS, incorporate some additional mechanisms that cannot be described by the AS-based algorithmic framework shown in Figure 3.3. Two ACO algorithms that belong to this category are the Ant Colony System (ACS) and the Hyper-Cube Framework for ACO (HCF). The additional features of these algorithms are briefly pointed out below.

Ant Colony System (ACS)

Ant Colony System (ACS) was proposed by Dorigo *et al*. (1997a; 1997b) based on the Ant-Q algorithm (Gambardella *et al.*, 1995). ACS differs from AS in three main aspects:

1. The random proportional rule (Eq. 3.1) utilized by AS for the construction of trial solutions is modified in ACS so that stronger exploitation of an ant's search experience is achieved. The algorithm uses

a new pseudorandom proportional rule, as part of which an ant selects the path with the best $[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}$ value by a probability of q_0. It can be envisaged that a high q_0-value will result in strong exploitation, as the best available option is chosen with a high probability. The value of q_0 can thus be adjusted in accordance with the desired level of exploitation.

2. In ACS, both pheromone evaporation and deposition are applied only to the best-so-far trial solution, s^{bsf}. Moreover, the quantity of pheromone deposited on the arcs belonging to s^{bsf} is discounted by the pheromone evaporation coefficient, ρ, which results in the new pheromone trails being a weighted average of the old pheromone value and the amount of pheromone deposited.

3. A local pheromone updating rule applies such that when an ant chooses to travel on an arc, the pheromone trail of the arc is reduced by a factor. This mechanism reduces the attractiveness of a travelled arc to subsequent ants.

Dorigo *et al*. (2004b) pointed out a very interesting observation that in order to manipulate the exploitation and exploration level of the algorithms, both MMAS and ACS implement upper and lower limits for pheromone trails However, the specification of such limits is explicit in MMAS and implicit in ACS. It can be seen from the unique features of ACS that there are many additional parameters, in addition to those already involved in AS, that need to be defined prior to the implementation of ACS. This is, undoubtedly, a major drawback of this algorithm.

<u>Hyper-Cube Framework for ACO (HCF)</u>

The Hyper-Cube Framework (HCF) for ACO was proposed by Blum *et al*. (2001). The main difference introduced by HCF is the normalization of pheromone trails such that they always lie in the interval [0,1]. This is implemented by the following pheromone update equations:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{3.12}$$

where

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{1/OFC^{k}}{\sum_{h=1}^{m}\left(1/OFC^{h}\right)}, & \text{if arc } (i,j) \text{ is used by ant } k. \\ 0, & \text{otherwise.} \end{cases} \tag{3.13}$$

Dorigo *et al*. (2004b) pointed out that the resulting pheromone vector can be seen as "*a shift of the old pheromone vector toward the vector given by the weighted average of the solutions used in the pheromone update*".

Table 3.2 summarizes the distinguising features of the ACO algorithms presented in this section.

Table 3.2: A summary of the distinguishing features of the ACO algorithms discussed

ACO variants	**Main features**
AS	o The earliest ACO algorithm o All trial solutions apply pheromone update
AS_{rank}	o Similar to AS, but only the best *g* trial solutions are rewarded
EAS	o Similar to *AS_{rank}*, but only the iteration-best trial solutions are rewarded
MMAS	o Only the iteration-best or best-so-far trial solutions are rewarded o Minimum and maximum pheromone trails apply
ACS	o Use of *pseudorandom proportional rule* during construction of trial solutions o Pheromone evaporation and reward are applied only to the best-so-far trial solutions. o A local pheromone updating rule applies
HCF	o Pheromone trails always lie in the interval [0,1]

3.4 Ant Colony Optimisation Applications

Since its first application to TSP and the encouraging results obtained, the ACO metaheuristics have been applied to a wide range of combinatorial optimisation problems, including benchmark and real-world problems. In this thesis, benchmark problems refer to those used by researchers in the field of evolutionary computation in testing the effectiveness of new optimisation algorithms. Among the most famous benchmark problems are the Travelling Salesman Problem (TSP), the Quadratic Assignment Problem (QAP) etc. Previous applications of ACO to various benchmark optimisation problems are detailed in Dorigo *et al*. (2004b).

As the optimisation problem addressed by the research presented in this thesis is a scheduling problem, previous implementations of ACO metaheuristics to some benchmark scheduling optimisation problems are reviewed. ACO applications to three real-world optimisation problems are discussed subsequently.

3.4.1 Benchmark scheduling optimisation problems

(A) Resource-Constrained Project Scheduling Problem (RCPSP)

The Resource-Constrained Project Scheduling Problem (RCPSP) is a scheduling problem where the set of activities of a project are scheduled such that the total makespan, which is the completion time of the last scheduled operation of the project, is minimized, subject to resource and precedence constraints amongst activities. The Elitist-Ant System-based algorithm proposed by Merkle *et al*. (2002), EAS-RCPSP, is at this time, the best performing approach for RCPSP (Dorigo *et al.*, 2004b). In the ACO formulation for RCPSP, the following components are defined:

Construction of trial solutions: The construction graph is comprised of (*act* + 2) fully connected nodes, where *act* is the total number of activities to be scheduled and the additional two nodes represent dummy start and end nodes. Starting from a dummy start node, an ant is considered to travel to each node (activity) once. Given an ant currently at position (i-1), a schedule generation method is utilized to generate a set of optional and feasible activities to be visited next (position i), based on the current partial solution. The probability of activity j being scheduled at position i

follows the random proportional rule (Eq. 3.1). A complete trial solution, which comprises a sequence of scheduled activities, is obtained when all (*act* + 2) construction steps are visited exactly once.

Definition of pheromone trails: The pheromone trail τ_{ij} refers to the desirability of scheduling activity j as the i-th activity (that is, putting activity j in position i).

Heuristic: η_{ij} refers to the desirability of scheduling activity j as the ith activity based on some user-defined information. The best-known heuristic formulation for this problem is based on the normalized version of the latest start time heuristic (Dorigo *et al.*, 2004b), given by: $\eta_{ij} = \max_{l \in AL} LS_l - LS_j + 1$, where LS_j is the latest possible start time of activity j and AL is the set of activities that are available given a partial schedule.

Local search: A 2-opt local search algorithm that considers swapping the position of two activities in a trial solution is adopted (Dorigo *et al.*, 2004b).

(B) Group-shop scheduling problem

In a Group Shop Scheduling problem (GSP), a set of *act* operations O is partitioned into (Dorigo *et al.*, 2004b):

- A set of subsets $M = \{M_1, \ldots, M_m\}$, where each M_i corresponds to the operations to be processed by machine i; and

- A set of subsets $J = \{J_1, \ldots, J_n\}$, where each set J_j corresponds to the operations belonging to job j. Subset J is further partitioned into groups $G = \{G_1, \ldots, G_p\}$ where G_k corresponds to operations belonging to group k.

The objective of GSP is to minimize the makespan of operations. The constraints to be satisfied include (Dorigo *et al.*, 2004b):

  - Each machine i can process at most one operation at a time.

  - Operations must be processed without pre-emption.

- Operations within one group can be processed in any order but the groups of a job are totally ordered.

GSP can be seen as a general shop scheduling problem, with special cases including the Job Shop Scheduling problem (JSP), the Open Shop Scheduling problem (OSP) and the Mixed Shop Scheduling problem (MSP). Among the ACO approaches previously used for shop scheduling problems (Colorni *et al*., 1994; Pfahringer, 1996; Blum, 2002), a Max-Min Ant System-based algorithm, namely the ACO-MMAS-HC-GSP proposed by Blum (2002), is the current best-performing ACO algorithm for GSP. The formulation adopted by the ACO-MMAS-HC-GSP algorithm is now described.

Construction of trial solutions: The construction graph and the way a trial solution to this problem is constructed are identical to those of RCPSP.

Definition of pheromone trails: The pheromone model used in MMAS-HC-GSP assigns a pheromone value to a pair of related operations. Two operations are related if they belong to the same group or must be processed on the same machine. A high pheromone value of two related operations o_i and o_j means operation o_i is favoured to be processed before (but not necessarily immediately before) operation o_j. This model was claimed to be the best pheromone representation for this problem type, where relative positioning rather than absolute positioning of operations is more important.

Heuristic: An earliest start heuristic that favors operations with the earliest valid starting time with respect to the partial schedule is adopted. Given a partial schedule, the heuristic information is calculated based on the inverse of the earliest possible starting time of an operation and then normalized over all eligible operations.

Local search: The best local search algorithm used in conjunction with MMAS-HC-GSP applies an iterative improvement algorithm to each trial solution, and then applies a tabu search algorithm to the best local optimum. A local neighborhood definition introduced by Nowicki *et al*. (1996) for JSP, where a job placed at the a-th position is moved to a b-th position, is utilized in the local search algorithm.

(C) Single-machine total weighted tardiness scheduling problem

In a single machine total weighted tardiness scheduling problem (SMTWTP), n jobs have to be processed on a single machine, without interruption. Each job is given a known processing time p_j, a weight w_j, and a due date d_j and all jobs are available to be scheduled from time zero. The objective of SMTWTP is to schedule all the jobs in a sequence such that the sum of the weighted tardiness, $\sum_{j=1}^{n} w_i T_i$, is minimized. Given a trial solution (schedule), the tardiness of a job i is defined as $T_j = \max\{0, CT_j - d_j\}$, where CT_j is its completion time in the schedule under consideration. ACO has been applied to SMTWTP concurrently by den Besten *et al*. (2000) and Merkle *et al*. (2003) and the formulations proposed in the two studies are similar for many characteristics.

Construction of trial solutions: The construction graph comprises of fully connected components C representing the n positions to which the n jobs are assigned. In order to construct a trial solution, an ant chooses a job for position 1, another for position 2 until all n jobs are scheduled.

Definition of pheromone trails: τ_{ij} indicates the attractiveness of placing job j at position i. A problem with solely using τ_{ij} when choosing the next job to be placed on a schedule was identified by Merkle *et al*. (2003), and the problem was resolved by implementing a new pheromone summation rule. The new rule takes into account the pheromone values of placing job j at positions [i, i-1, …, 1] when estimating the desirability of placing job j at position i. In this way, even if a job j with high τ_{ij} was not placed at position i, it is highly likely to be placed at a position close to i influenced by the high value of τ_{ij}.

Heuristic: Three different formulations were tested by den Besten *et al*. (2000) to compute the heuristic information:

1. Earliest Due Date (EDD): $\eta_{ij} = 1/d_j$, where d_j is the due date of job j.

2. Modified Due Date (MDD): Similar to EDD, but the sum of processing times of already scheduled jobs (partial schedule) is taken into account (η_{ij}

$= 1/mdd_j$, where $mdd_j = \max\{C + p_j, d_j\})$, where C is the total processing time of a partial schedule.

3. Apparent Urgency (AU): In this heuristic, the average processing time of the remaining jobs, $\overline{p}$, is taken into consideration ($\eta_{ij} = 1/au_j$, where $au_j = (w_j/p_j).\exp(-\max\{d_j - C_j, 0\})/k\,\overline{p}$), k is a parameter set).

Merkle *et al*. (2003) identified a problem with using the MDD heuristic, whereby the value of $\max\{C + p_j, d_j\}$ is too large due to the total processing time, C, which is large when the sequence of a partial schedule becomes too long. As a consequence, the heuristic difference becomes insufficiently apparent to ants when choosing the next job to schedule. In order to rectify this problem, Merkle *et al*. (2003) improved the MDD heuristic by subtracting the total processing time, C, such that $\eta_{ij} = 1/mdd_j$, where $mdd_j = \max\{C + p_j, d_j\} - C$.

Local search: Two neighborhood definitions have been used for the SMTWTP, including:

1. Exchanging the pair of jobs placed at the i-th and j-th positions (interchange).

2. Removal of the job at the i-th position and inserting it to the j-th position of the schedule (insertion).

3.4.2 Real-world optimisation problems

In spite of some very encouraging results obtained by ACO for benchmark optimisation problems, there are not many applications to real-world problems, or operational research. Given the context of the research work presented in this thesis, this section is devoted to the review of ACO applications to three different real-world optimisation problems.

(A) Design of Water Distribution System (WDS)

A formulation based on ACO was proposed by Maier *et al*. (2003b) to minimise the costs associated with the size of pipelines for a water distribution system (WDS), subject to constraints such as demand and pressure criteria. The adaptations made in order to apply ACO to the WDS optimisation problem can be summarized as follows:

Construction of trial solutions: The ACO construction graph is formed by a set of nodes representing the set of pipes whose sizes are to be optimized, fully connected by arcs representing optional pipe diameters. A complete trial solution to the WDS problem comprises a size for each pipeline of the pipeline system considered. In order to construct a trial solution, a single ant agent visits each node (pipe) in a random (or heuristically defined) order. An ant at node i (pipe i) considers a randomly chosen, unvisited node k ($k \neq i$), and in order to travel to node k, the ant needs to make a decision, based on the random proportional rule (Eq. 3.1), about which pipe size to choose.

Definition of pheromone trails: τ_{ij} represents the desirability of pipe size index j being used for pipe i.

Heuristic: $\eta_{ij} = {}^{1}\!/_{cost_j}$ is a myopic value of using pipe size index j for pipe i, based on the users' experience, where $cost_j$ is the cost per unit of pipe size index j.

Local search: Local search was not considered in the formulation.

The ACO formulation for the WDS optimisation problem has been tested with two case studies: a 14-pipe Problem and the New York City Water Supply Tunnels Problem (Maier *et al.*, 2003a). The performance of the ACO approach for the 14-pipe Problem is comparable to those of a genetic algorithm, both in terms of the ability of finding the global optimum and the computational time required. On the other hand, the ACO approach found a better solution to the New York City Water Supply Tunnels Problem than that given by a GA. A later study by Zecchin *et al.* (2006) confirmed the performance of the ACO formulation against a genetic algorithm when applied to the New York City Water Supply Tunnels Problem and the Hanoi Problem. In addition, experiments conducted in the latter study indicated that MMAS is a more robust ACO algorithm compared to its ancestor, AS, due to the upper and lower limits imposed on pheromone trails.

A study conducted by Afshar (2006) investigated the optimisation of the layout of pipe networks using ACO. This optimisation problem can be seen as a special instance of the generalised water distribution design specification introduced by Zecchin *et al.* (2005).

(B) Pump Scheduling

The optimal scheduling of pump operations problem is described by Goldman *et al*. (2000) as:

"*Given a water distribution network, where diurnal demands, initial tank levels and electricity tariffs are known, the goal of this problem is to find the optimal pump schedules over a time period, typically 24 hours, such that the operational costs are minimized and constraints are satisfied.*"

This problem has been studied by using Simulated Annealing (Goldman *et al.*, 2000) and genetic algorithms (Mackle *et al.*, 1995; Savic *et al.*, 1997; Kazantzis *et al.*, 2002; van Zyl *et al.*, 2004). The ACO formulation proposed by Prasad *et al*. (2006) to solve the pump scheduling problem is comprised of the following elements:

Construction of trial solutions: A representation of trial solutions based on time triggers proposed by Lopez-Ibanez *et al*. (2005) was adopted, where a trial solution is comprised of a set of strings, each associated with the operational schedule of a pump. A string is formed by a finite pair of integers representing the number of hours a pump is off and remains on when it is switched, respectively. If a pump switch is defined by switching a pump from off to on, and the status of a pump (being on or off) as an interval, the length of a string (number of integers) is 2 x maximum allowable number of pump switches, S. When an ant constructs a trial solution, it travels to each interval (randomly) and chooses a duration for the interval from a set of available options. It should be noted that this type of solution representation enables maximum pump switch constraints to be satisfied explicitly during the construction of trial solutions, avoiding the need for a penalty factor. Once a duration is chosen for all intervals, a complete trial solution is obtained.

Definition of pheromone trails: τ_{ij} represents the utility of assigning a duration *j* to interval *i* for a particular pump.

Heuristic: Heuristic information was not considered in the formulation.

Local search: Local search was not considered in the formulation.

An AS-based algorithm was implemented to solve two pump scheduling case studies – a test case study proposed by van Zyl *et al*. (2004) and a real system in the United Kingdom (Prasad *et al.*, 2006). At each iteration, trial solutions are ranked and only the best-iteration trial solutions are rewarded. While the results obtained for the test network were shown to be comparable with those given by a hybrid GA approach proposed by van Zyl *et al*. (2004), the ACO algorithm was found to be inferior to the genetic algorithm for the real system.

(C) Optimal Siting of New Fire Stations

An ACO algorithm (ANT) has been coupled with a geographical information system (GIS) to determine the optimum locations of six new fire stations, aiming at increasing the effectiveness of the fire stations in covering the transportation routes of hazardous materials (HAZMATS) through Singapore (Liu *et al.*, 2006). Using GIS, the map of Singapore is represented by a grid coordinate system by means of a finite number of discrete cells. Each cell is assigned a coordinate (i, j).

Construction of trial solutions: Each of the six ants in ANT is used to search for the optimal location of a fire station.

Definition of pheromone trails: τ_{ij} represents the desirability of locating a new fire station in a discrete cell of coordinate (i, j).

Heuristic: Heuristic information was not considered in the ACO formulation.

Local search: Each trial solution (a complete set of proposed locations of all six fire stations) is applied to a 2-phase local search. In phase 1, a neighborhood random search (NRS) strategy is used, where all ants randomly move from their current coordinate to other cells within a certain distance (eg. 3km). The current solution is replaced if a local solution associated with a better objective function value is found. The NRS search is repeated for a predefined number of iterations before the second phase of local search is triggered. In phase 2, an adaptive enumeration neighborhood search (AENS) is activated, where in an AENS routine, each of the six ants moves to every cell within a certain distance from its current cell while keeping the other five ants fixed at their original cells. Similar to the phase-1 search, a current solution is replaced by an improved local solution. The AENS search is repeated until all local solutions are evaluated.

The performance of ANT was found to be superior to those of a genetic algorithm (GA) and a random start 2-phase local search procedure (RANDOM LS).

3.5 Motivation for Applying ACO to PPMSO

As part of the research work presented in this thesis, a formulation based on the ACO metaheuristic is proposed for the power plant maintenance optimisation problem (PPMSO). The choice of the ACO metaheuristic for PPMSO is mainly motivated by the following:

- The decision tree-based solution construction mechanism of ACO fits in well with PPMSO, which is naturally an optimisation problem with sequential decisions. By using the decision-tree based structure, many constraints commonly encountered in PPMSO problems can be explicitly addressed, eliminating the need to use penalty factors. In addition, the search space of an optimisation problem can be greatly reduced by progressively eliminating optional solution components that no longer satisfy problem constraints, given a current partial solution.

- As a population-based metaheuristic, ACO is highly suitable for real-world optimisation problems that usually have a large search space and involve complex mathematical functions. As a global optimisation method, ACO performs coarse-grained search to quickly identify decision space regions where promising solutions are located. Local search strategies can then be used to search within local neighbourhoods of trial solutions generated throughout the ACO optimisation process. Secondly, existing simulation models corresponding to the case study in hand can be easily incorporated into an ACO formulation, without the need for simplifying complex mathematical equations.

- Heuristic information that reflects the experience of a user can be optionally incorporated into the formulation.

- As a population-based metaheuristic, ACO searches different regions of a problem search space and is thus able to produce different solutions of similar criteria quality (objective function cost). Taking PPMSO as an example, a list of the best 20 maintenance schedules produced during an ACO run might be recorded. The decision maker can then consider each of these schedules based on other non-quantifiable criteria, which were not included in the optimisation run.

- Since ACO explores many feasible as well as infeasible trial solutions to a problem, the increased speed of modern computers enables real-world-sized problems to be solved in reasonable run-times.

- For real-world problems, it is unrealistic to aim for a globally optimal solution. Near-optimal, or reasonable good, solutions, which can normally be obtained by global optimisation metaheuristics such as ACO, are sufficient for practical purposes.

- The ACO metaheuristic has been applied to both benchmark and real-world optimisation problems and the results obtained are promising when compared with other metaheuristics.

Chapter 4
Proposed Approach to Maintenance Scheduling Optimisation

In this chapter, the main contributions of the research work presented in this thesis are covered, which are:

- The definition of power plant maintenance scheduling optimisation (PPMSO) is generalized. In particularly, the options of outage duration shortening and deferral of maintenance tasks are incorporated.
- A new formulation based on Ant Colony Optimisation (ACO) is proposed for a more generalized PPMSO problem.
- Different constraints commonly encountered in maintenance scheduling problems are categorized. Methods for addressing these constraints are also proposed.
- A new heuristic formulation is developed for ACO to solve PPMSO problems more effectively.
- Two local search operators are developed to refine the rather coarse-grained search of ACO in a problem search space.
- The ACO-PPMSO algorithm is coded in the Fortran 90 programming language.

4.1 Definition of power plant maintenance scheduling optimisation (PPMSO)

The requirements of an optimisation problem have to be defined before any proposed optimisation methods can be properly formulated to solve the problem. The power plant maintenance scheduling optimisation (PPMSO) problem has been defined previously as an optimisation problem that involves the determination of the optimum timing of the maintenance periods of each of the generating machines (units) used for power generation, assuming maintenance durations are fixed (Dopazo *et al.*, 1975; Yamayee *et al.*, 1983; Mukerji *et al.*, 1991; Satoh *et al.*, 1991; Kim *et al.*, 1997; Aldridge *et al.*, 1999; Dahal *et al.*, 1999; Dahal *et al.*, 2000; El-Amin *et al.*, 2000; Foong *et al.*, 2005a; Foong *et al.*, 2005b). Such a PPMSO definition is insufficient, as there are times when certain generating machines cannot be taken offline much longer than a certain period of time in order to meet system demand or to achieve system reliability. In this case, maintenance duration of these tasks can be shortened by employing more personpower, or maintenance tasks can be deferred. As part of the contribution of the research presented in this thesis, the PPMSO problem definition is generalized to include the options of 'maintenance duration shortening' and 'deferral of maintenance tasks'. As a result, not only the optimum commencement time, but also the optimum duration is sought for each maintenance task to be scheduled within a planning horizon.

PPMSO is generally considered as a minimization problem (S, f, Ω), where S is the set of all maintenance schedules, f is the objective function which assigns an objective function value $f(s)$ to each trial maintenance schedule $s \in S$, and Ω is a set of constraints. Mathematically, PPMSO can be defined as the determination of a set of globally optimal maintenance schedules $S^* \subset S$, such that the objective function is minimized $f(s^* \in S^*) \leq f(s \in S)$ (for a minimization problem) subject to a set of constraints Ω. Specifically, PPMSO has the following characteristics:

- It consists of a finite set of decision points $D = \{d_1, d_2, \ldots, d_N\}$ comprised of N maintenance tasks to be scheduled;

- Each maintenance task $d_n \in D$ has a normal (default) duration $NormDur_n$ and is carried out during a planning horizon T_{plan};

Two decision variables $start_n$ and $chdur_n$ need to be defined for each task d_n, including:

1. The start time for the maintenance task, $start_n$, with the associated set of options: $T_{n,chdur_n} = \{ t \in T_{plan}; chdur_n \in K_n: ear_n \leq t \leq lat_n - chdur_n + 1\}$ where the terms in brackets denote the set of time periods when maintenance of unit d_n may start; ear_n is the earliest time for maintenance task d_n to begin;

lat_n is the latest time for maintenance task d_n to end and $chdur_n$ is the chosen maintenance duration for task d_n.

2. The duration of the maintenance task, $chdur_n$, with the associated finite set of decision paths: $K_n = \{0, s_n, 2s_n, \ldots, NormDur_n - s_n, NormDur_n\}$, where the terms in brackets denote the set of optional maintenance durations for task d_n, and s_n is the time step considered for maintenance duration shortening.

A trial maintenance schedule, $s \in S = \langle(start_1, chdur_1), (start_2, chdur_2), \ldots, (start_N, chdur_N)\rangle$ is comprised of maintenance commencement times, $start_n$, and durations, $chdur_n$, for all N maintenance tasks that are required to be scheduled.

Binary variables, which can take on values 0 or 1, are used to represent the state of a task in a given time period in the mathematical equations of the PPMSO problem formulation. $X_{n,t}$ is set to 1 to indicate that task $d_n \in D$ is scheduled to be carried out during period $t \in T_{plan}$. Otherwise, $X_{n,t}$ is set to a value of 0, as given by:

$$X_{n,t} = \begin{cases} 1 & \text{if task } d_n \text{ is being maintained in period } t \\ 0 & \textit{otherwise} \end{cases} \tag{4.1}$$

In addition, the following sets of variables are defined:

$S_{n,t} = \{k \in T_{n,chdur_n}, chdur_n \in K_n: t - chdur_n + 1 \le k \le t\}$ is the set of start times k, such that if maintenance task d_n starts at time k for a duration of $chdur_n$, that task will be in progress during time t;

$D_t = \{d_n: t \in T_n\}$ is the set of maintenance tasks that is considered for period t.

Objectives and constraints

Traditionally, cost minimization and maximization of reliability have been the two objectives commonly used when optimizing power plant maintenance schedules. Two examples of reliability objectives are evening out the system reserve capacity throughout the planning horizon, and maximizing the total reservoir storage water volumes at the end of the planning horizon, in the case of a hydropower system. An additional objective associated with the more generalized definition of PPMSO presented in this thesis is the minimization of the total maintenance duration shortened/deferred. The rationale behind this objective is that shortening of maintenance duration (i.e. speeding up the completion of maintenance tasks) requires additional personnel and equipment, whereas deferral of maintenance tasks might result in unexpected breakdown of generating units, and in both events, additional costs are incurred by the power utility operator.

Constraints specified in PPMSO problems are also power plant specific. The formulation of some common constraints including the allowable maintenance window, continuity, load, availability of resources, precedence of maintenance tasks, reliability and the minimum maintenance duration required, which are presented in Eqs. 4.2 to 4.8.

The timeframes within which individual tasks in the system are required to start and finish maintenance form maintenance window constraints, which can be formulated as:

$$ear_n \leq start_n \leq lat_n - chdur_n + 1 \qquad \text{for all } d_n \in D. \tag{4.2}$$

where $start_n$ and $chdur_n$ are the start time and maintenance duration, T respectively, chosen for task d_n.

The continuity constraint states that once a maintenance task d_n commences, it should not finish before completion and the time corresponding to the chosen outage duration $chdur_n$ has elapsed, and is given by:

$$X_{n,t} = \begin{cases} 1 & \text{for } t = \left[start_n, \ldots, start_n + chdur_n - 1\right] \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

where $start_n$ and $chdur_n$ are the start time and maintenance duration chosen for task d_n.

Load constraints (Eq. 4.4) are usually rigid/hard constraints in PPMSO problems, which ensure that feasible maintenance schedules that do not cause demand shortfalls throughout the whole planning horizon are obtained:

$$\sum_{d_n \in D} P_{n,t} - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \geq L_t \quad \text{for all } t \in T_{plan} \tag{4.4}$$

where L_t is the anticipated load for period t and P_n is the loss of generating capacity associated with maintenance task d_n.

Resource constraints are specified in the case where the availability of certain resources, such as highly skilled technicians, is limited. In general, resources of all types assigned to maintenance tasks should not exceed the associated resource capacity at any time period, as given by:

$$\sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} Res_{n,k}^{r} \leq ResAvai_t^{r} \quad \text{for all } t \in T_{plan}, r \in R. \tag{4.5}$$

where $Res_{n,k}^{r}$ is the amount of resource of type r available that is required by task d_n at period k; $ResAvai_t^{r}$ is the associated capacity of resource of type r available at period t and R is the set of all resource types.

Precedence constraints that reflect the relationships between the order of maintenance of generating units in a power system are usually specified in PPMSO problems. An example of such a constraint is a case where task 2 should not commence before task 1 is completed, as given by:

$$start_2 > start_1 + chdur_1 - 1 \tag{4.6}$$

where $start_n$ is the start time chosen for task d_n.

Depending on particular system characteristics and requirements, reliability constraints can be formulated in various ways, including provision of reserve generation capacity as a certain proportion of demand throughout the planning horizon. This is given by:

$$\sum_{d_n \in D} P_{n,t} - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \geq L_t + f_{res} \cdot L_t \quad \text{for all } t \in T_{plan} \tag{4.7}$$

where L_t is the anticipated load for period t; P_n is the loss of generating capacity associated with maintenance task d_n and f_{ews} is the factor of load demand required for reserve.

In the case of maintenance duration shortening, there is usually a practical limit to the extent that the duration can be shortened. Due to the different characteristics of maintenance tasks, minimum maintenance durations may vary with individual tasks:

$$NormDur_n \geq chdur_n \geq MinDur_n, \text{ for all } d_n \in D. \tag{4.8}$$

where $chdur_n$ is the maintenance duration of task d_n; $MinDur_n$ is the minimum shortened outage duration for task d_n; $NormDur_n$ is the normal duration of maintenance task d_n.

4.2 Proposed ACO formulation for PPMSO

Before the PPMSO problem can be optimized using ACO, it has to be mapped onto a graph, which is expressed in terms of a set of decision points consisting of the N maintenance tasks that need to be scheduled $D = \{d_1, d_2, d_3, \ldots, d_N\}$. In accordance with the formulation introduced, there are three variables that need to be defined $V = \{v_1, v_2, v_3\}$ for each maintenance task:

- Variable 1, v_1: the overall state of the maintenance task under consideration (i.e. if maintenance currently being carried out or not),
- Variable 2, v_2: the duration of the maintenance task, and
- Variable 3, v_3: the commencement time for the maintenance task.

For maintenance task d_n, a set of decision paths $DP_{c,n}$ is associated with decision variable $v_{c,n}$ (where subscript c = 1, 2 or 3) (shown as dashed lines in Figure 4.1). For decision variable $v_{1,n}$, these correspond to the options of carrying out the maintenance tasks d_n at normal duration, shortening the maintenance duration and deferring maintenance tasks. For decision variable $v_{2,n}$, these correspond to the optional shortened durations available for the maintenance tasks. For decision variable $v_{3,n}$, these correspond to the optional start times for maintenance tasks d_n. It should be noted that, as the latest finishing time of maintenance tasks is usually fixed, there are different sets of start time decision paths, each corresponding to a maintenance duration decision path (Figure 4.1). This graph can then be utilized to construct trial solutions using the ACO-PPMSO algorithm introduced in Section 4.3.

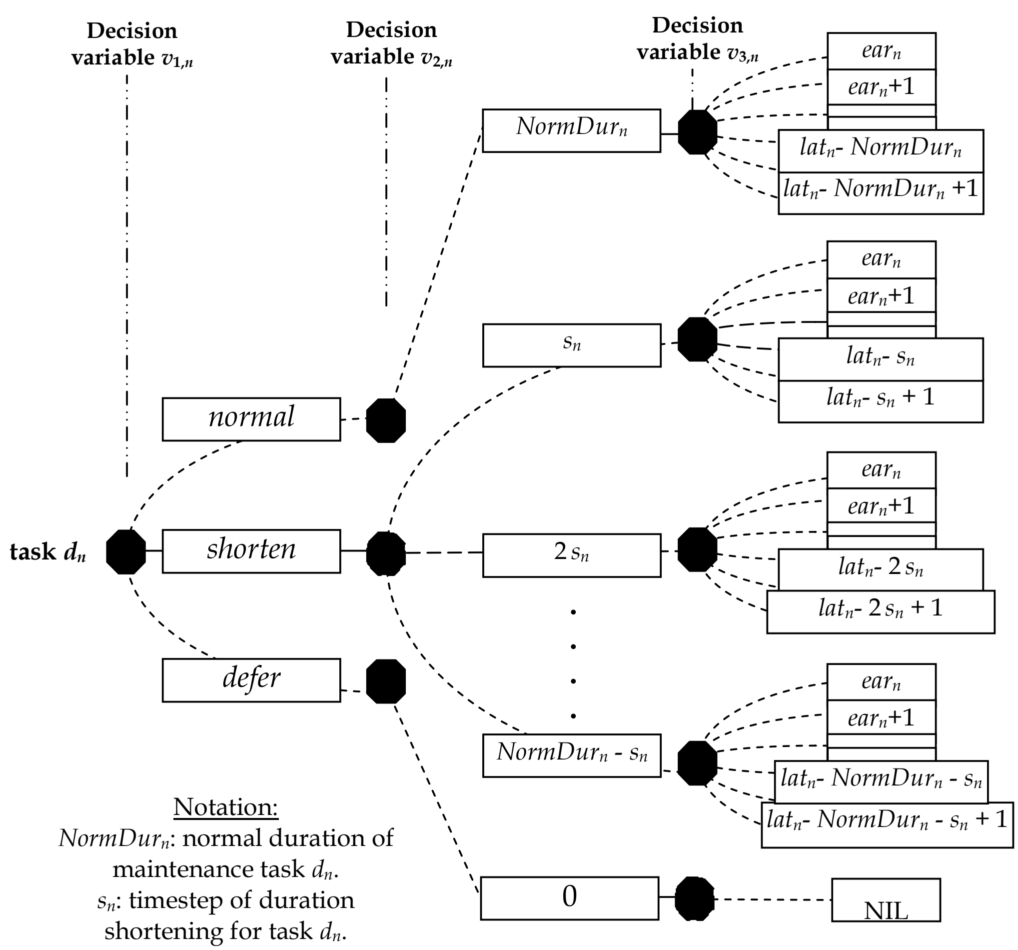


Figure 4.1: Proposed ACO-PPMSO graph

4.3 The ACO-PPMSO algorithm

The new formulation proposed in this research for power plant maintenance scheduling using Ant Colony Optimisation is implemented via an ACO-PPMSO algorithm, represented by the flowchart given in Figure 4.2. The mechanisms involved in each procedure of the proposed ACO-PPMSO algorithm are detailed in Sections 4.3.1 to 4.3.6.

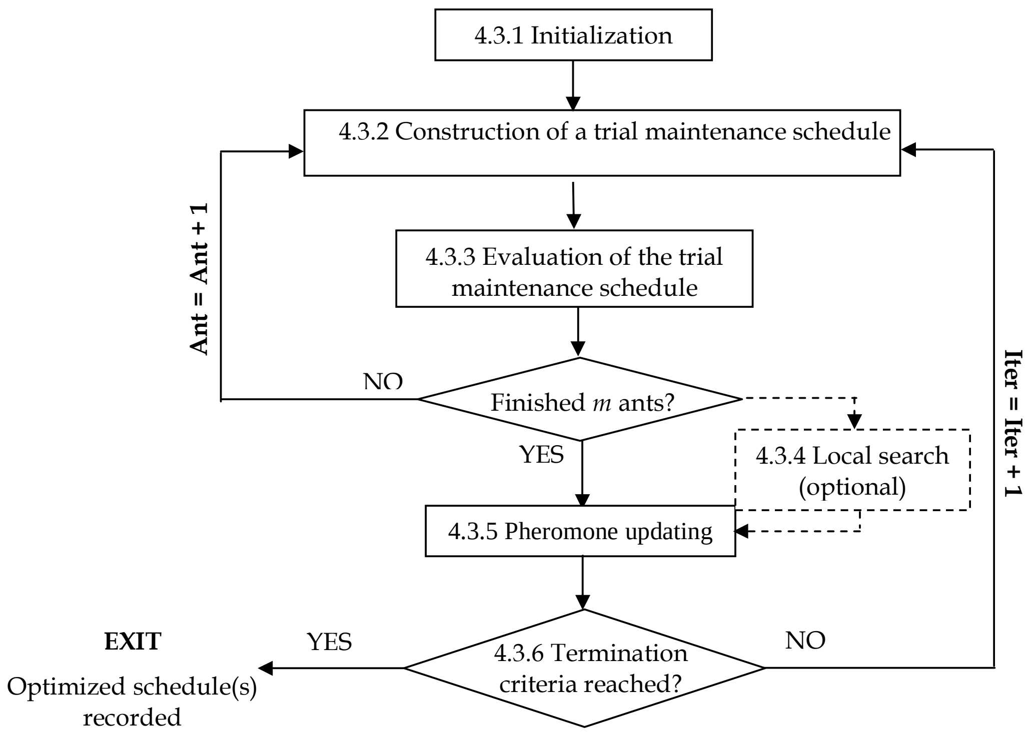


Figure 4.2: Proposed ACO-PPMSO algorithm

4.3.1 Initialization

The optimisation process starts by reading details of the power system under consideration (eg. generating capacity of each unit, daily system demands, time step for duration shortening etc.). In addition, various ACO parameters (eg. initial pheromone trails (τ_0), number of ants used, pheromone evaporation rate etc.) need to be defined.

4.3.2 Construction of a trial maintenance schedule

A trial maintenance schedule is constructed using the ACO-PPMSO graph shown in Figure 4.1. In order to generate one trial maintenance schedule, an ant travels to one of the decision points (maintenance tasks) at a time. At each decision point, d_n, a three-stage selection process that corresponds to the three decision variables, $v_{1,n}$, $v_{2,n}$ and $v_{3,n}$, is performed.

At each stage, the probability that decision path *opt* is chosen for maintenance of task d_n in iteration t is given by:

$$p_{n,opt}(t) = \frac{\left[\tau_{n,opt}(t)\right]^{\alpha} \cdot \left[\eta_{n,opt}\right]^{\beta}}{\sum_{y \in DP_{c,n}} \left[\tau_{n,y}(t)\right]^{\alpha} \cdot \left[\eta_{n,y}\right]^{\beta}}. \tag{4.9}$$

subscripts c = 1, 2 and 3 refer to the three decision variables, $v_{1,n}$, $v_{2,n}$ and $v_{3,n}$; $\tau_{n,opt}(t)$ is the pheromone intensity deposited on the decision path *opt* for task d_n in iteration t; $\eta_{n,opt}$ is the heuristic value of decision path *opt* for task d_n; α and β are the relative importance of pheromone intensity and the heuristic, respectively.

It should be noted that the term *opt* in Eq. 4.9 represents the decision path under consideration, of all decision paths contained in set $DP_{c,n}$. When used for stages 1, 2 and 3, respectively, the terms *opt* and $DP_{c,n}$ are substituted with those associated with the decision variable considered at the corresponding stage (Table 4.1). The pheromone level associated with a particular decision path (e.g. deferral of a particular maintenance task) is a reflection of the quality of the maintenance schedules that have been generated previously that contain this particular option. The heuristic associated with a particular decision path is related to the likely quality of a solution that contains this option, based on user-defined heuristic information. The following paragraphs detail the three-stage selection process for decision point (maintenance task) d_n, including the adaptations required when using Eq. 4.9 for each stage.

Table 4.1: Adaptations for Eq. 4.9 in stages 1, 2 and 3 of the selection process

	Stage 1	Stage 2	Stage 3
c	1	2	3
opt	$stat \in DP_{1,n}$	$dur \in DP_{2,n}$	$day \in DP_{3,n,chdur_n}$
$DP_{c,n}$	$DP_{1,n}$={*normal, shorten, defer*}	$DP_{2,n}$ = {0, s_n, $2s_n$,…, $NormDur_n$}	$DP_{3,n,chdur_n}$ = {$chdur_n \in DP_{2,n}$: ear_n, ear_n+1,…, $lat_n - chdur_n + 1$}
$\tau_{n,opt}$	$\tau_{n,stat}$	$\tau_{n,dur}$	$\tau_{n,chdur_n,day}$
$\eta_{n,opt}$	$\eta_{n,defer} < \eta_{n,shorten} < \eta_{n,normal}$	$\eta_{n,dur_n} \propto dur$	$\eta_{n,chdur_n,day} = \left(\eta^{Res}_{n,chdur_n,day}\right)^w \cdot \eta^{Load}_{n,chdur_n,day}$

In stage 1, a decision needs to be made whether to perform the maintenance task under consideration at normal or shortened duration, or to defer it (decision variable $v_{1,n}$ in Figure 4.1). In this case, c = 1 and opt = $stat \in DP_{1,n}$={*normal, shorten, defer*} is the set of decision paths associated with decision variable $v_{1,n}$ for task d_n. The probability of each of these options being chosen is a function of the strength of the pheromone trails and heuristic value associated with the option (Eq. 4.9). For the PPMSO problem, the heuristic formulation should generally be defined such that normal maintenance durations are preferred over duration shortening, and deferral is the least favored option (Eq. 4.10). However, real costs associated with duration shortening and deferral options can be used if the extra costs incurred associated with these options are quantifiable and available. The adaptations required for Eq. 4.9 to be used at the stage 1 selection process are summarized in Table 4.1. It is suggested that values of the heuristics should be selected such that:

$$\eta_{n,defer} < \eta_{n,shorten} < \eta_{n,normal} \tag{4.10}$$

Once a decision has been made at stage 1, the selection process proceeds to stage 2 (decision variable $v_{2,n}$ in Figure 4.1), where the duration of the maintenance task under consideration, d_n, is required to be selected from a set of available decision paths $DP_{2,n}$ = {0, s_n, $2s_n$, . . . , $NormDur_n$}. The symbols s_n and $NormDur_n$ denote the time step for maintenance duration shortening, and the normal maintenance duration, respectively. For Eq. 4.9 to be used at stage 2, the terms c and opt in the equation are substituted by the values 2 and $dur \in DP_{2,n}$, respectively. It should be noted that if the 'normal' or 'defer' options were chosen at stage 1, the normal duration of the maintenance task, or a duration of 0, respectively, are automatically chosen for the task. In the case of duration shortening, a constraint is normally specified where each maintenance task has a minimum duration at which the completion of the task cannot be further accelerated due to limitations such as the availability of highly specialized technicians. This constraint can be addressed at this stage such that only feasible trial maintenance schedules (with regard to this constraint) are constructed (see Section 4.4 for details of such constraint-handling techniques). The pheromone trails and heuristic values associated with optional durations are used to determine the probability that these durations are chosen. In order to favor longer maintenance durations (i.e. the smallest amount of shortening compared with the normal maintenance duration), it is suggested that the heuristic value associated with a decision path should be directly proportional to the maintenance duration (Eq. 4.11).

$$\eta_{n,dur} \propto dur \tag{4.11}$$

The substitutions for the various terms in Eq. 4.9 when used in stage 2 are summarized in Table 4.1.

Once a maintenance duration has been selected, the solution construction process enters stage 3 (decision variable $v_{3,n}$ in Figure 4.1), where a start time for the maintenance task is selected from the set of optional start times available $DP_{3,n,chdur_n}$ = {$chdur_n \in DP_{2,n}$: ear_n, $ear_n+1,\ldots, lat_n - chdur_n + 1$}, given a chosen duration of $chdur_n$. In order to utilize Eq. 4.9 at stage 3, adjustments are made such that $c = 3$ and $opt = day \in DP_{3,n,chdur_n}$. It should be noted that this stage is skipped if the 'defer' option is chosen at stage 1. The probability that a particular start day is chosen is a function of the associated pheromone trail and heuristic value. The suggested heuristic formulation for selection of the maintenance start day is given by Eqs. 4.12 to 4.17.

$$\eta_{n,chdur_n,day} = \left(\eta^{Res}_{n,chdur_n,day}\right)^{w} \cdot \eta^{Load}_{n,chdur_n,day} \tag{4.12}$$

$$\eta^{Res}_{n,chdur_n,day} = \frac{\sum_{k \in J_{n,chdur_n,day}} Y_{ResV(k)=0} \cdot R_{n,chdur_n,day}(k)}{\sum_{k \in J_{n,chdur_n,day}} (Y_{ResV(k)=0} - 1) \cdot R_{n,chdur_n,day}(k)} \tag{4.13}$$

$$\eta^{Load}_{n,chdur_n,day} = \frac{\sum_{k \in J_{n,chdur_n,day}} Y_{LoadV(k)=0} \cdot C_{n,chdur_n,day}(k)}{\sum_{k \in J_{n,chdur_n,day}} (Y_{LoadV(k)=0} - 1) \cdot C_{n,chdur_n,day}(k)} \quad (4.14)$$

$$Y_{ResV(k)=0} = \begin{cases} 1 & \text{if no violation of resource constraints in time period } k \\ 0 & \text{otherwise} \end{cases} \quad (4.15)$$

$$Y_{LoadV(k)=0} = \begin{cases} 1 & \text{if no violation of load constraints in time period } k \\ 0 & \text{otherwise} \end{cases} \quad (4.16)$$

$$w = \begin{cases} 1 & \text{if resource constraints are considered} \\ 0 & \text{otherwise} \end{cases} \quad (4.17)$$

where $\eta_{n,chdur_n,day}(t)$ is the heuristic for start time $day \in DP_{3,n,chdur_n}$ for task d_n, given a chosen duration $chdur_n$; $R_{n,chdur_n,day}(k)$ represents the prospective resources available in reserve in time period k if task d_n is to commence at start time *day* and takes $chdur_n$ to complete (less than 0 in the case of resource deficits); $C_{n,chdur_n,day}(k)$ is the prospective power generation capacity available in reserve in time period k if task d_n is to commence at start time *day* and takes $chdur_n$ to complete (less than 0 in the case of power generation reserve deficits); $J_{n,chdur_n,day} = \{day \in DP_{3,n,chdur_n} : day \le k \le day + chdur_n - 1\}$ is the set of time periods k such that if task d_n starts at start time *day*, that task will be in maintenance during period k.

As mentioned above, the heuristic formulation in Eq. 4.12 includes a resource-related term, $\eta^{Res}_{n,chdur_n,day}$, and a load-related term, $\eta^{Load}_{n,chdur_n day}$. These two terms are expected to evenly distribute maintenance tasks over the entire planning horizon, which potentially maximizes the overall reliability of a power system. For PPMSO problem instances that do not consider resource constraints, the value of w in Eq. 4.12 can be set to 0 (Eq. 4.17). In order to implement the heuristic, each ant is provided with a memory matrix on resource reserves and another matrix on generation capacity reserves prior to construction of a trial solution. This is updated every time a unit maintenance commencement time is added to the partially completed schedule.

The three-stage selection process is then repeated for another maintenance task (decision point). A complete maintenance schedule is obtained once all maintenance tasks have been considered.

4.3.3 Evaluation of trial maintenance schedule

Once a complete trial maintenance schedule, $s \in S$, has been constructed by choosing a maintenance commencement time and duration at each decision point (i.e. for each maintenance task to be scheduled), an ant-cycle has been completed. The trial schedule's objective function cost (*OFC*) can then be determined by an evaluation function, which is a function of the values of objectives and constraint violations:

$$OFC(s) = f\left(obj_1(s), obj_2(s), \ldots, obj_{Z_T}(s), vio_1(s), vio_2(s), \ldots, vio_{C_T}(s)\right) \qquad (4.18)$$

where *OFC*(*s*) is the objective function cost associated with a trial maintenance schedule, *s*; $obj_1(s)$ is the value of the first objective; $vio_1(s)$ is the degree of violation of the first constraint; Z_T is the total number of objectives; C_T is the total number of constraints that cannot be satisfied during the construction of trial solutions.

It should be noted that not all constraints specified in a problem are accounted for using Eq. 4.18. Maintenance windows, precedence and minimum duration constraints, just to name a few, can be satisfied during the construction of a trial solution and would not appear in Eq. 4.18. In other words, a complete trial solution would have satisfied these constraints already before the evaluation process is carried out. On the other hand, load and reserve constraints can only be checked upon completion of a complete trial solution and therefore the violation of these constraints, if there is any, can only be reflected through penalty terms in the objective function (Eq. 4.18). Detailed categorizations of constraints commonly encountered in PPMSO problems, as well as the appropriate methods of handling them, are presented in Section 4.4. In general, the trial schedule has to be run through a simulation model in order to calculate some elements of the objective function and whether certain constraints (those accounted for through penalty terms) have been violated.

After *m* ants have performed procedures 4.3.2 and 4.3.3, where *m* (the number of ants) is predefined in procedure 4.3.1, an iteration cycle has been completed. At this stage, a total of *m* maintenance schedules have been generated for this iteration. It should be noted that all ants in an iteration can generate their trial solutions concurrently, as they are working on the same set of pheromone trail distributions in decision space.

4.3.4 Local search

Recently, local search has been utilized to improve the optimisation ability of ACO. While it has been found to result in significant improvements in some applications (Dorigo *et al.*, 1997b; den Besten *et al.*, 2000), little success has been obtained in others (Merkle *et al.*, 2002). Local search has also been found useful for some problems where the formulation of heuristics is difficult (Dorigo *et al.*, 2004b).

In this research, local search is coupled with ACO to solve the PPMSO problem. As part of the local search algorithm proposed in this thesis, a 'target maintenance schedule' is selected from the trial solutions generated by the ACO algorithm, an example being the best maintenance schedule obtained in each iteration. A 'neighbor maintenance schedule' is then generated by performing local search based on the neighborhood definition, which must be specified beforehand, as discussed later. Satisfaction of constraints that can be checked during the construction of trial maintenance schedules (see Section 4.4), such as the allowable maintenance window and precedence constraints, are then checked. A simulation model is used to assess the quality of the 'neighbor maintenance schedule'. If the neighbor results in a better objective function cost (*OFC*), the original 'target maintenance schedule' is replaced. Based on the definition of neighborhood, more 'neighbor maintenance schedules' are generated until a termination criterion, which must be predefined, is met. A common termination criterion is the maximum number of 'neighbor maintenance schedules' allowable per 'target maintenance schedule'. By the end of the local search, the best-found 'neighbor maintenance schedule', or the original 'target maintenance schedule' in an event where no better local solution can be found, is adopted to proceed to the next step of the ACO-PPMSO algorithm (Figure 4.3).

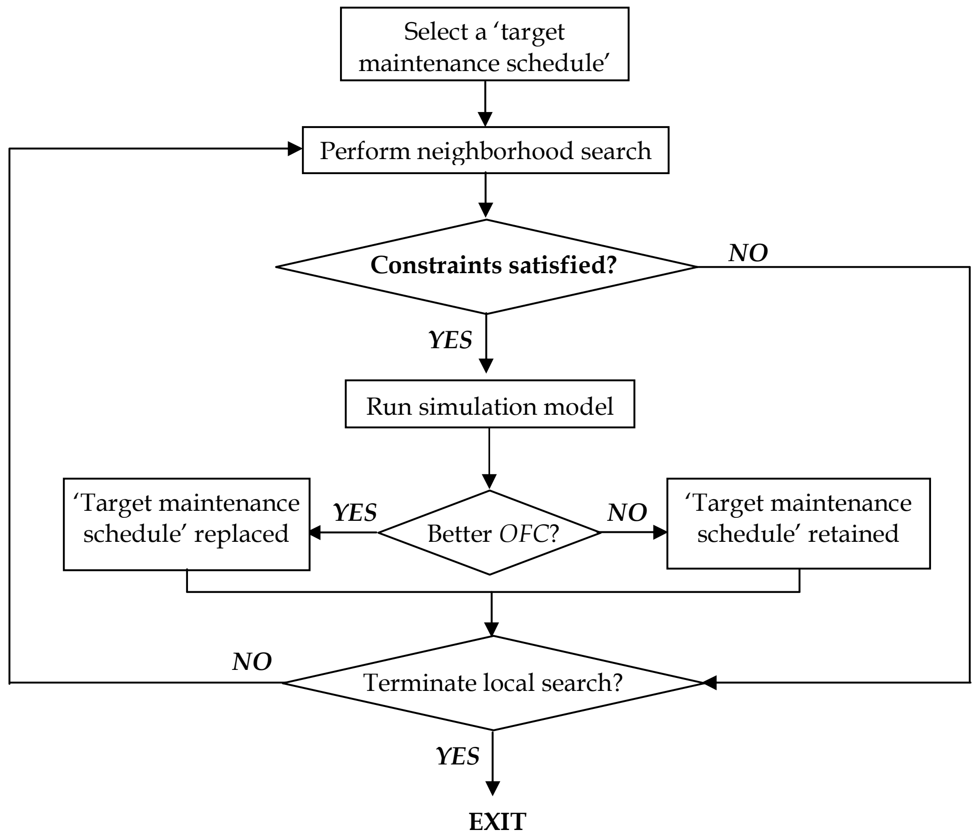


Figure 4.3: Local search framework for ACO-PPMSO algorithm

The definition of the neighborhood is problem-specific, and therefore must be carefully considered when applied to new optimisation problems. In this research, two local search operators are defined for the ACO-PPMSO algorithm, namely the *Duration Extender* and the *PPMSO-2-opt*, respectively. These operators search in different neighborhoods of the 'target maintenance schedule'.

(1) The ***Duration Extender*** operator is developed to increase the robustness of the ACO metaheuristic by dealing directly with the optimisation objectives. In particular, the operator looks for a reduced number of solutions that have shortened and/or deferred durations, which in turn, results in better *OFC*s.

As part of the *Duration Extender*, if the 'target maintenance schedule' does not include any shortening or deferral decisions, the local search routine is aborted. However, if this is not the case, local search is applied, as part of which a shortened or deferred task, $chosen\_d_n$, is randomly selected. If the selected task, *chosen_*d_n was originally shortened, local search will be performed in two neighborhoods: **(i)** The maintenance duration of the chosen task, *chosen_*d_n, is extended by s_n time periods, where s_n is the maintenance duration time step of task d_n. **(ii)** The maintenance duration of the chosen shortened task, *chosen_*d_n, is rescheduled by s_n periods earlier and s_n time periods are added to its maintenance duration. Otherwise, if the selected task was originally deferred, the minimum non-deferral maintenance duration is chosen and a start time is randomly selected for the task. For the *Duration Extender*, a termination criterion can be specified such that local search is aborted when all shortened/deferred task(s) in the 'target maintenance schedule' is/are considered.

(2) The ***PPMSO-2-opt*** operator is developed by modifying the 2-opt strategy used when solving the Travelling Salesman Problem (TSP) (Stützle *et al.*, 1997), where two edges of connected cities are exchanged. In *PPMSO-2-opt*, the maintenance start times of a pair of randomly selected tasks of the 'target maintenance schedule' are exchanged. It should be noted that the maximum number of possible 'neighbor maintenance schedules' formed based on a 'target maintenance schedule' ($^{N}C_2 = \frac{N!}{2! \cdot (N-2)!}$) can be specified as the termination criterion of the local search. Otherwise, a smaller number of local solutions can be defined as the stopping criterion.

4.3.5 Pheromone updating

As described previously in Section 3.2.2, two mechanisms, namely pheromone evaporation and pheromone rewarding, are involved in the pheromone updating process. Pheromone evaporation reduces all pheromone trails by a factor. In this way, exploration of the search space is encouraged by preventing a rapid increase in pheromone on frequently-chosen paths. Pheromone rewarding is performed in a way that reinforces good solutions.

In Section 3.3, various ACO algorithms were reviewed. As pointed out in the same section, these algorithms are distinguished from each other in the way pheromone updating is performed. In the ACO-PPMSO formulation, pheromone updating is performed on the pheromone matrices used for the three-stage selection process. A general pheromone updating formulation (regardless of the ACO algorithm adopted) is introduced for this purpose:

$$\tau_*(t+1) = \rho \cdot \tau_*(t) + \Delta\tau_*(t) \tag{4.19}$$

$$\Delta\tau_*(t) = \sum_{s \in Sol_{update}} q = \begin{cases} \dfrac{Q}{OFC(s_{update})} & \text{if } * \in s_{update} \\ 0 & \text{otherwise} \end{cases} \tag{4.20}$$

where t is the index of iteration; (1 - ρ) is the pheromone evaporation rate; lower asterisk * of τ_* denotes the element of the pheromone matrix under consideration ($\tau_{n,opt}$, $\tau_{n,dur}$ and $\tau_{n,dur,day}$ for decision variables v_1, v_2 and v_3, respectively); s_{update} is any trial schedule contained in $Sol_{update}(t)$, which is the set of trial schedules chosen to be rewarded in iteration t; $\Delta\tau_*(t)$ is the amount of pheromone rewarded to pheromone trail τ_* by the end of iteration t; $OFC(s_{update})$ is the objective function cost associated with the trial schedule s_{update} that contains element *; Q is the reward factor (a user-defined parameter).

In order to apply the different ACO algorithms reviewed in Section 3.3 to the PPMSO problem, additional specifications are made to the general pheromone updating rules:

(A) Ant System (AS)

In AS, the trial maintenance schedules obtained by all ants are rewarded by an amount of pheromone (Eq. 4.21), which is a function of the individual objective function cost (*OFC*).

$$Sol_{update}(t) = Sol_{all}(t) \tag{4.21}$$

where $Sol_{update}(t)$ is the set of trial maintenance schedules for which schedule components are rewarded by pheromone; $Sol_{all}(t)$ is the set of all trial maintenance schedules generated in iteration t.

(B) Elitist-Ant System (EAS)

In EAS, only the least-OFC schedule(s) in every iteration is/are rewarded (Eq. 4.22).

$$Sol_{update}(t) = s_{iter-best}(t) \tag{4.22}$$

where $s_{iter-best}(t)$ is the best maintenance schedule evaluated in iteration t.

(C) Max-Min Ant System (MMAS)

Similarly to EAS, MMAS only rewards iteration-best trial solution(s) (Eq. 4.22). Additionally, upper and lower bounds are imposed on the pheromone trails in order to prevent premature convergence and greater exploration of the solution surface. These bounds are given by:

$$\tau_{max}(t+1) = \frac{1}{1-\rho} \cdot \frac{Q}{OFC_{iter-best}(t)}. \tag{4.23}$$

$$\tau_{c,min}(t+1) = \frac{\tau_{max}(t+1)(1-\sqrt[n_c]{p_{best}})}{(avg_c - 1)\sqrt[n_c]{p_{best}}} \tag{4.24}$$

where n_c is the number of decision points for decision variable v_c; avg_c is the average number of decision paths available at each decision point for decision variable v_c; subscript c = 1, 2 and 3 refers to the three decision variables considered in procedure 4.3.2; p_{best} is the probability that the paths of the current iteration-best-solution, $s_{iter\text{-}best}(t)$, 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 lower and upper bound of pheromone are applied to all decision paths in the search space:

$$\tau_{c,min}(t) \leq \tau_{n,opt}(t) \leq \tau_{max}(t)\ ; opt \in DP_{c,n}\ c = 1,2,3 \text{ for all } t, n. \tag{4.25}$$

4.3.6 Termination of run

Procedures 4.3.2 to 4.3.5 are repeated until the termination criterion of an ACO run is met, e.g. either the maximum number of evaluations allowed has been reached or stagnation of the objective function cost has occurred. A set of maintenance schedules resulting in the minimum *OFC* is the final outcome of the optimisation run.

4.4 Constraint handling techniques in ACO-PPMSO

ACO is an unconstrained optimisation metaheuristic. As constraints are inevitable in PPMSO problems, there is a need to find ways of incorporating constraints during optimisation. In this research, two different constraint handling techniques are adopted. In order to decide which of the two techniques should be used, constraints encountered in PPMSO problems have been characterized using the following classification scheme:

Direct vs. indirect constraints: Constraints can be characterized based on the earliest stage at which they can be addressed during optimisation. The maintenance window (Eq. 4.2), continuity (Eq. 4.3), precedence (Eq. 4.6) and minimum maintenance duration (Eq. 4.8) constraints can be addressed when trial solutions are being generated during ant cycles (procedure described in Section 4.3.2). On the other hand, the violation of load (Eq. 4.4), reliability (Eq. 4.7) and resource (Eq. 4.5) constraints often cannot be identified from a partially built trial maintenance schedule. As part of the classification scheme introduced in this paper, the former constraints are referred to as direct constraints and the latter as indirect constraints.

Rigid vs. soft constraints: Constraints can also be classified based on their "rigidity". For rigid constraints, such as maintenance windows, continuity, minimum maintenance duration, precedence and load constraints, even the slightest violations are generally intolerable. On the other hand, constraints, such as resource and reliability constraints, may be able to be violated to a degree specified by decision makers and are therefore referred to as "soft" constraints.

The two constraint handling techniques used in the ACO-PPMSO formulation and the constraint types they are able to accommodate include:

Graph-based technique: This technique utilizes candidate lists during ant cycles when trial solutions are being constructed (Figure 4.1). Given a partially built trial schedule, a candidate list consists of the optional start times that are available for a maintenance task, such that the constraints under consideration are not violated. Direct and some rigid constraints, such as the maintenance window, precedence and minimum duration constraints, can be accounted for using this technique. During the construction of a trial maintenance schedule, an ant incrementally adds start times to a partially built schedule. By dynamically updating the candidate lists of 'unvisited units', only start times that would result in solutions that satisfy the maintenance window and precedence constraints are considered.

In order to illustrate the mechanism of the graph-based technique, the following example is considered. As part of a case study system, two maintenance tasks, namely task 1 and 2, are required to be scheduled over year 2006. Each task normally takes 16 days, which can be shortened by a time step of 4 days or deferred altogether if necessary. In addition, the following constraints must be satisfied:

Constraint 1 – Each task can be shortened only up to 50% of normal duration.

Constraint 2 – Both tasks can start as early as in 1 Jan 2006 and must finish no later than 30 June 2006.

Constraint 3 – No maintenance task should start on a public holiday.

Constraint 4 – Task 1 must precede task 2.

During the construction of trial maintenance schedules, either task 1 or task 2 can be considered first. For demonstration purposes, let us assume that task 1 is being considered first (Figure 4.4). As detailed in section 4.3.2, the selection of a maintenance duration and start time for task 1 is a three-stage process. At stage 1, decision has to be made whether maintenance task 1 is carried out as normal, is shortened in duration, or is deferred (Figure 4.4).

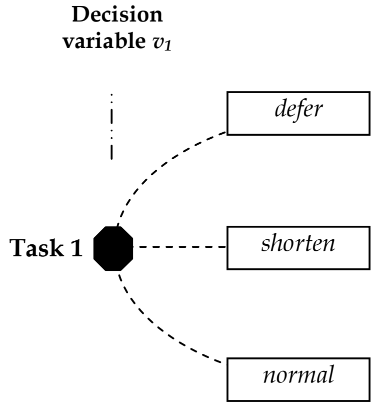


Figure 4.4: Stage-1 selection process

Once a decision has been made at stage 1, the decision path set available at stage 2 is updated correspondingly. If the 'normal' option was chosen at stage 1, the normal maintenance duration (16 days) would automatically be assigned to task 1. Similarly, a duration of 0 day is assigned if deferral was the decision made for task 1 at stage 1. Alternatively, if the 'shorten' option was chosen, decision paths of shortened durations 12, 8 and 4 days are available at stage 2. However, due to constraint 1, only a maximum of 50% of normal maintenance duration can be shortened, hence the 4-day duration decision path is no longer a valid decision path) and is therefore crossed out (as otherwise, an infeasible maintenance schedule that violates constraint 1 could be constructed (Figure 4.5)).

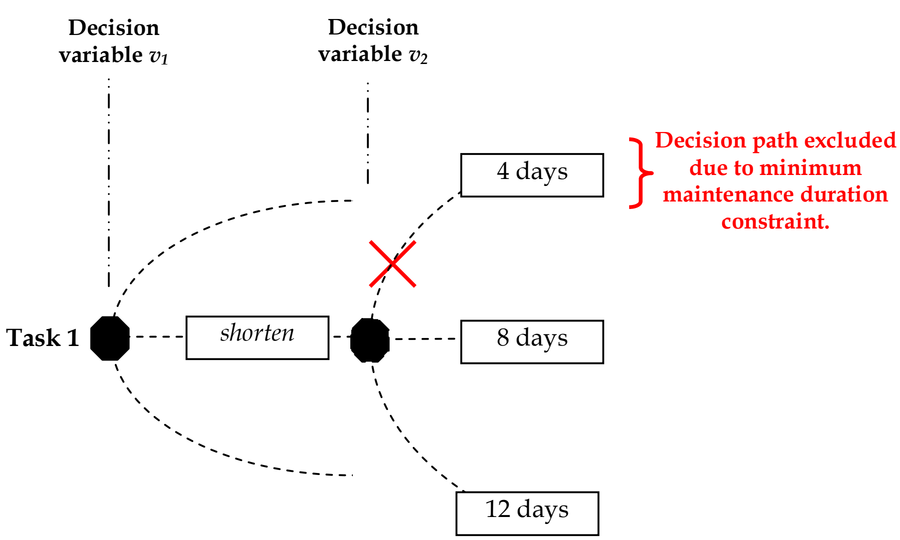


Figure 4.5: Handling of constraint 1

Once a decision has been made at stage 2, stage-3 selection is carried out. As the earliest start time and the latest finish time of task 1 are fixed (constraint 2), the start day decision paths available for task 1 are adjusted dynamically corresponding to the maintenance duration chosen at stage 2. Let us assume that a 12-day maintenance duration was chosen at stage 2. Consequently, the earliest and latest start days for task 1 are 1 Jan 2006 and 19 Jun 2006, respectively (Figure 4.6). In contrast, if a duration of 8 days was selected, the latest start day available at stage 3 would be 23 June 2006 (Figure 4.6).

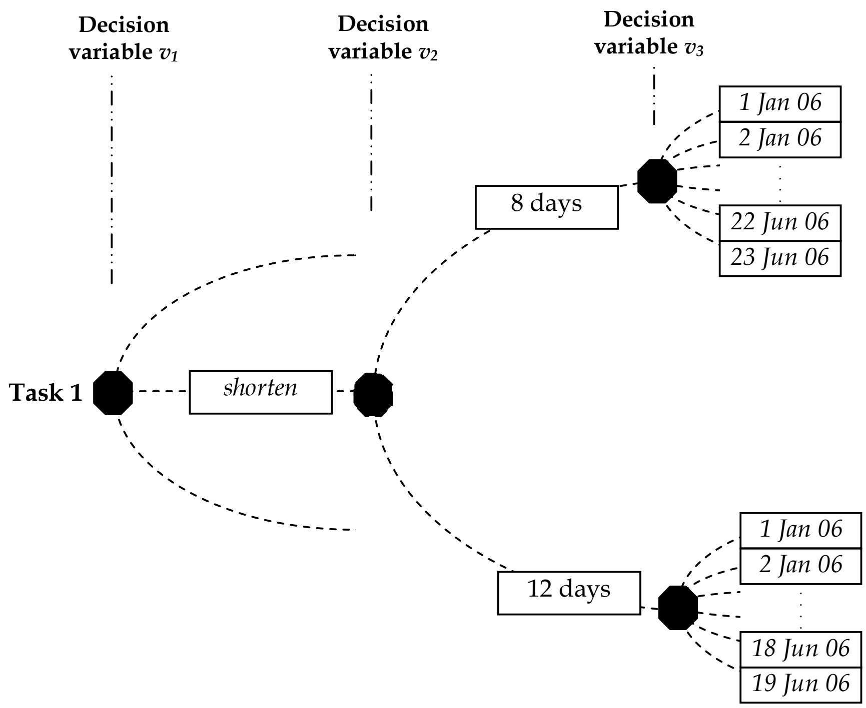


Figure 4.6: Handling of constraint 2

At stage 3, a start day is required to be chosen from the decision paths corresponding to start days of 1 Jan 2006 to 19 Jun 2006. However, Easter holidays fall on 14-17 April in 2006 and due to constraint 3, these decision paths are eliminated (Figure 4.7) so that only feasible trial schedules with regard to the public-holiday constraints are built.

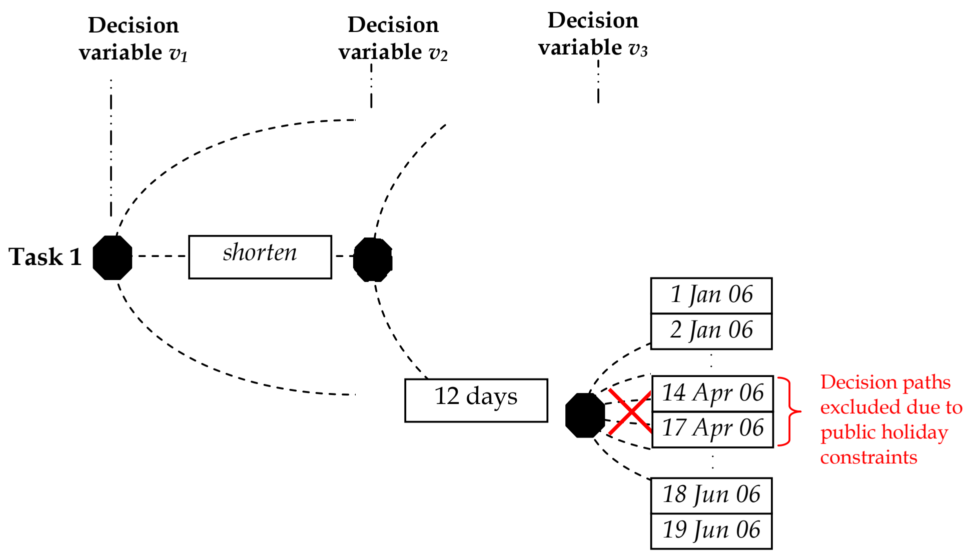


Figure 4.7: Handling of constraint 3

Once the decisions regarding the maintenance duration and start day for task 1 have been made, the three-stage selection process is repeated for task 2. It should be noted , however, that tasks 1 and 2 are related due to constraint 4. Therefore, the decisions made for task 1 have an impact on the options available for task 2. For example, if the maintenance of task 1 starts on Jan 1 2006, the earliest optional start day available for task 2 is 13 Jan 2006, if task 2 is carried out at normal duration (Figure 4.8).

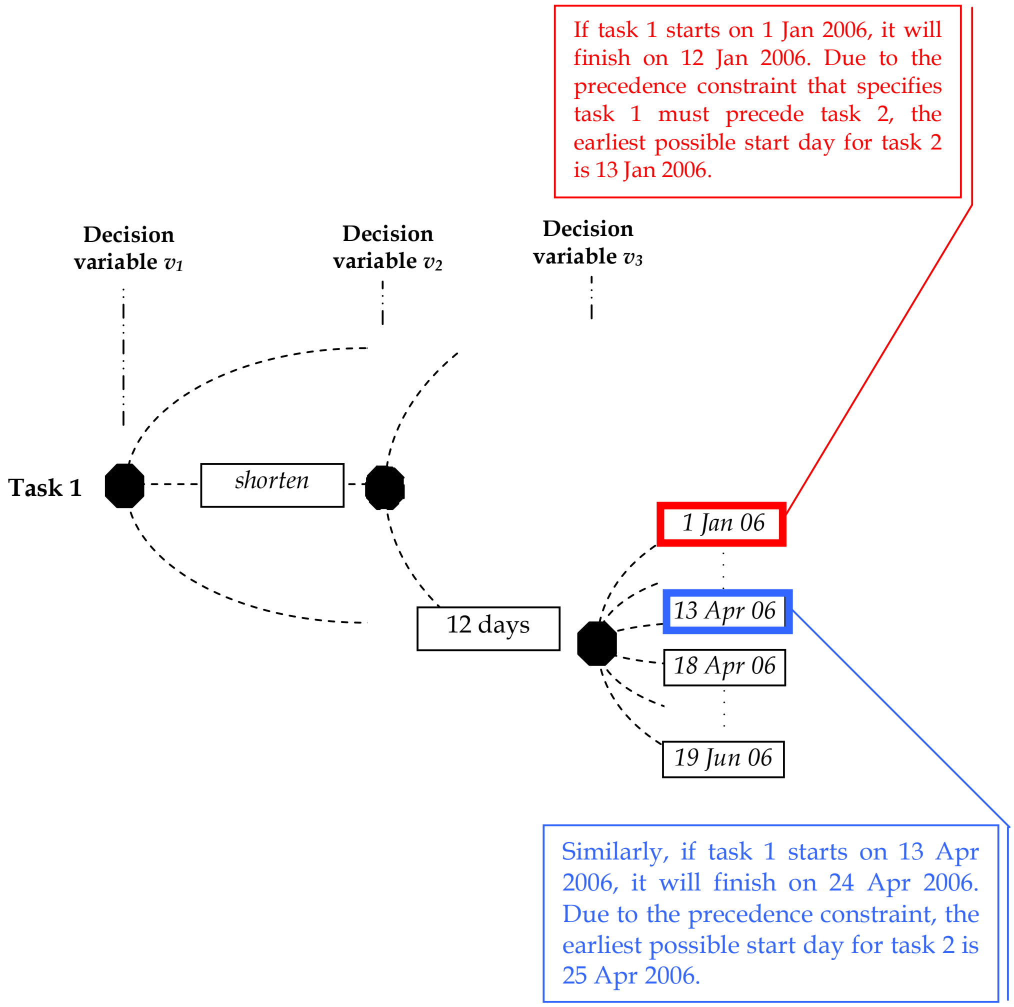


Figure 4.8: Handling of constraint 4

In short, the graph-based constraint handling technique dynamically adjusts the ACO-PPMSO graph (Figure 4.1) as trial maintenance schedules are being constructed incrementally.

Penalty-based technique: Penalties are the most common technique used for constraint handling when using metaheuristics (Coello Coello, 2002). A penalty function is used to transform a constrained optimisation problem into an unconstrained problem by adding or subtracting a value to/from the objective function cost based on the degree of constraint violation (Coello Coello, 2002). When applying ACO to the design of water distribution systems, Maier *et al.* (2003a) used penalty functions to negatively reinforce pipe diameters that result in trial solutions that violate pressure constraints. In ACO-PPMSO, penalty functions are used to address indirect or potentially soft constraints, such as the availability of personpower to perform the maintenance and load constraints. When dealing with soft constraints, penalty factors may be varied to reflect the amount of constraint violation that may be tolerated. Penalty costs also have to be used to account for indirect constraints, as the degree of constraint violation is not known until a complete trial solution has been constructed, as discussed earlier. In such cases, the degree of violation generally has to be obtained with the aid of a simulation model.

Using the last example, if load constraints were considered, they would have to be addressed using the penalty-based technique. This is because whether or not the system load could be met is unknown until a complete trial maintenance schedule has been constructed and run through a simulation model.

The ability to implement direct and some rigid constraints using the graph-based technique is one of the attractive features of using ACO for PPMSO. Firstly, by preventing the generation of infeasible solutions, the number of simulation model runs required is reduced. This is advantageous for real-world PPMSO problems, as the number of times the simulation model has to be run is a major source of computational overhead. Moreover, there are difficulties associated with the use of penalty-based techniques that remain unresolved at the time of writing, in spite of extensive research into this area (Coello Coello, 2002). For example, hand tuning is required for assigning appropriate penalty factors to each constraint and objective term in the objective function. Many researchers have proposed automated approaches for estimating penalty factors (Coello Coello, 2002). However, these approaches often introduce additional parameters for which appropriate values have to be provided. For the reasons outlined above, the graph-based approach is preferred over the penalty-based technique. However, in some instances, such as for indirect constraints, this approach cannot be used, as the degree of constraint violation can only be ascertained once a complete trial schedule has been generated. In such situations, the penalty-based technique has to be used (as discussed above).

4.5 Software development

A program has been coded in the Fortran 90 programming language to implement the ACO-PPMSO formulation. The full source code and sample of input files are attached in Appendix A. The random number generator subroutine used in the program was written by Nishimura (1997).

4.6 Summary

In this chapter, important contributions of the research work presented in this thesis have been detailed. The definition of a power plant maintenance scheduling optimisation (PPMSO) problem has been generalized by incorporating the options of duration shortening and maintenance deferral. Shortening of the duration and deferral of maintenance tasks are inevitable when scheduling for maintenance in a real world power system, such as in the event of anticipated demand increase. Incorporation of these options before proposing a new optimisation formulation allows PPMSO problems to be solved more practically.

A new formulation has been proposed to enable Ant Colony Optimisation (ACO) to be applied to PPMSO. Several issues with regard to the practical utilization of the proposed formulation have been resolved. These include the constraint-handling techniques, heuristic information and local search algorithms (optional in the formulation).

Constraints commonly encountered in PPMSO have been categorized based on whether they can be accounted for during the construction of a trial solution and whether they can be violated to achieve better objective values. Techniques for handling different constraint types have been proposed correspondingly. In particular, an advantage of using ACO for PPMSO is the possibility of incorporating some constraints during the construction of trial solutions, eliminating the need for complicated penalty functions in the formulation.

In order to improve the performance of the ACO formulation, a new heuristic formulation has been proposed. The heuristic formulation guides the optimisation algorithm to search in promising regions of a problem space, which should be extremely useful in the earlier stage of an optimisation run when the pheromone intensity is uniformly distributed over the search space. The algorithms adopted by two different local search operators, namely the *Duration Extender* and the *PPMSO-2-opt*, have been presented. Given a 'target maintenance schedule', the Duration Extender searches the neighborhood of the schedule for trial solutions that include less duration shortening, whereas the *PPMSO-2-opt* exchanges the maintenance start times of two randomly chosen tasks. These local search operators are designed to conduct a more refined search within the neighborhood of iteration-best maintenance schedules given by ACO, which were obtained using pheromone and heuristics.

Chapter 5
Testing On Benchmark Case Studies

Power plant maintenance scheduling optimisation (PPMSO) case studies can have completely different fitness landscapes, depending on the objectives, constraints, and number of variables of a particular problem. Adopting the representation of combinatorial optimisation problems outlined in Section 3.2.1, the definition of fitness landscapes given by Merz (2000) is as follows:

"*The fitness landscape L = (S, f, d) of a problem instance for a given combinatorial optimisation problem consists of a set of trial solutions, S, an objective function f : X → R, which assigns a real-valued fitness to each of the trial solutions in S, and a distance measure d, which defines the spatial structure of the landscape.*"

In other words, the fitness landscape is a characteristic of the search space of an optimisation problem, which is defined by the fitness function evaluated over the spectrum of different solutions to the problem. Therefore, it is important to test whether the new ACO-PPMSO formulation, the new heuristic formulation and the local search operator, developed as part of the contribution of this research can be effectively used for PPMSO case studies with different characteristics.

To test the utility of the new ACO-PPMSO formulation, two benchmark case studies that have been previously published in the literature (Escudero *et al.*, 1980; Yamayee *et al.*, 1983; Aldridge *et al.*, 1999; Dahal *et al.*, 1999; Dahal *et al.*, 2000; El-Amin *et al.*, 2000), as well as modified versions of both case studies, are considered. These case studies involve finding optimum maintenance schedule(s) for a 21-unit and a 22-unit power system, respectively. Despite the similarity in the number of generating units, the case studies are different in objectives and constraint requirements. The system specification and the application of the proposed ACO-PPMSO formulation to the original and the modified versions of the 21- and 22-unit case studies are detailed in Sections 5.1 and 5.2, respectively. Experimental procedures, results and analysis follow in Sections 5.3 and 5.4. A summary of the chapter is given in Section 5.5.

5.1 Benchmark case studies

As mentioned above, in order to test the new ACO-PPMSO formulation developed as part of this research, two benchmark case studies from the literature, namely a 21- and 22-unit case study, are utilized. The motivation for choosing these case studies is the availability of results obtained by other optimisation methods, with which the results obtained using the new ACO-PPMSO formulation can be compared. The specifications of the two case studies are detailed in Sections 5.1.1 and 5.1.2, respectively.

5.1.1 21-unit system

The first case study considered in this research is the 21-unit power plant maintenance problem investigated by Aldridge *et al*. (1999) and Dahal *et al*. (1999; 2000) using a number of metaheuristics. This case study is a modified version of the 21-unit problem introduced by Yamayee *et al*. (1983), and consists of 21 generating facilities, of which 20 units are thermal and one is hydropower. System details are listed in Table 5.1. All of the machines are to be scheduled for maintenance either in the first or second half of a year's planning horizon, which results in a combinatorial optimisation problem with approximately 5.18×10^{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.

Constraints to be satisfied include:

1. Maintenance window constraints: The earliest start time and latest finish time of maintenance tasks for each machine are detailed in Table 5.1.

2. Resource constraints: A limit of 20 maintenance personpower is available each week.

3. Demand constraints: A single peak load of 4739 MW has to be met.

As mentioned previously, a number of metaheuristics have been applied to this problem. Aldridge *et al*. (1999) 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*. (1999) obtained a maintenance schedule that resulted in a better objective function value than the best-known solution given by Aldridge *et al*. (1999), although this required slight violations of personpower constraints. In another study, Dahal *et al*. (2000) 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 5.4.

Table 5.1: Details of 21-unit system (Aldridge et al., 1999)

Problem formulation

Mathematically, this optimisation 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:

$$Min\left\{ SSR = \sum_{t \in T_{plan}} \left(\sum_{n=1}^{N} P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n - L_t \right)^2 \right\} \tag{5.1}$$

where P_n is the generating capacity of unit d_n; L_t is anticipated load for period t.

subject to the maintenance window, load and personpower constraints, as given by:

$$ear_n \leq start_n \leq lat_n - NormDur_n + 1 \qquad \text{for all } d_n \in D. \tag{5.2}$$

$$\sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} Res_{n,k} \leq ResAvai_t \quad \text{for all } t \in T_{plan} \tag{5.3}$$

$$\left(\sum_{n} P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \right) \geq L_t \quad \text{for all } t \in T_{plan} \tag{5.4}$$

where ear_n is the earliest start time for unit d_n; lat_n is the latest start time for unit d_n; $NormDur_n$ is the outage duration (week) for unit d_n; $start_n$ is the maintenance start time for unit d_n and $ResAvai_t$ is the personpower available at period t.

It should be noted that personpower is considered as a type of resource constraint. The maintenance window constraints are taken into account by the construction graph-based technique (Section 4.4), whereas both load and personpower constraints are indirect and are therefore taken into account by using penalty-based techniques (Section 4.4).

When applying the ACO-PPMSO formulation to this case study, the heuristic developed as part of this research (Eqs. 4.12 to 4.17) was used together with pheromone for selection of start times when generating trial maintenance schedules. It should be noted that the value of w in Eq. 4.12 was set to 1, as utilization of resource (personpower) constraints is involved in this case. Upon completion of a trial maintenance schedule, a simulation model was used to calculate the SSR value and any violations of personpower or load constraints associated with schedule *s*. The quality of individual maintenance schedules in this problem is given by an objective function cost (OFC), which is a function of the value of SSR and the total violation of personpower and load constraints (Eq. 5.5).

$$OFC(s) = SSR(s)\cdot\left(ManVio_{tot}(s)+1\right)\cdot\left(LoadVio_{tot}(s)+1\right) \tag{5.5}$$

where $OFC(s)$ is the objective function cost ($) associated with schedule *s*; $SSR(s)$ is the sum of squares of reserve generation capacity (MW^2) associated with schedule *s*; $ManVio_{tot}(s)$ is the total personpower shortfall (person) associated with schedule *s*; $LoadVio_{tot}(s)$ is the total demand shortfall (MW) associated with schedule *s*.

The calculation of constraint violations is given in Eqs. 5.6 to 5.9. For a trial maintenance schedule, the total personpower shortfall associated with schedule *s*, $ManVio_{tot}(s)$, is given by summation of the personpower shortage in all periods within the planning horizon:

$$ManVio_{tot}(s) = \sum_{t \in T_{MV}} \left(\sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} Res_{n,k} - ResAvai_t \right) \tag{5.6}$$

where T_{MV} is the period where personpower constraints are violated, and is given by:

$$T_{MV} = \left(t : \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} Res_{n,k} > ResAvai_t \right) \tag{5.7}$$

The total demand shortfall associated with schedule *s*, $LoadVio_{tot}(s)$, is the summation of demand shortfall in all periods within the planning horizon. The calculation of this value may be represented by the following equation.

$$LoadVio_{tot}(s) = \sum_{t \in T_{LV}} \left(\sum_{n} P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \right) \tag{5.8}$$

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

$$T_{LV} = (t : \sum_{n} P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n < L_t) \tag{5.9}$$

The *OFC* can be viewed as the virtual cost associated with a maintenance schedule.

5.1.2 22-unit system

The 22-unit power plant maintenance scheduling optimisation problem was first solved by Escudero *et al*. (1980) using an implicit enumeration algorithm and later by El-Amin *et al*. (2000) using tabu search. In this problem, each generating unit is required to be scheduled for maintenance once within a planning horizon of 52 weeks. Details of the system are shown in Table 5.2. The objective when scheduling for maintenance is to even out reserve generation capacity over the planning horizon subject to the following constraints:

(1) The maintenance window constraints specify that all units can be maintained anytime within the planning horizon and have to finish maintenance by week 52, except for unit 10, which can only be taken offline between weeks 6 and 22.

(2) Load constraints require peak demands (Table 5.3) to be met.

(3) The reliability constraint requires a minimum reserve of 20% of the peak demand throughout the planning horizon.

(4) The two precedence constraints specify that maintenance of units 2 and 5 has to be carried out before that of units 3 and 6, respectively.

(5) Units 15 and 16, as well as units 21 and 22, cannot be maintained simultaneously due to personpower constraints.

Table 5.2: Details of 22-unit system (Escudero et al., 1980)

NOTE: This table is included on page 104 of the print copy of the thesis held in the University of Adelaide Library.

Table 5.3: Weekly peak load of the 22-unit system (El-Amin *et al.*, 2000)

Week	Demand (MW)	Week	Demand (MW)	Week	Demand (MW)
1	1694	19	1695	37	2089
2	1714	20	1675	38	1989
3	1844	21	1805	39	1999
4	1694	22	1705	40	1982
5	1684	23	1766	41	1672
6	1763	24	1946	42	1782
7	1663	25	2116	43	1772
8	1583	26	1916	44	1556
9	1543	27	1737	45	1706
10	1586	28	1927	46	1806
11	1690	29	2137	47	1826
12	1496	30	1927	48	1906
13	1456	31	1907	49	1999
14	1396	32	1888	50	2109
15	1443	33	1818	51	2209
16	1273	34	1848	52	1779
17	1263	35	2118		
18	1655	36	1879		

<u>Problem formulation</u>

In order to even out reserve generation capacity, the formulation used in both Escudero *et al*. (1980) and El-Amin *et al*. (2000) for the 22-unit problem was designed to minimize the summed deviation of generation reserve from the average reserve over the entire planning horizon, LVL. Mathematically, the optimisation of this case study can be described as the minimization of the summed deviation of generation reserve from the average reserve over the planning horizon (Eqs. 5.6 to 5.8):

$$Min\left\{ LVL = \sum_{t \in T_{plan}} \left| Res_{avg} - Res_t \right| \right\} \tag{5.6}$$

where the generation reserve (Res_t) and average reserve (Res_{avg}) are given by:

$$Res_t = \sum_{n=1}^{N} P_n - \sum_{n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n - L_t \tag{5.7}$$

$$Res_{avg} = \frac{\sum_{t \in T_{plan}} Res_t}{T} \tag{5.8}$$

where L_t is the anticipated load demand for period t; P_n is the generating capacity of unit d_n; T is the total number of time indices, *subject to* the following constraints:

$$ear_n \le start_n \le lat_n - NormDur_n + 1 \qquad \text{for all } d_n \in D. \tag{5.9}$$

$$\left(\sum_n P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \right) \ge L_t \quad \text{for all } t \in T_{plan} \tag{5.10}$$

$$\left(\sum_n P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \right) \ge 1.2 L_t \quad \text{for all } t \in T_{plan} \tag{5.11}$$

$$\begin{cases} start_3 > start_2 + NormDur_2 - 1 \\ start_6 > start_5 + NormDur_5 - 1 \end{cases} \tag{5.12}$$

$$\begin{cases} X_{15,k} = 0 \text{ for } k = \left[start_{16}, \ldots, start_{16} + NormDur_{16} - 1 \right] \\ X_{16,k} = 0 \text{ for } k = \left[start_{15}, \ldots, start_{15} + NormDur_{15} - 1 \right] \\ X_{21,k} = 0 \text{ for } k = \left[start_{22}, \ldots, start_{22} + NormDur_{22} - 1 \right] \\ X_{22,k} = 0 \text{ for } k = \left[start_{21}, \ldots, start_{21} + NormDur_{21} - 1 \right] \end{cases} \tag{5.13}$$

It is interesting to note that, given the same objective, the objective formulations used by Escudero *et al*. (1980) and El-Amin *et al*. (2000) are quite different from that of Aldridge *et al*. (1999).

As there is no resource utilization throughout the planning horizon, there is no need for the inclusion of the resources term in the heuristic formulation (Eq. 4.12) for this case study (thus w may be set to 0). The precedence and maintenance window constraints of this system are direct and rigid constraints, which can be incorporated by using the graph-based technique, whereas the load and reliability constraints were taken into account using penalty functions. The objective function cost (OFC) used in this case study is a function of the reserve generation capacity LVL value and the total violation of load and reliability constraints (Eq. 5.14).

$$OFC(s) = LVL(S) \cdot \left(LoadResVio_{tot}(s) + 1\right) \tag{5.14}$$

where $OFC(s)$ is the objective function cost ($) associated with schedule s; $LVL(s)$ is the level of reserve generation capacity (MW) associated with schedule s; $LoadResVio_{tot}(s)$ is the total demand and reserve shortfall (MW) associated with schedule s.

It should be noted that the inclusion of a load constraint violation term in Eq. 5.14 is not necessary because violation of load constraints would be reflected as violation of reserve constraints. The calculation of constraint violations is given by Eqs. 5.15 and 5.16. The total load and reserve shortfall associated with schedule s, $LoadResVio_{tot}(s)$, is the summation of load and reserve shortfall in all periods within the planning horizon:

$$LoadResVio_{tot}(s) = \sum_{t \in T_{LV}} \left(\sum_{n} P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \right) \tag{5.15}$$

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

$$T_{LV} = (t : \sum_{n} P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n < 1.2 L_t) \tag{5.16}$$

5.2 Modified case studies

The general approach to PPMSO presented in this research includes options for maintenance duration shortening and deferral of maintenance tasks (Section 4.2). However, these options were not considered in previous studies that investigated the two case studies presented in Section 5.1 (Escudero *et al.*, 1980; Yamayee *et al.*, 1983; Aldridge *et al.*, 1999; Dahal *et al.*, 1999; Dahal *et al.*, 2000; El-Amin *et al.*, 2000). Therefore, in order to test the utility of the new ACO-PPMSO formulation, especially with regard to the impact of having the options of shortening and deferring maintenance tasks, modifications have been made to the 21- and 22-unit case study systems.

With their original system load, neither the 21-unit nor the 22-unit case study system require shortening or deferral of maintenance tasks. In order to create a need for these options, the system loads of both case studies have been increased and the options of shortening and deferral have been made available. Details of the modified case study systems, as well as the modifications made to the formulation for the application of ACO-PPMSO, are given in the following sections.

5.2.1 Modified 21-unit case study

The 21-unit case study system described in Section 5.1.1 is modified in the following ways:

(1) As shown in Figure 5.1, the original system load (4739MW) is increased by 5% throughout the whole planning horizon, and another 5% increment for weeks 15 to 25.

(2) Some maintenance tasks can be carried out in durations shorter than the original outage duration or deferred altogether (shown in Table 5.4). Essentially, outage durations can be shortened by a time step of 2 weeks to a certain minimum duration for each individual task (Table 5.4). The personpower requirements for shortened durations are also detailed in Table 5.4.

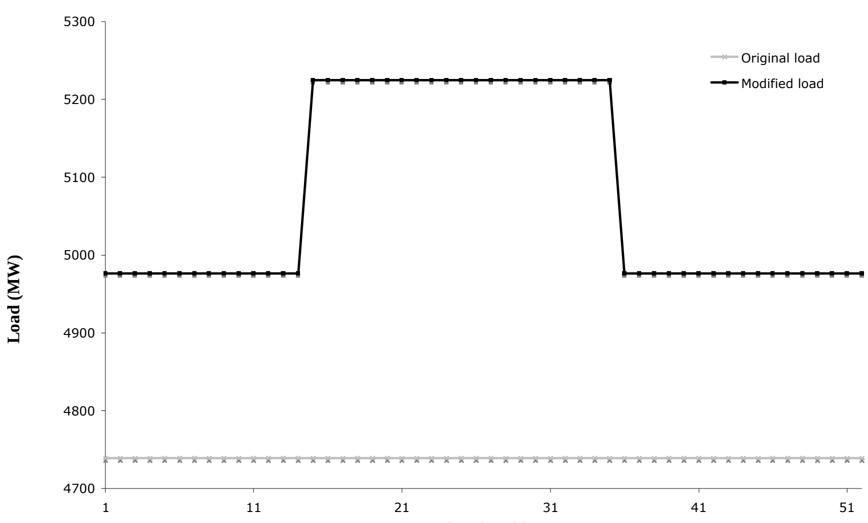


Figure 5.1: Original and modified system load for the 21-unit case study

Table 5.4: Personpower utilization for the modified 21-unit case study system

Unit No., n	Option	Optional Outage Duration, (weeks)	Personpower required for each week, $Res_{n,wk(wk=1,2,\ldots,\ chdur_n)}$ (person)
1	Normal	7	10, 10, 5, 5, 5, 5, 3
	Shorten	5	10, 10, 10, 8, 5
		3	15, 14, 14
	Defer	0	NIL
2	Normal	5	10, 10, 10, 5, 5
	Shorten	3	15, 15, 10
	Defer	0	NIL
3	Normal	2	15, 15
	Defer	0	NIL
4	Normal	1	20
	Defer	0	NIL
5	Normal	5	10, 10, 10, 10, 10
	Shorten	3	17, 17, 16
	Defer	0	NIL
6	Normal	3	15, 15, 15
	Defer	0	NIL
7	Normal	3	15, 15, 15
	Defer	0	NIL
8	Normal	6	10, 10, 10, 5, 5, 5
	Shorten	4	13, 13, 13, 6
	Defer	0	NIL
9	Normal	10	3, 2, 2, 2, 2, 2, 2, 2, 2, 3
	Shorten	8	3, 3, 3, 2, 2, 3, 3, 3
		6	4, 4, 3, 3, 4, 4
		4	6, 5, 5, 6
		2	11, 11
	Defer	0	NIL
10	Normal	4	10, 10, 5, 5
	Shorten	2	15, 15

	Defer	0	NIL
11	Normal	1	20
	Defer	0	NIL
12	Normal	3	10, 15, 15
	Defer	0	NIL
13	Normal	2	15, 15
	Defer	0	NIL
14	Normal	4	10, 10, 10, 10
	Shorten	2	20, 20
	Defer	0	NIL
15	Normal	2	15, 15
	Defer	0	NIL
16	Normal	2	15, 15
	Defer	0	NIL
17	Normal	1	20
	Defer	0	NIL
18	Normal	2	15, 15
	Defer	0	NIL
19	Normal	1	15
	Defer	0	NIL
20	Normal	4	10, 10, 10, 10
	Shorten	2	20, 20
	Defer	0	NIL
21	Normal	3	10, 10, 10
	Defer	0	NIL

Problem formulation

Despite the possibility of shortening and deferral options in this case study, they are unfavorable from both an economic and operations points of view. Therefore, the objective function used for the original version of this case study (Eq. 5.5) has been modified to:

$$\begin{aligned} OFC(s) = SSR(s) \cdot \left(ManVio_{tot}(s)+1\right) \cdot \left(LoadVio_{tot}(s)+1\right) \\ \cdot (DurCut_{tot}(s)+1) \end{aligned} \tag{5.17}$$

where $OFC(s)$ is the objective function cost (\$) associated with schedule s; $SSR(s)$ is the sum of squares of reserve generation capacity (MW^2) associated with schedule s; $ManVio_{tot}(s)$ is the total personpower shortfall (person) associated with schedule s; $LoadVio_{tot}(s)$ is the total demand shortfall (MW) associated with schedule s; $DurCut_{tot}(s)$ is the total reduction in maintenance duration (weeks) due to shortening and deferral associated with schedule s.

While the calculation of total demand shortfall associated with schedule s, $LoadVio_{tot}(s)$, total personpower shortfall associated with schedule s, $ManVio_{tot}(s)$, and the sum of squares of reserve generation capacity associated with schedule s, $SSR(s)$, are detailed in Section 5.1.1, the value of $DurCut_{tot}(s)$ is given by:

$$DurCut_{tot}(s) = \sum_{n=1}^{total\_n} (NormDur_n - chdur_n(s)) \tag{5.18}$$

where n is the index of maintenance task d_n, n = 1, 2, 3, . . . , $total\_n$, where $total\_n$ is the total number of maintenance tasks to be scheduled ($total\_n$ = 21 in this case); $NormDur_n$ is the normal duration of maintenance task d_n, and $chdur_n(s)$ is the maintenance duration (week) of task d_n associated with schedule s.

It should be noted that by using Eq. 5.17 to direct the search during an ACO run, a trial maintenance schedule that includes shortened and/or deferred maintenance tasks is being assigned a higher OFC, which represent an unfavorable solution to ACO during pheromone update.

As an additional constraint in this modified case study, the minimum-duration constraints can be addressed during the stage-2 selection process when a trial solution is being constructed (Section 4.3.2) by allowing only optional durations that are greater than the minimum duration for each maintenance task. In this way, trial solutions constructed will not violate the minimum duration constraints. For example, machine unit 1 that normally requires 7 days to be maintained, can be shortened to 5 or 3 days, or be deferred altogether (Table 5.4).

5.2.2 Modified 22-unit case study

The 22-unit case study detailed in Section 5.1.2 has been modified as follows:

1. The weekly loads for the modified 22-unit case study system are increased by 60% (Figure 5.2).

2. As shown in Table 5.5, the maintenance tasks are allowed to be performed within either the first or second half of the planning horizon (except for unit 10).

3. In the case of duration shortening, outage duration is reduced by a time step of two weeks until the corresponding minimum outage duration of a machine unit is reached (Table 5.5).

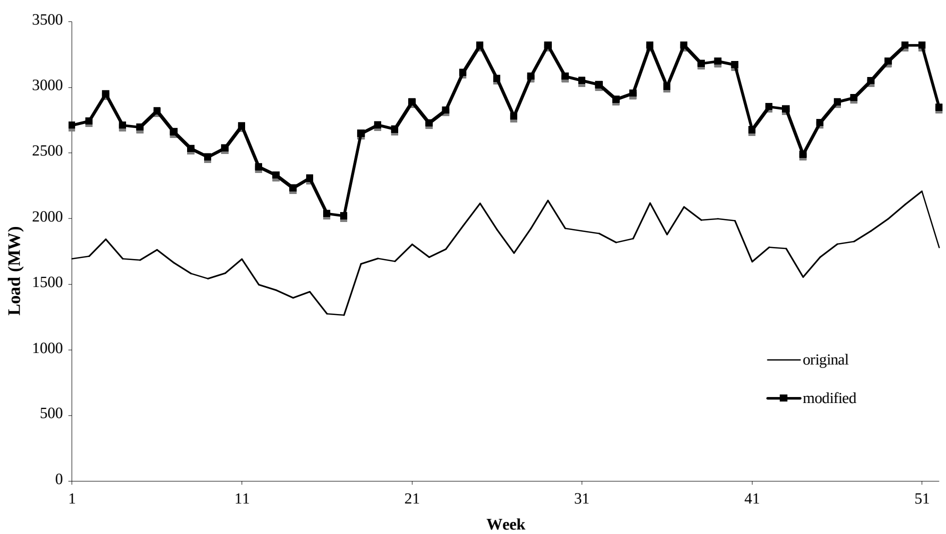


Figure 5.2: Modified 22-unit case study system – Weekly system load

<u>Problem formulation</u>

The objective function used for the original 22-unit case study (Eq. 5.14) has been modified to accommodate the options of shortening and deferral, and is given by:

$$OFC(s) = LVL(s) \cdot \left(LoadResVio_{tot}(s) + 1\right) \cdot (DurCut_{tot}(s) + 1) \tag{5.19}$$

where $OFC(s)$ is the objective function cost ($) associated with schedule s; $LVL(s)$ is the level of reserve generation capacity (MW) associated with schedule s; $LoadResVio_{tot}(s)$ is the total load constraint violation (MW) associated with schedule s; $DurCut_{tot}(s)$ is the total reduction in maintenance duration (weeks) due to shortening and deferral associated with schedule s.

The calculation of the total load constraint violation associated with schedule s, $LoadResVio_{tot}(s)$, and the level of reserve generation capacity associated with schedule s, $LVL(s)$ have been detailed previously in Section 5.1.2, whereas the value of the total duration shortened and deferred associated with schedule s, $DurCut_{tot}(s)$, is given by Eq. 5.18, where $total\_n$ = 22 in this case.

Table 5.5: Details of the modified 22-unit system

Unit No., n	Capacity, P_n (MW)	Normal outage duration, $NormDur_n$ (weeks)	Earliest start, ear_n (week)	Latest finish, lat_n (week)	Shortening allowed? [Optional shortened durations (week)]	Deferral allowed?
1	100	6	1	26	Y [4, 2]	Y
2	100	3	1	26	N	Y
3	100	3	1	26	N	Y
4	100	3	1	26	N	Y
5	90	6	1	26	Y [4, 2]	Y
6	90	4	1	26	Y [2]	Y
7	95	3	1	26	N	Y
8	100	4	1	26	Y [2]	Y
9	650	5	1	26	Y [3]	Y
10	610	12	6	22	Y [10, 8, 6, 4]	Y
11	91	4	1	26	Y [2]	Y
12	100	8	1	26	Y [6, 4]	Y
13	100	3	1	26	N	Y
14	100	6	27	52	Y [4]	Y
15	220	5	27	52	Y [3]	Y
16	220	6	27	52	Y [4]	Y
17	100	5	27	52	Y [3]	Y
18	100	5	27	52	Y [3]	Y
19	220	3	27	52	N	Y
20	220	3	27	52	N	Y
21	240	3	27	52	N	Y
22	240	5	27	52	Y [3]	Y

5.3 Experimental Procedure

Experiments have been conducted on both the original and modified versions of the 21-unit and 22-unit case studies to assess the utility of the proposed ACO-PPMSO formulation. Particular emphasis was given to assessing the usefulness of the heuristics developed, the impact of the two local search operators and the overall performance of the proposed ACO-PPMSO formulation.

A. Usefulness of heuristic formulation

The effectiveness of the new heuristic formulations for general PPMSO problems (Eqs. 4.10 to 4.12) introduced in Section 4.3.2 was examined by conducting optimisation runs with and without the heuristics (the latter was achieved by setting the relative weight of the heuristic, β, in Eq. 4.9 to 0). In addition, the sensitivity of optimisation results to increasing values of β was checked. It should be noted that, as a control, the value of α in Eq. 4.9 was fixed at 1.

B. Impact of local search operators

The impact of local search on the performance of the ACO-PPMSO algorithm was also investigated, both with and without heuristic. While the *PPMSO-2-opt* local search operator (see Section 4.3.4) can be tested with both original and modified versions of the 21- and 22-unit case studies, the *Duration Extender* local search operator (see Section 4.3.4) can only be tested with the modified version of these case studies, due to the availability of shortening and deferral options. The total number of trial solutions evaluated in the ACO runs with local search was identical to those without local search.

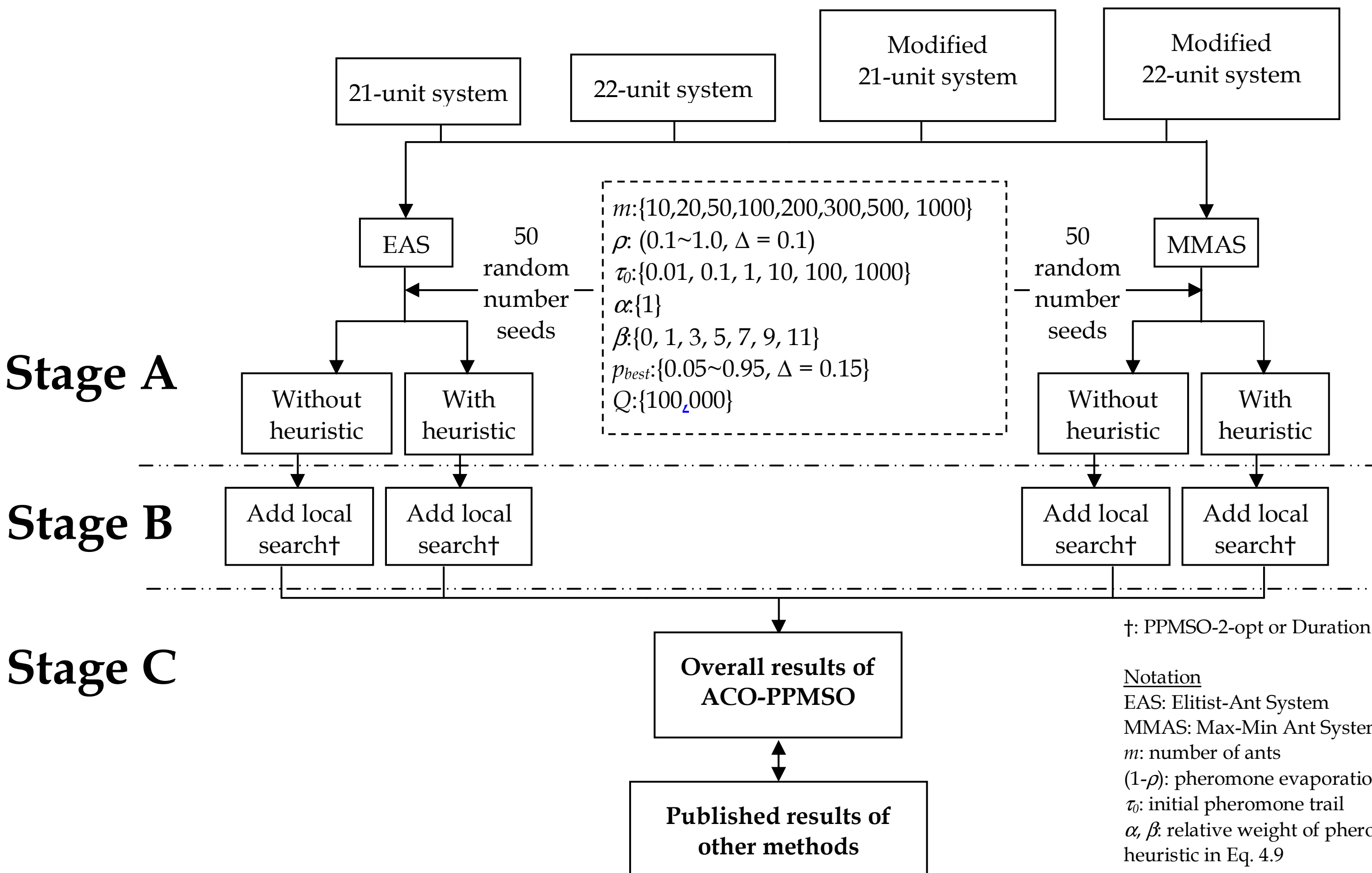
C. Overall performance of ACO-PPMSO

In order to check the overall performance of ACO for solving PPMSO problems, the results obtained for the two original case studies were compared with those obtained using other optimisation methods in previous studies. The optimised maintenance schedules obtained for the modified case studies were analysed and are discussed in detail. In addition, the ability of ACO-PPMSO to handle soft constraints was investigated.

In order to achieve the objectives outlined above, the testing procedure shown in Figure 5.3 was implemented separately for each of the four case studies. Items A, B and C mentioned above were investigated at Stages A, B and C in the testing procedure, respectively. To minimize the impact the ACO algorithm and parameters used have on the evaluation of the effectiveness of the heuristic, local search and overall performance of the ACO-PPMSO algorithm, two ACO algorithms, namely Elitist-Ant System parameters (shown in the dashed box in Figure 5.3) were used to solve the problem instance under consideration. In addition, each run was repeated 50 times with different random number seeds in order to minimize the influence of random starting values in the solution space on the results obtained and to enable Student's *t*-tests to be conducted to determine whether any differences in the results obtained were significant. In total, 3,024 different combinations of parameters, each with 50 different starting random number seeds, were evaluated as part of this study. In order to facilitate fair comparisons, the same number of evaluations per optimisation run were used as in previous studies that investigated the 21-unit case problem (30,000 evaluations). In this research, 'one ACO run' is defined as the use of an ACO algorithm with or without using heuristic information, with or without local search and with a defined set of parameters to solve a PPMSO instance. An example of an ACO run is the use of EAS to solve the modified 21-unit case study with heuristic information and *Duration Extender* local search and a defined parameter set of m = 200; ρ = 0.9; τ_0 = 0.1; Q = 500,000; α = 1, β = 11, repeated for 50 random number seeds. The overall performance of a parameter set is then assessed based on the objective function cost (OFC) averaged over the 50 simulations using different random number seeds. An analysis of the results obtained with the testing procedure outlined in Figure 5.3 is given in Section 5.4.

5.4 Results and analysis

The experimental results obtained for the original 21- and 22-unit case studies are summarized in Tables 5.6 to 5.9, while those for the modified case studies are presented in Tables 5.10 to 5.13. The detailed results can be found in Appendices B and C.



†: PPMSO-2-opt or Duration Extender.

Notation
EAS: Elitist-Ant System
MMAS: Max-Min Ant System
m: number of ants
(1-ρ): pheromone evaporation rate
τ_0: initial pheromone trail
α, β: relative weight of pheromone and heuristic in Eq. 4.9
p_{best}: see Eq. 4.24
Q: reward factor

Figure 5.3: Testing procedure

Table 5.6: Results for the 21-unit unit problem instance given by Elitist-Ant System (EAS) [deviation from best-known OFC of $13.66M]

Heur -istic	Local search	Best OFC ($M)	Average OFC ($M)	Worst OFC ($M)	Std dev. ($M)	Average evaluations[a]	Best parameter settings $\{m; \rho; \tau_0; \beta\}$[b]
✗	✗	14.84 [8.64%]	140.49 [928.48%]	365.13 [2572.99%]	86.00	28,841	{300; 0.9; 0.01; 0}
✓	✗	13.68 [0.15%]	13.71 [0.37%]	13.85 [1.39%]	0.03	20,692	{200; 0.9; 0.01; 9}
✗	*PPMSO -2-opt*	13.74 [0.59%]	51.62 [277.89%]	138.80 [916.11%]	33.72	25,494	{300; 0.8; 0.1; 0}
✓	*PPMSO -2-opt*	13.66 [0%]	13.70 [0.29%]	13.82 [1.17%]	0.03	22,434	{200; 0.9; 0.01; 9}

[a] Number of evaluations to reach the best solution in one run averaged over 50 runs with different random starting positions.

[b] m: number of ants; $(1-\rho)$: pheromone evaporation rate; τ_0: initial pheromone trail; β: relative weight of heuristic in Eq. 4.9.

Table 5.7: Results for the 21-unit unit problem instance given by Max-Min Ant System (MMAS) [deviation from best-known OFC of $13.66M]

Heur -istic	Local search	Best OFC ($M)	Average OFC ($M)	Worst OFC ($M)	Std dev. ($M)	Average evaluations[c]	Best parameter settings $\{m; \rho; p_{best}; \beta\}$[d]
✗	✗	13.86 [1.46%]	16.11 [17.94%]	43.35 [217.35%]	5.95	16,480	{10; 0.3; 0.2; 0}
✓	✗	13.66 [0%]	13.68 [0.15%]	13.72 [0.44%]	0.01	13,593	{20; 0.4; 0.35; 5}
✗	*PPMSO -2-opt*	13.80 [1.02%]	17.90 [31.04%]	69.04 [405.42%]	10.51	18,089	{50; 0.2; 0.05; 0}
✓	*PPMSO -2-opt*	13.66 [0%]	13.69 [0.22%]	13.78 [0.88%]	0.02	15,867	{50; 0.5; 0.5; 11}

[c] Number of evaluations to reach the best solution in one run averaged over 50 runs with different random starting positions.

[d] m: number of ants; $(1-\rho)$: pheromone evaporation rate; p_{best}: refer to Eq. 4.24; β: relative weight of heuristic in Eq. 4.9.

Table 5.8: Results for the 22-unit unit problem instance given by Elitist-Ant System (EAS)
[deviation from best-known OFC of $52.06]

Heur-istic	Local search	Best OFC ($M)	Average OFC ($M)	Worst OFC ($M)	Std dev. ($M)	Average evaluations[a]	Best parameter settings $\{m; \rho; \tau_0; \beta\}$[b]
✗	✗	63.41 [21.80%]	72.27 [38.82%]	81.15 [55.88%]	4.17	29,294	{200; 0.9; 100; 0}
✓	✗	58.41 [12.20%]	64.31 [23.53%]	73.25 [40.70%]	3.21	28,384	{300; 0.9; 1; 11}
✗	*PPMSO -2-opt*	58.91 [13.16%]	67.03 [28.76%]	79.99 [53.65%]	4.70	25,858	{300; 0.8; 1; 0}
✓	*PPMSO -2-opt*	55.67 [6.93%]	60.55 [16.31%]	67.97 [30.56%]	2.90	26,931	{300; 0.8; 10; 11}

[a] Number of evaluations to reach the best solution in one run averaged over 50 runs with different random starting positions.

[b] m: number of ants; $(1-\rho)$: pheromone evaporation rate; τ_0: initial pheromone trail; β: relative weight of heuristic in Eq. 4.9.

Table 5.9: Results for the 22-unit unit problem instance given by Max-Min Ant System (MMAS)
[deviation from best-known OFC of $52.06M]

Heur-istic	Local search	Best OFC ($M)	Average OFC ($M)	Worst OFC ($M)	Std dev. ($M)	Average evaluations[c]	Best parameter settings $\{m; \rho; p_{best}; \beta\}$[d]
✗	✗	59.91 [15.08%]	66.90 [28.51%]	76.17 [46.31%]	3.67	24,597	{100; 0.9; 0.5; 0}
✓	✗	55.72 [7.03%]	62.22 [19.52%]	68.65 [31.87%]	2.97	28,433	{200; 0.9; 0.2; 11}
✗	*PPMSO -2-opt*	57.64 [10.72%]	64.81 [24.49%]	76.65 [47.23%]	4.27	27,455	{200; 0.8; 0.5; 0}
✓	*PPMSO -2-opt*	54.56 [4.80%]	59.42 [14.14%]	66.56 [27.85%]	2.87	24,537	{200; 0.8; 0.35; 11}

[c] Number of evaluations to reach the best solution in one run averaged over 50 runs with different random starting positions.

[d] m: number of ants; $(1-\rho)$: pheromone evaporation rate; p_{best}: refer to Eq. 4.24; β: relative weight of heuristic in Eq. 4.9.

Table 5.10: Results for the Modified 21-unit unit problem instance given by Elitist-Ant System (EAS) [deviation from best-known OFC of $15.71M]

Heu-ristic	Local search	Best OFC ($M)	Average OFC ($M)	Worst OFC ($M)	Std dev. ($M)	Average $DurCut_{tot}$ (weeks)	Average evaluations[a]	Best parameter settings $\{m; \rho; \tau_0; \beta\}$[b]
✗	✗	65.61 [317.63%]	120.39 [666.33%]	209.05 [1230.68%]	39.16	17.6	27,538	{300; 0.9; 0.01; 0}
✓	✗	16.15 [2.80%]	24.42 [55.44%]	31.06 [97.71%]	5.16	6.4	29,029	{500; 0.9; 0.01; 1}
✗	*Duration Extender*	51.17 [225.72%]	105.02 [568.49%]	216.85 [1280.33%]	37.63	16.6	20,226	{200; 0.9; 0.01; 0}
✓	*Duration Extender*	15.94 [1.46%]	25.73 [63.78%]	47.65 [203.31%]	7.22	6.6	28,929	{500; 0.9; 0.01; 1}
✗	*PPMSO-2-opt*	68.42 [335.52%]	135.13 [760.15%]	219.07 [1294.46%]	36.67	19.3	28,784	{300; 0.9; 0.01; 0}
✓	*PPMSO-2-opt*	16.12 [2.61%]	26.87 [71.04%]	41.24 [162.51%]	5.17	6.9	28,213	{500; 0.9; 0.01; 1}

[a] Number of evaluations to reach the best solution in one run averaged over 50 runs with different random starting positions.

[b] m: number of ants; $(1-\rho)$: pheromone evaporation rate; τ_0: initial pheromone trail; β: relative weight of heuristic in Eq. 4.9.

Table 5.11: Results for the Modified 21-unit problem instance given by Max-Min Ant System (MMAS) [deviation from best-known OFC of $15.71M]

Heu-ristic	Local search	Best *OFC* ($M)	Average OFC ($M)	Worst OFC ($M)	Std dev. ($M)	Average $DurCut_{tot}$ (weeks)	Average evaluations[c]	Best parameter settings $\{m; \rho; p_{best}; \beta\}$[d]
✗	✗	28.69 [82.62%]	61.32 [290.32%]	119.15 [658.43%]	19.54	11.8	16,934	{20; 0.2; 0.2; 0}
✓	✗	15.97 [1.65%]	19.69 [25.33%]	29.03 [84.79%]	4.02	5.6	18,551	{50; 0.2; 0.05; 1}
✗	*Duration Extender*	27.25 [73.46%]	59.48 [278.61%]	106.45 [577.59%]	17.46	11.3	27,207	{300; 0.9; 0.2; 0}
✓	*Duration Extender*	15.74 [0.19%]	20.13 [28.13%]	29.72 [89.18%]	4.32	5.7	18,871	{50; 0.1; 0.05; 1}
✗	*PPMSO-2-opt*	33.64 [114.13%]	71.67 [356.21%]	132.10 [740.87%]	24.64	12.6	24,898	{500; 0.1; 0.05; 0}
✓	*PPMSO-2-opt*	15.71 [0%]	22.04 [40.29%]	29.66 [88.80%]	4.86	6.1	23,713	{500; 0.7; 0.05; 1}

[c] Number of evaluations to reach the best solution in one run averaged over 50 runs with different random starting positions.

[d] m: number of ants; $(1-\rho)$: pheromone evaporation rate; p_{best}: refer to Eq. 4.24; β: relative weight of heuristic in Eq. 4.9.

Table 5.12: Results for the Modified 22-unit unit problem instance given by Elitist-Ant System (EAS) [deviation from best-known OFC of $916.12]

Heu-ristic	Local search	Best OFC ($)	Average OFC ($)	Worst OFC ($)	Std dev. ($)	Average $DurCut_{tot}$ (weeks)	Average evaluations[a]	Best parameter settings $\{m; \rho; \tau_0; \beta\}$[b]
✗	✗	2186.22 [138.64%]	2797.85 [205.40%]	4267.31 [365.80%]	410.33	21.9	27,896	{300; 0.9; 0.01; 0}
✓	✗	1365.60 [49.06%]	1756.34 [91.72%]	2153.97 [135.12%]	175.55	13.8	28,648	{500; 0.9; 0.01; 11}
✗	*Duration Extender*	1953.99 [113.29%]	2529.19 [176.08%]	4140.45 [351.96%]	454.71	19.3	26,844	{300; 0.9; 0.01; 0}
✓	*Duration Extender*	1194.27 [30.36%]	1652.63 [80.39%]	2135.76 [133.13%]	167.85	12.7	27,448	{500; 0.9; 0.01; 11}
✗	*PPMSO-2-opt*	2331.92 [154.54%]	2876.16 [213.95%]	4357.14 [375.61%]	501.14	23.2	26,187	{300; 0.9; 0.01; 0}
✓	*PPMSO-2-opt*	1174.10 [28.16%]	1724.37 [88.23%]	2238.34 [144.33%]	172.63	13.7	21,718	{300; 0.9; 0.01; 11}

[a] Number of evaluations to reach the best solution in one run averaged over 50 runs with different random starting positions.

[b] m: number of ants; (1-ρ): pheromone evaporation rate; τ_0: initial pheromone trail; β: relative weight of heuristic in Eq. 4.9.

Table 5.13: Results for the Modified 22-unit unit problem instance given by Max-Min Ant System (MMAS) [deviation from best-known OFC of $916.12]

Heu-ristic	Local search	Best *OFC* ($)	Average *OFC* ($)	Worst *OFC* ($)	Std dev. ($)	Average $DurCut_{tot}$ (weeks)	Average evaluations	Best parameter settings $\{m; \rho; p_{best}; \beta\}$[d]
✗	✗	1439.33 [57.11%]	2076.43 [126.65%]	3998.67 [336.78%]	440.16	15.6	26,219	{300; 0.6; 0.2; 0}
✓	✗	1008.13 [10.04%]	1489.54 [62.59%]	2017.44 [120.22%]	280.45	12.1	23,329	{20; 0.3; 0.35; 11}
✗	*Duration Extender*	1632.25 [78.17%]	2099.93 [129.22%]	4085.77 [345.99%]	467.37	15.2	21,676	{20; 0.3; 0.2; 0}
✓	*Duration Extender*	1009.47 [10.19%]	1492.6 [62.93%]	2049.26 [123.69%]	267.38	12.3	22,254	{50; 0.3; 0.2; 11}
✗	*PPMSO-2-opt*	1614.39 [76.22%]	2068.8 [125.82%]	3936.71 [329.72%]	425.87	15.0	20,767	{20; 0.3; 0.2; 0}
✓	*PPMSO-2-opt*	1001.12 [9.28%]	1513.86 [65.25%]	2084.59 [127.55%]	306.26	12.4	21,347	{50; 0.1; 0.35; 11}

[a] Number of evaluations to reach the best solution in one run averaged over 50 runs with different random starting positions.

[b] m: number of ants; (1-ρ): pheromone evaporation rate; p_{best}: refer to Eq. 4.24; β: relative weight of heuristic in Eq. 4.9.

Stage A: Impact of heuristic

The effectiveness of the new heuristic formulations (as detailed in Section 4.3.2) was checked using a Student's *t*-test at a 95% significance level (calculations are shown in Appendices B2 and C2). Overall, the new heuristic formulation for applying ACO to PPMSO problems significantly improved the results obtained for all four case studies, with and without the use of a local search operator and for both ACO algorithms (Table 5.14). It can be seen that when the heuristic was used, not only were the average OFCs improved, but the standard deviations of the OFCs were also significantly smaller for all case studies (Tables 5.6 to 5.13), indicating that use of the new heuristic formulation enables good solutions to be found consistently.

Table 5.14: Impact of the new heuristic formulation with and without using local search

	21-unit system		*22-unit system*		*Modified 21-unit system*		*Modified 22-unit system*	
	EAS	**MMAS**	**EAS**	**MMAS**	**EAS**	**MMAS**	**EAS**	**MMAS**
Without local search	+	+	+	+	+	+	+	+
Duration Extender (see Section 4.3.4)	NT	NT	NT	NT	+	+	+	+
PPMSO-2-opt (see Section 4.3.4)	+	+	+	+	+	+	+	+

Notation:
+: Significant positive impact; −: Significant negative impact; NIL: Insignificant impact; NT: Not tested.

In order to gain a better understanding of the searching behavior of the ACO algorithms in solving each of the four case studies with and without heuristic, the optimisation process of ACO runs was examined. The investigation is firstly facilitated by comparing the iteration-best objective function value (SSR and LVL) curves (referred to as IB-SSR and IB-LVL curves hereafter) of ACO runs with and without heuristic. Secondly, in order to investigate constraint satisfaction during ACO runs with and without heuristic, the ability of ACO in accessing the feasible and infeasible regions of the case studies' solution space was assessed with the aid of a measure proposed in this research called the "infeasibility ratio", ψ. ψ is defined as the ratio of the number of infeasible solutions to the total number of solutions evaluated in a particular iteration t and is given by:

$$\psi(t) = \frac{\text{number of infeasible trial solutions}(t)}{\text{total number of trial solutions evaluated}(t)} \tag{5.20}$$

For the four case studies investigated, the following curves generated by selected EAS and MMAS runs when heuristics were and were not used are shown in Figures 5.4 to 5.11:

- Objective function values (SSR, LVL and $DurCut_{tot}$) associated with iteration-best schedules (referred to as IB-SSR, IB-LVL and IB-$DurCut_{tot}$ hereafter)

- Violation of various constraints (demand and personpower shortfall) associated with iteration-best schedules (referred to as IB-$LoadVio_{tot}$, IB-$ManVio_{tot}$ and IB-$LoadResVio_{tot}$ hereafter)

- Infeasibility ratio, ψ

It should be noted that the curves plotted in Figures 5.4 to 5.11 are given by the ACO runs using the best parameter settings (shown in last rows of Tables 5.6 to 5.13) obtained during the test. The random number seeds used in those runs are also shown (Figures 5.4 to 5.11).

Overall, the ACO-PPMSO algorithm is shown to search the problem search space effectively by minimizing the objective function values (SSR, LVL and $DurCut_{tot}$) for the four case studies investigated. This is illustrated by the decreasing trends of IB-SSR, IB-LVL and IB-$DurCut_{tot}$ over iterations during ACO runs as shown in (a) and (b) of Figures 5.4 to 5.11. In addition, the process of evolution of feasible trial solutions (i.e. solutions that do not violate constraints) using ACO-PPMSO is clearly shown in (c) and (d) of Figures 5.4 to 5.11.

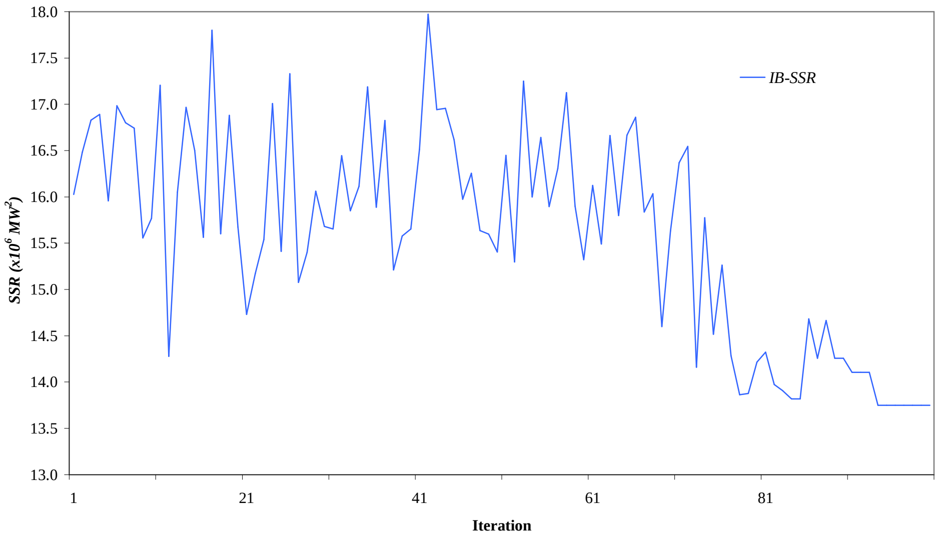
It can also be observed that convergence occurs at latter stages of runs where relatively larger ant populations (m > 50) are used. Interestingly, when smaller ant populations are used (m ≤ 50), the ACO search seems to restart several times during a run, as depicted by the multiple spikes followed by decreasing IB-SSR and IB-LVL curves shown in Figure 5.5, 5.9 and 5.11b.

By interpreting Figures 5.4 to 5.11 in detail, it can also be deduced that the four case studies investigated have quite different fitness landscapes. The original 21-unit case study is more highly constrained, as indicated by the large number of demand and personpower shortfalls associated with trial solutions constructed during the early stages of the optimisation runs (Figures 5.4c and 5.5c). On the other hand, the load constraints of the original 22-unit case study are easily satisfied, as the infeasibility ratio is approaching or equal to zero throughout the ACO run (Figures 5.6c and 5.7c). As for the modified versions of case studies, it can be observed that the optimisation process of the modified 21-unit case study is dominated by personpower constraints (Figures 5.8c and 5.9c), whereas the load constraints of the modified 22-unit case study are tighter compared to its original counterpart (Figures 5.10c and 5.11). It is interesting to find that for MMAS, smaller populations of ants are found to be more effective for more highly constrained problems, such as the original 21-unit problem (Table 5.7), the modified 21-unit problem (Table 5.11) and the modified 22-unit problem (Table 5.13). The ability of smaller ant populations to solve more highly constrained problems might be attributed to the occasional selection of non-best solutions after convergence, as explained previously. In addition, smaller ant populations results in a larger number of iterations, which is equivalent to a larger number of pheromone updates during an ACO run. Given more information from past searching experience (via pheromone updates), feasible regions of a search space may be better identified.

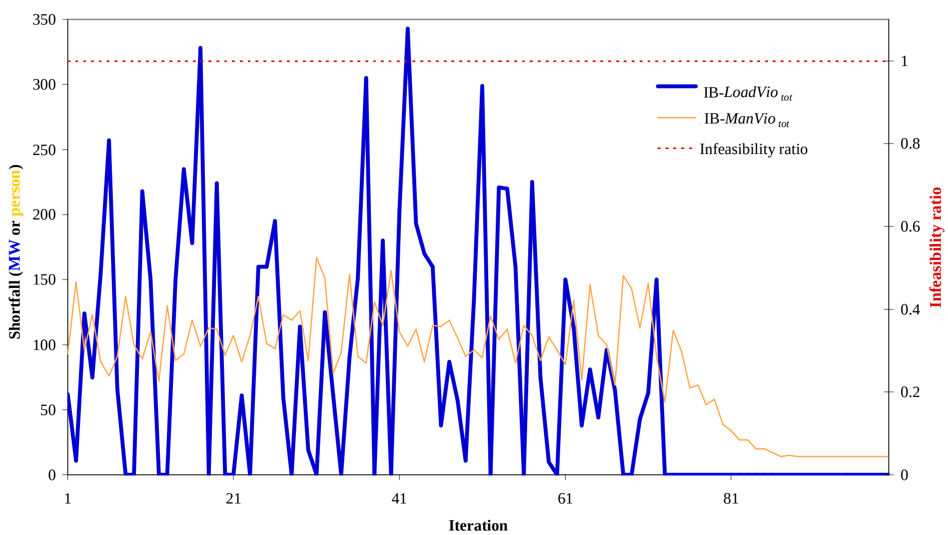
An interesting observation made from Figure 5.11d is that the average of the infeasibility ratio during iterations 200 to 600 is higher than that of previous iterations. This is corresponding well with the iteration-best total duration shortened/deferred (IB-$DurCut_{tot}$) of those stages during the ACO run (Figure 5.11b). The decrease in the total duration shortened/deferred means more maintenance tasks are performed, resulting in tighter constraints and thus a larger portion of trial solutions constructed is infeasible.

For all case studies, it can be seen that when the heuristic is used, the IB-SSR and IB-LVL obtained during the early stages of the optimisation runs were substantially lower (compare (a) and (b) of Figures 5.4 to 5.11). In addition, it can be observed that at the early stages of the ACO runs, fewer trial solutions that violated constraints were constructed when the heuristic was utilized (compare (c) and (d) of Figures 5.4 to 5.11). It was also found that the improvement in OFCs obtained when the heuristic is used for the modified 21- and 22-unit case studies is partly attributed to a significant reduction in duration shortened. This is clearly shown in the comparison between (a) and (b) of Figures 5.8 to 5.11 by the fact that the IB-DurCut$_{tot}$ curve is consistently lower throughout an ACO run when the heuristic formulation is used.

In view of the experimental results, the heuristic formulation is useful for ACO-PPMSO in three ways. Firstly, as the distribution of pheromone intensity within the search space of a problem is uniform at the beginning of an ACO run (assuming a single initial pheromone value is used), the optimisation process initially resembles a random search. During this period, the heuristic formulation can guide the algorithm to search in regions where feasible solutions are located with a higher probability. In this way, the number of infeasible solutions being constructed and rewarded with pheromone can be reduced. Secondly, even if a heuristic is not essential for constructing feasible/near feasible trial solutions (as is the case when the PPMSO problem is not highly constrained), the heuristic can assist with constructing trial solutions that consist of fewer overlapping tasks. In this way, the generation capacities throughout the planning horizon associated with trial maintenance schedules being constructed are more evenly distributed, which is one of the common objectives of PPMSO problems. Thirdly, when shortening and deferral options are allowed, use of the heuristic increases the probabilities that longer outage durations are chosen throughout an entire ACO run. This is particularly useful when shortening and deferral options are frequently chosen at random during the early stage of an ACO run.

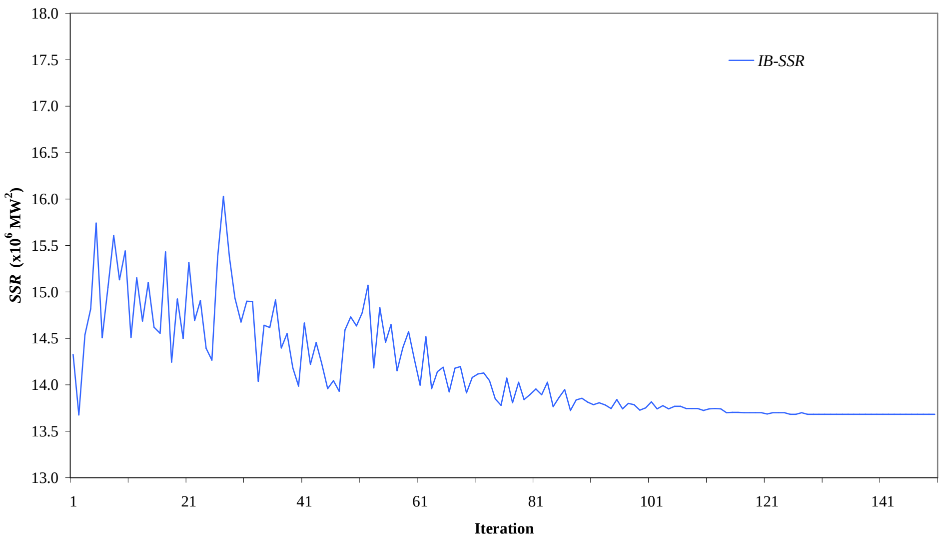


(a) Without heuristic†

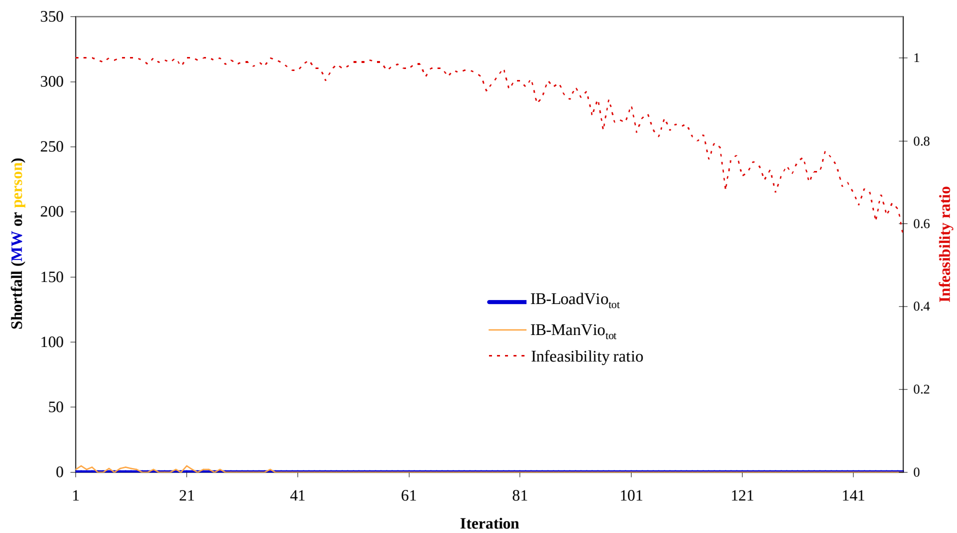


(c) Without heuristic†

† Parameter settings used shown in the first row, last column of Table 5.6 (random number seed = 888888)
†† Parameter settings used shown in the second row, last column of Table 5.6 (random number seed = 888888)
IB-SSR: Sum of squares of reserve associated with iteration-best schedules; IB-LoadVio$_{tot}$: Demand shortfalls associated with iteration-best schedules; IB-ManVio$_{tot}$: Personpower shortfalls associated with iteration-best schedules



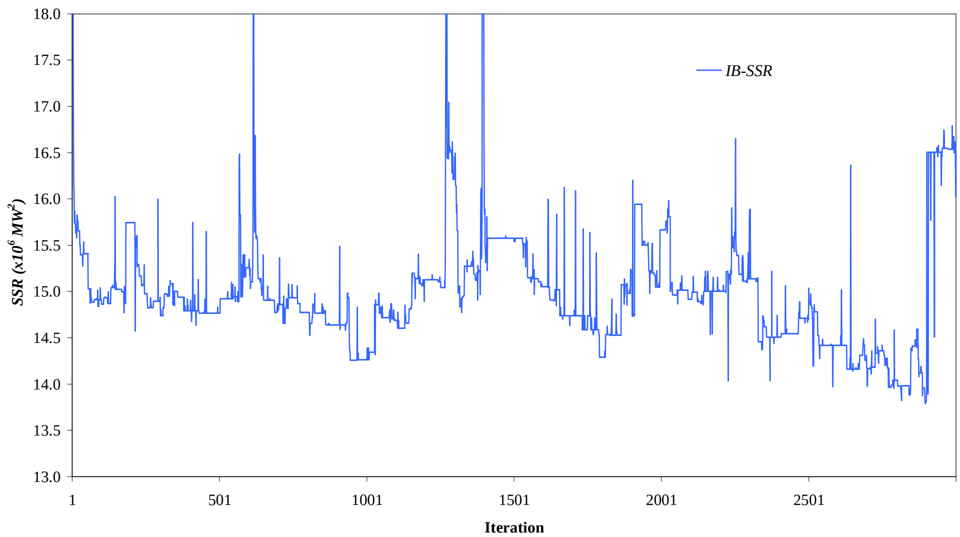
(b) With heuristic††



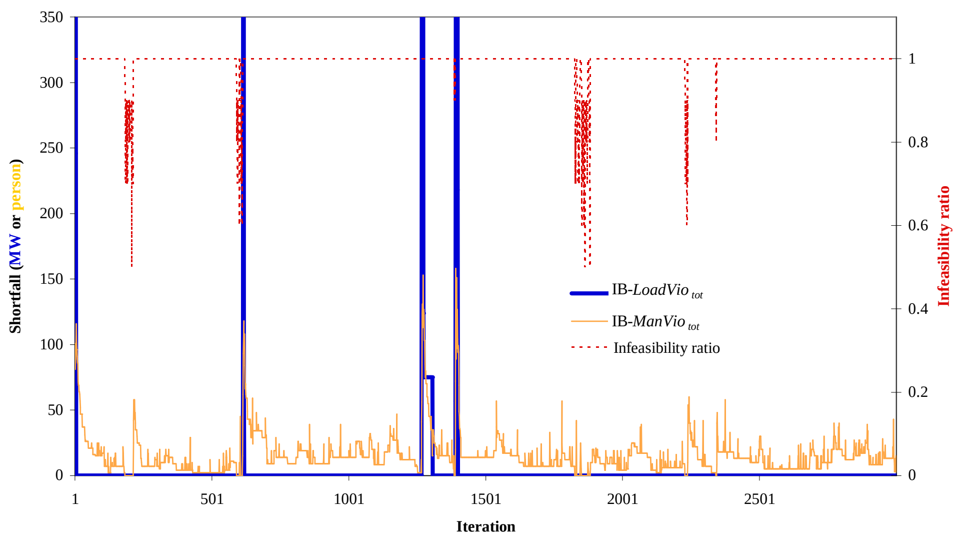
(d) With heuristic††

Figures 5.4(a) & (b): Performance of Elitist-Ant System (EAS) in solving the original 21-unit case study with and without heuristic (Comparison of the SSR-values associated with iteration-best schedules during optimisation run; Best-known SSR = 13.36 x 10^6 MW^2)

Figures 5.4(c) & (d): Performance of Elitist-Ant System (EAS) in solving the original 21-unit case study with and without heuristic (Comparison of the violation of constraints associated with iteration-best schedules and infeasibility ratio during optimisation run)

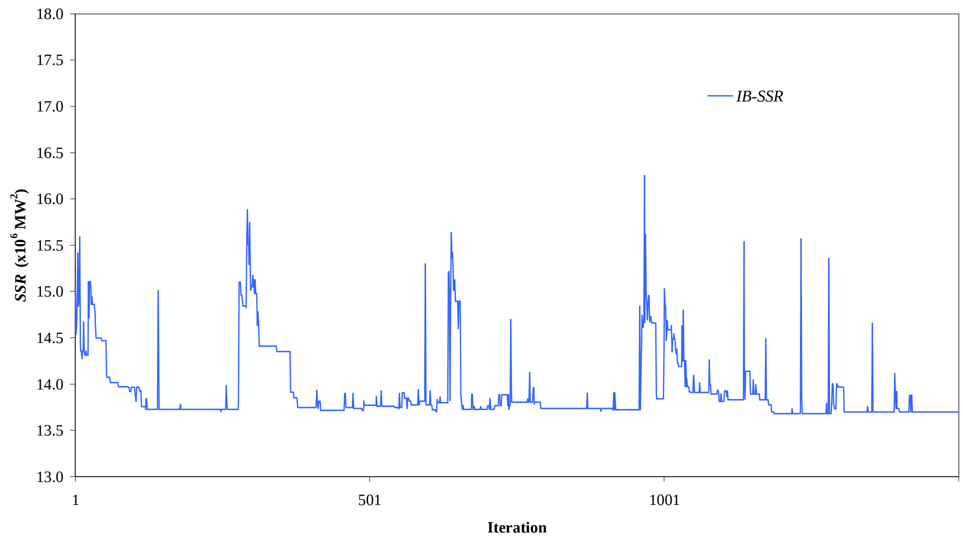


(a) Without heuristic†

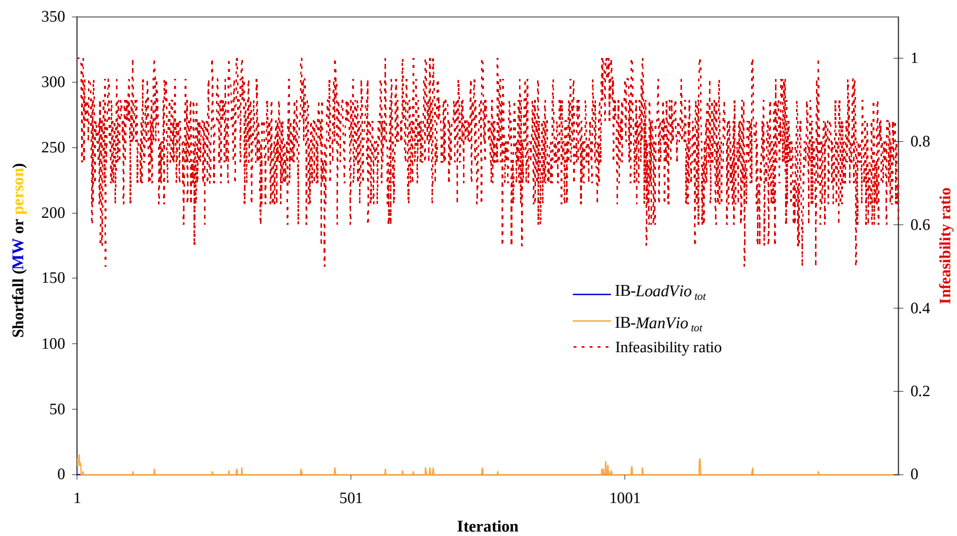


(c) Without heuristic†

† Parameter settings used shown in the first row, last column of Table 5.7 (random number seed = 356060)
†† Parameter settings used shown in the second row, last column of Table 5.7 (random number seed = 888888)
IB-SSR: Sum of squares of reserve associated with iteration-best schedules; IB-LoadVio$_{tot}$: Demand shortfalls associated with iteration-best schedules; IB-ManVio$_{tot}$: Personpower shortfalls associated with iteration-best schedules



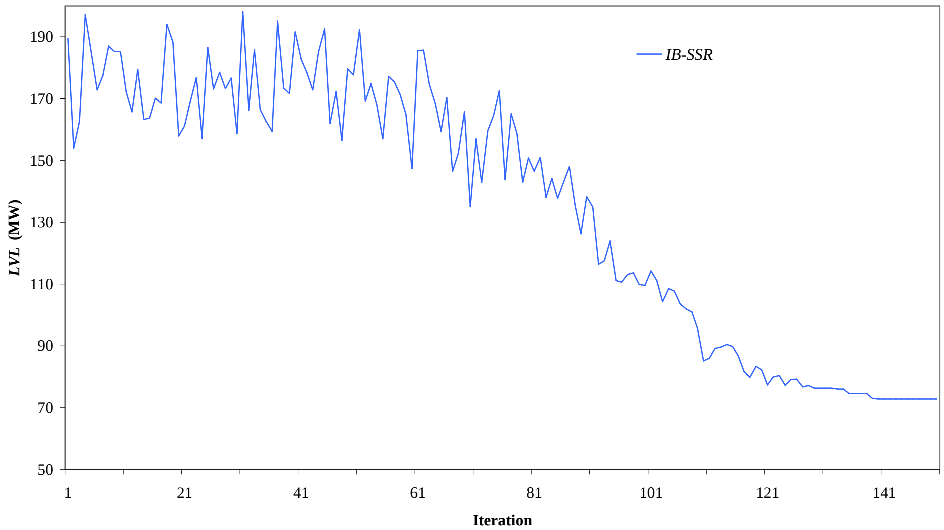
(b) With heuristic††



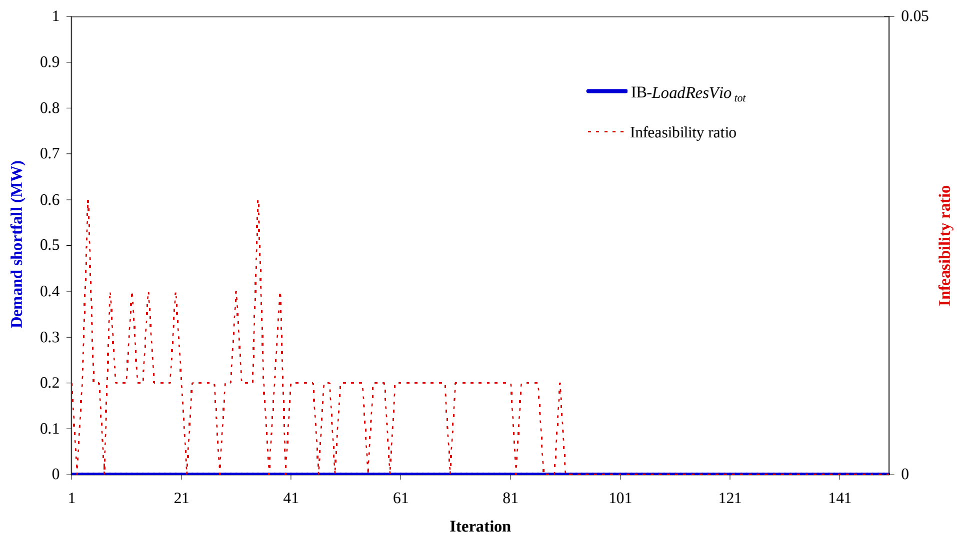
(d) With heuristic††

Figures 5.5(a) & (b): Performance of Max-Min Ant System (MMAS) in solving the original 21-unit case study with and without heuristic (Comparison of the SSR-values associated with iteration-best schedules during optimisation run; Best-known SSR = 13.36 x 10^6 MW^2).

Figures 5.5(c) & (d): Performance of Max-Min Ant System (MMAS) in solving the original 21-unit case study with and without heuristic (Comparison of the violation of constraints associated with iteration-best schedules and infeasibility ratio during optimisation run).

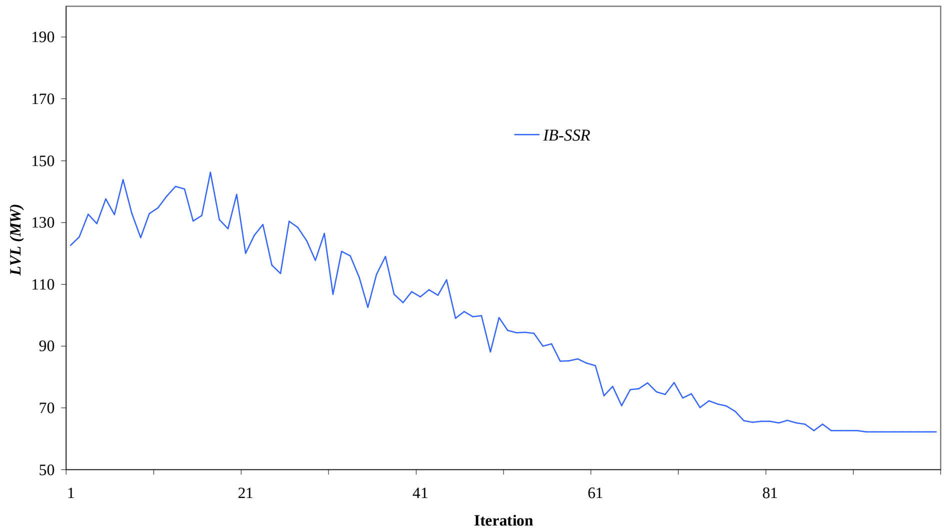


(a) Without heuristic†

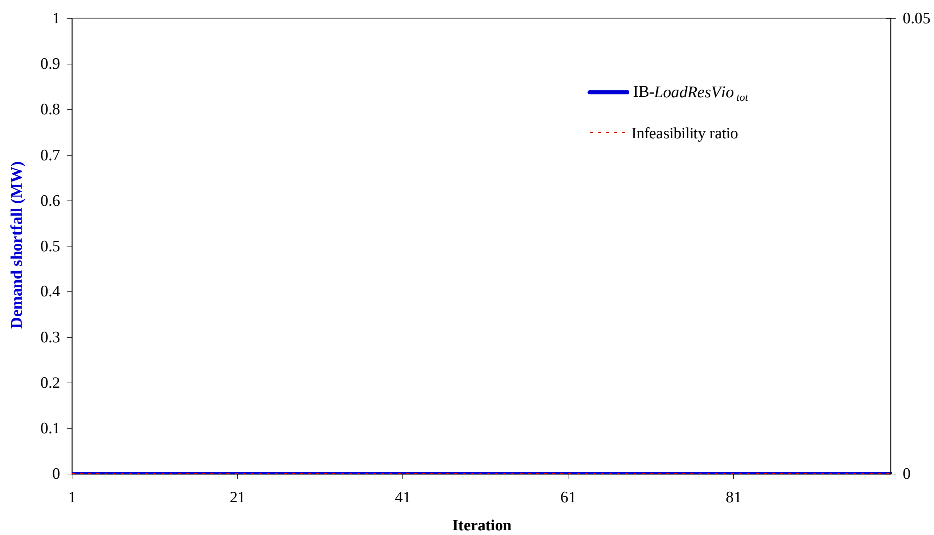


(c) Without heuristic†

† Parameter settings used shown in the first row, last column of Table 5.8 (random number seed = 888888) †† Parameter settings used shown in the second row, last column of Table 5.8 (random number seed = 888888) IB-LVL: The levels of reserve generation associated with iteration-best schedules; IB-LoadResVio$_{tot}$: Demand and reliability shortfalls associated with iteration-best schedules



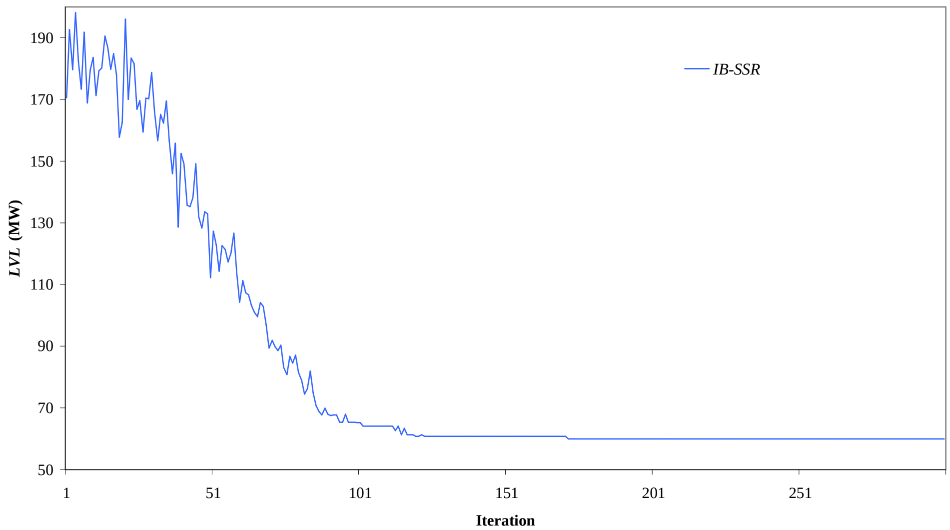
(b) With heuristic††



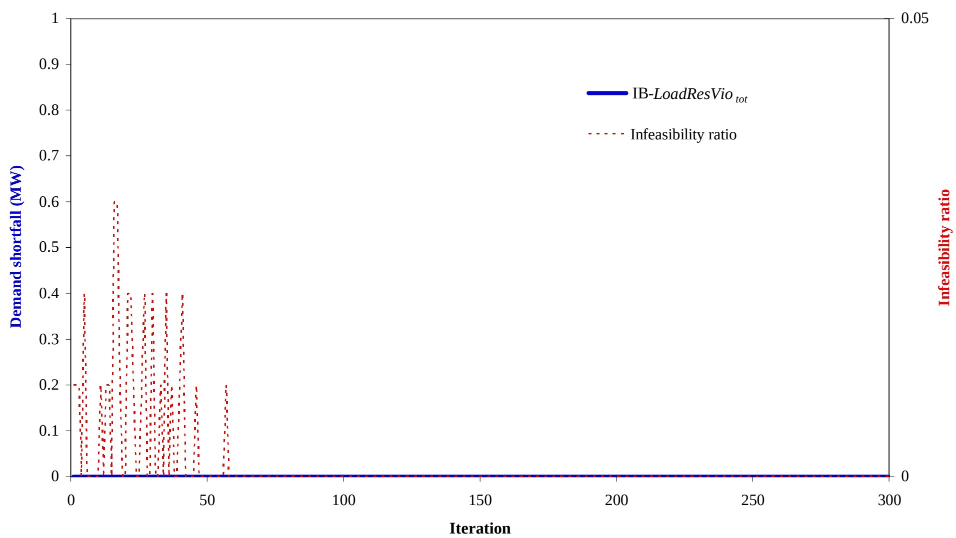
(d) With heuristic††

Figures 5.6(a) & (b): Performance of Elitist-Ant System (EAS) in solving the original 22-unit case study with and without heuristic (Comparison of the LVL-values associated with iteration-best schedules during optimisation run; Best-known LVL = 52.06 MW).

Figures 5.6(c) & (d): Performance of Elitist-Ant System (EAS) in solving the original 22-unit case study with and without heuristic (Comparison of the violation of constraints associated with iteration-best schedules and infeasibility ratio during optimisation run).

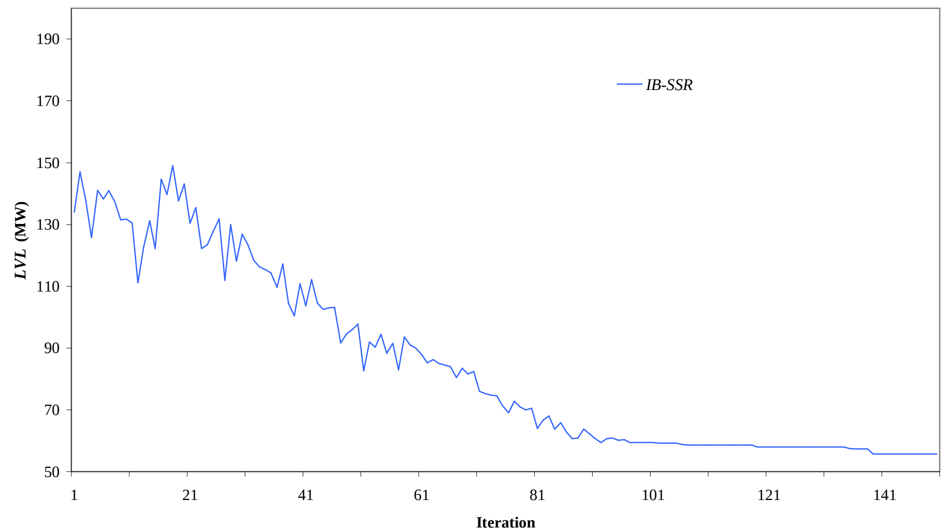


(a) Without heuristic†

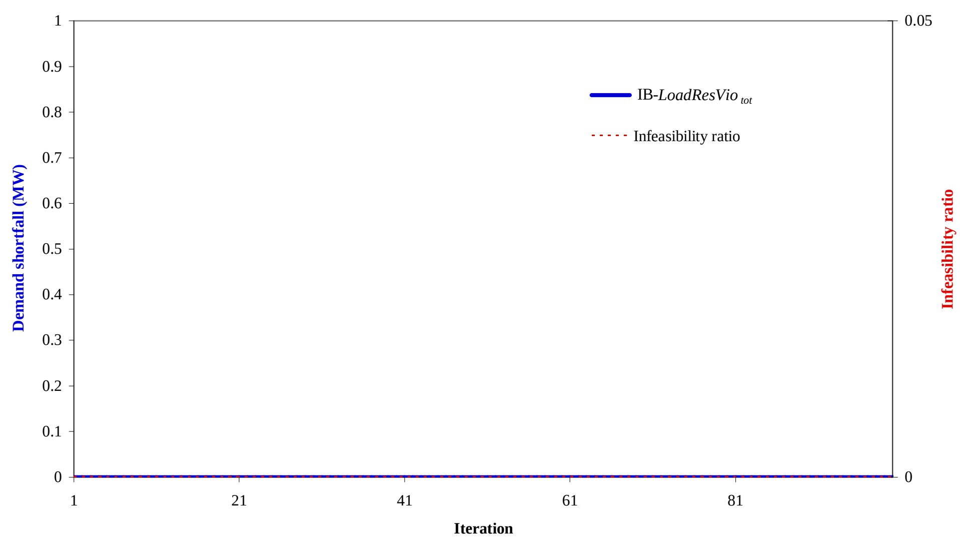


(c) Without heuristic†

† Parameter settings used shown in the first row, last column of Table 5.9 (random number seed = 955632)
†† Parameter settings used shown in the second row, last column of Table 5.9 (random number seed = 955632)
IB-LVL: The levels of reserve generation associated with iteration-best schedules; IB-LoadResVio$_{tot}$: Demand and reliability shortfalls associated with iteration-best schedules



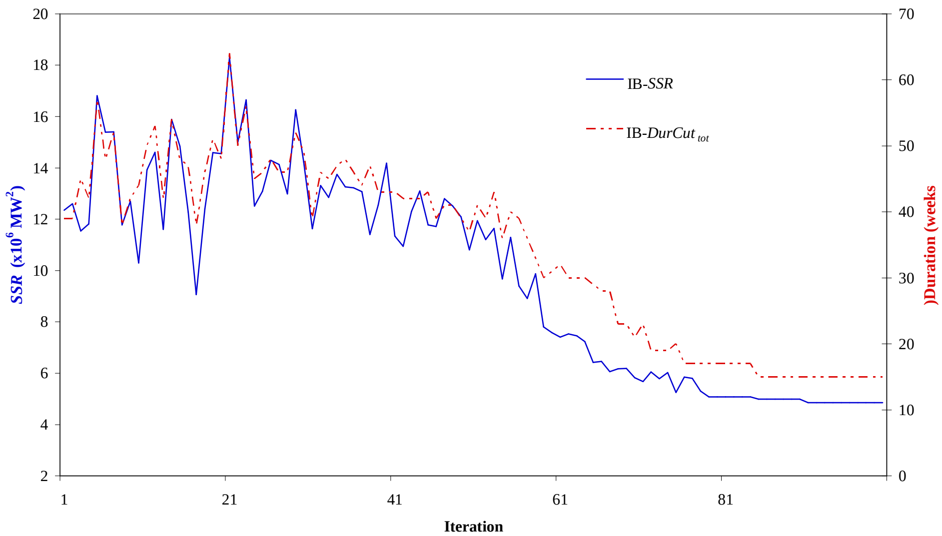
(b) With heuristic††



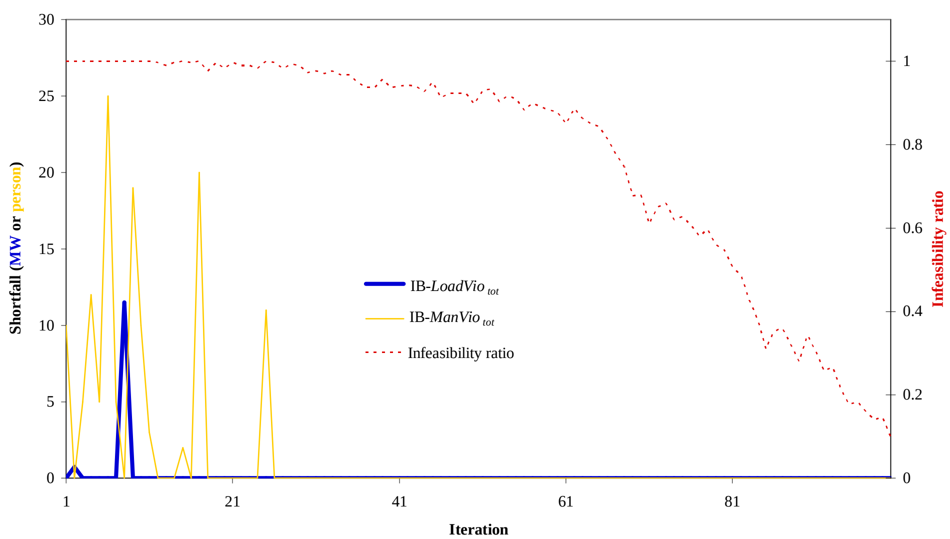
(d) With heuristic††

Figures 5.7(a) & (b): Performance of Max-Min Ant System (MMAS) in solving the original 22-unit case study with and without heuristic (Comparison of the LVL-values associated with iteration-best schedules during optimisation run; Best-known LVL = 52.06 MW).

Figures 5.7(c) & (d): Performance of Max-Min Ant System (MMAS) in solving the original 22-unit case study with and without heuristic (Comparison of the violation of constraints associated with iteration-best schedules and infeasibility ratio during optimisation run).

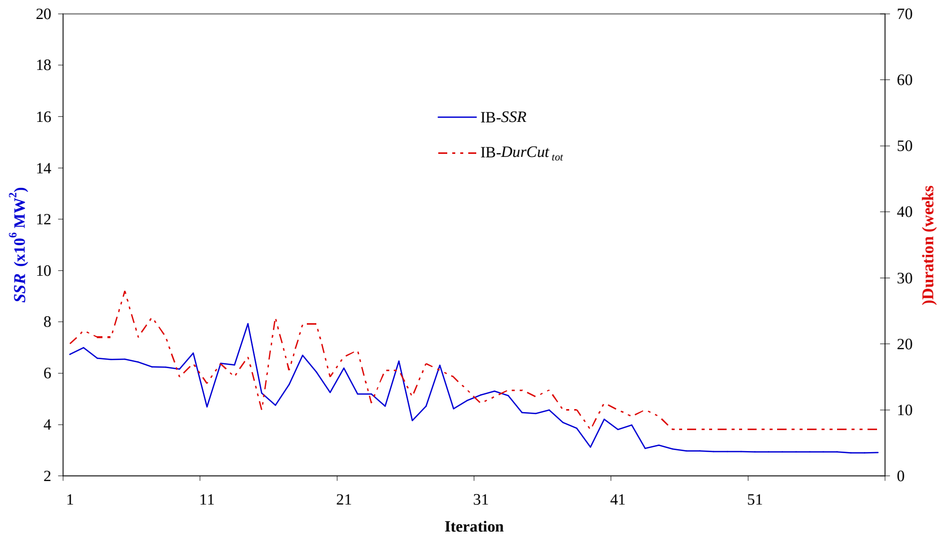


(a) Without heuristic†

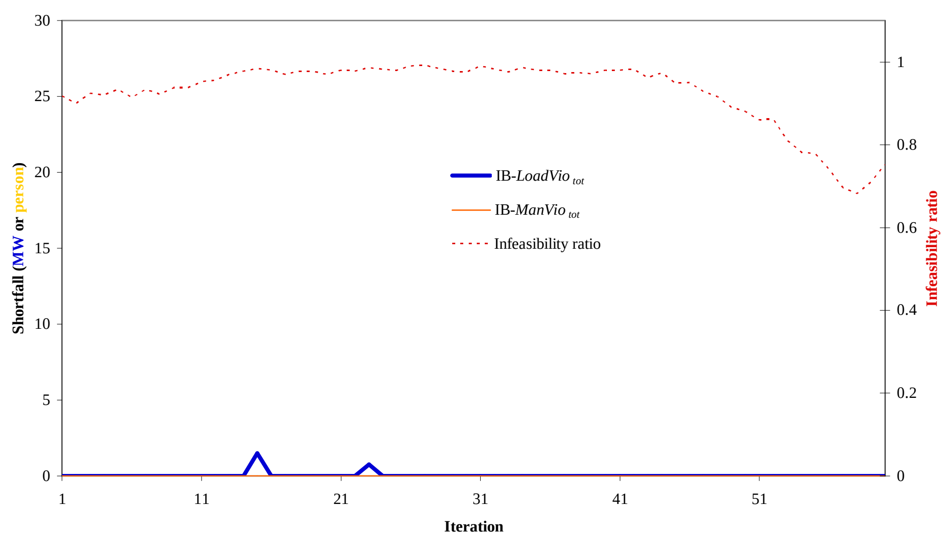


(c) Without heuristic†

† Parameter settings used shown in the first row, last column of Table 5.10 (random number seed = 888888)
†† Parameter settings used shown in the second row, last column of Table 5.10 (random number seed = 888888)
IB-SSR: Sum of squares of reserve associated with iteration-best schedules; IB-DurCut$_{tot}$: Total reduction in outage duration due to shortening and deferral associated with iteration-best schedules; IB-LoadVio$_{tot}$: Demand shortfalls associated with iteration-best schedules; IB-ManVio$_{tot}$: Personpower shortfalls associated with iteration-best schedules



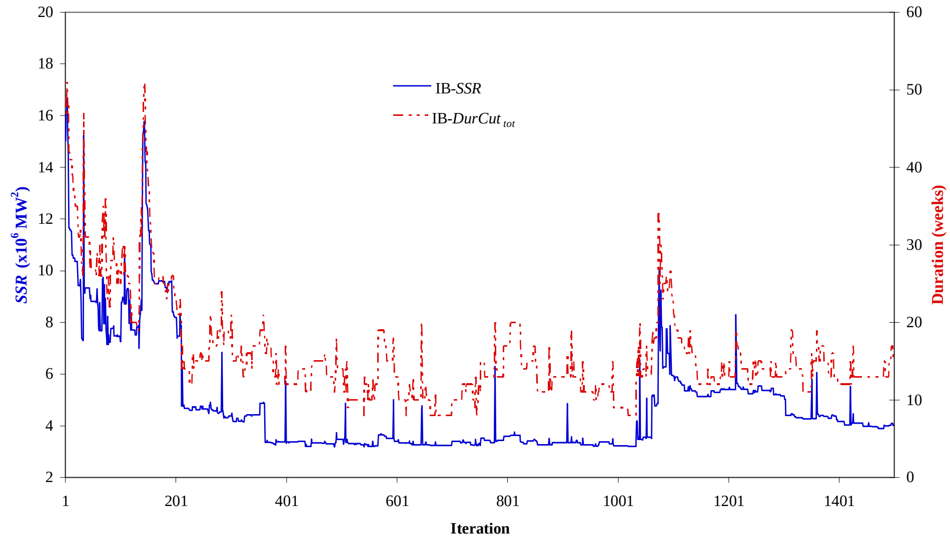
(b) With heuristic††



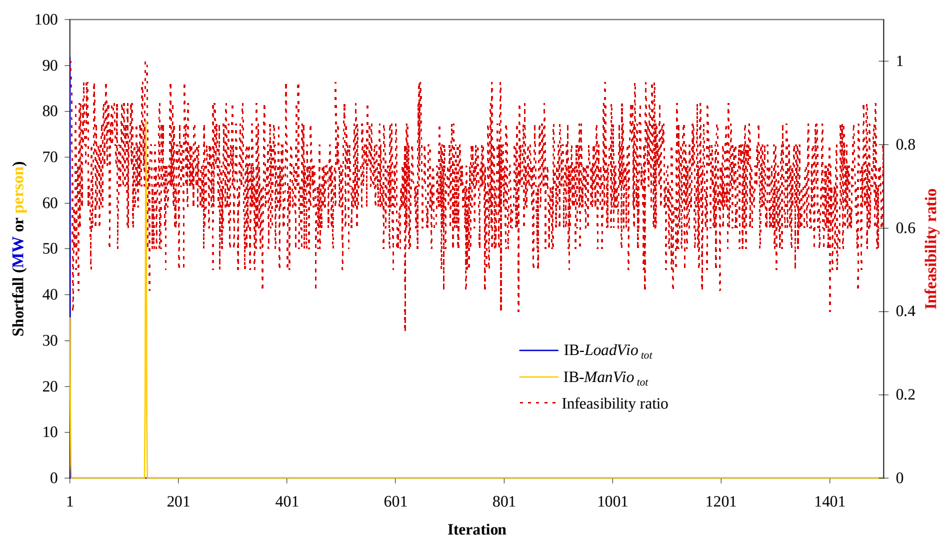
(d) With heuristic††

Figures 5.8(a) & (b): Performance of Elitist-Ant System (EAS) in solving the modified 21-unit case study with and without heuristic (Comparison of the SSR- and total duration shortened values associated with iteration-best schedules during optimisation run; Best-known SSR = 2.62 x 106 MW² with 5-week deferral).

Figures 5.8(c) & (d): Performance of Elitist-Ant System (EAS) in solving the modified 21-unit case study with and without heuristic (Comparison of the violation of constraints associated with iteration-best schedules and infeasibility ratio during optimisation run).

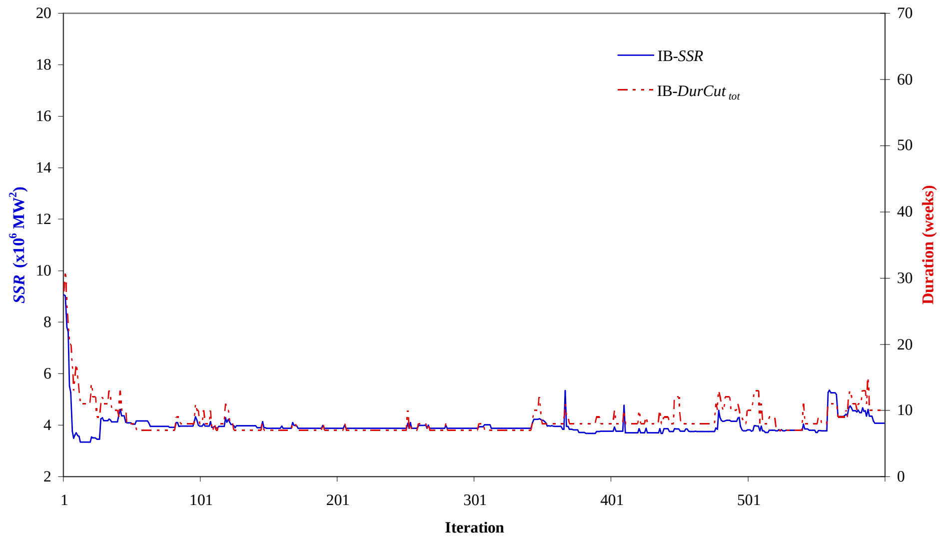


(a) Without heuristic†

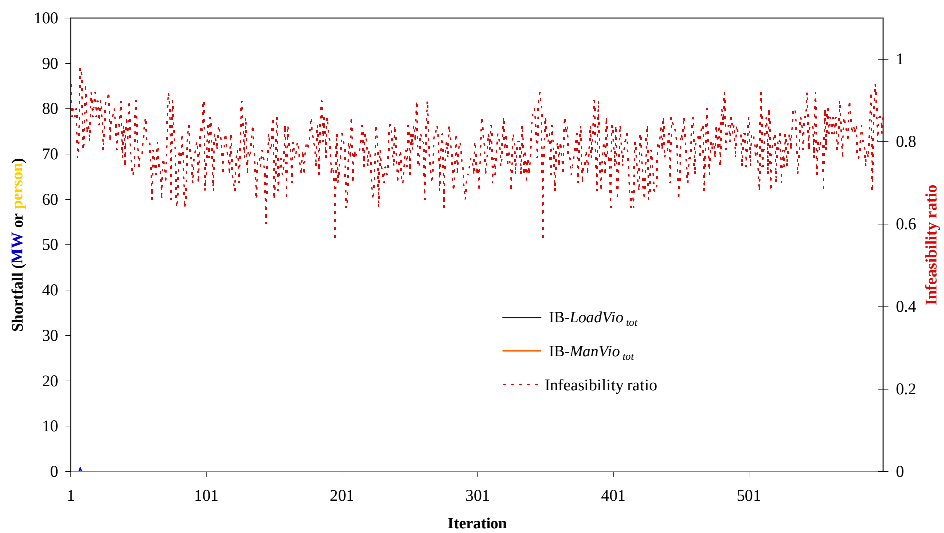


(c) Without heuristic†

† Parameter settings used shown in the first row, last column of Table 5.11 (random number seed = 655)
†† Parameter settings used shown in the second row, last column of Table 5.11 (random number seed = 655)
IB-SSR: Sum of squares of reserve associated with iteration-best schedules; IB-DurCut$_{tot}$: Total reduction in outage duration due to shortening and deferral associated with iteration-best schedules; IB-LoadVio$_{tot}$: Demand shortfalls associated with iteration-best schedules; IB-ManVio$_{tot}$: Personpower shortfalls associated with iteration-best schedules



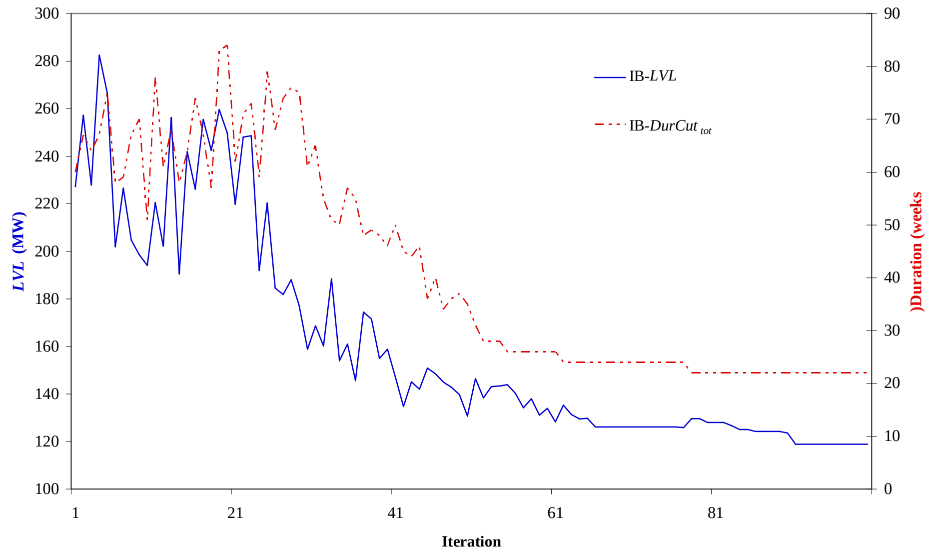
(b) With heuristic††



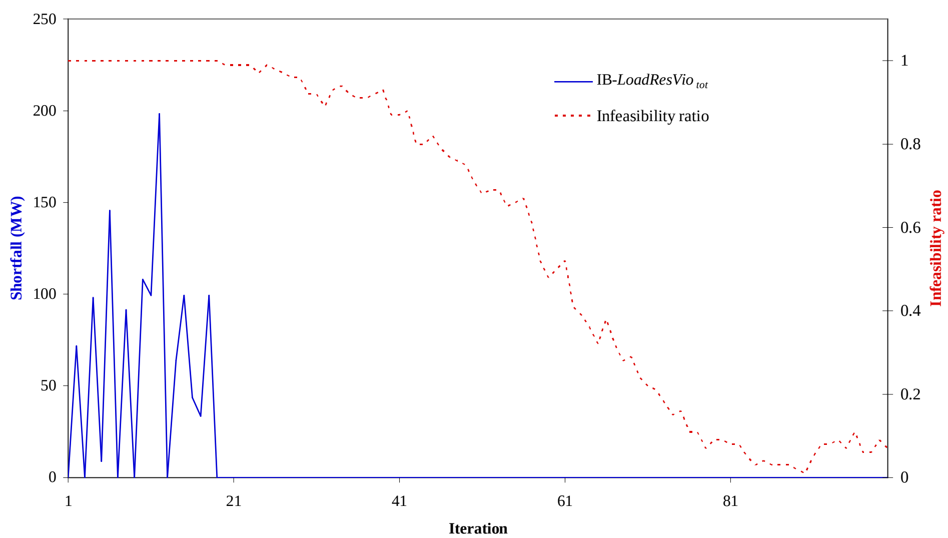
(d) With heuristic††

Figures 5.9(a) & (b): Performance of Max-Min Ant System (MMAS) in solving the modified 21-unit case study with and without heuristic (Comparison of the SSR- and total duration shortened values associated with iteration-best schedules during optimisation run; Best-known SSR = 2.62 x 106 MW² with 5-week deferral).

Figures 5.9(c) & (d): Performance of Max-Min Ant System (MMAS) in solving the modified 21-unit case study with and without heuristic (Comparison of the violation of constraints associated with iteration-best schedules and infeasibility ratio during optimisation run).



(a) Without heuristic†

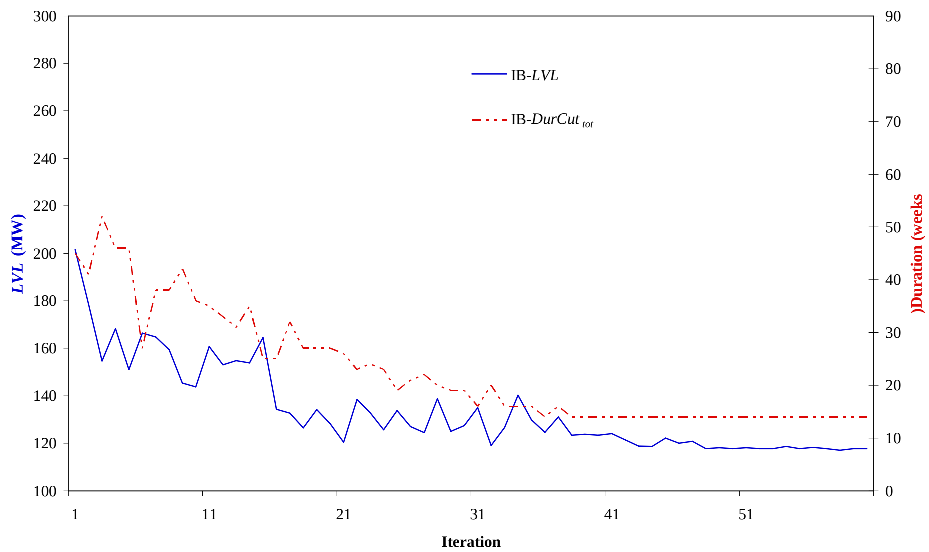


(c) Without heuristic†

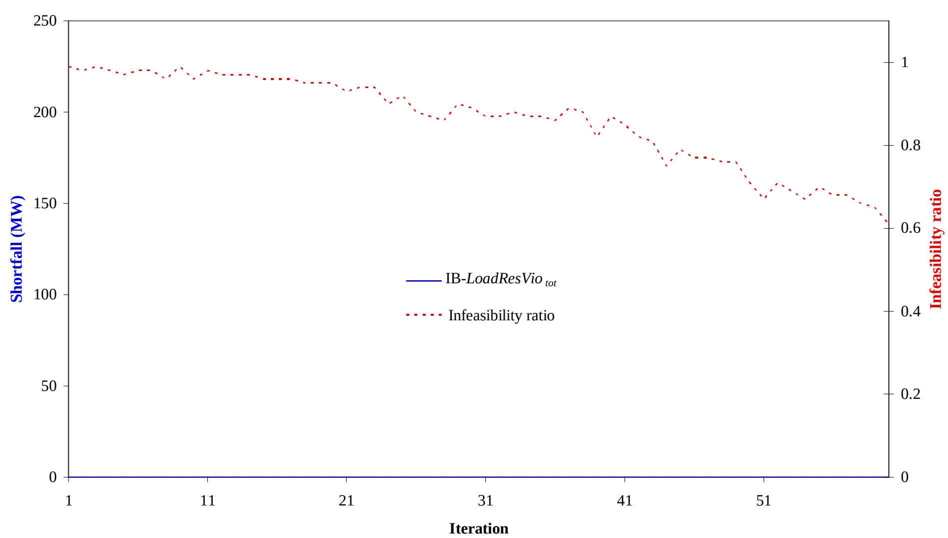
† Parameter settings used shown in the first row, last column of Table 5.12 (random number seed = 888888)

†† Parameter settings used shown in the second row, last column of Table 5.12 (random number seed = 888888)

IB-LVL: The levels of reserve generation associated with iteration-best schedules; IB-$DurCut_{tot}$: Total reduction in outage duration due to shortening and deferral associated with iteration-best schedules; IB-$LoadResVio_{tot}$: Demand and reliability shortfalls associated with iteration-best schedules



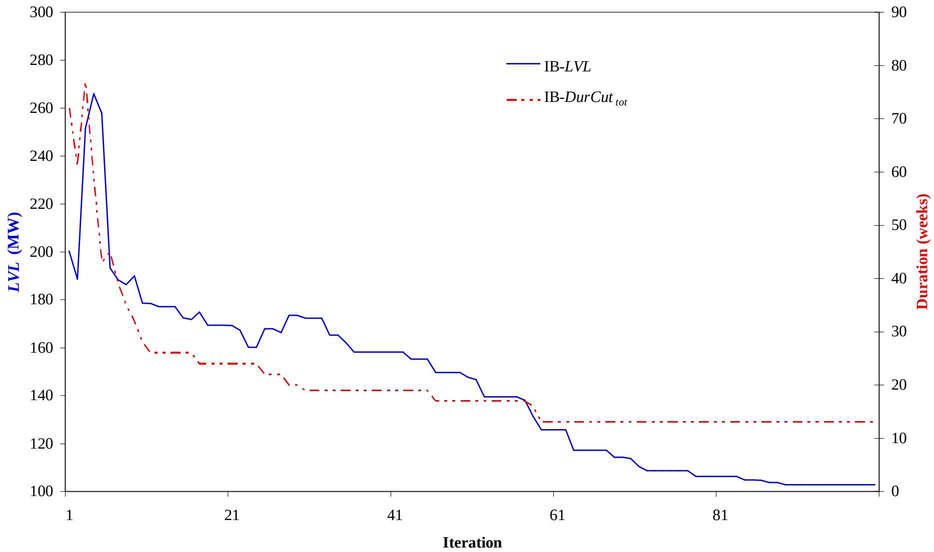
(b) With heuristic††



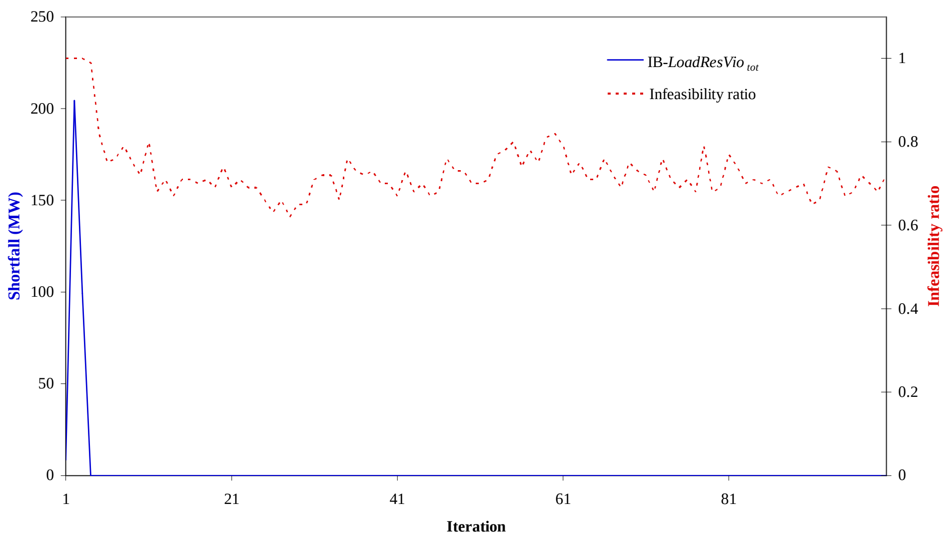
(d) With heuristic††

Figures 5.10(a) & (b): Performance of Elitist-Ant System (EAS) in solving the modified 22-unit case study with and without heuristic (Comparison of the LVL- and total duration shortened values associated with iteration-best schedules during optimisation run; Best-known LVL = 101.791 MW with 8-week shortening).

Figures 5.10(c) & (d): Performance of Elitist-Ant System (EAS) in solving the modified 22-unit case study with and without heuristic (Comparison of the violation of constraints associated with iteration-best schedules and infeasibility ratio during optimisation run).

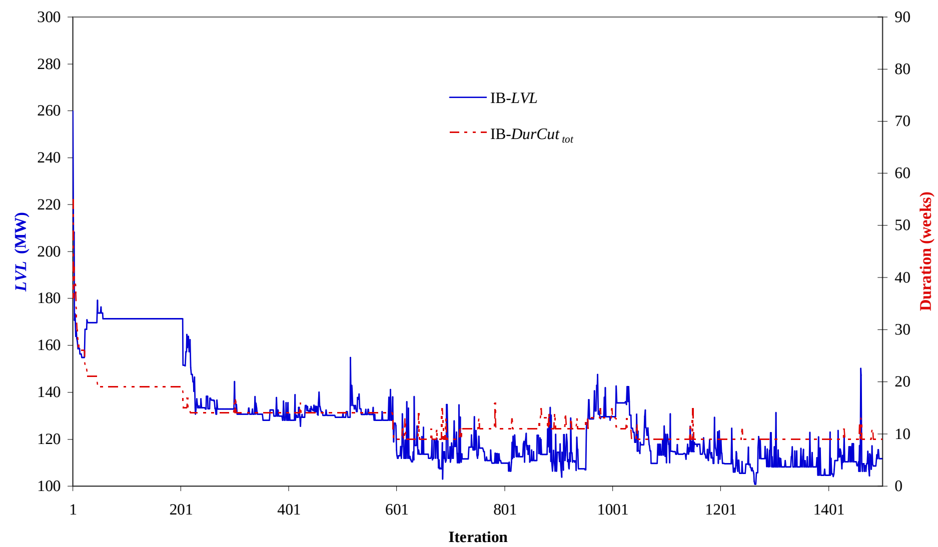


(a) Without heuristic†

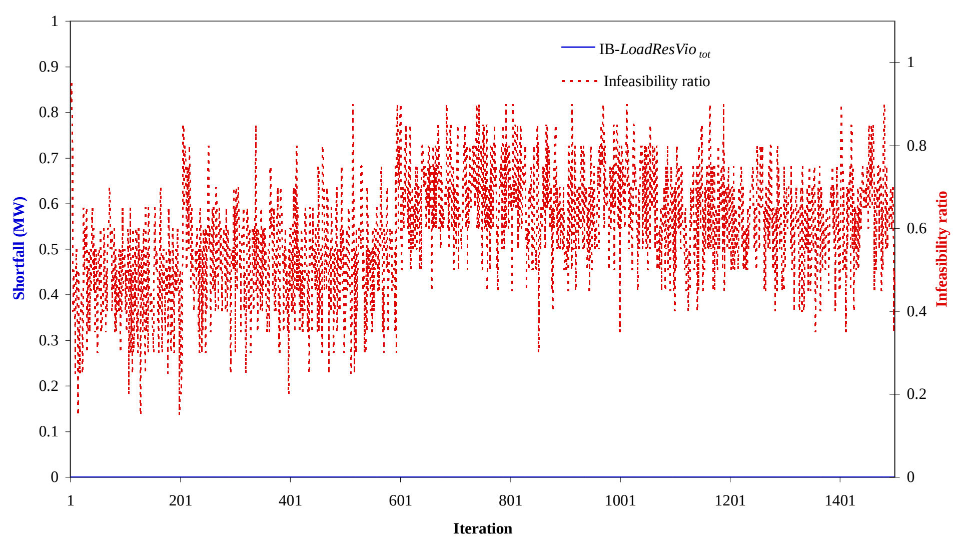


(c) Without heuristic†

† Parameter settings used shown in the first row, last column of Table 5.13 (random number seed = 36565)
†† Parameter settings used shown in the second row, last column of Table 5.13 (random number seed = 33552454)
IB-LVL: The levels of reserve generation associated with iteration-best schedules; IB-DurCut$_{tot}$: Total reduction in outage duration due to shortening and deferral associated with iteration-best schedules; IB-LoadResVio$_{tot}$: Demand and reliability shortfalls associated with iteration-best schedules



(b) With heuristic††



(d) With heuristic††

Figures 5.11(a) & (b): Performance of Max-Min Ant System (MMAS) in solving the modified 22-unit case study with and without heuristic (Comparison of the LVL- and total duration shortened values associated with iteration-best schedules during optimisation run; Best-known LVL = 101.791 MW with 8-week shortening).

Figures 5.11(c) & (d): Performance of Max-Min Ant System (MMAS) in solving the modified 22-unit case study with and without heuristic (Comparison of the violation of constraints associated with iteration-best schedules and infeasibility ratio during optimisation run).

In relation to the two ACO algorithms investigated (EAS and MMAS), the results obtained indicate that the heuristic has a significant positive impact on both EAS and MMAS. This is probably due to the ability of heuristic information to identify regions of the search space where high-quality initial solutions lie, reducing the number of low-quality trial solutions being reinforced at the beginning of an optimisation run. In addition, the results indicate that the ACO-PPMSO heuristic has a bigger positive impact on EAS compared to MMAS. EAS tends to stagnate after a number of iterations, which increases the impact of the quality of the initial solutions. The importance of the regions where the ants initially search using EAS is also highlighted by the relatively larger number of ants found for the best parameter settings than those for MMAS (Tables 5.6, 5.8, 5.10 and 5.12), implying that a search with more ants in each iteration (resulting a smaller number of iterations during an optimisation run, as the total number of function evaluations is fixed) works better than one with fewer ants (resulting a larger number of iterations during an optimisation run, as the total number of function evaluations is fixed). On the other hand, relatively smaller ant populations are found to perform best for MMAS (Tables 5.7, 5.8, 5.11 and 5.13), which might be attributed to the continuous exploration during an MMAS run (Figures 5.5, 5.7, 5.9 and 5.11) as a result of the lower and upper bound for pheromone values. It is interesting to observe that despite the expected overall downward trends throughout an optimisation run, the IB-*SSR* and IB-*LVL* curves spike occasionally throughout a run when a small population of ants is used (Figures 5.5, 5.9 and 5.11b). This phenomenon is found to be caused by the choice of non-best solutions after a short convergence (stagnation in OFC), which altered the distribution of pheromone over the problem search space. It should be noted that the possibility of having an iteration-best solution that is not the best-so-far solution is higher when a smaller population of ants is used.

B. Impact of local search

The optimisation results obtained by coupling two different local search operators, namely the *PPMSO-2-opt* (Section 4.3.4) and *Duration Extender* (Section 4.3.4), with the ACO algorithms investigated (Stage B of the testing procedure in Figure 5.3) are tabulated in Tables 5.5 to 5.13. The unpaired Student's *t*-test (calculations are shown in Appendices B2 and C2) was used to check the significance of the impact of the two local search operators in solving the four case studies with and without heuristic (Tables 5.15 and 5.16). It should be noted that while *PPMSO-2-opt* was applied to all case studies, *Duration Extender* was only tested with the modified case studies for reasons given previously.

Table 5.15: Impact of *PPMSO-2-opt* local search operator with and without heuristic

	21-unit system		*22-unit system*		*Modified 21-unit system*		*Modified 22-unit system*	
Heuristic	**EAS**	**MMAS**	**EAS**	**MMAS**	**EAS**	**MMAS**	**EAS**	**MMAS**
✗	+	NIL	+	+	−	−	NIL	NIL
✓	NIL	NIL	+	+	−	−	NIL	NIL

Notation:
+: Significant positive impact; −: Significant negative impact; NIL: Insignificant impact.

Table 5.16: Impact of *Duration Extender* local search operator with and without heuristic

	Modified 21-unit system		*Modified 22-unit system*	
Heuristic	**EAS**	**MMAS**	**EAS**	**MMAS**
✗	+	NIL	+	NIL
✓	NIL	NIL	+	NIL

Notation:
+: Significant positive impact; −: Significant negative impact; NIL: Insignificant impact.

<u>PPMSO-2-opt</u>

Overall, the impact of the local search *PPMSO-2-opt* operator ranges from being insignificant, to significantly improving or degrading the performance of the ACO algorithm investigated. While having a positive impact on solving the original 22-unit case study regardless of which of the two ACO algorithms was used, the *PPMSO-2-opt* local search operator was found to improve only the performance of EAS when the heuristic was not used for solving the original 21-unit case study. As for the modified case studies, the performance of ACO in solving the modified 21-unit case study was reduced significantly when the *PPMSO-2-opt* local search operator was adopted, while the impact of the local search was not significant when applied to the modified 22-unit case study.

From the results of the Stage B testing, it is interesting to observe that despite the similarity in the number of generating units for the 21- and 22-unit case study systems, the impact of the *PPMSO-2-opt* local search algorithm on the optimisation results of these case studies was quite different, which is likely to be caused by the difference in the problem characteristics of the two systems.

In order to better understand the results obtained, a series of tests was carried out to investigate the mechanism of *PPMSO-2-opt* in detail. The satisfaction of constraints associated with iteration-best solutions (target solutions) used for the local search operation and the % of infeasible local solutions generated when using EAS and MMAS were examined and are plotted in Figures 5.12 to 5.19. It should be noted that the results were obtained using the proposed heuristic formulation. It can be seen that for the original 21-unit case study (Figures 5.12 and 5.13) and the modified version of both the 21- and 22-unit case studies (Figures 5.16 to 5.19), a large number of infeasible local solutions were generated by *PPMSO-2-opt* in every iteration, even with feasible iteration-best solutions (target maintenance schedules). As discussed previously, these three case studies are highly constrained. A local solution generated by simply exchanging the maintenance start time of two randomly chosen generating units without any guidelines is likely to result in infeasible solutions in such a highly constrained search space. As a result, *PPMSO-2-opt* seems to have an insignificant or even detrimental impact when coupled with ACO for solving the aforementioned case studies. This is particularly evident for the modified 21-unit case study, where as many as 50% to 80% of the local solutions generated by *PPMSO-2-opt* in every iteration are infeasible with regard to both load and personpower constraints, which is responsible for the significant decrease in ACO performance. These results suggest that the *PPMSO-2-opt* local search operator is not well suited to problems with highly constrained search spaces.

On the other hand, it can be seen that the local solutions generated by *PPMSO-2-opt* in solving the original 22-unit case study are all feasible, as the iteration-best solutions are also feasible (Figures 5.14 and 5.15). In fact, this is the only case study for which *PPMSO-2-opt* has been found to be effective in improving the optimisation ability of ACO. As discussed previously, the constraints of the original 22-unit case study are easily satisfied. Therefore, the results obtained indicate that *PPMSO-2-opt* can be useful for solving problems that are not highly constrained.

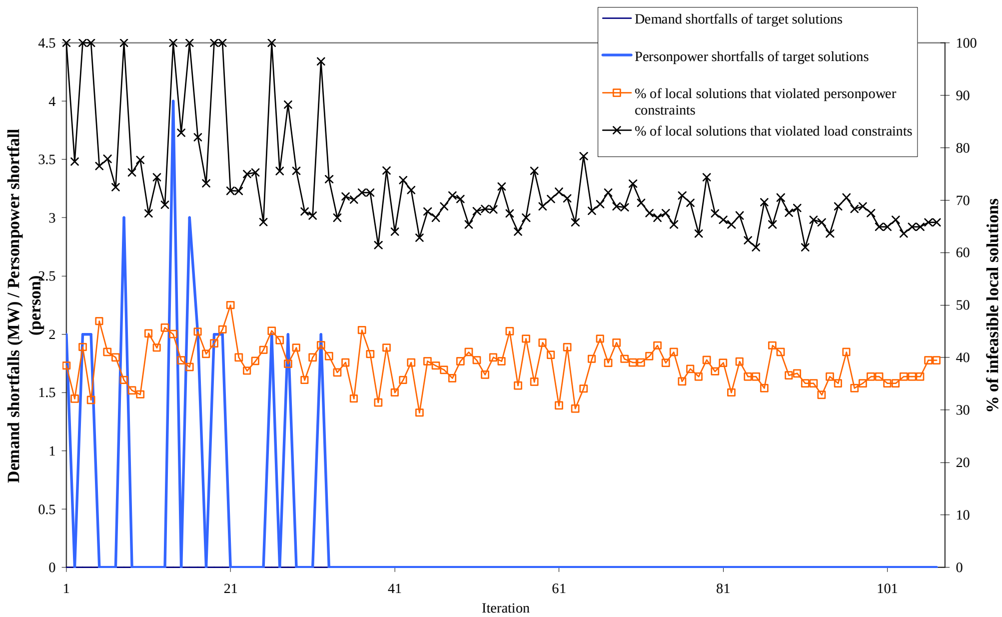


Figure 5.12: Infeasible local solutions obtained using *PPMSO-2-opt* (original 21-unit case study using EAS)

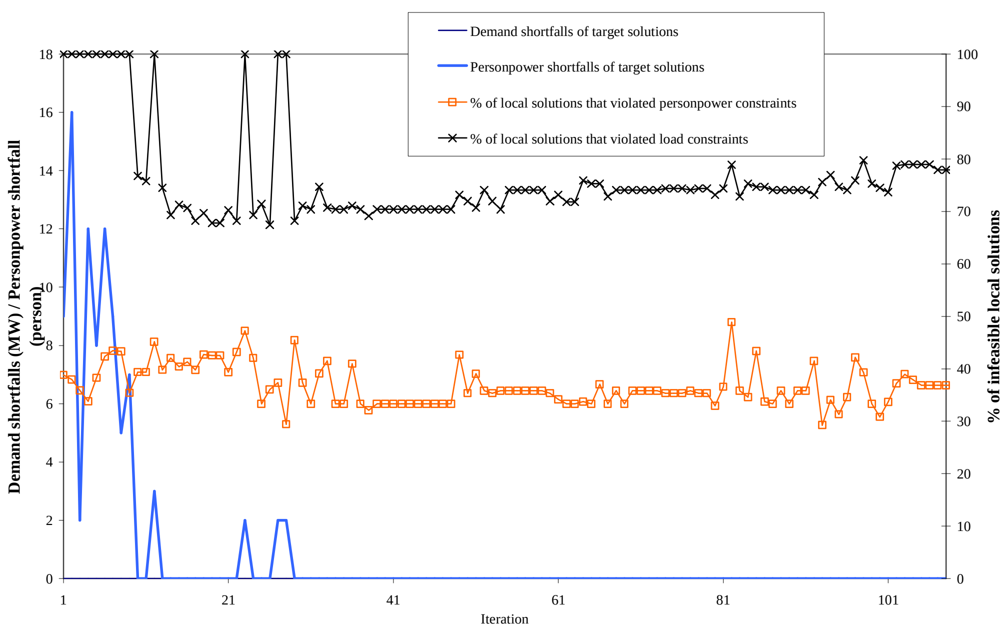


Figure 5.13: Infeasible local solutions obtained using *PPMSO-2-opt* (original 21-unit case study using MMAS)

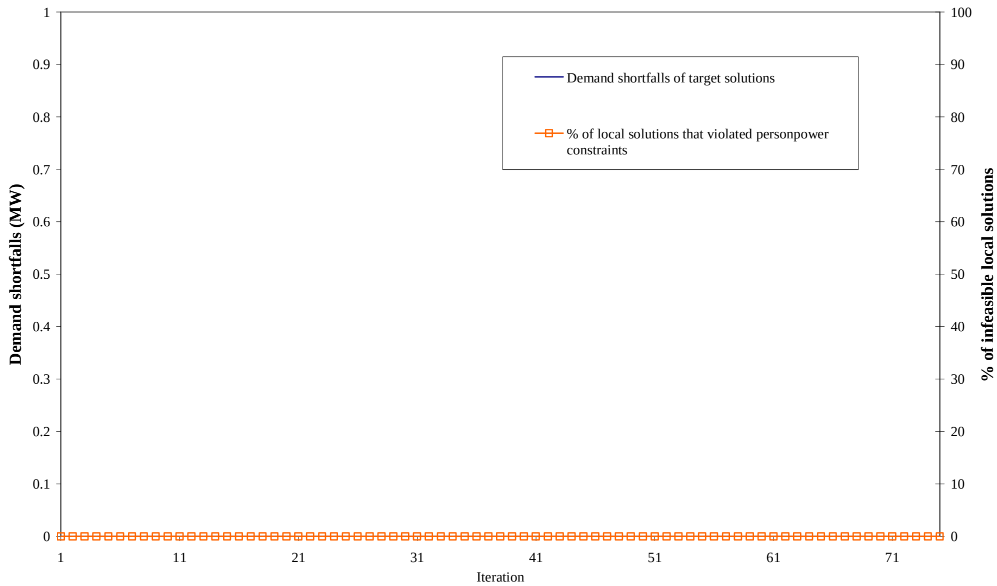


Figure 5.14: Infeasible local solutions using *PPMSO-2-opt* (original 22-unit case study using EAS)

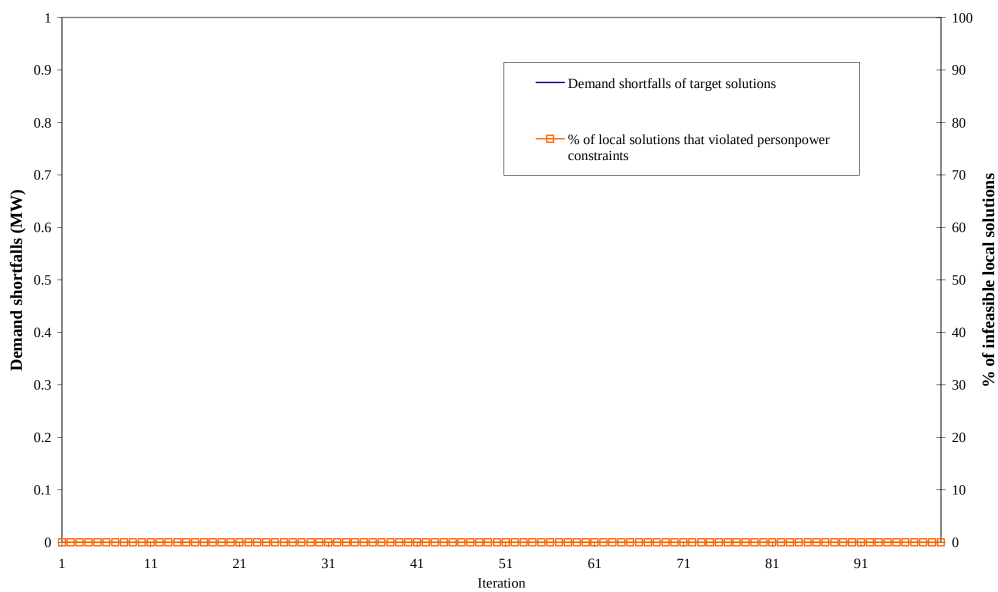


Figure 5.15: Infeasible local solutions using *PPMSO-2-opt* (original 22-unit case study using MMAS)

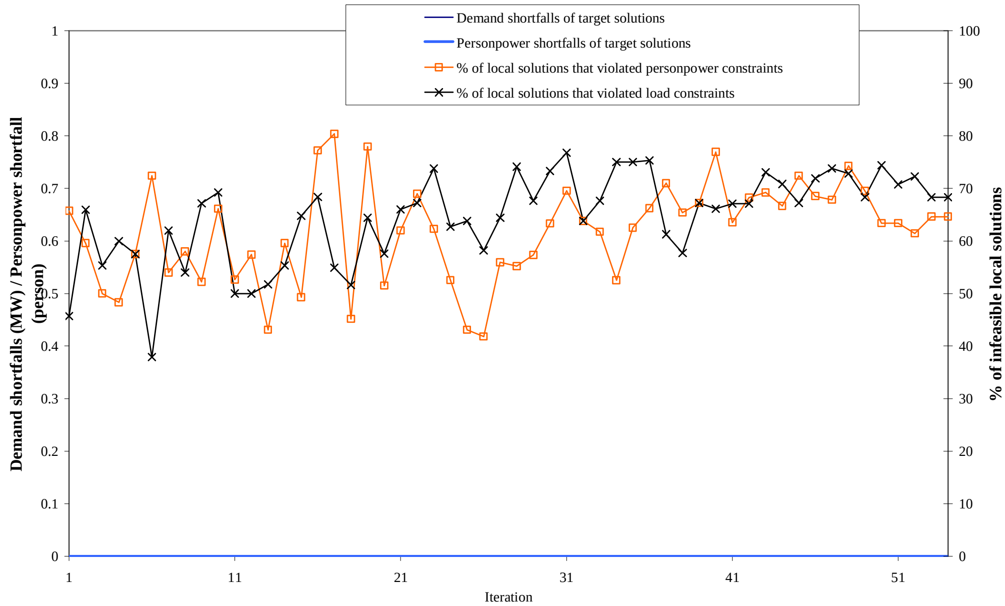


Figure 5.16: Infeasible local solutions using *PPMSO-2-opt* (modified 21-unit case study using EAS)

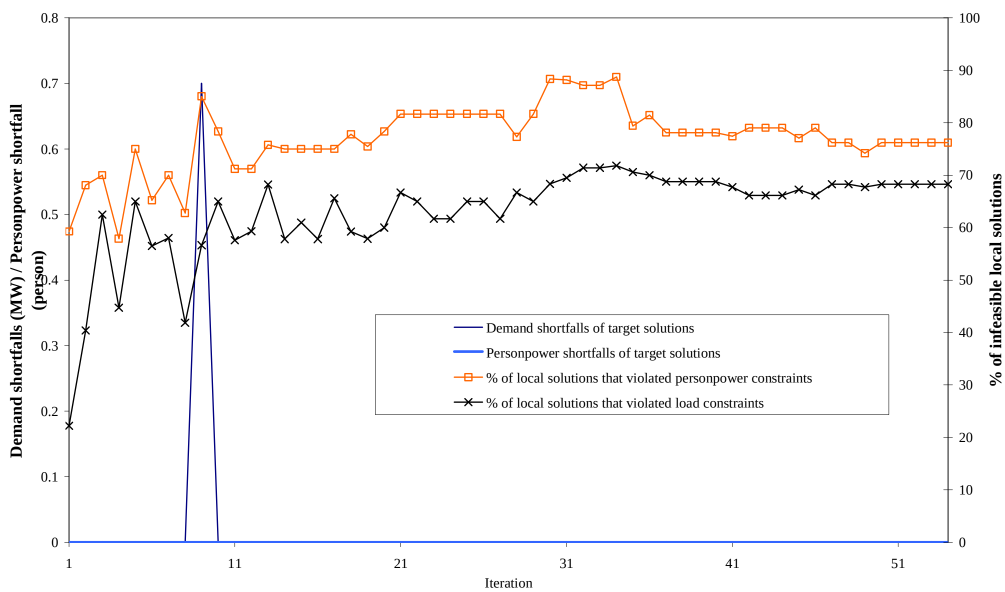


Figure 5.17: Infeasible local solutions using *PPMSO-2-opt* (modified 21-unit case study using MMAS)

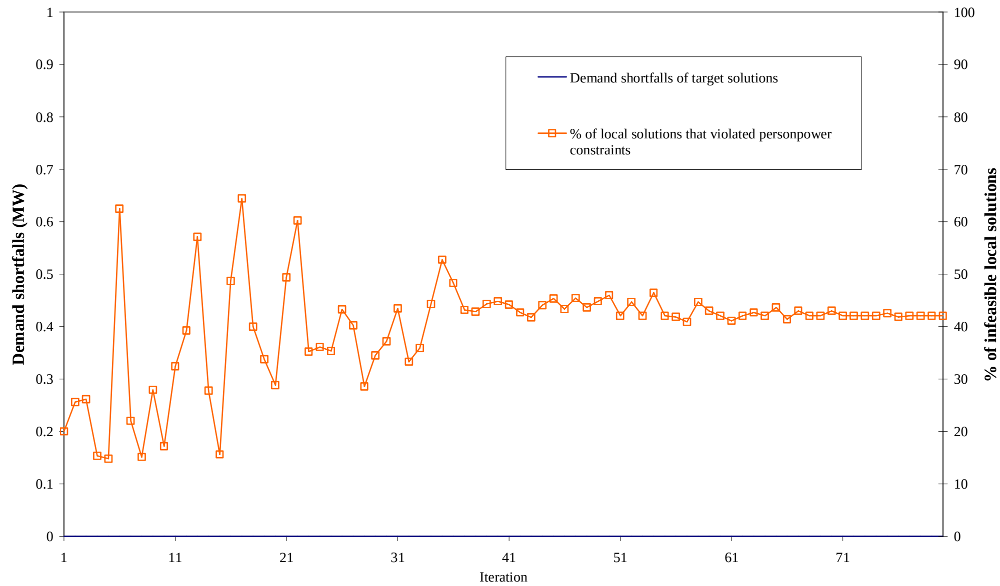


Figure 5.18: Infeasible local solutions using *PPMSO-2-opt* (modified 22-unit case study using EAS)

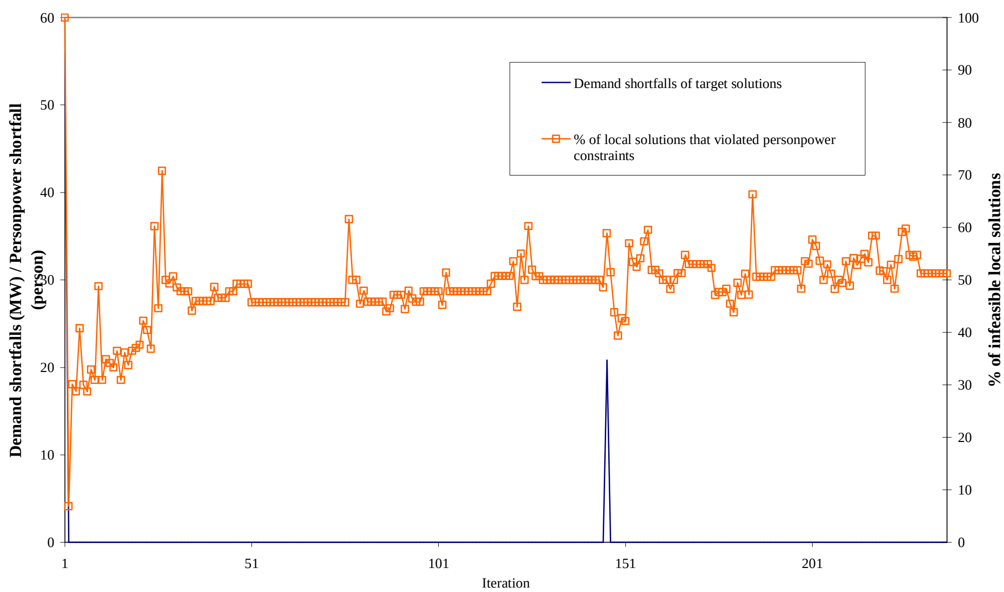


Figure 5.19: Infeasible local solutions using *PPMSO-2-opt* (modified 22-unit case study using MMAS)

Duration Extender

The *Duration Extender* local search operator, which is only applicable to PPMSO problems for which duration shortening or deferral options are available, was found to improve the performance of EAS in solving the modified 21-unit case study when the heuristic is not used, and also to produce significantly better results with and without heuristic when applied to the modified 22-unit case study (Table 5.16). On the other hand, the local search operator has an insignificant impact when the MMAS algorithm is used to solve both modified case studies (Table 5.16). The difference in the impacts the *Duration Extender* local search operator has on the performance of EAS and MMAS may be attributed to the different searching mechanisms involved in the algorithms. As MMAS is equipped with a robust explorative mechanism, it exhibits relatively stronger optimisation ability than EAS, thus the improvement of results using local search is less or insignificant.

Without the presence of the heuristic, it was observed that the average duration shortened or deferred decreased when the *Duration Extender* local search operator was used for both case studies and both ACO algorithms (comparing rows 1 and 3 of Tables 5.10 to 5.13), indicating the usefulness of the local search algorithm in improving the performance of ACO-PPMSO when heuristic information is not readily available.

Duration Extender is mainly used to locally optimize ACO solutions with regard to shortening/deferral decisions. Compared to real PPMSO case studies, the modified 21- and 22-unit case studies have only a small number of shortening and deferral options available, which may make a rigorous examination of the performance of the *Duration Extender* algorithm difficult. Therefore, the usefulness of the *Duration Extender* algorithm was further investigated with real PPMSO case studies (Chapter 6).

C. Overall performance of ACO-PPMSO

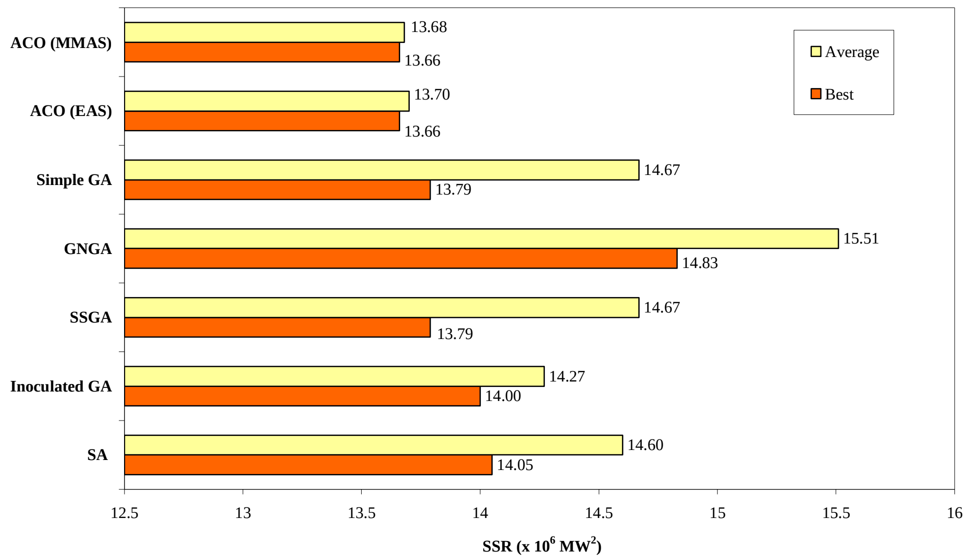
Original 21-unit and 22-unit case studies

By using the ACO-PPMSO algorithm, a new best-known objective value has been found for both the original 21-unit case study (SSR = 13.66 x 10^6 MW^2) and the original 22-unit case study (LVL = 52.06 MW).

A comparison of the results obtained by ACO-PPMSO with those obtained by various metaheuristics in other studies for the 21-unit case study is shown in Figure 5.20. As mentioned previously, the number of evaluations (trial solutions) allowed in the ACO runs and those of the other metaheuristics was identical. In particular, the best and average results of the metaheuristics were compared. While the best and average results given by the simple GA, SSGA, GNGA, inoculated GA and SA were obtained by 10 runs with different starting positions (Aldridge *et al.*, 1999; Dahal *et al.*, 1999; Dahal *et al.*, 2000), those of EAS and MMAS were obtained using 50 runs.

It can be seen that the EAS and MMAS algorithms have outperformed the algorithms that have been applied to this case study previously. It should be noted that a new best-found solution (SSR = 13.66 x 10^6 MW^2) for the 21-unit case study has been found by EAS and MMAS using the new ACO-PPMSO formulation. In addition, it can be seen that the differences between the average and best results of the ACO algorithms are much smaller than those for other metaheuristics (Figure 5.20), which indicates a consistent performance of the ACO-PPMSO formulation.

Among the metaheuristics previously used for solving the 21-unit case study, the inoculated GA, where the initial population is generated using a heuristic that ranks the generating units in order of decreasing capacity, was found to perform best in terms of the average results obtained. This indicates the potential of the benefit of a heuristic in solving PPMSO problems.



Notation

ACO: Ant Colony Optimisation

EAS: Elitist-Ant System

MMAS: Max-Min Ant System

Simple GA: Simple Genetic Algorithm

GNGA: Generational Genetic Algorithm

SSGA: Steady State Genetic Algorithm

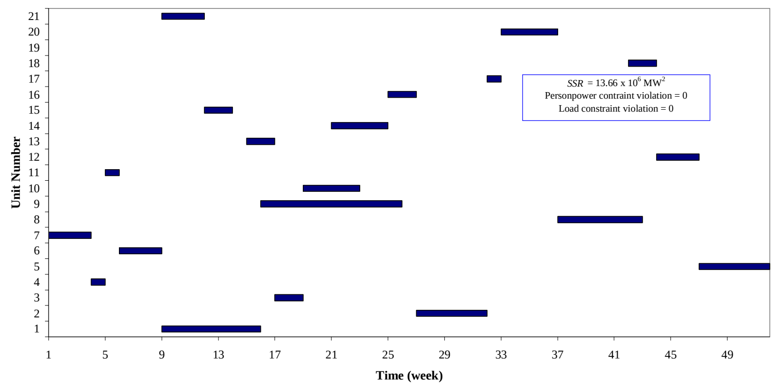
Inoculated GA: Inoculated Genetic Algorithm

SA: Simulated Annealing

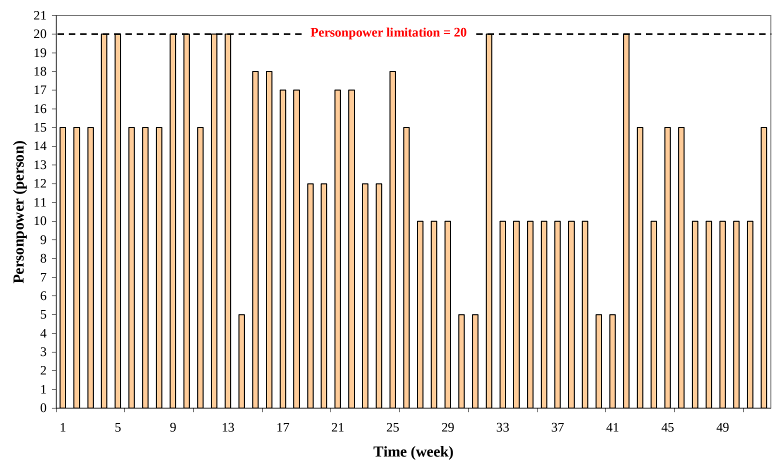
Figure 5.20: Comparison between results obtained using other optimisation methods (Aldridge *et al.*, 1999; Dahal *et al.*, 1999; Dahal *et al.*, 2000) and the ACO algorithms used in this thesis

As mentioned previously, a new best-found solution (SSR = 13.66 x 10^6 MW^2) has been found by the ACO-PPMSO formulation proposed in this thesis. In fact, different maintenance schedules were found that are associated with the new best-found SSR solution. Two maintenance schedules associated with the best-found SSR of 13.66 x 10^6 MW^2, along with the corresponding generation reserve levels and personpower utilization over the planning horizon, are presented in Figures 5.21 and

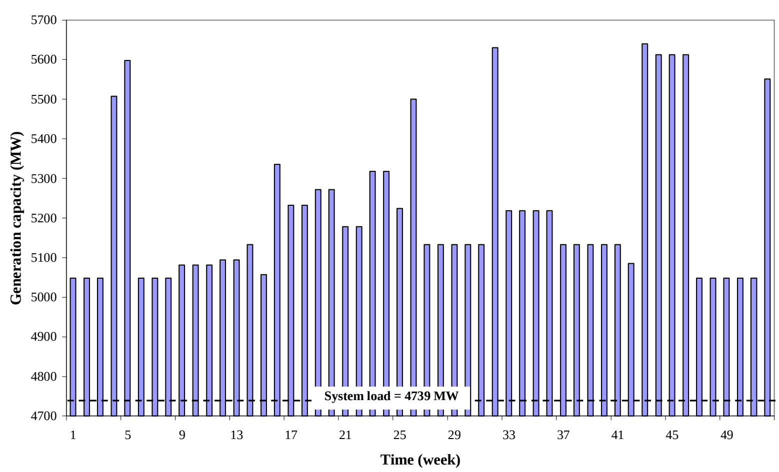
5.22. It can be seen that there is no demand or personpower shortfall associated with the two schedules. Despite the identical SSR-values, the two schedules are different, indicating there is more than one optimal solution in the problem search space. The two schedules that result in different personpower allocation profiles provides great flexibility during a negotiation with the asset manager. Maintenancs schedules associated with sub-optimal SSR values were also investigated, as such schedules may sometimes be preferred when some non-quantifiable criteria are taken into account. The maintenance schedule associated with a near-best-known SSR solution (SSR = 13.68 x 10^6 MW^2) is shown in Appendix B3.



(a)

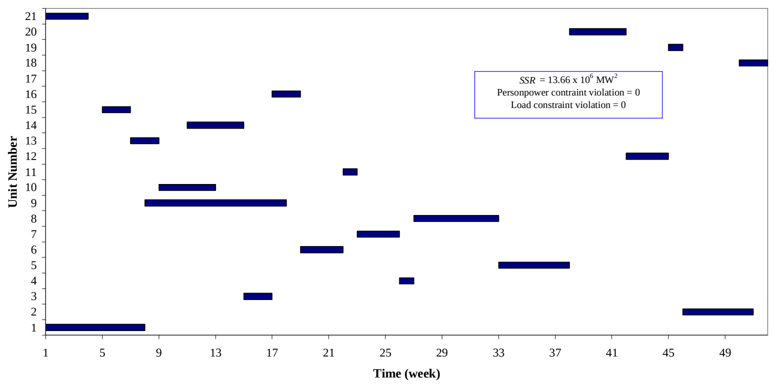


(b)

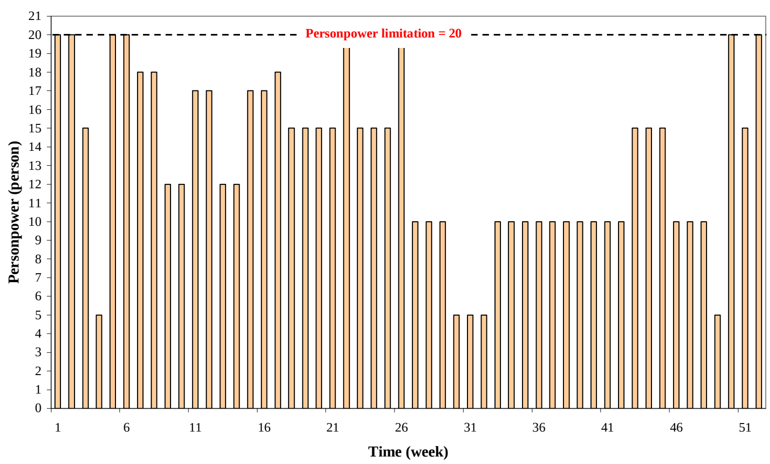


(c)

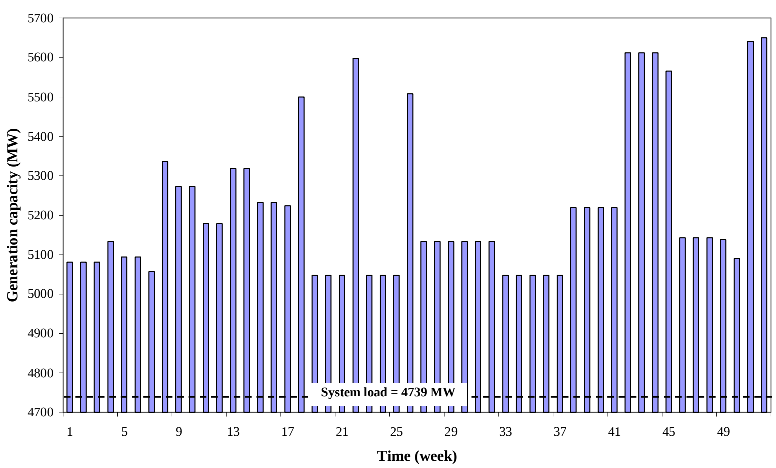
Figure 5.21: The (a) maintenance schedule of the 21-unit case study best-found-SSR solution A, the associated (b) personpower allocation and (c) reserve capacity levels



(a)



(b)



(c)

Figure 5.22: The (a) maintenance schedule of the 21-unit case study best-found-SSR solution B, the associated (b) personpower allocation and (c) reserve capacity levels

In Figure 5.23, the reserve level across the planning horizon associated with the best-known schedule found by ACO-PPMSO for the original 22-unit case study is compared with those obtained by implicit enumeration (Escudero *et al.*, 1980) and tabu search (El-Amin *et al.*, 2000). It can be seen that the reserve level given by the ACO schedule is more evenly spread out (summed deviation of generation reserve from the average reserve, LVL = 52.06 MW) than those obtained with implicit enumeration (LVL = 118.81 MW) and tabu search (LVL = 256.93 MW). It should be noted that due to insufficient information about the optimum solution in El-Amin *et al*. (2000), the LVL value of tabu search shown in Figure 5.23 was calculated using the best available published information (including a maintenance schedule shown in Appendix C3).

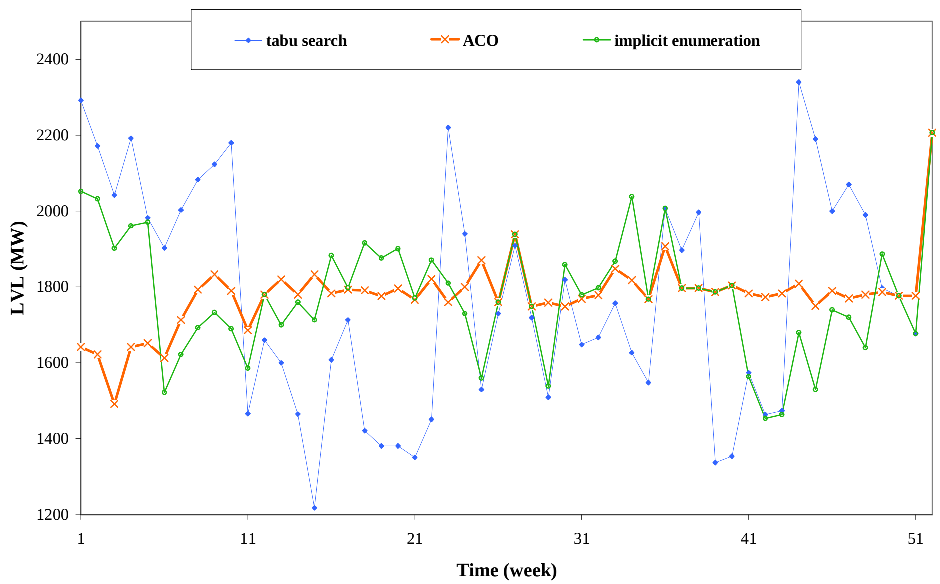
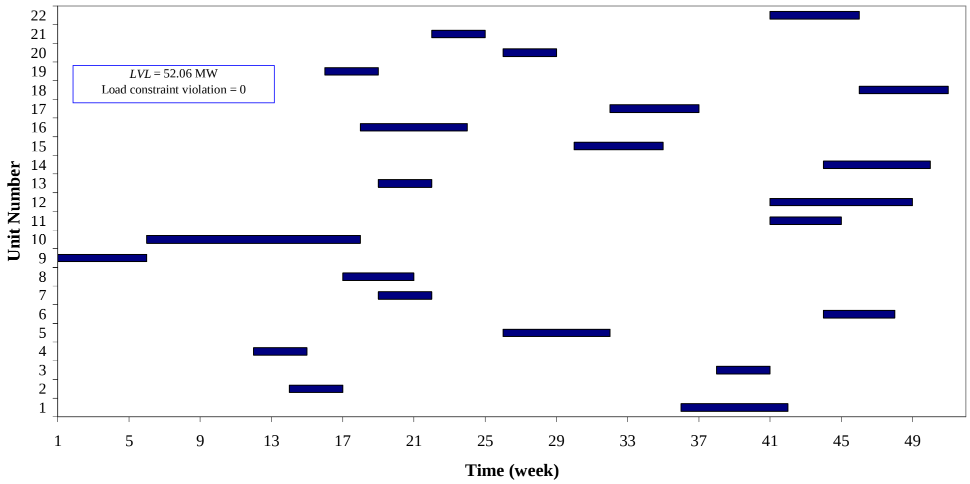
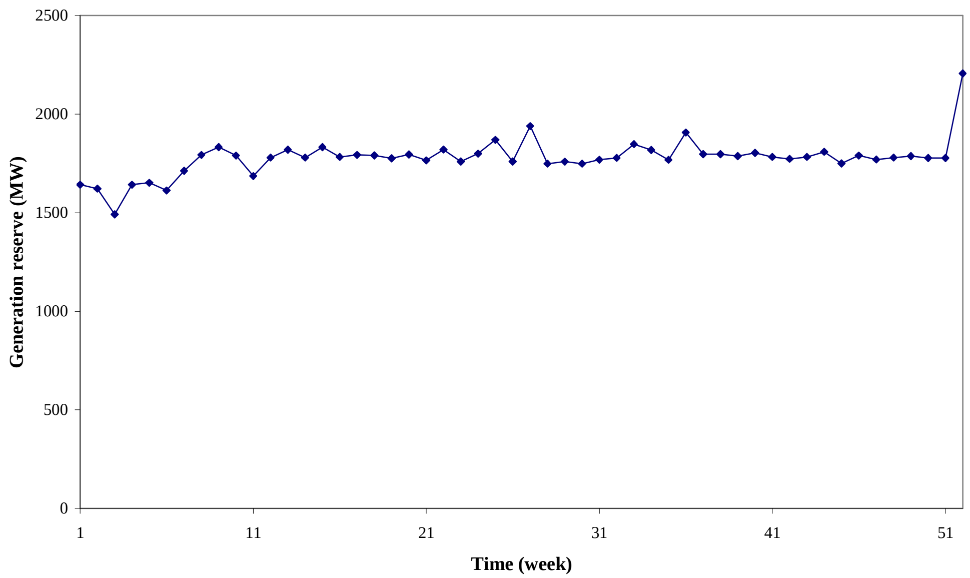


Figure 5.23: Comparison of reserve levels obtained using ACO, implicit enumeration (Escudero *et al.*, 1980) and tabu search (El-Amin *et al.*, 2000)

The best maintenance schedule found by ACO-PPMSO for this case study is shown in Figure 5.24, which is associated with the ACO reserve level presented in Figure 5.23. Another maintenance schedule associated with a near-best-known objective function value is presented in Figure 5.25. It can be seen that the two schedules are different but the objective function values associated with these schedules differ only by less than 1% (52.06 MW and 53.02 MW). In general, more than two different schedules can be produced by an ACO run according to the requirements specified by the user. For example, the best ten schedules obtained throughout an ACO run are examined. The availability of a wide range of different schedules that share similar objective function values provide a great flexibility to a scheduler when dealing with non-quantitative criteria (eg. operation and trading protocols, availability of resources etc.).

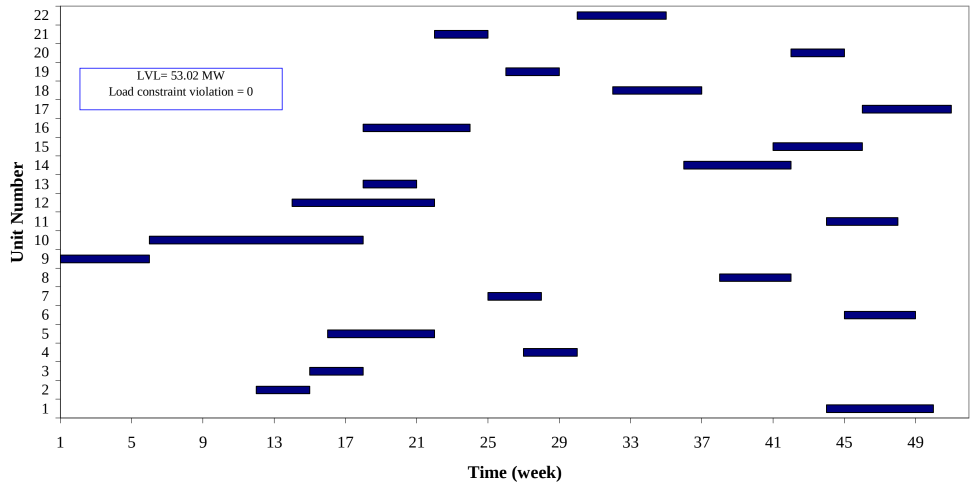


(a)

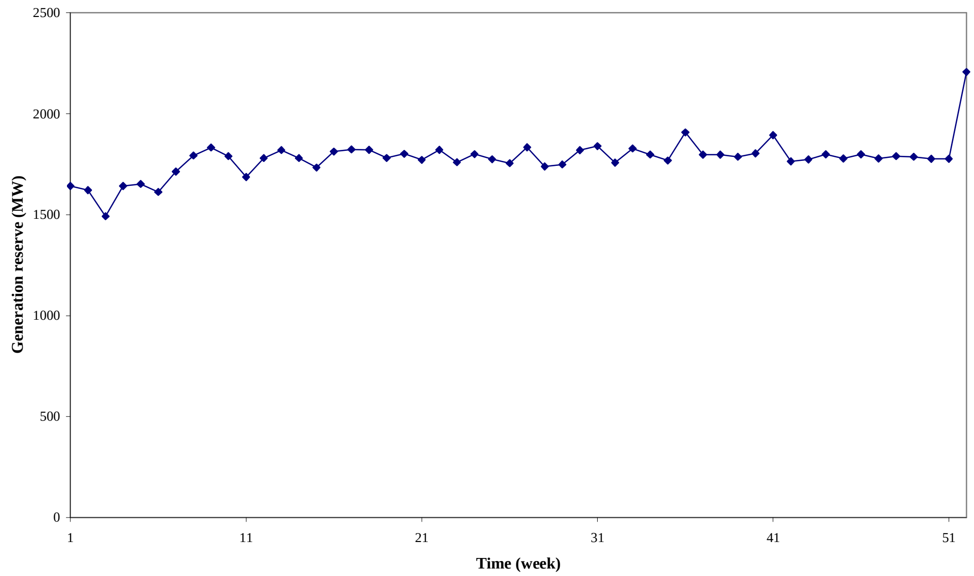


(b)

Figure 5.24: Best-known (a) schedule and (b) the associated generation reserve levels for the 22-unit case study



(a)



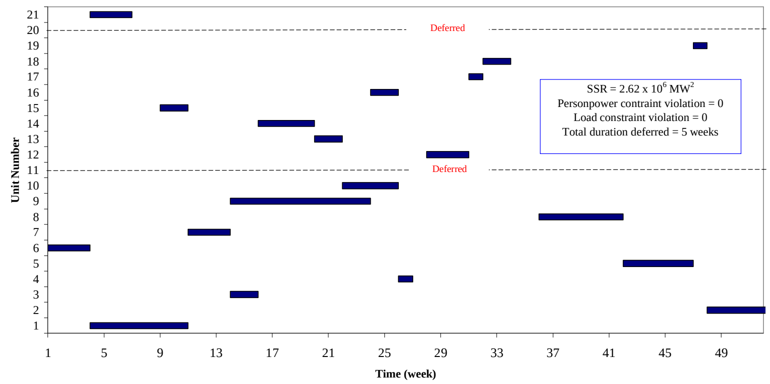
(b)

Figure 5.25: A near best-known (a) schedule and (b) the associated generation reserve levels for the 22-unit case study

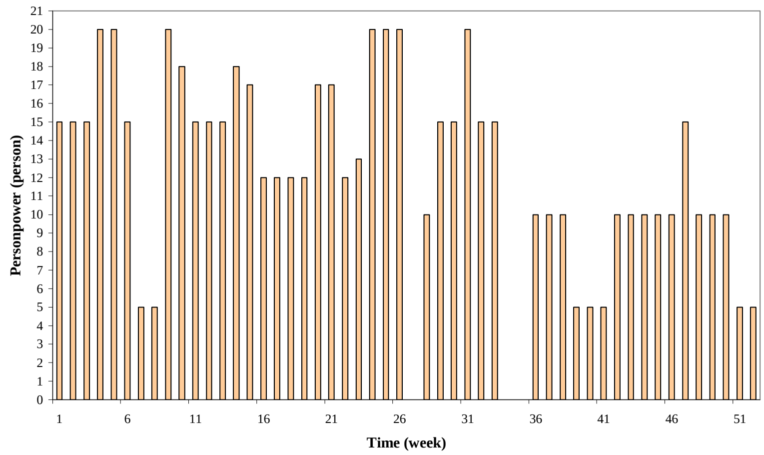
<u>Modified 21-unit and 22-unit case studies</u>

As the modified versions of the 21- and 22-unit case studies have been introduced in this research to test the developed ACO-PPMSO formulation, there are no previous results available for comparison purposes. As can be seen in Tables 5.10 to 5.13, the optimized maintenance schedules of both the modified 21- and 22-unit case studies include the shortening and/or deferral of maintenance tasks (average duration shortened/deferred > 0). Two maintenance schedules associated with the best-found objective function cost for the modified 21-unit case study are shown in Figures 5.26 and 5.27 (OFC = $15.71M) and for the modified 22-unit case study in Figures 5.28 and 5.29 (OFC = $916.12). In both schedules for the modified 21-unit case study, the maintenance tasks for generating units 11 and 21 are deferred, while all other tasks are carried out as normal. For the modified 22-unit case study, maintenance tasks for generating units 10, 16 and 17 are shortened by 2, 4 and 2 weeks, respectively. It should be noted that all constraints are satisfied by the schedules presented in Figures 5.26 to 5.29.

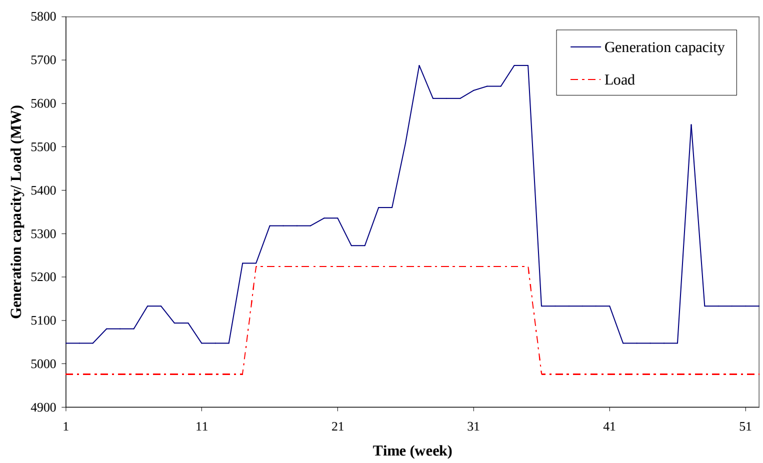
The results for the modified versions of the 21-unit and 22-unit case studies indicate that the ACO-PPMSO formulation introduced in this thesis is able to provide maintenance schedules that satisfy hard system constraints (eg. system demands) by shortening and deferring maintenance tasks. More importantly, the shortening and deferral options were not used if not necessary, as only a few, but not all, maintenance tasks were shortened/deferred (Figures 5.26 to 5.29).



(a)



(b)



(c)

Figure 5.26: The (a) maintenance schedule of the modified 21-unit case study best-found-SSR solution A, the associated (b) personpower allocation and (c) reserve capacity levels

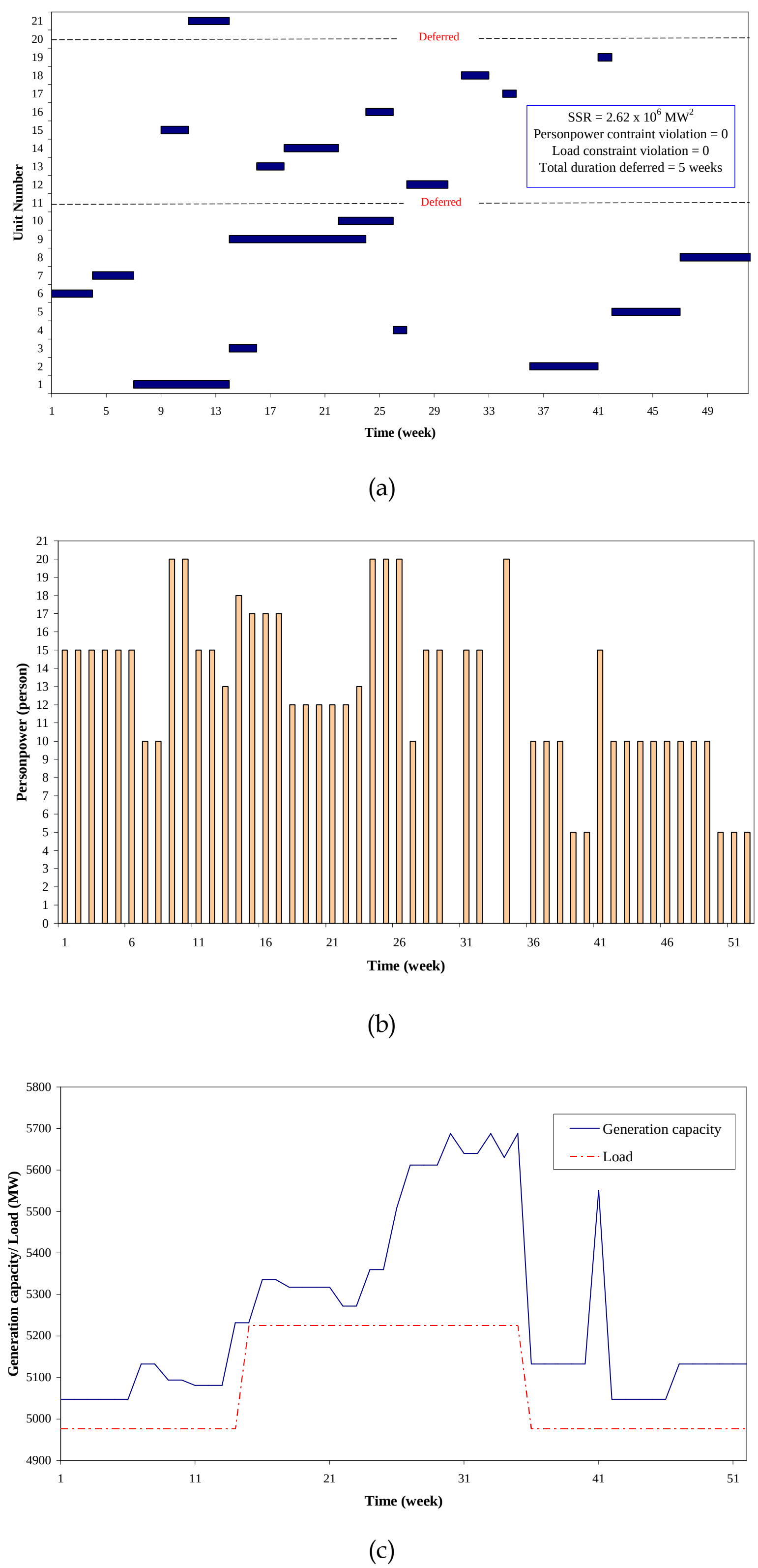
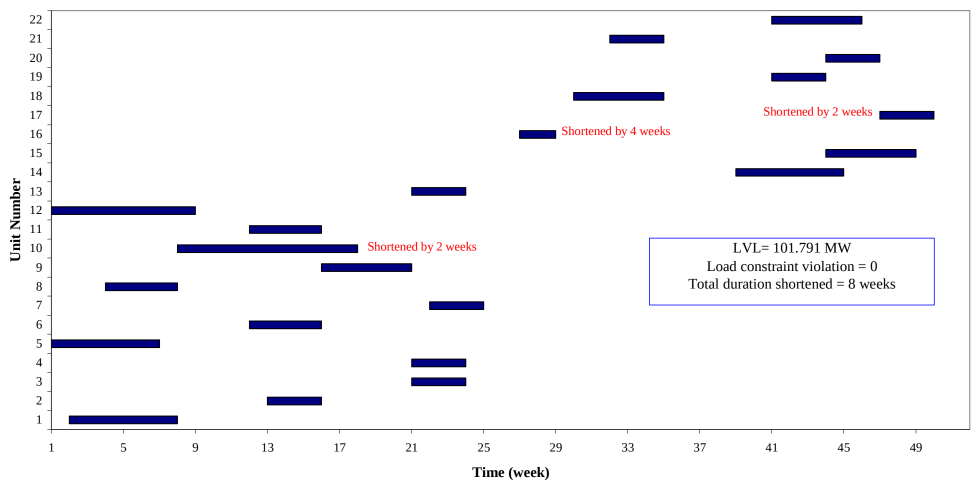
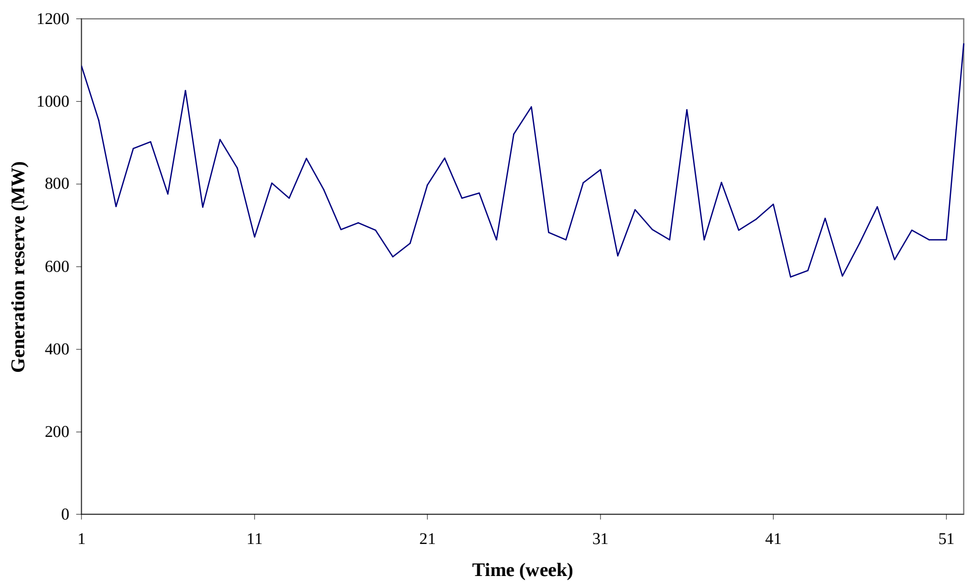


Figure 5.27: The (a) maintenance schedule of the modified 21-unit case study best-found-SSR solution B, the associated (b) personpower allocation and (c) reserve capacity levels

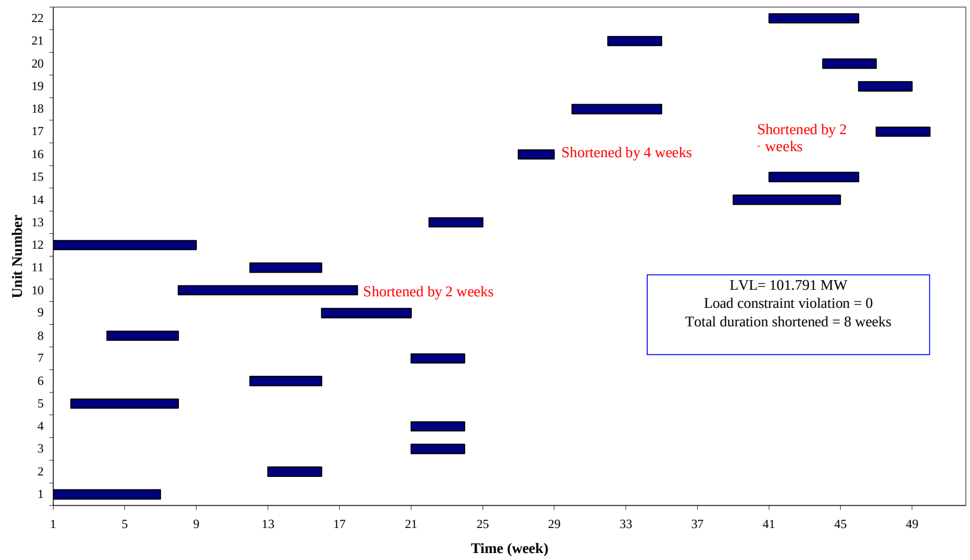


(a)

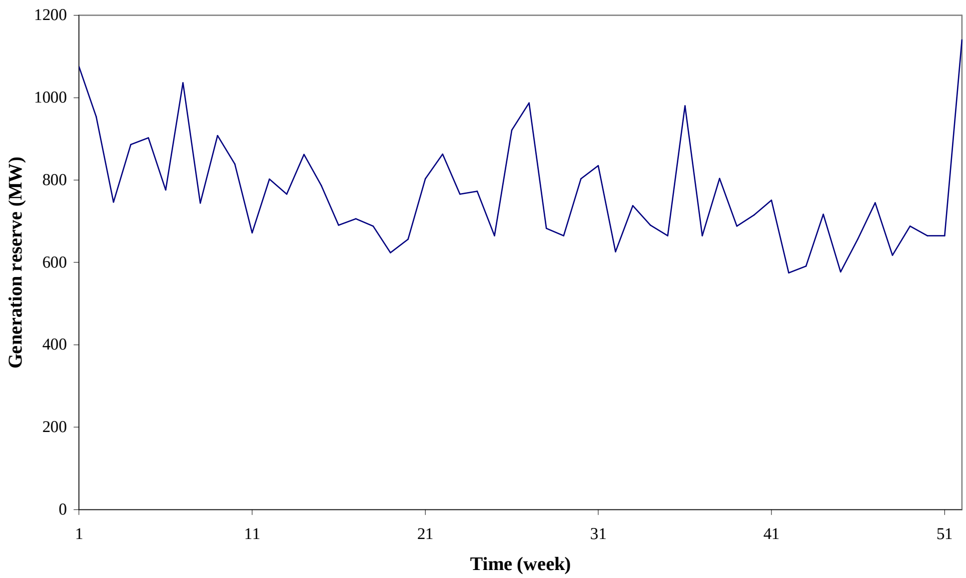


(b)

Figure 5.28: (a) Maintenance schedule A associated with the best-found OFC for the modified 22-unit case study and (b) the associated generation reserve levels



(a)



(b)

Figure 5.29: (a) Maintenance schedule B associated with the best-found OFC for the modified 22-unit case study and (b) the associated generation reserve levels

5.5 Summary and conclusions

In this chapter, the new ACO-PPMSO formulation introduced in Chapter 4 was tested on four case studies (original and modified versions of two benchmark case studies from the literature). In particularly, the performance of the heuristic formulation developed, the two local search algorithms introduced and the overall utility of the ACO-PPMSO formulation were investigated.

A testing procedure consisting of three stages was used to assess the utility of the proposed ACO-PPMSO formulation. In order to examine the impact that the ACO algorithms and parameters have on the performance of the ACO-PPMSO formulation, two ACO algorithms, namely Elitist-Ant System (EAS), Max-Min Ant System (MMAS) and a wide range of ACO parameters were utilized as part of the testing procedure. In addition, each ACO run was repeated with 50 random number seeds to minimize the impact of different initial positions in the search space. The significance of the experimental results obtained were checked using the Student's *t*-test.

Results of the testing have shown that the heuristic formulation improves the performance of the ACO-PPMSO algorithm significantly when applied to the four case studies investigated. It was found that while the *PPMSO-2-opt* local search operator seems to work well for unconstrained problems, it is not suitable for highly-constrained PPMSO problems. On the other hand, the performance of the *Duration Extender* local search operator has resulted in significant improvements in cases where duration shortening is applicable. Lastly, the results obtained by ACO-PPMSO for the two original case studies were better than those obtained by other optimisation methods, such as various genetic algorithm (GAs) formulations and simulated annealing (SA). For the 21-unit and 22-unit case studies, a new optimal solution has been found by the ACO-PPMSO formulation. In addition, the results given by ACO-PPMSO were more consistent compared with those obtained using other metaheuristics previously applied to the two benchmark case studies. The maintenance schedules found for the modified case studies have also been examined and it was found that the ACO-PPMSO formulation is able to meet hard system constraints by shortening and deferring maintenance.

The results of experiments carried out using the original and modified versions of the 21-unit and 22-unit case studies indicated that the ACO-PPMSO formulation developed as part of the research work presented in this thesis has potential for solving real-world PPMSO problems.

Chapter 6
Hydroelectric Power Case Studies

In this chapter, a five-station case study derived from the Hydro Tasmania hydropower system is used to test the utility of the ACO-PPMSO formulation for real-world maintenance scheduling problems. As part of the testing procedure, the impacts of shortening and deferral options on practical maintenance scheduling are investigated. In addition, the usefulness of a local search operator (*Duration Extender*) introduced in Chapter 4 is examined. The ACO-PPMSO formulation is subsequently utilized to schedule the 2006 maintenance tasks for the full Hydro Tasmania system, which consists of 55 generating units and a total of 118 maintenance tasks. In addition, four different scenarios are investigated, which represent various circumstances that a decision maker might encounter during maintenance scheduling, including routine maintenance scheduling, an increase in system demand, the unavailability of a major generating unit and the late return of a major generating unit from maintenance.

6.1 Background

Tasmania is the smallest and the only island state of Australia, lying south of the south-east corner of the Australian mainland (Figure 6.1). Tasmania has a total area of 68,332 km^2 (Wikipedia, 2006b) and a total population of 487,185 (Jackson, 2005). With its high rainfall and mountainous terrain, Tasmania has abundant water resources for renewable energy production. Having harnessed Tasmania's water for energy production for over 80 years, Hydro Tasmania is Australia's largest renewable energy generator with 29 small- to medium-sized hydroelectric power stations and one thermal power station. The thermal stations are used in emergency situations where there is a demand shortfall and their use is avoided if possible as a result of the high costs incurred. With an installed generating capacity of 2,260 MW, the Hydro Tasmania system produces over 10,000 GWh of renewable energy on an annual basis, which is approximately 60% of Australia's total renewable energy production` (Beswick *et al.*, 2003).

Figure 6.1: Geographical location of Tasmania

6.2 Five-Station Hydropower System

In order to further test the utility of the proposed formulation, a subset of the Hydro Tasmania power system was investigated in this study, which includes two catchment areas (Pieman-Anthony and Gordon-Pedder) and five power stations. The five power stations considered include eight generating units with an installed generating capacity of 893 MW (Figure 6.2).

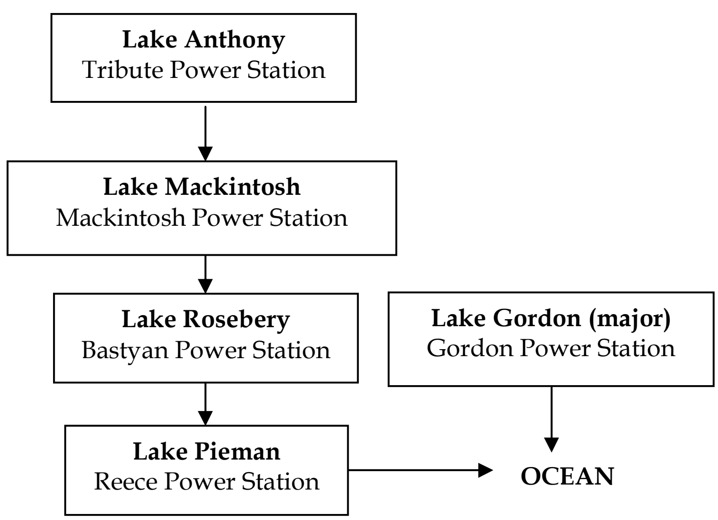


Figure 6.2: Schematic diagram of the five-station hydropower system

Of the five storages where water is drawn for power generation, three are run-of-the-river storages (Lakes Anthony, Rosebery and Pieman), while the other two are major storages (Lakes Mackintosh and Gordon). Run-of-river storages have limited storage capacity and in order to avoid spilling, they are given priority to operate, especially during high-inflow periods. On the other hand, major storages can store large volumes of water, and are normally relied upon for power generation during low inflow periods. Details of the five storages and the associated power stations are given in Table 6.1.

Table 6.1: Power station and headwater data

Power station	*Tribute*	*Mackintosh*	*Bastyan*	*Reece*	*Gordon*
Number of generators	1	1	1	2	3
Generating capacity of each generator (MW)	83	80	80	115	140
Maximum discharge (cumec)	34	145	145	144	86
Average efficiency factor (MW/cumec)	2.42	0.55	0.55	0.8	1.62
Headwater storage	Lake Anthony	Lake Mackintosh	Lake Rosebery	Lake Pieman	Lake Gordon
Storage capacity (10^6 m^3)	22	336	51	100	10,990

6.2.1 Problem specification

This case system requires a total of 14 maintenance tasks to be scheduled once over a planning horizon of 365 days from Jan 1, 2006 (Table 6.2). The task IDs denoted by "Inv" are investigative tasks in which the condition of generators is examined prior to the actual maintenance (task IDs denoted by "Act"). Among all maintenance tasks, the biggest loss of generation capacity occurs during the upgrade of the Gordon power station, when all three generating units of the station are inoperable. The starting levels of Lake Gordon and other storages are assumed to be 60% and 75% full, respectively, in this problem (Stolp, S., personal communication, 2004).

Table 6.2: Details of maintenance tasks

Power Station	Machine number	Maintenance type	Task ID	Normal mainte-nance duration (days)	Loss of generating capacity (MW)
Tribute	1	Investigative	Tri_Inv	5	83
	1	Actual	Tri_Act	12	83
Mackin-tosh	1	Investigative	Mac_Inv	5	80
	1	Actual	Mac_Act	19	80
Bastyan	1	Investigative	Bst_Inv	5	80
	1	Actual	Bst_Act	12	80
Reece	1	Investigative	Rce#1_Inv	5	115
	1	Actual	Rce#1_Act	19	115
	2	Investigative	Rce#2_Inv	5	115
	2	Actual	Rce#2_Act	19	115
Gordon	1	Actual	Gor#1_Act	19	140
	2	Actual	Gor#2_Act	19	140
	3	Actual	Gor#3_Act	19	140
	Station upgrade	Actual	Gor_stn	42	420

The aim of this optimisation problem is to determine a commencement time and duration for each maintenance task in the hydropower case system, such that the system reliability is maximized (Eq. 6.1) and the total duration shortened/deferred is minimized (Eq. 6.2), subject to a number of constraints. In this case study, the maximization of system reliability is achieved by maximizing the expected total final energy in storage of the two major storages at the end of the planning horizon:

$$\textbf{Objective 1}: Max\ \{ETFEIS(s) = EFEIS_{Mackintosh}(s) + EFEIS_{Gordon}(s)\} \qquad (6.1)$$

where $ETFEIS(s)$ is the expected total final energy in storage (GWh) of Lakes Mackintosh and Gordon associated with maintenance schedule s, at the end of the planning horizon; $EFEIS_{Mackintosh}(s)$ and $EFEIS_{Gordon}(s)$ are the expected energy in storage (GWh) of Lakes Mackintosh and Gordon, respectively, associated with maintenance schedule s at the end of the planning horizon.

$$\textbf{Objective 2}: Min\{DurCut_{tot}(s)\} \qquad (6.2)$$

where the value of the total duration shortened and deferred associated with schedule s, $DurCut_{tot}(s)$, is given by Eq. 5.18, where $total\_n$ = 14 in this case.

The constraints to be satisfied are (Stolp, S., personal communication, 2004):

1. The earliest time a maintenance task can start is January 1 and all tasks should be finished by December 31.

2. An investigative task has to finish between 4 to 6 weeks prior to the commencement of the actual maintenance task.

3. There is no maintenance during the Easter, Christmas and New Year public holidays.

4. The maintenance duration of all tasks can be shortened by a time step of 2 days.

5. The system power demands (Figure 6.3) have to be met throughout the planning horizon. The total expected unserved energy (EUE) over the planning horizon should not be greater than 0.002% of the total annual energy demand.

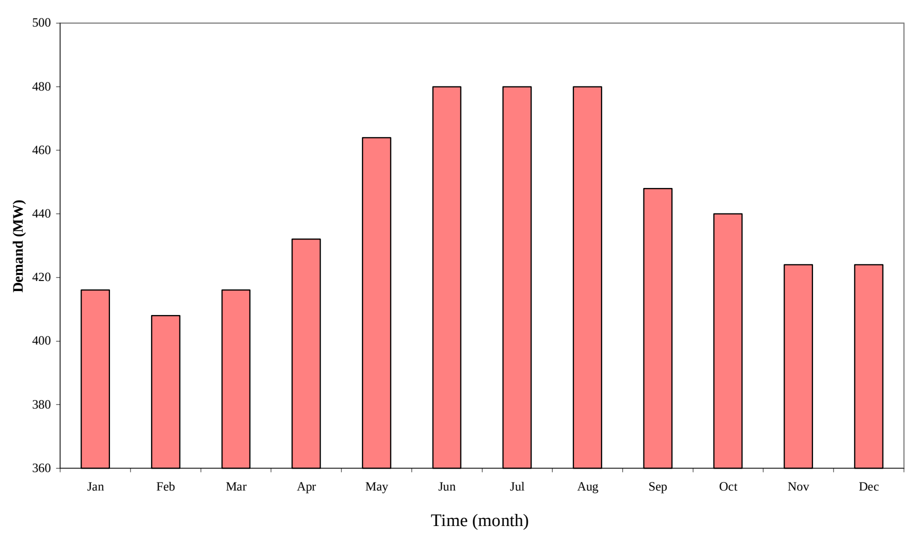


Figure 6.3: Forecasted system demand for 2006

6.2.2 Problem formulation

In the ACO-PPMSO formulation, constraints are incorporated at the earliest possible stage during the optimisation process, using either the graph-based or penalty-based techniques introduced in Section 4.4. In the five-station case study system, constraints 1, 2, 3 and 4 are related to the timeframe during which maintenance tasks are allowed to commence. Therefore, it is more computationally effective to take these constraints into account during the construction of trial solutions, so that the trial solutions generated are feasible with regard to these constraints (construction graph-based technique in Section 4.4). For example, in order to incorporate constraints 1 and 2, the decision paths associated with investigative and actual tasks are dynamically updated during construction of each trial maintenance schedule. In the construction of a trial maintenance schedule, if May 18 was chosen as the commencement date for the actual maintenance task of the unit at Tribute power station, the corresponding investigative task will be dynamically assigned optional start days from April 1 to April 15 (Figure 6.4). It should be noted that if the investigative task was assigned a start time first, the optional start days for the corresponding actual task would be updated dynamically in the same way. Similarly, constraint 3 is handled by eliminating the optional start days associated with public holidays during the construction of trial solutions. Constraint 4 is addressed by allowing only durations that are greater than the minimum maintenance durations during the construction of trial maintenance schedules.

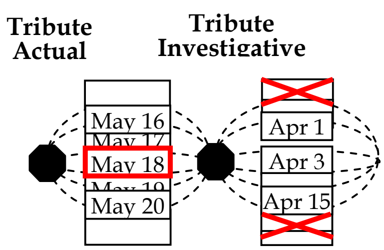


Figure 6.4: Handling of constraints 1 and 2

Unlike constraints 1 to 4, whether or not constraint 5 (system demand) is satisfied by a trial maintenance schedule is not known until the complete schedule has been constructed and a simulation model has been run, necessitating the use of penalty-based techniques in order to meet this constraint (see Section 4.4).

Adapting Eq. 4.18, the objective function used for this problem is comprised of the actual objective terms i.e. the expected total final energy in storage (ETFEIS) and the total duration cut down ($DurCut_{tot}$), as well as an additional term to address the violation of load constraints (EUE), and is given by:

$$OFC(s) = (c_{EUE} \cdot EUE(s) + \frac{c_{ETFEIS}}{ETFEIS(s)}) \cdot DurCut_{tot}(s)^2 \tag{6.3}$$

where $OFC(s)$ is the objective function cost (\$) associated with a trial maintenance schedule, s; $EUE(s)$ is the total annual expected unserved energy (GWh) associated with a trial maintenance schedule, s; $ETFEIS(s)$ is the expected total final energy in storage (GWh) associated with a trial maintenance schedule, s; c_{EUE} is the penalty cost per unit EUE (\$/GWh); c_{ETFEIS} is the cost per unit of the inverse of $ETFEIS$ (\$GWh); $DurCut_{tot}$ (s) is the total reduction in maintenance duration (day) associated with a trial maintenance schedule, s due to shortening and deferral.

The OFC can be viewed as the virtual cost associated with a trial maintenance schedule. It should be noted that the values of c_{EUE} and c_{ETFEIS} in the objective function (Eq. 6.3) can be varied to reflect the relative importance of the objectives and constraints, as perceived by the decision maker. Hard constraints (demand constraints in this case) are usually assigned relatively higher costs, such that trial solutions that violate these constraints are more heavily penalized. It can also be seen that the greater the reduction in maintenance duration in a trial maintenance schedule, the higher the associated OFC. The values of c_{EUE} and c_{ETFEIS} used in the optimisation runs for this problem were chosen to be 1000 and 10000, respectively. As a hard constraint, the penalty cost associated with violation of demand constraints ($c_{EUE} \cdot EUE(s)$ in Eq. 6.3) should be much higher than that associated with an objective term, which is the total final energy in storage term ($\frac{c_{ETFEIS}}{ETFEIS(s)}$ in Eq. 6.3) in this case.

The value of $DurCut_{tot}$ (Eq. 6.3) associated with a trial schedule can be easily calculated once the complete schedule has been obtained, or even during the construction of the schedule using Eq. 5.18. However, the values of expected unserved energy (EUE) and expected total final energy in storage (ETFEIS) associated with a trial maintenance schedule are calculated using a simplified version of the SYSOP (SYStems-OPeration) simulation model currently used by Hydro Tasmania for the assessment of proposed maintenance schedules for its full system. In SYSOP, dispatching rules that specify the order in which storages are used for power generation when meeting demands are employed. For example, run-of-river storages that have exceeded certain storage levels are given higher priority during dispatch to avoid spilling. During the ACO-PPMSO optimisation process, the trial maintenance schedule generated by individual ants, along with the system demand, storage inflows, and the initial level of storages at the start of the planning horizon are input into the simplified SYSOP model. The outputs of the simplified SYSOP model, including the expected total final energy in storage of the major storages and the expected unserved energy over the planning horizon, are used to calculate the objective function cost, OFC, associated with a trial maintenance schedule using Eq. 6.3.

6.2.3 Analysis conducted

An experiment has been conducted to assess the utility of the proposed ACO-PPMSO formulation for real PPMSO problems. Particular emphasis was given to assessing the utility of the shortening and deferral options, the impact of the *Duration Extender* local search operator and the overall performance of the proposed ACO-PPMSO formulation.

The optimisation runs in the experiment described above are performed on a Linux Symmetric Multi Processor Kernel (Memory: 1GB; CPU: AMD Athlon(tm) MP 2600+) utilizing the ACO-PPMSO program described in Section 4.5. The simulation model routine used in the program is a simplified version of the SYSOP model that caters only for the five power stations investigated in this case study. In order to facilitate its use within the ACO-PPMSO program, the simplified SYSOP model originally written in the PASCAL language has been translated to the Fortran 90 language as part of this research.

A. Utility of shortening and deferral options

The impact of shortening and deferral options in the ACO-PPMSO formulation for real-world PPMSO problems was investigated by considering the following scenarios:

Scenario 1: All maintenance tasks must be completed at normal outage duration within the specified planning horizon. In other words, neither shortening nor deferral are allowed.

Scenario 2: The options of shortening outage duration and deferral of maintenance tasks are considered.

For both scenarios, the optimum maintenance schedules obtained as a result of different storage inflows were examined. The three storage inflow conditions tested were extracted from 80 years of historical inflow data at the 13th percentile (dry year), 64th percentile (intermediate year) and 92nd percentile (wet year) (Stolp, S., personal communication, 2004). The monthly average inflows of individual storages for dry, intermediate and wet years are shown in Figures 6.5 to 6.7.

B. Performance of the *Duration Extender* local search operator

The performance of the *Duration Extender* local search strategy (see Section 4.3.4) was examined by carrying out separate ACO runs with and without using the local search. The effectiveness of the *Duration Extender* was then checked using a Student's t-test at a 5% significance level. It should be noted that *Duration Extender* is only applicable to cases where shortening and deferral options are considered.

C. Overall performance of ACO-PPMSO for real-world PPMSO problems

The results obtained by ACO-PPMSO were compared with those found by a random evaluation method and a maintenance schedule proposed by Hydro Tasmania practitioners based on conventional techniques and engineering judgment. In a random evaluation run, the number of maintenance schedules generated was identical to the number of schedules generated in an equivalent ACO run.

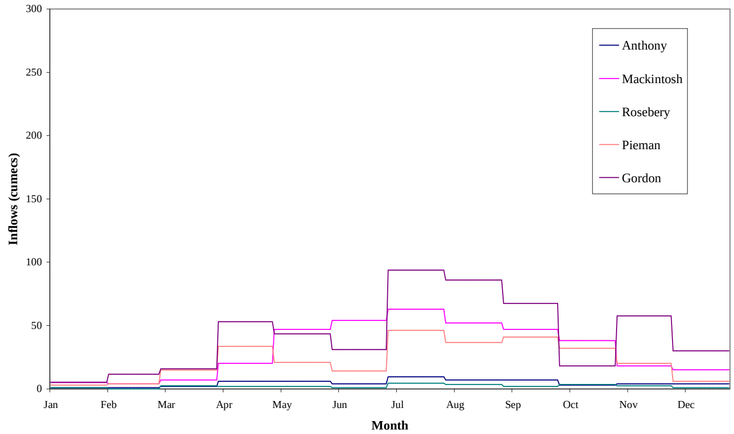


Figure 6.5: Dry year storage inflows

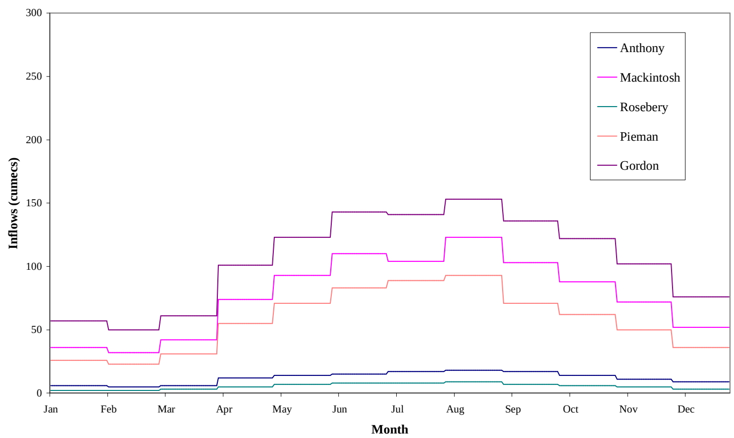


Figure 6.6: Intermediate year storage inflows

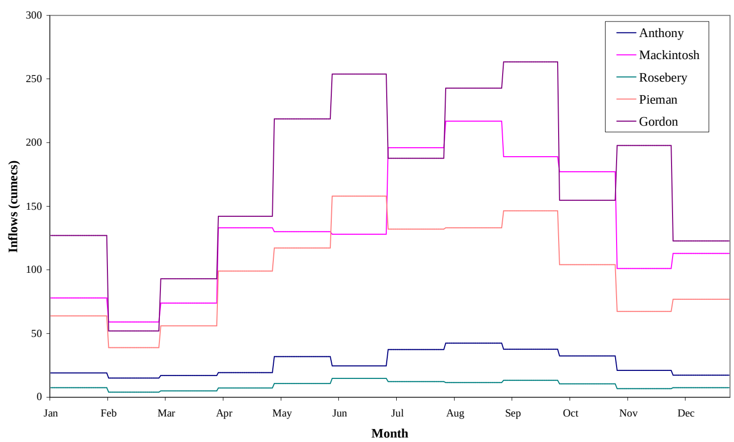


Figure 6.7: Wet year storage inflows

In order to achieve the objectives outlined above, the testing procedure shown in Figure 6.8 was implemented. Items A, B and C aforementioned were investigated at stages A, B and C in the testing procedure, respectively. The Max-Min Ant System (see Section 3.3.1), which was found to be superior to the Elitist-Ant System (see Section 3.3.1) for the 21-unit and 22-unit case problems, was adopted for this problem.

To minimize the impact the parameters used have on the evaluation of the effectiveness of the shortening and deferral options, local search and overall performance of the ACO-PPMSO algorithm, a wide range of ACO parameters (shown in the dashed box in Figure 6.8) was used to solve the problem. It should be noted that investigations into the effect of the reward factor Q (Eq. 4.23) and initial pheromone τ_0 (Section 4.3.1) are not considered in this study, as they were found to have no impact on algorithm performance. The values of both α and β were set to 1.0. In addition, each run was repeated 30 times with different random number seeds in order to minimize the influence of random starting values in the solution space on the results obtained and to enable statistical significance testing of the results to be conducted. In each ACO run, a maximum of 100,000 trial solutions were generated, where 'an ACO run' is defined as the use of a particular set of parameters (for example, m = 800; ρ = 0.9; p_{best} = 0.01) to solve the hydropower case study system maintenance scheduling problem, given a storage inflow condition (for example, wet year inflow), using a specified random number seed (for example, 8998). The performance of a parameter setting is then gauged by the best *OFC* obtained in each run, averaged over 30 ACO runs with different random number seeds.

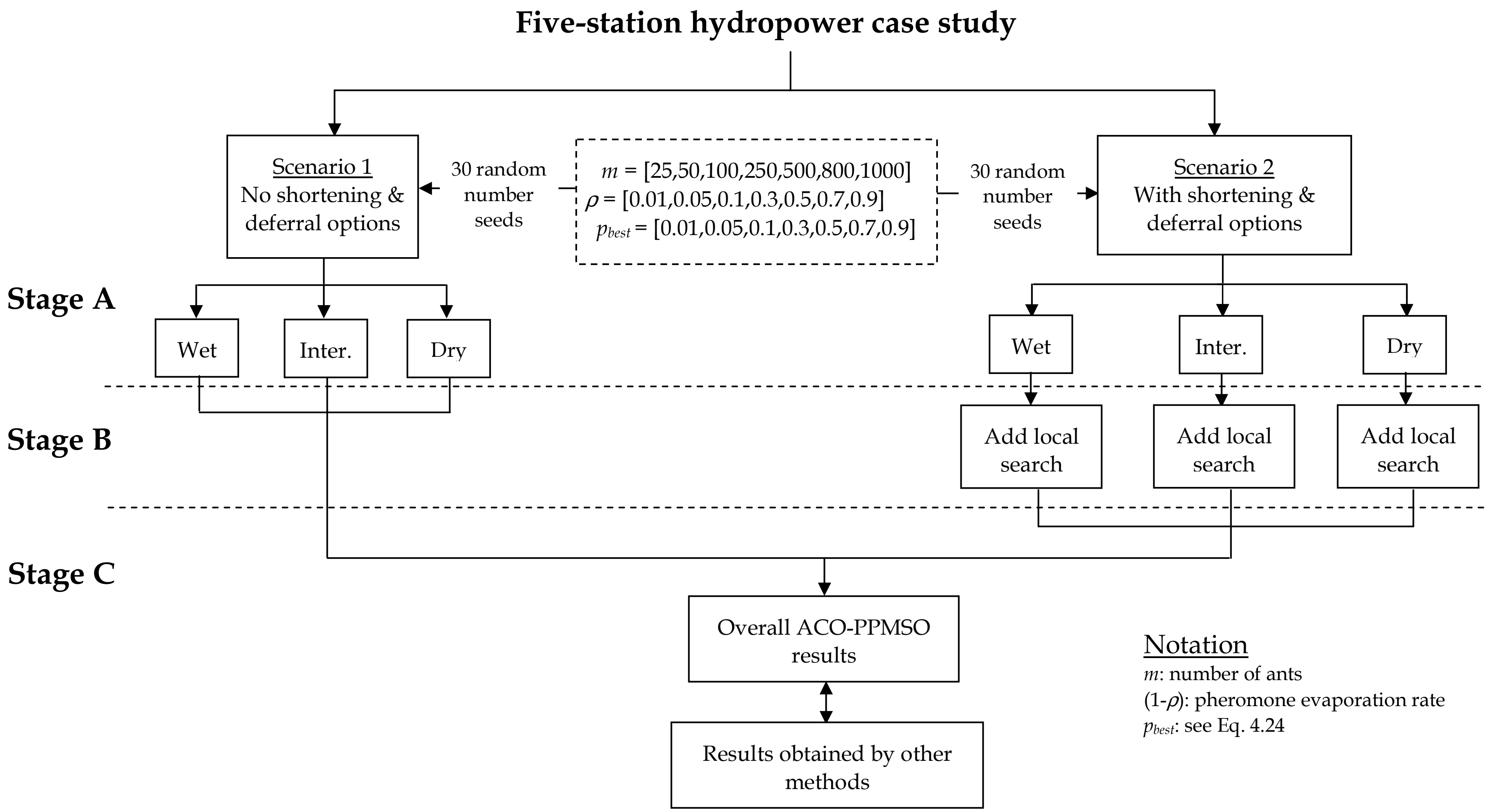


Figure 6.8: Experimental procedure for the five-station hydropower plant maintenance scheduling optimisation case study

6.2.4 Results and discussion

The results of stages A and B of the testing procedure shown in Figure 6.8 are summarized in Table 6.3 (detailed results are shown in Appendix D). For each inflow condition depicted in Table 6.3, the first and second rows are the results for scenarios 1 and 2 of stage-A testing, respectively, while the third rows contain the results for stage-B testing. The results for the parameter sets that resulted in the best overall performance (averaged over 30 simulations with different random number seeds) are also shown in Table 6.3. The various statistics presented were calculated using the OFC values obtained in the 30 trials with 30 different random starting positions in decision variable space. It should be noted that in this problem, maintenance schedules that violate load constraints (i.e. expected unserved energy (*EUE*) > 0) are considered to be infeasible).

Table 6.3: Results given by ACO-PPMSO for different inflow conditions investigated

In-flow	Local search	Avg. *EUE* (GWh)	Avg. *ETFEIS* (GWh)	Avg. $DurCut_{tot}$ (day)	Avg. *OFC* ($)	Avg. evalua-tion	Std dev. of *OFC*	Best parameter setting {m; ρ; p_{best}}*
	✗	131.06⁺	631.80	0	131,078	**76,700**	2,270	{800; 0.7; 0.3}
Dry	✗	0	542.35	34.1	22,679	84,987	546	{1000; 0.7; 0.01}
	✓	0	**543.50**	33.7	22,204	77,918	843	{50; 0.99; 0.3}
	✗	32.45⁺	2523.76	0	32,455	90,241	785	{500; 0.95; 0.3}
Int	✗	0	2527.77	29.9	3,525	83,614	336	{1000; 0.9; 0.05}
	✓	0	2531.65	27.1	3,115	51,784	213	{50; 0.7; 0.05}
	✗	0.00	4710.11	0	2.12	**68,731**	0.00	{500; 0.3; 0.3}
Wet	✗	0	4699.33	0	**2.12**	51,223	0.003	{100; 0.3; 0.5}
	✓	0	4713.45	0	2.12	65,935	0.001	{100; 0.3; 0.5}

⁺ Expected unserved energy (*EUE*) > 0 i.e. load constraints violated

Notation: *EUE*: Expected unserved energy, *ETFEIS*: Expected total final energy in storage, $DurCut_{tot}$: Total reduction in maintenance duration due to duration shortening and deferral of maintenance tasks; *OFC*: Objective function cost.

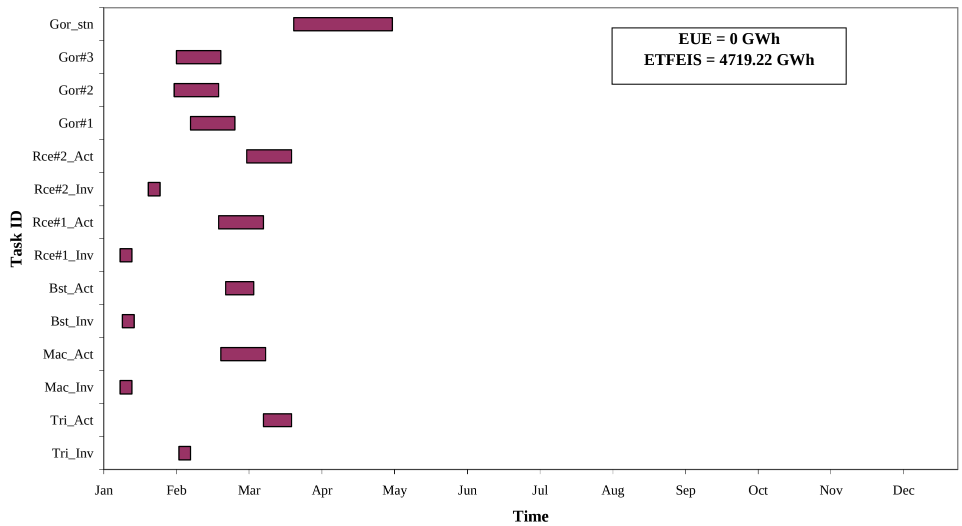
* m: number of ants; (1-ρ): pheromone evaporation rate; p_{best}: see Eq. 4.24.

Scenario 1: No shortening and deferral options

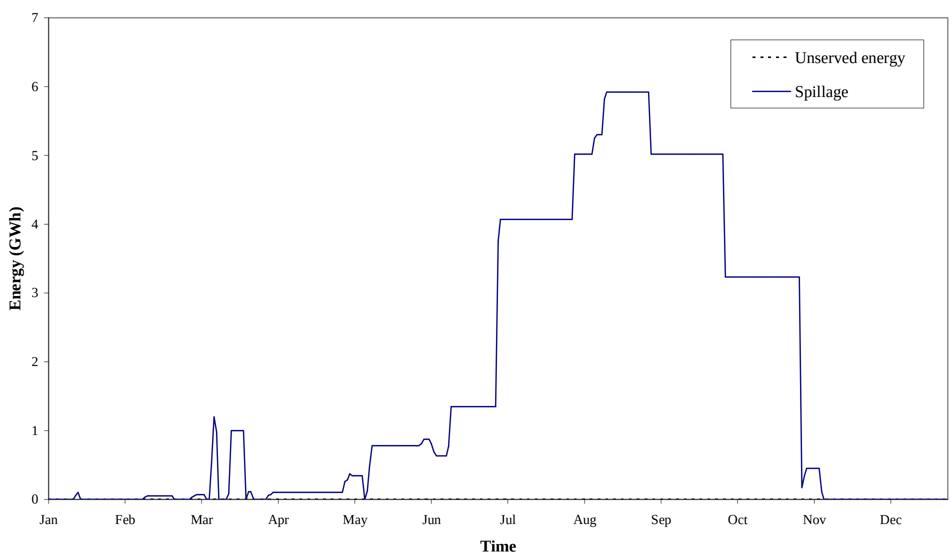
Stage-A results

It is shown in Table 6.3 that infeasible solutions (expected unserved energy > 0.002% of total annual energy demand) were obtained for the dry and intermediate inflow conditions. Further investigations have shown that, given the storage inflows in dry and intermediate years, it is not possible to meet demand constraints if all maintenance tasks are performed within the given planning horizon.

The best-found schedules obtained by ACO-PPMSO for the three inflow conditions investigated, as well as the associated unserved energy and spillage conditions, are shown in Figures 6.9 to 6.11. In order to better understand the optimisation process of the ACO-PPMSO algorithm, the objective function costs (IB-OFC), expected unserved energy (IB-EUE) and expected total final energy in storage (IB-ETFEIS) associated with the iteration-best schedules recorded throughout the runs that produced the schedules in Figures 6.9 to 6.11 are shown in Figures 6.12 to 6.14.

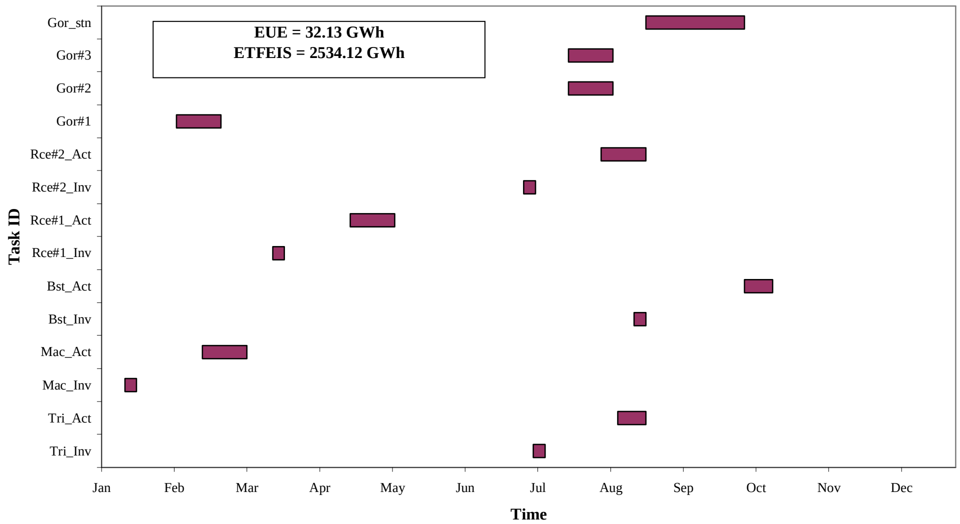


(a)

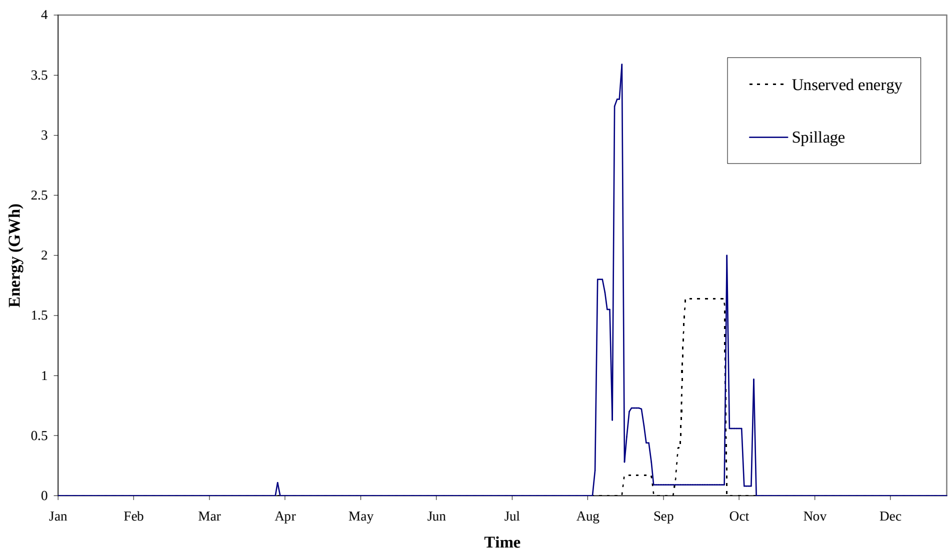


(b)

Figure 6.9: (a) The best-OFC schedule for wet inflow conditions and (b) the associated unserved energy and spillage conditions

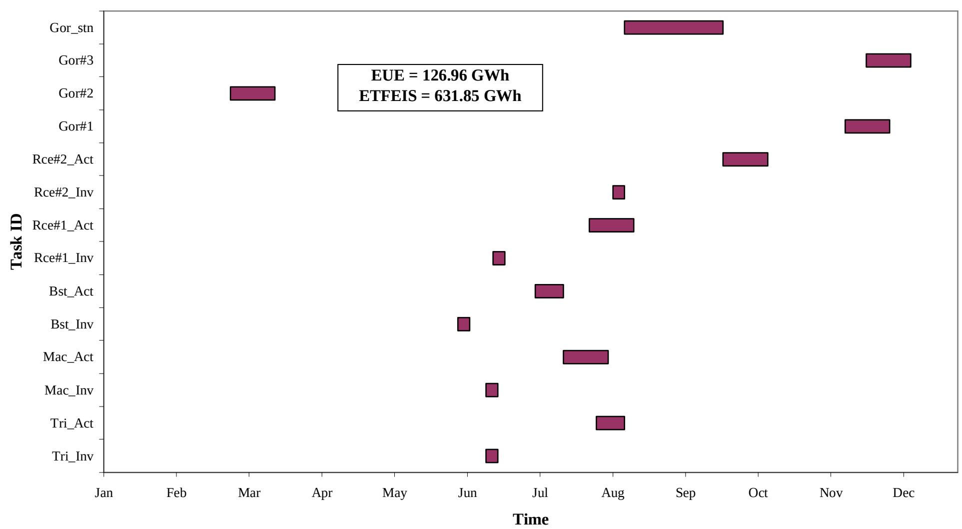


(a)

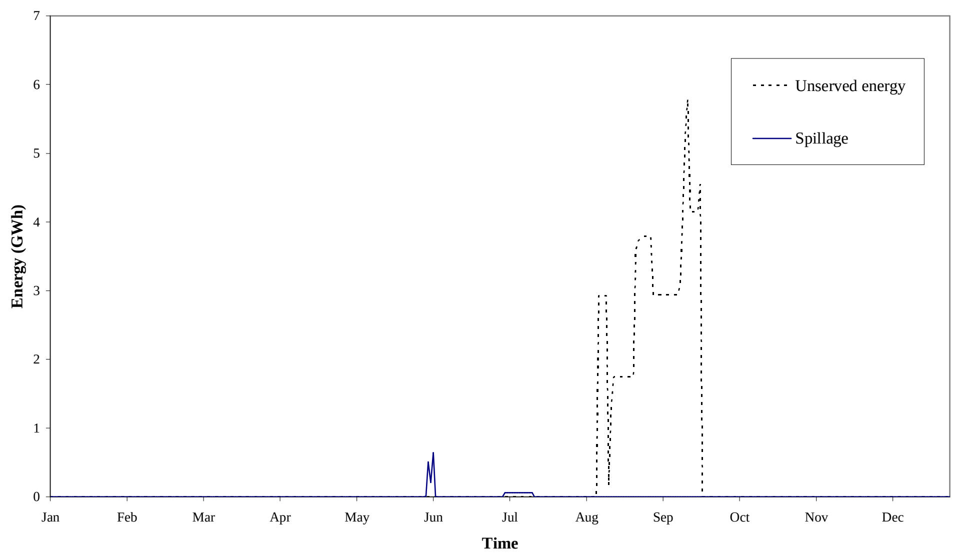


(b)

Figure 6.10: (a) The best-OFC schedule for intermediate inflow conditions and (b) the associated unserved energy and spillage conditions

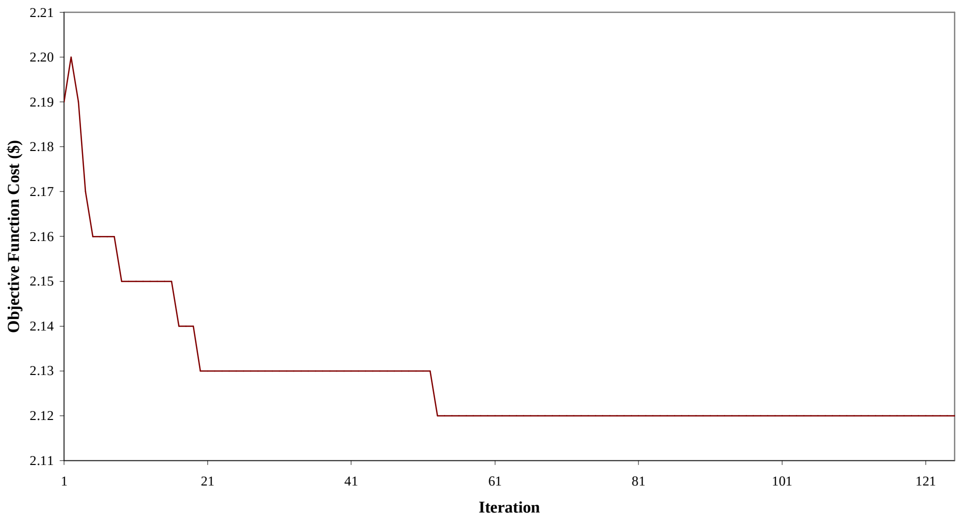


(a)

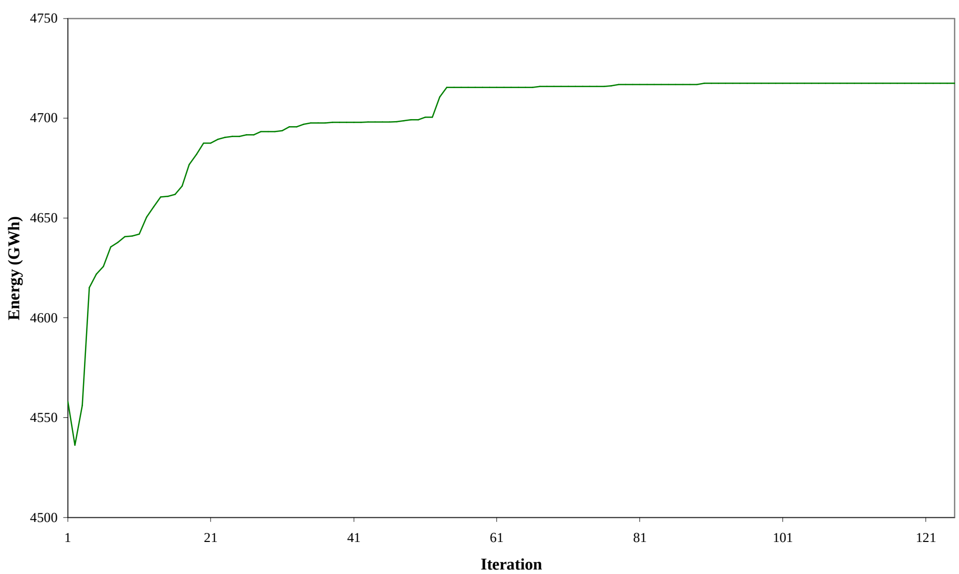


(b)

Figure 6.11: (a) The best-OFC schedule for dry inflow conditions and (b) the associated unserved energy and spillage conditions

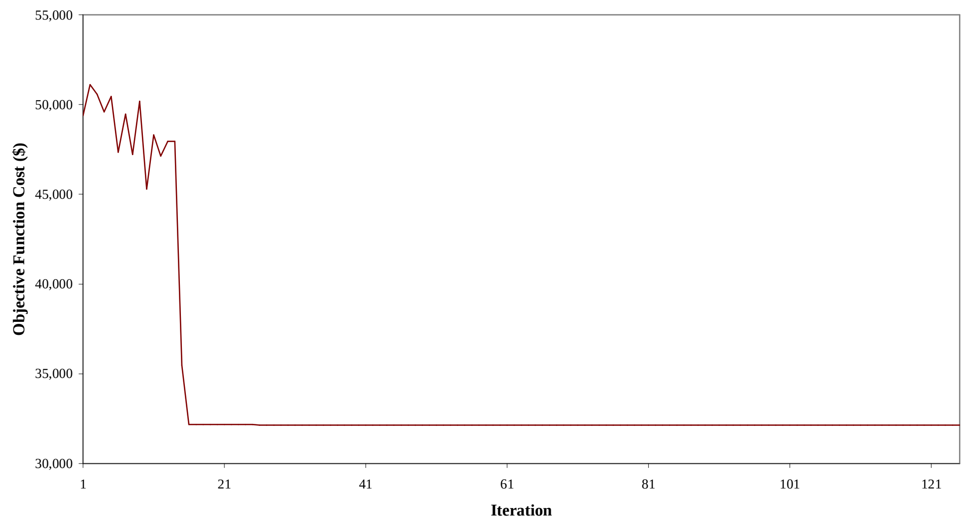


(a)



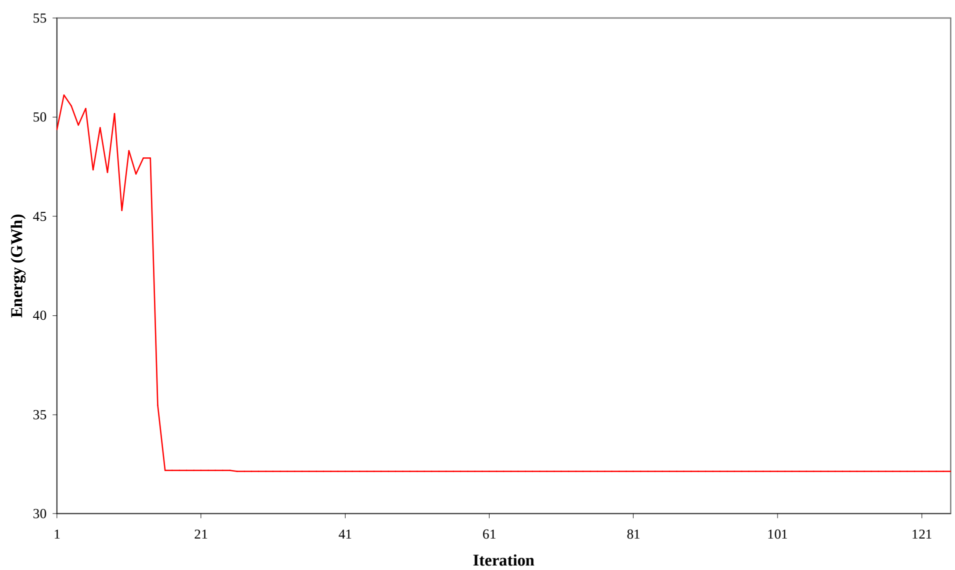
(b)

Figure 6.12: (a) IB-OFC and (b) IB-ETFEIS associated with iteration-best schedules recorded throughout the ACO run that produced the schedule in Figure 6.9 (wet inflow condition)



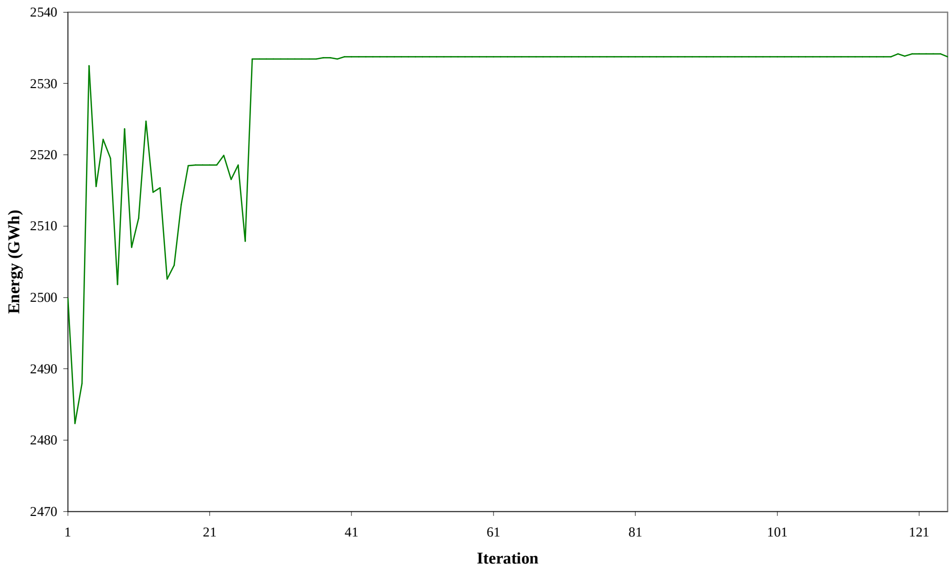
55,000
50,000
45,000
40,000
35,000
30,000
Objective Function Cost ($)
1
21
41
61
81
101
121
Iteration

(a)



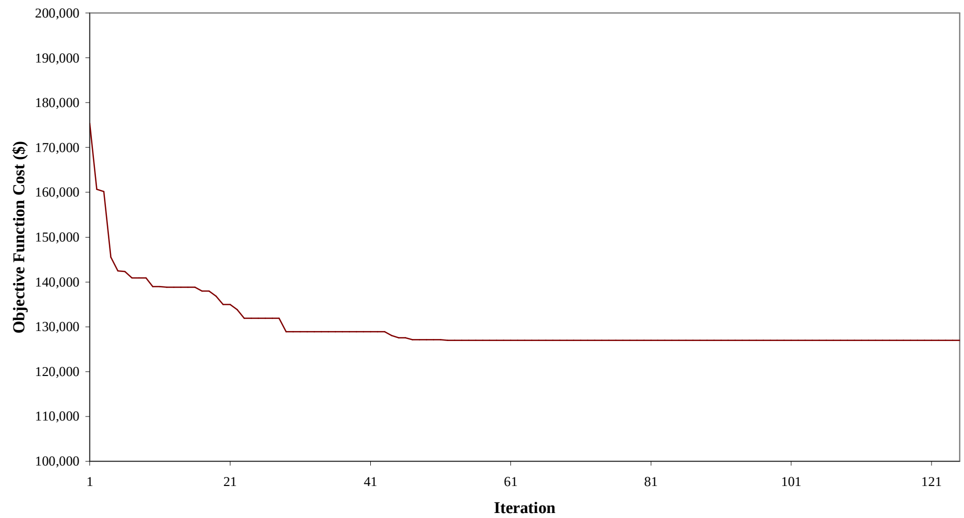
55
50
45
40
35
30
Energy (GWh)
1
21
41
61
81
101
121
Iteration

(b)

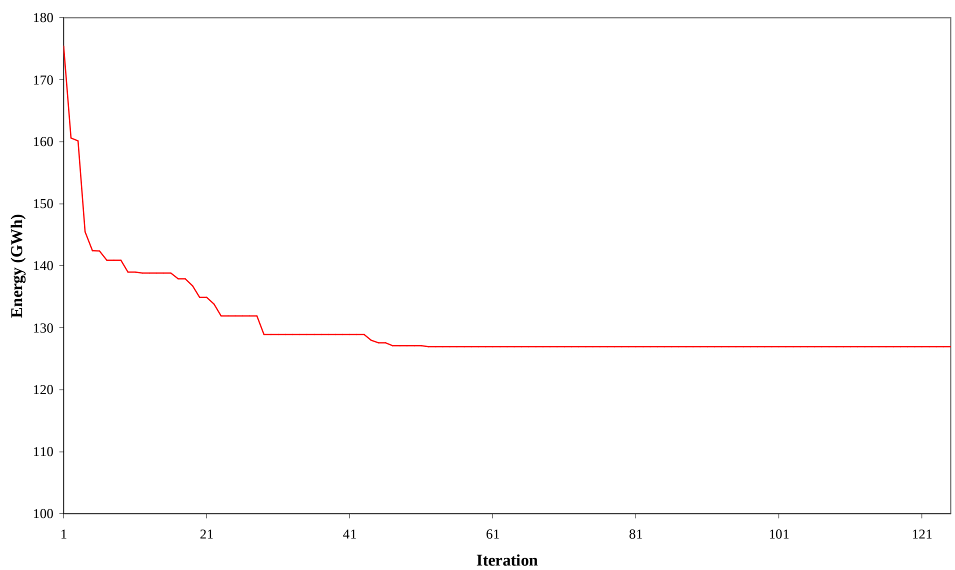


(c)

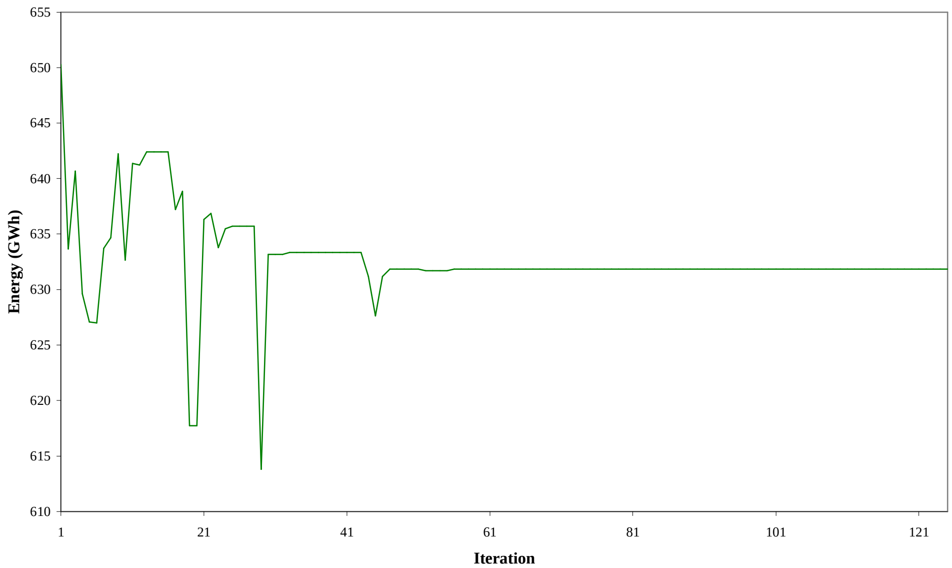
Figure 6.13: (a) IB-OFC, (b) IB-EUE and (c) IB-ETFEIS associated with iteration-best schedules recorded throughout the ACO run that produced the schedule in Figure 6.10 (intermediate inflow condition)


200,000
190,000
180,000
170,000
160,000
150,000
140,000
130,000
120,000
110,000
100,000
Objective Function Cost ($)
1
21
41
61
81
101
121
Iteration

(a)


180
170
160
150
140
130
120
110
100
Energy (GWh)
1
21
41
61
81
101
121
Iteration

(b)



(c)

Figure 6.14: (a) IB-OFC, (b) IB-EUE and (c) IB-ETFEIS associated with iteration-best schedules recorded throughout the ACO run that produced the schedule in Figure 6.11 (dry inflow condition)

The optimized schedules for each inflow condition (Figures 6.9 to 6.11) were examined in relation to the rationale of the optimisation outcome. It was found that, given wet inflow conditions, meeting energy demand is not difficult and the driving force behind the optimisation process is the maximization of total energy in storage of the system. This is clearly shown in the decreasing IB-OFC values (Figure 6.12a) as a result of increasing IB-ETFEIS (Figure 6.12b), while the IB-EUE curve is not presented, as IB-EUE = 0 throughout the ACO-PPMSO run.

It can be seen that during wet inflow conditions, maintenance tasks are scheduled for the early periods of the year, when the storage inflows are relatively lower and all storages are not full yet. The Gordon power station upgrade and the maintenance of its generators are performed during the low-demand, low-inflow periods (Jan to May) so that small storages can be emptied to cater for the inflows later in the year. In this way, the total final energy in storage of the system can be maximized.

On the other hand, as there were no feasible schedules for both dry and intermediate inflow conditions, the degree of load constraint violation was minimized in the optimisation runs for these scenarios. The down-sloping IB-OFC curves for both intermediate and dry inflow conditions (Figures 6.13a and 6.14a) correspond closely to the decreasing IB-EUE values shown in Figures 6.13b and 6.14b, indicating the minimization of expected unserved energy as the dominant objective in these optimisation runs. On the other hand, maximization of the ETFEIS objective for intermediate and dry inflow conditions is not apparent, as indicated by the fluctuating, and later rather stagnant, IB-ETFEIS values recorded during the corresponding runs.

It can be seen that in a dry inflow year, the run-of-river and Lake Mackintosh storages are available for power generation from January until June and are taken offline for maintenance from July to September (Figure 6.11a). The rationale behind this is that these smaller storages need to be emptied in summer (January to June) to be able to accommodate the much higher inflows in winter (July to September) without spilling when they are being maintained. In this way, these storages are full and able to operate at their maximum capacity when Gordon power station and its generators are being maintained in late September, which minimizes the total unserved energy over the planning horizon. For an intermediate inflow condition, the optimized maintenance schedule resembles that of the dry inflow condition, except that maintenance of the Gordon generators is performed before July, as the smaller storages are receiving sufficient inflows to meet the relatively low energy demand in that period.

Stage-C results

A schedule obtained by conventional techniques using engineering judgment (Stolp, S., personal communication, 2005) is shown in Figure 6.15. It can be seen that using traditional techniques, maintenance tasks for the units at Gordon power station and the upgrade of the station are scheduled during winter, assuming that run-of-river storages are receiving sufficient inflows to meet energy demands within that period. The values of objective function cost (OFC), expected total final energy in storage (ETFEIS) and expected unserved energy (EUE) associated with the schedule obtained by engineering judgment (Figure 6.15) and random evaluation are compared with those obtained by ACO-PPMSO (Figures 6.9 to 6.11) in Table 6.4. It should be noted that the ACO-PPMSO results presented correspond to the best results obtained from the 30 runs with different random number seeds, and are therefore slightly better than the results presented in Table 6.3. The detailed results obtained by random evaluation are shown in Appendix D2.

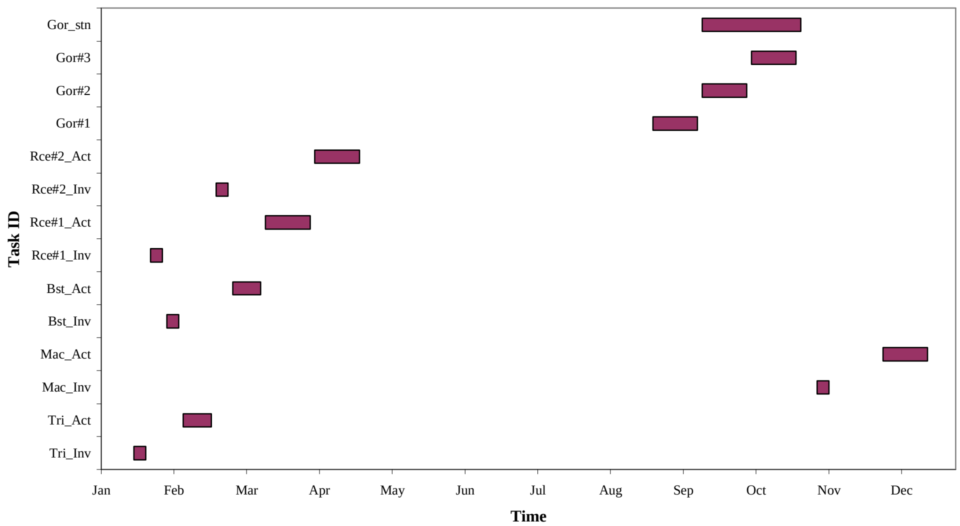


Figure 6.15: Schedule obtained by engineering judgment

Table 6.4: Comparison of results obtained by different methods

	Wet			Intermediate			Dry		
	EUE (GWh)	ETFEIS (GWh)	OFC ($)	EUE (GWh)	ETFEIS (GWh)	OFC ($)	EUE (GWh)	ETFEIS (GWh)	OFC ($)
Random evaluation	0.00	4,668.9	2.14	34.80	2,498.30	34,774.29	141.20	626.00	141,209
Engineering judgment	0.00	4,584.28	2.18	59.44	2,571.69	59,444.00	282.64	745.90	282,653
ACO-PPMSO	0.00	4,719.22	2.12	32.13	2,533.87	32,134.00	126.96	631.85	126,976

Notations: EUE: Expected unserved energy, ETFEIS: Expected total final energy in storage, OFC: Objective function cost.

It can be seen that the *OFC* associated with the best schedule obtained by ACO-PPMSO for each of the dry, intermediate and wet inflow conditions is lower than those obtained with engineering judgment and random evaluation (Table 6.4). However, it should be noted that the schedule obtained by conventional techniques was proposed based on the maintenance scheduler's experience on the full hydropower system, which might be different when applied to the simplified system considered in this study. In addition, the schedules obtained by ACO-PPMSO were the outcome of optimisation assuming perfect knowledge of inflow conditions. Nevertheless, the results obtained highlight the potential of using ACO for PPMSO in light of the context of this research. It can be seen that the schedules obtained by random evaluation are inferior to those obtained using ACO-PPMSO for all three inflow conditions. Based on the same inflow data and number of trial solutions evaluated, the results thus indicate that the ACO-PPMSO algorithm is useful in obtaining good solutions for maintenance scheduling problems.

Scenario 2: Options of shortening and deferral considered

Stage-A results

For dry and intermediate inflow conditions, it can be seen that the best-*OFC* maintenance schedules obtained are feasible (Average *EUE* = 0) when the durations of some maintenance tasks are shortened (Average $DurCut_{tot} > 0$) (last two rows of each inflow results in Table 6.3).

The best-*OFC* schedules for wet, intermediate and dry inflow conditions are presented in Figures 6.16 to 6.18. The rationale behind these schedules was analysed, by taking into account storage inflows and system demand, as well as the rules implemented in the simulation model (SYSOP) with regard to the priorities of power stations being called for generation.

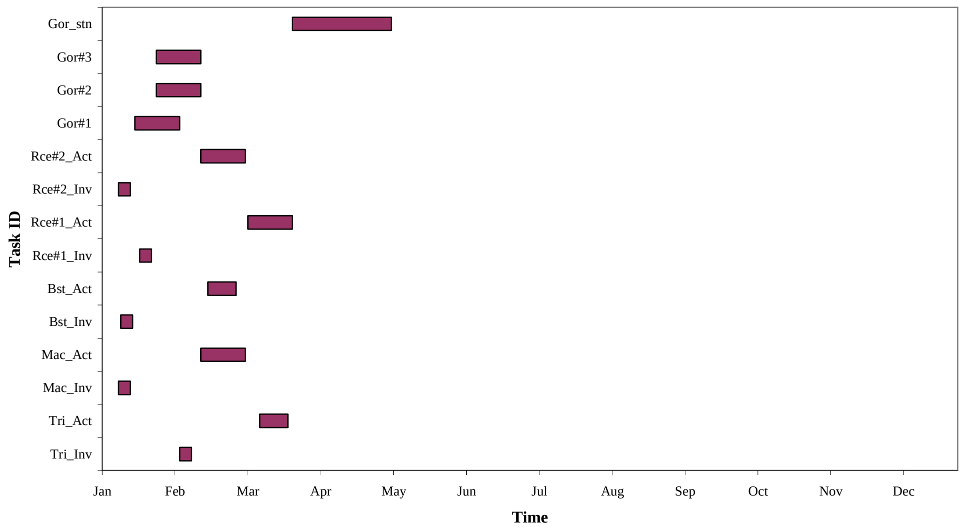


Figure 6.16: The best-*OFC* schedule for the wet year (EUE = 0 GWh; ETFEIS = 4718 GWh)

Figure 6.17: The best-OFC schedule for the intermediate year (EUE = 0 GWh; ETFEIS = 2539 GWh)

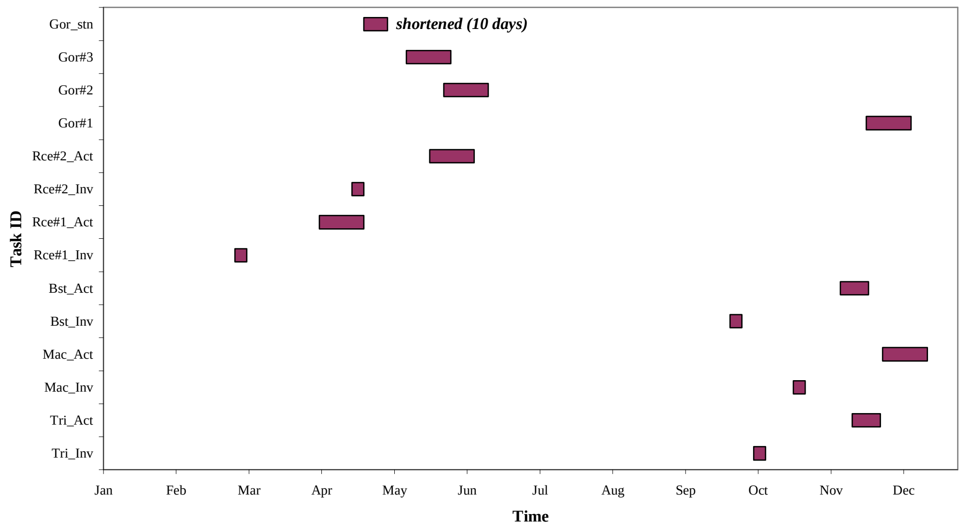
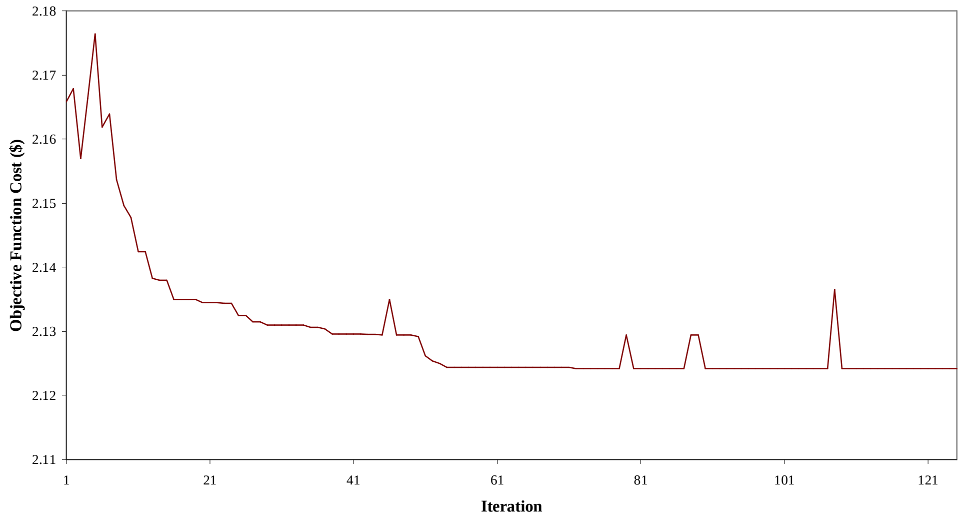
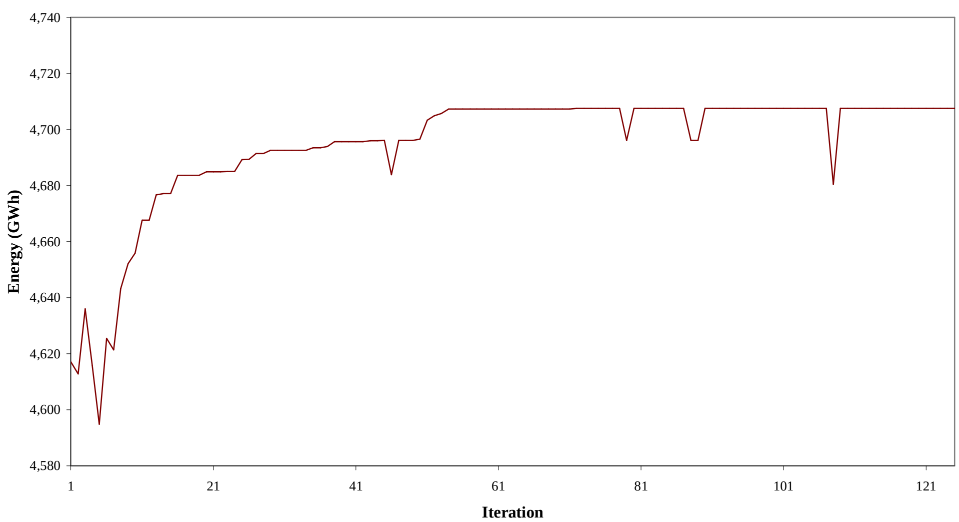


Figure 6.18: The best-OFC schedule for the dry year (EUE = 0 GWh; ETFEIS = 544 GWh)

For the wet inflow condition (Figure 6.16), neither duration shortening nor deferral of maintenance tasks is required, as demand constraints are easily satisfied. In addition, it can be seen that all maintenance tasks are scheduled in the first quarter of the planning horizon. All storages are 75% full at the start of the planning horizon, and are still able to accommodate inflows during maintenance. By winter, when storage inflows are even higher, run-of-river storages are almost full, if not spilling, and are able to provide the relatively high demand in this period without having to draw down major storages (Lakes Mackintosh & Gordon). In this way, generation from major storages is minimized and the expected total energy-in-storage is maximized. It can be seen that the iteration-best objective function cost (IB-OFC) decreases (Figure 6.19a) as a result of the increasing expected total final energy in storage (IB-ETFEIS) (Figure 6.15b) throughout the ACO run. It should be noted that none of the iteration-best schedules violate the demand constraints (ie. IB-EUE = 0 for all iterations).



(a)

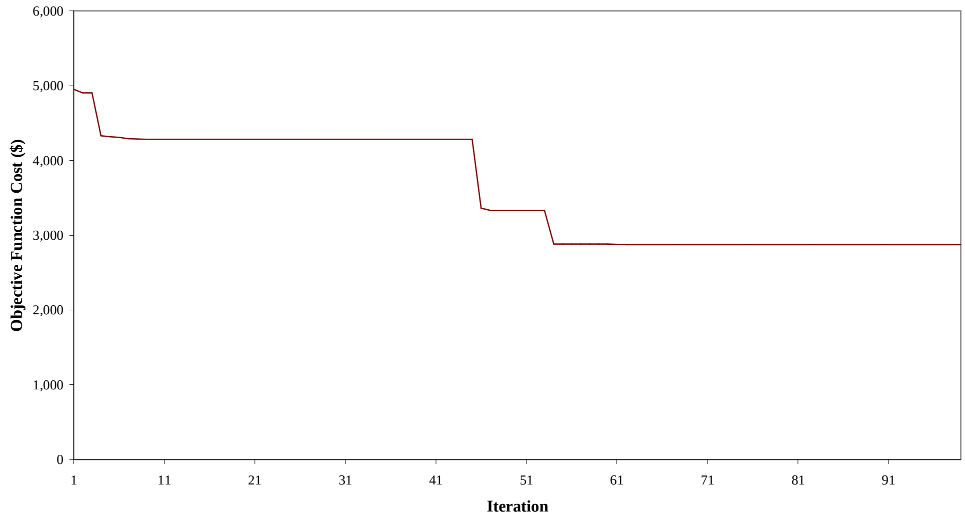


(b)

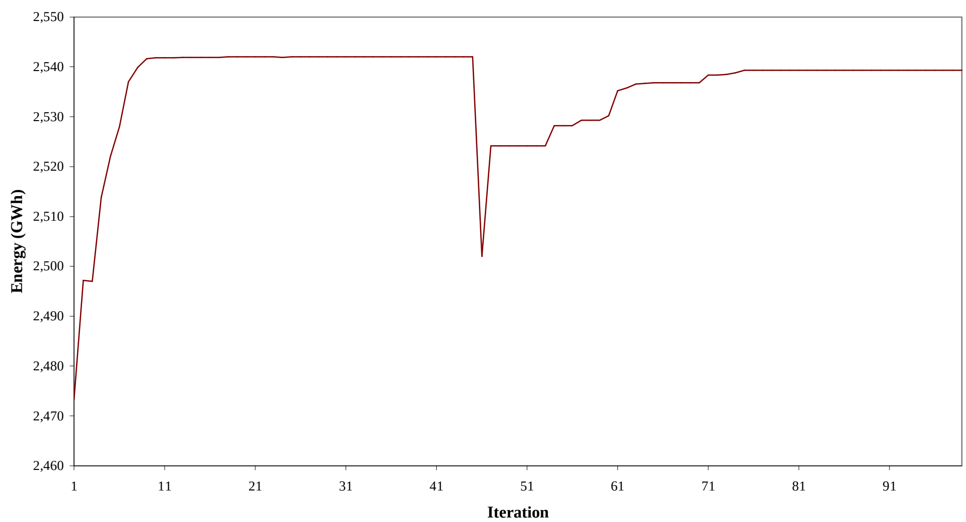
Figure 6.19: (a) IB-OFC and (b) IB-ETFEIS associated with iteration-best schedules recorded throughout the ACO run that produced the maintenance schedule shown in Figure 6.16 (wet inflow condition)

For the intermediate inflow condition (Figure 6.17), the Gordon station upgrade task, which normally takes 42 days to complete, had to be shortened by 61.9% in order to satisfy demand constraints. In addition, most of the maintenance tasks are not scheduled in the period from April to August. This is because although the highest storage inflows take place in August, run-of-river storages are still incapable of meeting winter demands (May to August, Figure 6.3), therefore requiring the major storages for generation. Only when the demand is relatively lower in September and the storage inflows are still quite high, Gordon station is taken offline for maintenance. However, as the run-of-river storage levels decrease rapidly as a result of the loss of Gordon, Gordon station had to be brought back on-line to avoid demand shortfalls. The schedules obtained also indicated that the maintenance tasks for Mackintosh, Gordon#2 and Gordon#3 machines are scheduled in a way such that Lake Mackintosh is emptied before its maintenance to reduce spilling.

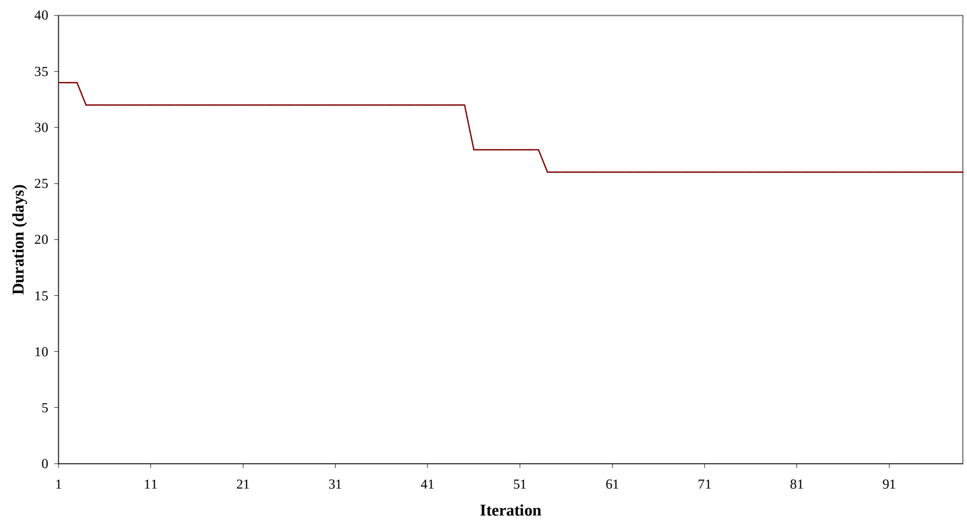
The ACO optimisation process that produced the Figure 6.17 maintenance schedule is shown in Figure 6.20. It can be seen that the IB-OFC curve (Figure 6.20a) decreases in stages, mainly corresponding to the reduction in total outage duration shortened/deferred (Figure 6.20c). Figure 6.20c also illustrates that when a new minimum IB-DurCut$_{tot}$ is found (eg. Iteration 2, 43 and 52), IB-ETFEIS undergoes maximization (Figure 6.20b). As IB-EUE = 0 throughout the run, it can be deduced that the optimisation process for the intermediate inflow condition was driven primarily by the total duration shortened/deferred and secondarily by the total final energy in storage.


6,000
5,000
4,000
3,000
2,000
1,000
0
Objective Function Cost ($)
1
11
21
31
41
51
61
71
81
91
Iteration

(a)


2,550
2,540
2,530
2,520
2,510
2,500
2,490
2,480
2,470
2,460
Energy (GWh)
1
11
21
31
41
51
61
71
81
91
Iteration

(b)

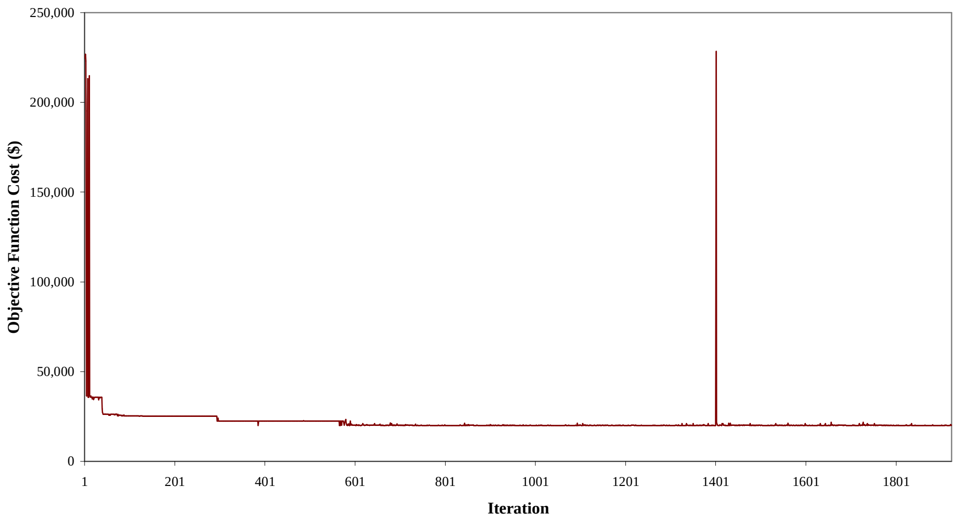


(c)

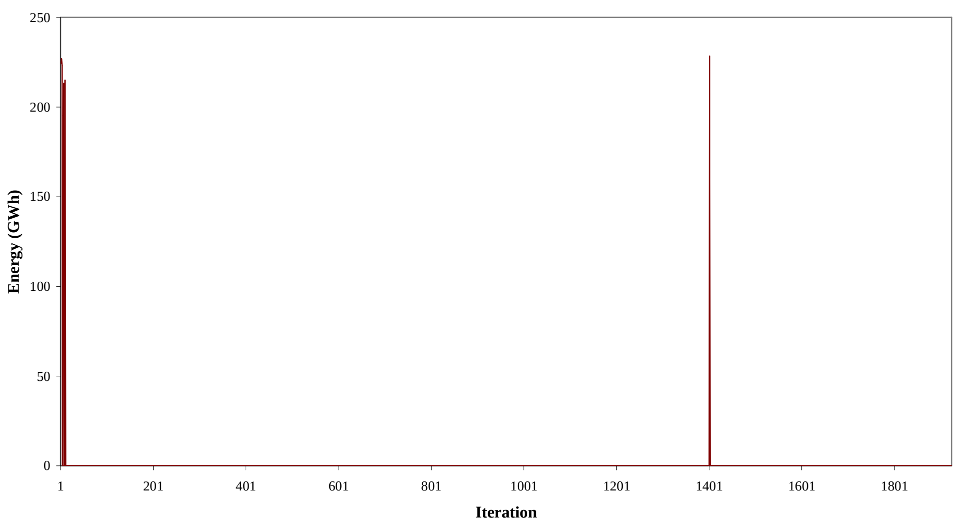
Figure 6.20: (a) IB-OFC, (b) IB-ETFEIS and (c) IB-DurCuttot associated with iteration-best schedules recorded throughout the ACO run that produced the maintenance schedule shown in Figure 6.17 (intermediate inflow condition)

Compared to the intermediate inflow condition, the duration of the Gordon station upgrade task is shortened even more (by 76%) for the dry inflow condition (Figure 6.18). This is as expected, as the expected unserved energy during dry conditions is higher than that during intermediate inflows. Similar to the intermediate inflow condition, all maintenance tasks are not scheduled in winter (May-September, Figure 6.3) when demand is the highest in a low-inflow year. Specifically, as inflows are exceptionally low in the Jan-Mar period (Figure 6.3), all storages are used to meet demand. Only in April, when storage inflows start to increase, are run-of-river storages fully relied on for meeting demand while the shortened upgrade task of Gordon station is carried out. In addition, the last quarter of the planning horizon is deemed to be the best period for maintaining the run-of-river stations, as these storages are already running quite low at that time.

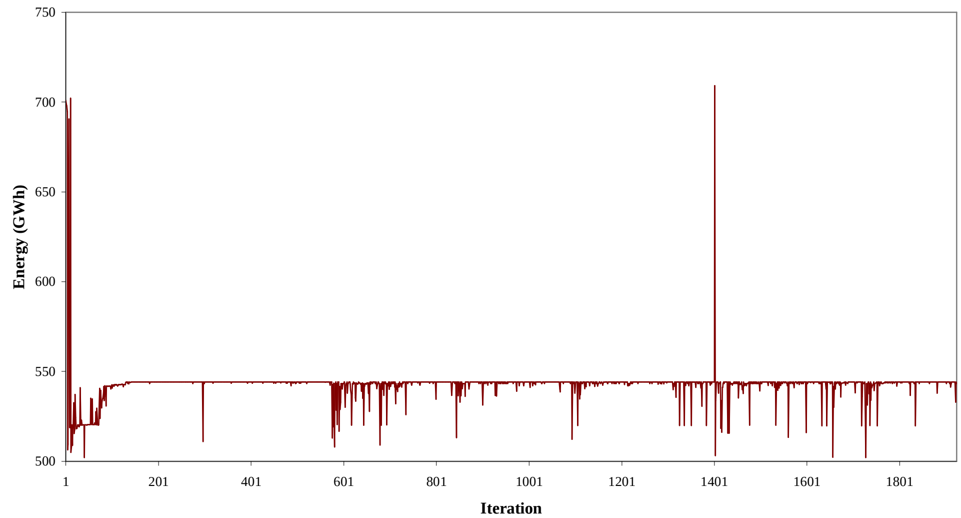
The optimisation process for the dry inflow condition (Figure 6.21) is similar to that for the intermediate inflow condition, except that IB-EUE > 0 for the first 11 iterations of the run that caused the high IB-OFCs at the beginning of the run (Figures 6.21a and b).


250,000
200,000
150,000
100,000
50,000
0
Objective Function Cost ($)
1
201
401
601
801
1001
1201
1401
1601
1801
Iteration

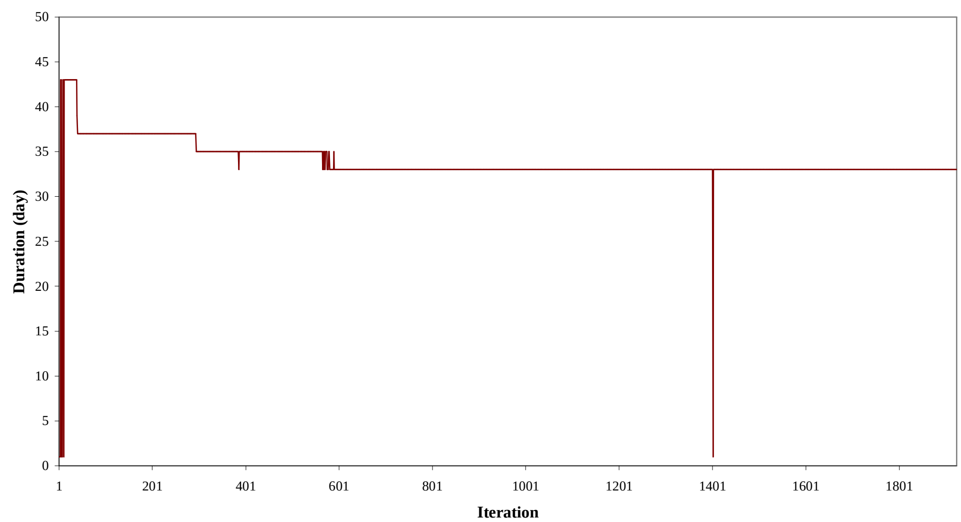
(a)


250
200
150
100
50
0
Energy (GWh)
1
201
401
601
801
1001
1201
1401
1601
1801
Iteration

(b)



(c)



(d)

Figure 6.21: (a) IB-OFC, (b) IB-EUE, (c) IB-ETFEIS and (d) IB-DurCuttot associated with iteration-best schedules recorded throughout the ACO run that produced the maintenance schedule shown in Figure 6.18 (dry inflow condition)

Stage-B results

The usefulness of the *Duration Extender* local search operator is shown to be statistically significant (p-value < 0.01) for both dry and intermediate inflow conditions when checked with an unpaired, 2-sided Student's t-test (calculations are shown in Appendix D5). The improvement in Average OFC when local search is used is mainly attributed to the reduction of total duration shortened and deferred (last row of each inflow result in Table 6.3). However, it should be noted that the local search strategy is only performed for iteration-best trial schedules that contain one or more decisions of shortening. Therefore, the local search was of little use, if any, during the optimisation for wet inflow conditions, as demand constraints are well satisfied in that scenario without the need for shortening and deferral of maintenance tasks.

Stage-C results

The best maintenance schedules obtained by ACO-PPMSO for the three inflow conditions are compared with those found by the random evaluation method. The detailed results obtained by random evaluation are shown in Appendix D4. Table 6.5 depicts that the results of ACO-PPMSO are superior as indicated by the much lower objective function costs for all inflow conditions. In addition, as a result of having the shortening and deferral options available, there were no demand shortfalls during the intermediate and dry inflow conditions. It should be noted that comparison with a practitioner's schedule is not made as maintenance tasks are not deferred or shortened as part of current Hydro Tasmania practice.

Table 6.5: Results obtained by different methods

Inflow	Method	EUE (GWh)	ETFEIS (GWh)	$DurCut_{tot}$ (day)	OFC ($)
Dry	Random	0.00	520.30	44	38921.37
	ACO-PPMSO	0.00	544.20	32	20011.77
Intermediate	Random	0.00	2489.30	42	7427.71
	ACO-PPMSO	0.00	2539.30	26	2870.85
Wet	Random	0.00	4562.60	2	19.73
	ACO-PPMSO	0.00	4718.37	0	**2.12**

Notation: EUE: Expected unserved energy, ETFEIS: Expected total final energy in storage; $DurCut_{tot}$: Total reduction in maintenance duration due to shortening and deferral; OFC: Objective function cost.

6.2.5 Summary

A testing procedure was carried out on a five-station hydropower maintenance scheduling case study. It was shown that the shortening and deferral options of maintenance tasks allow PPMSO problem to be solved practically, especially when not all maintenance tasks can be performed under unfavourable system conditions. In addition, the *Duration Extender* local search operator was shown to be statistically significant in improving the performance of ACO-PPMSO when shortening and deferral options are considered. Comparison with maintenance schedules obtained by other methods, including a random evaluation method and that based on conventional techniques and engineering judgments of maintenance schedulers, indicated that ACO-PPMSO is a competitive optimisation method for real-world PPMSO problems.

It should be noted that the five-station hydro PPMSO case study provided a platform for the development and refinement of the ACO-PPMSO formulation in this research. In particular, the availability of shortening and deferral options as part of the ACO-PPMSO formulation is an outcome of extensive analysis carried out on different scenarios and numerous discussions with Hydro Tasmania maintenance schedulers. In addition, the constraints handling methods used in ACO-PPMSO were tested extensively and undergone repeated modifications throughout the investigation of this case study.

6.3 Full Hydro Tasmania Maintenance Scheduling Case Study

6.3.1 Problem specification

The full Hydro Tasmania case system considers an integrated system of 38 storages (including run-of-river and major storages), 28 hydropower stations, 55 generating units and a total of 118 maintenance tasks to be performed over a planning horizon of 365 days from Jan 1, 2006. The maintenance scheduling problem aims to find a start date for each of the 109 maintenance tasks shown in Table 6.7, while the commencement date of nine other maintenance tasks have been fixed (Table 6.6) (Stolp, S., personal communication, 2006). A maintenance schedule is sought such that expected total final energy in storage (ETFEIS) of the hydropower system is maximized, thermal generation (THERM) is minimized and total reduction in maintenance duration due to shortening and deferral ($DurCut_{tot}$) is minimized, subject to the following constraints (Stolp, S., personal communication, 2006):

1. Demand constraints: Forecasted system demand must be met (i.e. Expected unserved energy = 0) under specified inflow conditions.

2. Reliability constraints: Reserve capacity = 30% of system demand at all times. However, violation of the constraints by a maximum of 2 days can be tolerated.

3. Timeframe window constraints: Timeframe window constraints of individual maintenance tasks are presented under the 'Earliest start date' and 'Latest finish date' headings in Table 6.7.

4. Precedence constraints: Precedence constraints are presented under the 'Other constraints' heading in Table 6.7.

5. Completion constraints: All maintenance tasks must be completed within the planning horizon.

As part of constraint 1, the forecasted system demand, as well as inflow conditions, are given beforehand. Based on a forecasted average demand of 1193.3 MW, hourly system demand is calculated by SYSOP using a series of load shape functions (not shown due to confidentiality requirements). Due to space limitations, the daily average demand is presented in this thesis (Figure 6.22). A total of 77 historical system inflows (1924~2000) are stored in SYSOP to assess the performance of a given maintenance schedule. Although daily inflows are used in SYSOP, the monthly average inflows from 2001 to 2005 are shown in Figure 6.23 for illustration. The total storage inflows are the highest from June to October during winter in Tasmania (Figure 6.23). The water levels of storages in the system at the start of the planning horizon (1 Jan 2006) are given in Table 6.8.

Table 6.6: Fixed-date maintenance tasks

Station	Task id	Number of machines involved	Machine #	Fixed maintenance commencement date	Outage duration (days)
Cluny	33613	1	1	13-Feb-06	1
Gordon	30630	1	2	1-Jan-06	67
Gordon	33485	3	1 2 3	14-Jan-06	1
Margaret	33543	7	1 2 3 4 5 6 7	9-Jan-06	2
Meadowbank	33528	1	1	1-Jan-06	5
Paloona	33611	1	1	30-Jan-06	1
Reece	33530	1	2	1-Jan-06	6
Rowallan	33539	1	1	19-Jan-06	1

Table 6.7: Maintenance tasks that need to be scheduled

Station	Number of tasks in the group	Task id	Machine #	Earliest start date	Latest finish date	Other constraints	Optional outage duration (days)
Bastyan	1	30555	1	18-Mar-06	18-Jun-06		3,0
Bastyan	1	30556	1	21-Apr-06	24-Jul-06		5,3,0
ButlersG	1	32095	1	1-Jan-06	31-Dec-06		16,14,12,10,8,0
ButlersG	1	32941	1	21-Apr-06	22-May-06		12,10,8,6,0
ButlersG	1	32111	1	23-Jun-06	31-Dec-06		5,3,0
Catagunya	1	30139	2	1-Jan-06	10-Jul-06		5,3,0
Catagunya	1	33628	1	1-Jan-06	31-Dec-06		2,0
Catagunya	1	33498	1	15-Jan-06	4-Feb-06		1,0
Catagunya	1	33578	2	5-Feb-06	26-Feb-06		2,0
Cethana	1	33462	1	1-Jan-06	31-Dec-06		3,0
Cethana	1	33468	1	1-Jan-06	27-Jan-06		1,0

Table 6.7: Maintenance tasks that need to be scheduled (cont)

Station	Number of tasks in the group	Task id	Machine #	Earliest start date	Latest finish date	Other constraints	Optional outage duration (days)
Cluny	1	33499	1	13-Jan-06	2-Feb-06		1,0
Devils_Gate	1	33463	1	1-Jan-06	31-Dec-06		3,0
Devils_Gate	1	33465	1	1-Jan-06	27-Jan-06		1,0
Echo	1	32970	1	1-Jan-06	31-Dec-06		19,17,15,13,11,0
Echo	1	33223	1	1-Jan-06	31-Dec-06		1,0
Echo	1	33548	1	1-Jan-06	31-Dec-06		1,0
Fisher	1	32460	1	1-Jan-06	10-Jul-06		11,9,7,0
Fisher	1	33459	1	1-Jan-06	31-Dec-06		3,0
Fisher	1	33470	1	1-Jan-06	22-Jan-06		2,0
Fisher	1	33605	1	1-Jan-06	31-Dec-06		1,0
Gordon	1	30072	1	1-Jan-06	31-Dec-06		213,200,190,180,170,160,150,140,130,120,110,0
Gordon	1	30773	1, 2, 3	1-Jan-06	31-Dec-06		22,20,18,16,14,12,0
Gordon	4	33601	1, 2, 3	1-Jan-06	31-Dec-06	At least two months apart from 30998,30999& 31002	3,0
		31002	2, 3	1-Jan-06	31-Dec-06	At least two months apart from 30998,30999& 33601	3,0
		30998	2, 3	1-Jan-06	31-Dec-06	At least two months apart from 30999,31002& 33601	3,0
		30999	1, 2, 3	1-Jan-06	31-Dec-06	At least two months apart from 30998,31002& 33601	3,0
Gordon	2	33456	1	1-Jan-06	29-Jan-06	Following 33456	2,0
		33457	3	1-Jan-06	29-Jan-06		1,0
JButters	1	33511	1	26-Jan-06	9-Feb-06		1,0
Lemonthyme	1	30553	1	1-Jan-06	19-Jul-06		14,12,10,8,0
Lemonthyme	1	33461	1	1-Jan-06	31-Dec-06		4,2,0
Lemonthyme	1	33467	1	1-Jan-06	27-Jan-06		1,0
Liapootah	2	32279	1	20-Jan-06	24-Apr-06	Precedes 33592	5,3,0
		33592	1	25-Apr-06	25-Jul-06		26,24,22,20,18,16,14,0

Table 6.7: Maintenance tasks that need to be scheduled (cont)

Station	Number of tasks in the group	Task id	Machine #	Earliest start date	Latest finish date	Other constraints	Optional outage duration (days)
Liapootah	3	33625	1	1-Jan-06	31-Dec-06		1,0
		33626	2	1-Jan-06	31-Dec-06	Following 33625	1,0
		33627	3	1-Jan-06	31-Dec-06	Following 33626	1,0
Liapootah	2	33480	3	1-Jan-06	27-Jan-06		1,0
		33481	1	1-Jan-06	27-Jan-06	Following 33480	1,0
Liapootah	1	33551	1	1-Jan-06	27-Jan-06		1,0
Liapootah	1	33579	2	7-Feb-06	27-Feb-06		1,0
Mackintosh	1	33474	1	1-Jan-06	28-Jan-06		1,0
Meadowbank	1	30134	1	1-Jan-06	11-Jul-06		12,10,8,6,0
Meadowbank	1	33505	1	1-Jan-06	27-Jan-06		1,0
Meadowbank	1	33600	1	13-Jan-06	27-Jan-06		1,0
Paloona	1	30560	1	1-Jan-06	31-Dec-06		26,24,22,20,18,16,14,0
Paloona	1	30558	1	1-Jan-06	31-Dec-06		5,3,0
Paloona	1	30557	1	1-Jan-06	31-Dec-06		3,0
Paloona	1	33464	1	1-Jan-06	31-Dec-06		3,0
Paloona	1	33466	1	1-Jan-06	27-Jan-06		1,0
Poatina	1	32512	2	7-Feb-06	28-Sep-06		54,50,46,42,38,34,30,0
Poatina	1	30185	6	20-May-06	10-Dec-06		26,24,22,20,18,16,14,0
Poatina	1	31446	3	1-Jan-06	31-Dec-06		54,50,46,42,38,34,30,0
Poatina	1	33473	2	1-Jan-06	31-Dec-06		16,14,12,10,8,0
Poatina	1	33559	1	20-Jan-06	12-Feb-06		4,2,0
Poatina	1	33285	4	1-Jan-06	29-Jan-06		3,0
Poatina	1	33286	5	1-Jan-06	28-Jan-06		2,0
Poatina	1	33612	3	5-Feb-06	26-Feb-06		2,0
Reece	1	33555	1	1-Jan-06	31-Jan-06		1,0
Repulse	1	32967	1	1-Jan-06	20-May-06		21,19,17,15,13,11,0
Repulse	1	33614	1	1-Jan-06	14-Jan-06		1,0
Repulse	1	33563	1	2-Feb-06	16-Feb-06		1,0
Rowallan	1	33469	1	1-Jan-06	27-Jan-06		1,0
Rowallan	1	33460	1	1-Jan-06	31-Dec-06		3,0

Table 6.7: Maintenance tasks that need to be scheduled (cont)

Station	Number of tasks in the group	Task id	Machine #	Earliest start date	Latest finish date	Other constraints	Optional outage duration (days)
Tarraleah	2	30792	1, 2, 3, 4	1-Jan-06	31-Dec-06	at least 10 months apart from 30793	5,3,0
		30793	1, 2, 3, 4	1-Jan-06	31-Dec-06	at least 10 months apart from 30792	6,4,0
Tarraleah	1	30076	5	6-Mar-06	11-Dec-06		102,100,90,80,70,60, 50,0
Tarraleah	5	33444	1, 2	1-Jan-06	31-Jan-06		1,0
		33446	3	1-Jan-06	31-Jan-06	following 33444	1,0
		33450	6	1-Jan-06	28-Jan-06		2,0
		33447	2	1-Jan-06	28-Jan-06	following 33450	1,0
		33448	3	1-Jan-06	28-Jan-06	following 33447	1,0
Tarraleah	2	33449	5	1-Jan-06	27-Jan-06		1,0
		32981	1	1-Jan-06	27-Jan-06	Following 33449	1,0
Tarraleah	1	32939	2	1-Jan-06	31-Dec-06		4,2,0
Tarraleah	1	32940	1	1-Jan-06	31-Dec-06		5,3,0
Tarraleah	1	33550	6	1-Jan-06	21-Jan-06		1,0
Tarraleah	1	33598	6	22-Jan-06	11-Feb-06		1,0
Tarraleah	1	33599	4	23-Jan-06	12-Feb-06		1,0
Trevallyn	1	33204	4	1-Jan-06	31-Dec-06		30,28,26,24,22,20,18 ,16,0
Trevallyn	1	30179	1	1-Jan-06	31-Dec-06		29,27,25,23,21,19,17 ,15,0
Trevallyn	1	32799	1	1-Jan-06	31-Dec-06		19,17,15,13,11,0
Trevallyn	1	33609	4	1-Jan-06	31-Dec-06		2,0
Trevallyn	1	33617	3	1-Jan-06	1-Mar-06		2,0
Trevallyn	1	33624	4	1-Jan-06	31-Dec-06		1,0
Tribute	1	30552	1	1-Jan-06	31-Dec-06		27,25,23,21,19,17,15 ,0
Tribute	1	33526	1	1-Jan-06	31-Jan-06		1,0
Tungatinah	1	30523	1, 2, 3, 4, 5	1-Jan-06	31-Dec-06		44,42,40,38,36,34,32 ,30,28,26,24,22
Tungatinah	1	30518	1	1-Jan-06	31-Dec-06		28,24,20,16,14,0
Tungatinah	1	30528	1	1-Jan-06	31-Dec-06		91,80,70,60,50,40,0
Tungatinah	1	30527	2	1-Jan-06	31-Dec-06		90,80,70,60,50,40,0
Tungatinah	1	30961	3	1-Jan-06	31-Dec-06		126,120,110,100,90, 80,70,60,0
Tungatinah	1	30519	2	1-Jan-06	31-Dec-06		92,80,70,60,50,40,0

Table 6.7: Maintenance tasks that need to be scheduled (cont)

Station	Number of tasks in the group	Task id	Machine #	Earliest start date	Latest finish date	Other constraints	Optional outage duration (days)
Tungatinah	1	30962	4	1-Jan-06	31-Dec-06		61,50,40,30,0
Tungatinah	1	33081	5	9-May-06	9-Nov-06		5,3,0
Tungatinah	1	33079	2	23-Jun-06	31-Dec-06		5,3,0
Tungatinah	1	33452	3	1-Jan-06	27-Jan-06		1,0
Tungatinah	1	33451	5	10-Jan-06	24-Jan-06		1,0
Tungatinah	1	33573	1	27-Jan-06	10-Feb-06		1,0
Tungatinah	1	33597	4	20-Feb-06	6-Mar-06		1,0
Wayatinah	1	30297	2	1-Jan-06	10-Jul-06		5,3,0
Wayatinah	1	30298	3	10-Jan-06	13-Jul-06		5,3,0
Wayatinah	1	33482	2	1-Jan-06	27-Jan-06		1,0
Wayatinah	1	33522	2	1-Jan-06	31-Dec-06		1,0
Wayatinah	1	33564	1	29-Jan-06	18-Feb-06		1,0
Wayatinah	1	33581	3	13-Feb-06	5-Mar-06		1,0
Wilmot	1	30133	1	1-Jan-06	31-Dec-06		76,72,68,64,60,56,52,48,44,40,0
Wilmot	1	33569	1	1-Jan-06	31-Dec-06		1,0

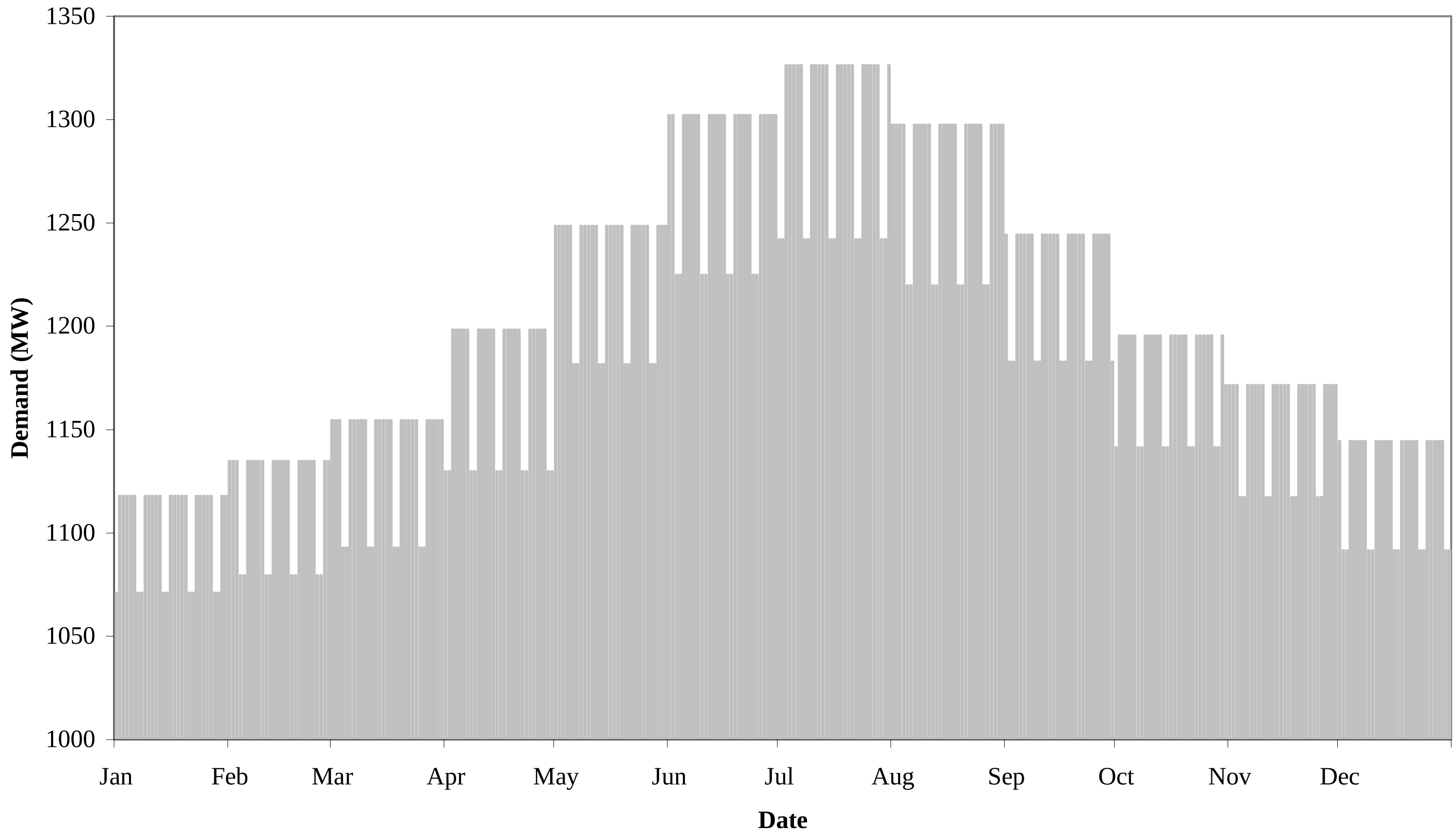


Figure 6.22: Forecasted 2006 daily average demand for Hydro Tasmania case study

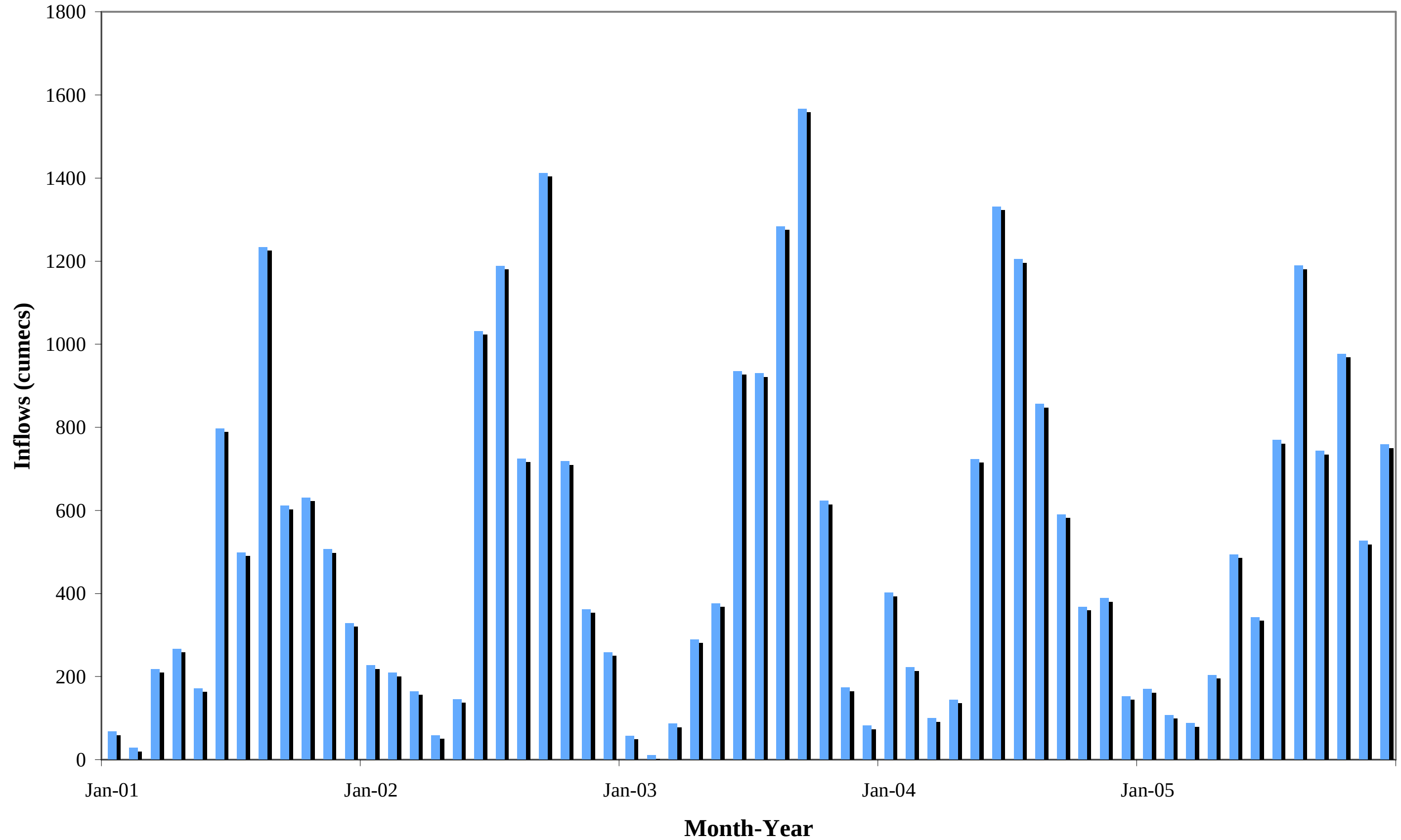


Figure 6.23: Historical total system inflows for 2001 ~ 2005 (monthly average)

Table 6.8: Storage levels on 1 Jan 2006

Storage	cumec.days
L_STCLAIR	301.5
L_KWILLIAM	6022.5
DERWENT_PUMPS_STORE	0
MOSSY_MARSH	0
L_ECHO	2919.8
DEE_LAG	4.2
BRONTE_LAG	137.9
BRADYS_L	246.8
L_BINNEY	157
TUNGATINAH_LAG	44.9
L_LIAPOOTAH	6.8
WAYATINAH_LAG	51.3
L_CATAGUNYA	31.6
L_REPULSE	46.9
CLUNY_LAG	31.4
MEADOWBANK_L	318.3
ARTHURS_L	3881.4
TODS_FOREBAY	0
GREAT_LAKE	11506.9
L_TREVALLYN	77
L_MACKENZIE	135.1
FISHER_FOREBAY	0
L_ROWALLAN	1350.4
L_PARANGANA	14.2
L_GAIRDNER	37
L_CETHANA	199.1
L_BARRINGTON	353.9
L_PALOONA	25.6
L_PEDDER	2140.1
L_GORDON	53509.1
L_NEWTON	18.2
L_PLIMSOLL	201.8
L_MURCHISON	629.5
L_MACKINTOSH	3149.5
L_ROSEBERY	563.9
L_PIEMAN	1089.7
L_MARGARET	136.1
L_BURBURY	4866.9

6.3.2 Problem formulation

The constraints of this case study are addressed using either the graph-based technique or the penalty-based technique introduced in Section 4.4. The timeframe window constraints and precedence constraints are addressed using the graph-based technique during construction of trial solutions, while demand constraints, reliability constraints and completion constraints are handled by the penalty-based technique after the violation of these constraints have been calculated by the SYSOP simulation model. Adapting Eq. 4.18, the objective function used by ACO-PPMSO includes all objective terms and the penalty associated with violation of constraints addressed by the penalty-based technique. The objective function used for this case study is therefore given by:

$$OFC(s) = \frac{c_{ETFEIS}}{ETFEIS(s)} + c_{short} * DurCut_{tot}(s) + c_{res} * ResVio(s) + c_{dev} * dev(s/s_0) \tag{6.4}$$

where $OFC(s)$ is the objective function cost ($) associated with a trial maintenance schedule, s; $ETFEIS(s)$ is the expected total final energy in storage (GWh) associated with a trial maintenance schedule, s; $DurCut_{tot}(s)$ is the total reduction in maintenance duration due to shortening and deferral (day) associated with a trial maintenance schedule, s; $ResVio(s)$ is the violation of reserve constraints (day) associated with a trial maintenance schedule, s; $dev(s/s_0)$ is the total deviation (day) of a maintenance schedule, s, from an original schedule, s_0; c_{ETFEIS}, c_{short}, c_{res} and c_{dev} are the weights given to $ETFEIS$ ($MWh), $DurCut_{tot}$ ($/day), $ResVio$ ($/day) and dev ($/day).

The values of $DurCut_{tot}(s)$ and $dev(s/s_0)$ associated with a trial schedule, s, can be calculated easily given a partially-completed or a complete schedule, given by Eqs 5.18 and 6.5, respectively:

$$dev(s/s_0) = \sum_{n=1}^{total\_n} \left| \left(start_n(s) + chdur_n(s)\right) - \left(start_n(s_0) + chdur_n(s_0)\right) \right| \tag{6.5}$$

where $start_n(.)$ and $chdur_n(.)$ are the start date and chosen duration (day) of maintenance task d_n associated with the schedule denoted in the bracket.

On the other hand, the values of ETFEIS and ResVio (Eq. 6.4) can only be calculated by the SYSOP simulation model once a complete maintenance schedule has been obtained. ETFEIS is the sum of final energy in storage of all major storages in the system. It should be noted that thermal generation throughout the planning horizon is dictated by the overall position of storages in the system, rather than the scheduling of maintenance tasks, thus eliminating the need for thermal generation to be included in the objective function (Eq. 6.4).

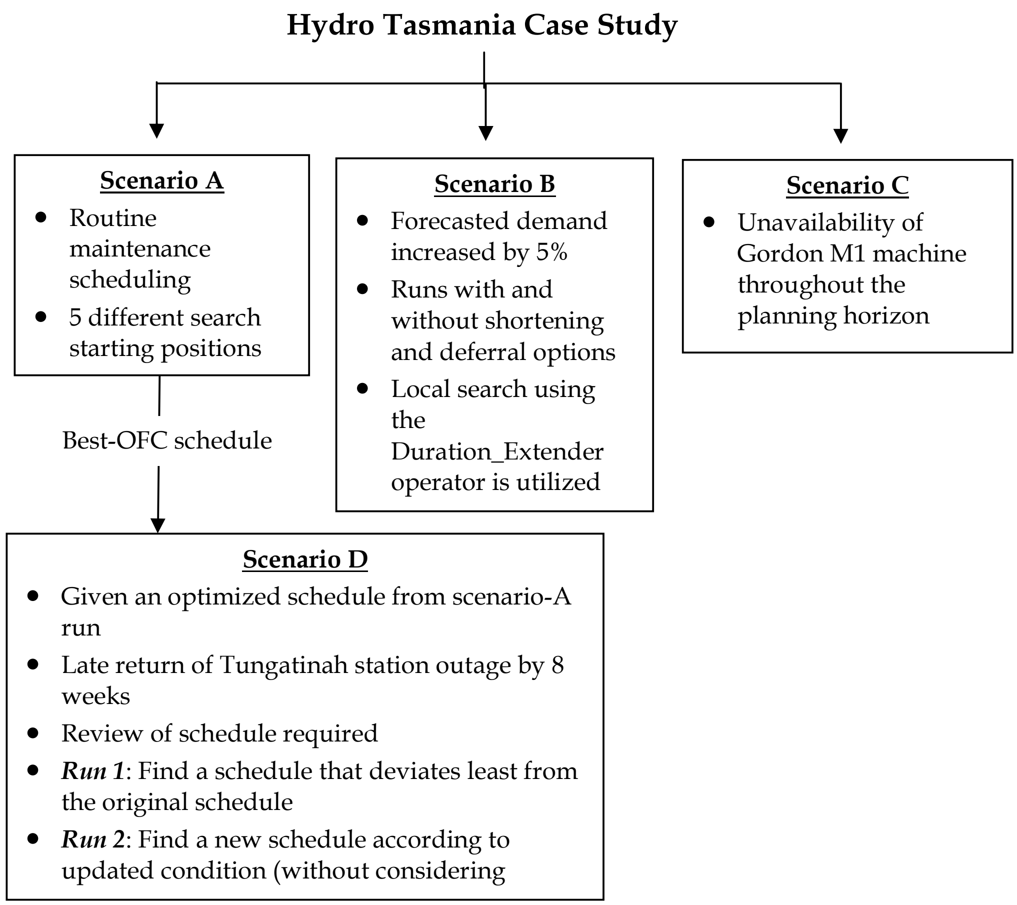


Figure 6.24: Optimisation runs for Hydro Tasmania system

6.3.3 Analysis conducted

In order to further evaluate the utility of the proposed ACO-PPMSO approach and to test the impact of changed conditions on the maintenance schedules obtained, four scenarios commonly encountered during maintenance scheduling are considered (see Figure 6.24).

6.3.3.1 Scenario A: Routine maintenance

In scenario A, the decision maker aims to find an optimum schedule for the maintenance tasks shown in Tables 6.6 and 6.7, given the forecasted demand (Figure 6.22). Each trial maintenance schedule is assessed using the 77 historical system inflows specified in Section 6.3.1. In order to check the sensitivity of the optimisation outcome to different starting positions within the problem search space, optimisation runs are carried out using different random number seeds. However, due to the long run-time (approximately three minutes) taken for each trial solution to be evaluated by the simulation model, only five runs were considered. It should be noted that in this scenario, options of shortening and deferral of maintenance tasks are not considered.

6.3.3.2 Scenario B: Increased system demand

In scenario B, the average system demand used in scenario A is increased by 5% due to population growth. As the demand constraints are now higher, the optimized schedule found by the ACO-PPMSO algorithm may not be feasible. Therefore, two optimisation runs are performed, one with and one without considering shortening and deferral options. For the run in which shortening and deferral options are considered, optional durations available for maintenance tasks to be scheduled are given in Figure 6.7.

6.3.3.3 Scenario C: Loss of a Gordon machine

Scenario C models the loss of the Gordon M1 generator throughout the planning horizon. As Gordon is one of the major stations of the Hydro Tasmania system, the unavailability of a Gordon machine essentially represents the loss of a portion of the power system generating capacity. As the forecasted system demand remains unchanged, the loss of generating capacity may affect the ability of the system to meet all of the hard constraints listed in Section 6.3.1. In this case, the decision maker of Hydro Tasmania may be interested in finding a good maintenance schedule that satisfies the tighter-than-usual hard constraints. An optimisation run using ACO-PPMSO is performed to find the best schedule available, given such adverse system conditions.

6.3.3.4 Scenario D: Late return of Tungatinah station

As part of this scenario, it is assumed that the best schedule obtained as a result of the scenario-A run is adopted by the decision maker. It is also assumed that two weeks into the Tungatinah station upgrade (Task id:

30523), the decision maker is informed that an extra eight weeks is required to complete the task. In this scenario, the utility of the original schedule (the one provided by the scenario-A run) is examined, taking into account the late return of Tungatinah station by eight weeks.

Late return of generating machines from maintenance is a common scenario in maintenance scheduling. As the consequence of such a scenario is a loss of the system generating capacity, the original (existing) maintenance schedule may need to be changed to ensure hard constraints are satisfied. However, disruptions to an existing schedule are not favorable, as high costs may incur as a result of changes to arrangements (such as personpower and machines) that have been made for some maintenance tasks. In view of this, a further scenario was conducted to determine if system reliability could be improved with minimal disruption to the original schedule. As a basis of comparison, another optimisation run was conducted to determine the schedule that results in the best system reliability without considering the original schedule. In order to achieve this, the following runs were carried out using the ACO-PPMSO algorithm:

Review run: A review optimisation run was performed using the ACO-PPMSO algorithm was performed. In addition to meeting the objectives and constraints specified previously, an optimum schedule that deviates the least from the original schedule was obtained.

New run: A new optimisation run was performed to find an optimum schedule using updated maintenance task details, in which the extended outage duration of the Tungatinah station maintenance was specified. It should be noted that the deviation from the start times contained in the original schedule was not taken into account in this run. The date the decision maker was informed about the late return was used as the start date of the planning horizon for this run. The water levels of storages in the system were assumed to be identical to those on 1 Jan 2006 problem (Stolp, S., personal communication, 2006).

The parameter settings used for the runs in this case study (Table 6.9) were derived from the author's experience based on the outcomes of the sensitivity analysis performed for previous case studies.

Table 6.9: Parameters for optimisation runs

Parameter	Value
Number of ants, m	100
1-pheromone evaporate rate, 1-r	0.8
p_{best}	0.05
Initial pheromone level, t_0	1000 units
Reward factor, Q	100 units/$
Weight given to the violation of reliability constraint, c_{resvio}	$200/day
Weight given to expected total final energy in storage, c_{ETFEIS}	$1000 per 1/MWy
Weight given to the total shortened/deferred duration, c_{short}	$1/day
Weight given to the total deviation from an original schedule, c_{dev}	$0.5/day

6.3.4 Results and discussion

The results obtained by ACO-PPMSO for scenarios A, B, C and D are summarized in Table 6.10. In the table, the objective values and satisfaction of various constraints associated with the best schedules are given.

6.3.4.1 <u>Scenario A</u>

The best schedule obtained by the five ACO-PPMSO runs for scenario A (referred to as the 'ACO schedule-A' hereafter) is tabulated in Table 6.11 and plotted in Figure 6.25. The schedule is compared to an actual schedule used by Hydro Tasmania (referred to as the 'Hydro schedule' hereafter), which was derived based on many years experience of the system (Table 6.11 and Figure 6.26). The total final energy in storage, total duration shortened/deferred, thermal generation and the satisfaction of various constraints associated with the two schedules are compared in Table 6.10.

Table 6.10: Summary of results for the Full Hydro Tasmania case

		Scenario A *Routine maintenance*		*Scenario B* *Demand increase*		*Scenario C* *Unavailability of Gordon M1 machine*	*Scenario D* *Late return of Tungatinah station*	
		Hydro schedule	**ACO schedule-A**	**Without shortening and deferral options**	**Shortening and deferral options considered**	-	**Review run: Deviation from original schedule considered**	**New run: Deviation from original schedule *NOT* considered**
Objectives	Expected total final energy in storage (MWy)	742.39	745.17	701.53	700.56	745.45	778.97	779.97
	Total duration shortened/de-ferred (day)	N/A	N/A	N/A	99	N/A	N/A	N/A
	Total deviation from original schedule (day)	N/A	N/A	N/A	N/A	N/A	35	2712
Constraints	Expected unserved energy (MWy)	0	0	0.046	0	0	0	0
	Violation of reserve constraints (days)	1	0	5	0	1	2	1

Table 6.11: The schedules proposed by ACO (^) and Hydro Tasmania (^^)

^ TFEIS = 745.174 MWY; Violation of reserve constraints = 0 days; Thermal generation = 130.69 (77 inflows)
^^ TFEIS = 742.39 MWy; Violation of reserve constraints = 1 day; Thermal generation = 130.44 MW (77 inflows)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
30555	Bastyan	Machine G1 (79.9MW) (94MVA-0.85PF) (14.5kV) (166.7rpm)	Machine Protection Tests.	7 Jun 2006	9 Jun 2006	2 May 2006 9:00	4 May 2006 17:00
30556	Bastyan	Machine G1 (79.9MW) (94MVA-0.85PF) (14.5kV) (166.7rpm)	Functional testing of Governor, AVR etc for Machine Condition Assessment.	28 May 2006	1 Jun 2006	5 Jun 2006 9:00	9 Jun 2006 16:00
32095	Butlers Gorge	Machine G1 (12.2MW) (14 MVA-0.8 PF) (11KV) (250RPM)	Transline protection upgrade.	26 Jun 2006	11 Jul 2006	10 Jul 2006 9:00	25 Jul 2006 9:00
32111	Butlers Gorge	Machine G1 (12.2MW) (14 MVA-0.8 PF) (11KV) (250RPM)	Class 2 Functional Test and Control & Protection Testing	4 Oct 2006	8 Oct 2006	25 Sep 2006 7:30	29 Sep 2006 16:30
32941	Butlers Gorge	Machine G1 (12.2MW) (14 MVA-0.8 PF) (11KV) (250RPM)	BG1 governor and alternator oil change out	23 Apr 2006	4 May 2006	1 May 2006 8:00	12 May 2006 16:30
30139	Catagunya	Machine G2 (24MW) (30MVA-0.8PF) (11KV) (187.5RPM)	Class 2 condition assessment. more...	13 Feb 2006	17 Feb 2006	13 Mar 2006 6:30	17 Mar 2006 15:30
33498	Catagunya	Machine G1 (24MW) (30MVA-0.8PF) (11KV) (187.5RPM)	Brushgear and minor mechanical maintenance.	31 Jan 2006	31 Jan 2006	25 Jan 2006 7:15	25 Jan 2006 15:15
33578	Catagunya	Machine G2 (24MW) (30MVA-0.8PF) (11KV) (187.5RPM)	Brushgear and minor mechanical maintenance. more...	5 Feb 2006	6 Feb 2006	15 Feb 2006 7:15	16 Feb 2006 15:15
33628	Catagunya	Machine G1 (24MW) (30MVA-0.8PF) (11KV) (187.5RPM)	Condition assessment of 11kV A52 machine circuit breaker.	18 Mar 2006	19 Mar 2006	28 Feb 2006 7:15	1 Mar 2006 15:15
33462	Cethana	Machine G1 (85 MW) (100MVA-0.85PF) (13.8kV) (200rpm)	Still water testing.	18 Mar 2006	20 Mar 2006	6 Feb 2006 8:00	8 Feb 2006 16:00

Table 6.11: The schedules proposed by ACO (^) and Hydro Tasmania (^^) (cont)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
33468	Cethana	Machine G1 (85 MW) (100MVA-0.85PF) (13.8kV) (200rpm)	Brushgear maintenance.	2 Jan 2006	2 Jan 2006	13 Jan 2006 8:00	13 Jan 2006 14:00
33499	Cluny	Machine G1 (17MW) (21.25 MVA-0.8PF) (11KV) (115.4RPM)	Brushgear and minor mechanical maintenance. more...	14 Jan 2006	14 Jan 2006	23 Jan 2006 7:15	23 Jan 2006 15:15
33613	Cluny	All Equipment	Urgent replacement of old damaged aircraft marker balls.	13 Feb 2006	13 Feb 2006	13 Feb 2006 9:00	13 Feb 2006 13:00
33463	Devils Gate	Machine G1 (60 MW) (75MVA-0.85PF) (11KV) (166.7RPM)	Still water testing.	12 Mar 2006	14 Mar 2006	13 Feb 2006 8:00	15 Feb 2006 16:00
33465	Devils Gate	Machine G1 (60 MW) (75MVA-0.85PF) (11KV) (166.7RPM)	Brushgear maintenance and governor compressor maintenance.	1 Jan 2006	1 Jan 2006	12 Jan 2006 7:30	12 Jan 2006 12:30
32970	Lake Echo	Machine G1 (32.4MW) (36 MVA-0.9PF) (11 kV) (428RPM)	Lake Echo headworks protection upgrade	22 Jul 2006	9 Aug 2006	1 May 2006 8:00	19 May 2006 16:30
33223	Lake Echo	Machine G1 (32.4MW) (36 MVA-0.9PF) (11 kV) (428RPM)	RTU upgrade - Commission points from Relay T60 from Echo to Tarraleah.	10 Sep 2006	10 Sep 2006	10 Jan 2006 9:00	10 Jan 2006 13:00
33548	Lake Echo	Machine G1 (32.4MW) (36 MVA-0.9PF) (11 kV) (428RPM)	Install new bolts on Fore Bay gates at cable swivel points	13 Oct 2006	13 Oct 2006	17 Mar 2006 7:30	17 Mar 2006 15:30
32460	Fisher	Canal/Flume No1	Flume repairs on vertical & horizontal seals.	3 Jan 2006	13 Jan 2006	20 Feb 2006 8:00	2 Mar 2006 15:00
33459	Fisher	Machine G1 (43.2 MW) (48MVA-0.9PF) (11kV) (500RPM)	Still water testing.	17 Dec 2006	19 Dec 2006	27 Feb 2006 8:00	1 Mar 2006 16:00
33470	Fisher	Machine G1 (43.2 MW) (48MVA-0.9PF) (11kV) (500RPM)	Brushgear maintenance and runner repair.	14 Jan 2006	15 Jan 2006	9 Jan 2006 8:00	10 Jan 2006 21:00
33605	Fisher	Machine G1 (43.2 MW) (48MVA-0.9PF) (11kV) (500RPM)	Install RTD's and accelerometers on the stator	5 Nov 2006	5 Nov 2006	6 Feb 2006 8:00	6 Feb 2006 14:30

The schedules proposed by ACO (^) and Hydro Tasmania (^^) (cont)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
30072	Gordon	Machine G1 (144 MW) (160MVA-0.9PF) (18kV) (272.2RPM)	Removal and installation of Alternator rotor poles and coolers. more...	2 Jan 2006	2 Aug 2006	2 Jun 2006 7:30	31 Dec 2006 19:00
30680	Gordon	Machine G2 (144 MW) (160MVA-0.9PF) (18kV) (272.2RPM).	Modernisation and upgrade program more...	1 Jan 2006	8 Mar 2006	1 Jan 2006 7:30	8 Mar 2006 19:00
30773	Gordon	All Equipment	2 yearly intake gate condition inspection	8 Nov 2006	29 Nov 2006	17 Oct 2006 9:00	7 Nov 2006 19:00
30998	Gordon	All Equipment	Downstream Basslink Studies as per Gordon River Work, Safety and Communications Plan to be developed, contact Andrew Jones.	27 Oct 2006	29 Oct 2006	6 Oct 2006 21:00	8 Oct 2006 18:00
30999	Gordon	All Equipment	Downstream Basslink Studies as per Gordon River Work, Safety and Communications Plan to be developed, contact Andrew Jones.	29 Dec 2006	31 Dec 2006	8 Dec 2006 21:00	10 Dec 2006 18:00
31002	Gordon	All Equipment	Downstream Basslink Studies as per Gordon River Work, Safety and Communications Plan to be developed, contact Andrew Jones.	25 Aug 2006	27 Aug 2006	7 Apr 2006 21:00	9 Apr 2006 18:00
33456	Gordon	Machine G1 (144 MW) (160MVA-0.9PF) (18kV) (272.2RPM)	Brushgear and minor mechanical maintenance. more...	23 Jan 2006	24 Jan 2006	18 Jan 2006 7:00	19 Jan 2006 17:00
33457	Gordon	Machine G3 (144 MW) (160MVA-0.9PF) (18kV) (272.2RPM)	Brushgear and minor mechanical maintenance.	25 Jan 2006	25 Jan 2006	20 Jan 2006 7:00	20 Jan 2006 15:00
33485	Gordon	All Equipment	Intake gate testing.	14 Jan 2006	14 Jan 2006	14 Jan 2006 14:00	14 Jan 2006 17:00
33601	Gordon	All Equipment	Downstream Basslink Studies as per Gordon River Work, Safety and Communications Plan to be developed, contact Andrew Jones.	21 Apr 2006	23 Apr 2006	10 Mar 2006 20:00	12 Mar 2006 17:00

The schedules proposed by ACO (^) and Hydro Tasmania (^^) (cont)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
33511	John Butters	Machine G1 (144 MW) (160MVA-0.9PF) (13.8 kV) (272.7rpm)	Operator training. more...	27 Jan 2006	27 Jan 2006	2 Feb 2006 8:00	2 Feb 2006 14:00
30553	Lemonthyme	Machine G1 (51 MW) (60MVA-0.85PF) (11KV) (300RPM)	Machine 6 Yearly Condition Assessment	13 Mar 2006	26 Mar 2006	6 Mar 2006 9:00	19 Mar 2006 17:00
33461	Lemonthyme	Machine G1 (51 MW) (60MVA-0.85PF) (11KV) (300RPM)	Still water testing.	5 Apr 2006	8 Apr 2006	30 Jan 2006 8:00	2 Feb 2006 17:00
33467	Lemonthyme	Machine G1 (51 MW) (60MVA-0.85PF) (11KV) (300RPM)	Brushgear maintenance.	20 Jan 2006	20 Jan 2006	11 Jan 2006 6:30	11 Jan 2006 11:30
32279	Liapootah	Machine G1 (27.9MW) (31MVA-0.9PF) (11KV) (300RPM)	Class 2 Functional Testing	9 Mar 2006	13 Mar 2006	6 Mar 2006 7:40	10 Mar 2006 16:30
33480	Liapootah	Machine G3 (27.9MW) (31MVA-0.9PF) (11KV) (300RPM)	Brushgear and minor mechanical maintenance. more...	9 Jan 2006	9 Jan 2006	10 Jan 2006 7:15	10 Jan 2006 15:15
33481	Liapootah	Machine G1 (27.9MW) (31MVA-0.9PF) (11KV) (300RPM)	Brushgear and minor mechanical maintenance. more...	10 Jan 2006	10 Jan 2006	11 Jan 2006 7:15	11 Jan 2006 15:15
33551	Liapootah	Machine G1 (27.9MW) (31MVA-0.9PF) (11KV) (300RPM)	Clean machine oil cooler.	2 Jan 2006	2 Jan 2006	9 Jan 2006 10:00	9 Jan 2006 14:30
33579	Liapootah	Machine G2 (27.9MW) (31MVA-0.9PF) (11KV) (300RPM)	Brushgear and minor mechanical maintenance.	8 Feb 2006	8 Feb 2006	17 Feb 2006 7:15	17 Feb 2006 15:15
33592	Liapootah	Machine G1 (27.9MW) (31MVA-0.9PF) (11KV) (300RPM)	Rectify bearing thrust pad failure, investigate bearing failure cause, runner inspection, bearing temp probe calibration & other oppurtunity maintenance	27 Apr 2006	22 May 2006	20 Mar 2006 7:45	14 Apr 2006 15:45
33625	Liapootah	Machine G1 (27.9MW) (31MVA-0.9PF) (11KV) (300RPM)	Testing circuit breaker trip coil and closing coils.	7 Oct 2006	7 Oct 2006	2 Feb 2006 9:00	2 Feb 2006 15:00

The schedules proposed by ACO (^) and Hydro Tasmania (^^) (cont)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
33626	Liapootah	Machine G2 (27.9MW) (31MVA-0.9PF) (11KV) (300RPM)	Testing circuit breaker trip coil and closing coils.	8 Oct 2006	8 Oct 2006	3 Feb 2006 9:00	3 Feb 2006 15:00
33627	Liapootah	Machine G3 (27.9MW) (31MVA-0.9PF) (11KV) (300RPM)	Testing circuit breaker trip coil and closing coils.	9 Oct 2006	9 Oct 2006	6 Feb 2006 9:00	6 Feb 2006 15:00
33474	Mackintosh	Machine G1 (79.9 MW) (94MVA-0.85PF) (14.5kV) (166.7rpm)	Brushgear maintenance and re-running earth for surge diverters.	3 Jan 2006	3 Jan 2006	18 Jan 2006 7:30	18 Jan 2006 15:30
33543	Lake Margaret	All Equipment	Urgent repairs to woodstave pipeline. Request by Scott Newett EXT:4002.	9 Jan 2006	10 Jan 2006	9 Jan 2006 6:00	10 Jan 2006 16:00
30134	Meadowbank	Machine G1 (40MW) (50MVA-0.8PF) (11kV) (150RPM)	Installation of the machine fire detection and suppression equipment capex	15 Feb 2006	26 Feb 2006	27 Feb 2006 8:15	10 Mar 2006 18:15
33505	Meadowbank	Machine G1 (40MW) (50MVA-0.8PF) (11kV) (150RPM)	Brushgear and minor mechanical maintenance. more...	6 Jan 2006	6 Jan 2006	17 Jan 2006 8:15	17 Jan 2006 16:15
33528	Meadowbank	Machine G1 (40MW) (50MVA-0.8PF) (11kV) (150RPM)	Investigate noisy running and source of turbine bearing high temperature	1 Jan 2006	5 Jan 2006	29 Dec 2005 15:15	5 Jan 2006 15:15
33600	Meadowbank	Machine G1 (40MW) (50MVA-0.8PF) (11kV) (150RPM)	Fix faulty cooling water flow relay	26 Jan 2006	26 Jan 2006	20 Jan 2006 13:00	20 Jan 2006 14:30
30557	Paloona	Machine G1 (28.0 MW) (35MVA-0.8PF) (11kV) (187.5rpm)	Machine Protection Testing.	6 May 2006	8 May 2006	16 May 2006 9:00	18 May 2006 17:00
30558	Paloona	Machine G1 (28.0 MW) (35MVA-0.8PF) (11kV) (187.5rpm)	Functional testing of Governor, AVR etc for Machine Condition Assessment.	20 Apr 2006	24 Apr 2006	19 Jun 2006 8:00	23 Jun 2006 16:00
30560	Paloona	Machine G1 (28.0 MW) (35MVA-0.8PF) (11kV) (187.5rpm)	Machine Condition Assessment.	5 Mar 2006	30 Mar 2006	20 Nov 2006 8:00	15 Dec 2006 16:00

The schedules proposed by ACO (^) and Hydro Tasmania (^^) (cont)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
33464	Palona	Machine G1 (28.0 MW) (35MVA-0.8PF) (11kV) (187.5rpm)	Still water testing.	12 Feb 2006	14 Feb 2006	20 Feb 2006 8:00	22 Feb 2006 16:00
33466	Palona	Machine G1 (28.0 MW) (35MVA-0.8PF) (11kV) (187.5rpm)	Brushgear maintenance.	1 Jan 2006	1 Jan 2006	12 Jan 2006 12:00	12 Jan 2006 16:00
33611	Palona	Machine G1 (28.0 MW) (35MVA-0.8PF) (11kV) (187.5rpm)	Investigate high temperature of governor oil system	30 Jan 2006	30 Jan 2006	30 Jan 2006 12:00	30 Jan 2006 15:00
30185	Poatina	Machine G6 (50mw) (62.5MVA-0.8PF) (16KV) (600RPM)	3 Yearly Maintenance schedule tasks. more...	8 Oct 2006	2 Nov 2006	11 Sep 2006 7:00	6 Oct 2006 14:00
31446	Poatina	Machine G3 (50mw) (62.5MVA-0.8PF) (16KV) (600RPM)	MIV remedial work on servo's. Also 3 yearly maintenance schedule tasks to be done. This allows the maint schedule to fit with modernisation outage requirements.	26 Jul 2006	17 Sep 2006	10 Jul 2006 8:00	1 Sep 2006 18:00
32512	Poatina110	Machine G2 (50mw) (62.5MVA-0.8PF) (16KV) (600RPM)	3 Yearly maintenance schedule tasks and MIV control upgrade of the elect-mech components.	14 Apr 2006	6 Jun 2006	8 May 2006 8:00	30 Jun 2006 18:00
33473	Poatina110	Machine G2 (50mw) (62.5MVA-0.8PF) (16KV) (600RPM)	Nozzle Repairs more...	16 Dec 2006	31 Dec 2006	12 Jan 2006 6:00	27 Jan 2006 16:00
33559	Poatina110	Machine G1 (50mw) (62.5MVA-0.8PF) (16KV) (600RPM)	Runner Inspection & Brushgear Maintenance more...	8 Feb 2006	11 Feb 2006	30 Jan 2006 6:00	2 Feb 2006 10:00
33285	Poatina220	Machine G4 (50mw) (62.5MVA-0.8PF) (16KV) (600RPM)	Runner Inspection & Brushgear Maintenance more...	25 Jan 2006	27 Jan 2006	4 Jan 2006 6:00	6 Jan 2006 15:00
33286	Poatina220	Machine G5 (50mw) (62.5MVA-0.8PF) (16KV) (600RPM)	Runner Inspection & Brushgear Maintenance more...	10 Jan 2006	11 Jan 2006	9 Jan 2006 6:00	10 Jan 2006 14:00

The schedules proposed by ACO (^) and Hydro Tasmania (^^) (cont)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
33612	Poatina220	Machine G3 (50mw) (62.5MVA-0.8PF) (16KV) (600RPM)	Runner Inspection & Brushgear Maintenance. more...	5 Feb 2006	6 Feb 2006	15 Feb 2006 6:00	16 Feb 2006 15:30
33555	Reece1	Machine G1 (115.6 MW) (136MVA-0.85PF) (14.5kV) (166.7rpm)	Repair significant leak in upper labyrinth cooling water pipe.	20 Jan 2006	20 Jan 2006	19 Jan 2006 7:00	19 Jan 2006 13:00
33530	Reece2	T2 250 / 13.8 kV - 136 MVA ONAN	Investigate buccholz flow trip 30/12/05 @ 21:43	1 Jan 2006	6 Jan 2006	30 Dec 2005 21:43	6 Jan 2006 21:43
31673	Repulse	All Equipment	Turbine major component risk inspections, blade seal replacement & other opportunity inspections/maintenance more...	7 Feb 2006	27 Feb 2006	6 Mar 2006 7:40	26 Mar 2006 16:40
32967	Repulse	Trash Rack No. 1	Replacement of Trash Rack screens by divers.	7 Feb 2006	27 Feb 2006	6 Mar 2006 7:30	26 Mar 2006 16:00
33563	Repulse	Machine G1 (28MW) (35MVA-0.8PF) (11KV) (136.4RPM)	Brushgear and minor mechanical maintenance. more...	2 Feb 2006	2 Feb 2006	9 Feb 2006 7:15	9 Feb 2006 15:15
33614	Repulse	All Equipment	Urgent replacement of old damaged aircraft marker balls.	1 Jan 2006	1 Jan 2006	13 Feb 2006 9:00	13 Feb 2006 13:00
33460	Rowallan	Machine G1 (10.45MW) (11MVA-0.95PF) (6.6kv) (250RPM)	Still water testing.	10 Jul 2006	12 Jul 2006	23 Jan 2006 8:00	25 Jan 2006 16:00
33469	Rowallan	Machine G1 (10.45MW) (11MVA-0.95PF) (6.6kv) (250RPM)	Brushgear maintenance.	26 Jan 2006	26 Jan 2006	9 Jan 2006 8:00	9 Jan 2006 14:00
33539	Rowallan	All Equipment	Installation of REC metering. As per request from Aurora's Gavin Knight 0417040038	19 Jan 2006	19 Jan 2006	19 Jan 2006 8:00	19 Jan 2006 10:30

The schedules proposed by ACO (^) and Hydro Tasmania (^^) (cont)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
30076	Tarraleah	Machine G5 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Class 3 Condition inspection. Capex MIV Replacement, Turbine & Spears Refurbishment and install PDA.	8 Aug 2006	17 Nov 2006	12 Sep 2006 8:00	22 Dec 2006 16:00
30792	Tarraleah	Canal/Flume No1	Cleaning of No 1 Canal - Lower section	15 Feb 2006	19 Feb 2006	1 May 2006 14:00	5 May 2006 14:00
30793	Tarraleah	Canal/Flume No1	Cleaning of No 1 Canal - Lower section	25 Dec 2006	30 Dec 2006	13 Nov 2006 14:00	15 Nov 2006 14:00
30796	Tarraleah	Canal/Flume No1	Cleaning of No 1 Canal - Upper section	15 Feb 2006	19 Feb 2006	1 May 2006 14:00	5 May 2006 14:00
30797	Tarraleah	Canal/Flume No1	Cleaning of No 1 Canal - Upper section	25 Dec 2006	30 Dec 2006	15 Nov 2006 14:01	18 Nov 2006 14:00
32939	Tarraleah	Machine G2 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	TA2 governor oil change out	12 Dec 2006	15 Dec 2006	6 Mar 2006 7:00	9 Mar 2006 15:30
32940	Tarraleah	Machine G1 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	TA1 governor oil change	7 Aug 2006	11 Aug 2006	27 Mar 2006 8:00	31 Mar 2006 16:30
32981	Tarraleah	Machine G1 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Brushgear change-out and minor mechanical maintenance.	7 Jan 2006	7 Jan 2006	12 Jan 2006 7:00	12 Jan 2006 15:00
33444	Tarraleah	Machine G1 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Full load test required by CSI to find out the cause of "outborad bearing high vibration" - 0 to 100% load test	23 Jan 2006	23 Jan 2006	30 Jan 2006 7:00	30 Jan 2006 11:00
33445	Tarraleah	Machine G2 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Full load test required by CSI to find out the cause of "outborad bearing high vibration" - 0 to 100% load test	23 Jan 2006	23 Jan 2006	30 Jan 2006 11:00	30 Jan 2006 15:00

The schedules proposed by ACO (^) and Hydro Tasmania (^^) (cont)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
33446	Tarraleah	Machine G3 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Full load test required by CSI to find out the cause of 'outborad bearing high vibration' - 0 to 100% load test	24 Jan 2006	24 Jan 2006	31 Jan 2006 7:00	31 Jan 2006 11:00
33447	Tarraleah	Machine G2 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Brushgear change-out and minor mechanical maintenance.	27 Jan 2006	27 Jan 2006	5 Jan 2006 7:00	5 Jan 2006 15:00
33448	Tarraleah	Machine G3 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Brushgear change-out and minor mechanical maintenance.	28 Jan 2006	28 Jan 2006	6 Jan 2006 7:00	6 Jan 2006 15:00
33449	Tarraleah	Machine G5 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Brushgear change-out and minor mechanical maintenance.	6 Jan 2006	6 Jan 2006	10 Jan 2006 7:00	10 Jan 2006 15:00
33450	Tarraleah	Machine G6 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Brushgear change-out and minor mechanical maintenance. more...	25 Jan 2006	26 Jan 2006	3 Jan 2006 7:00	4 Jan 2006 15:00
33550	Tarraleah	Machine G6 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Penstock drain and anchor slip training	10 Jan 2006	10 Jan 2006	15 Jan 2006 7:00	15 Jan 2006 19:00
33598	Tarraleah	Machine G6 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Brushgear and minor mechanical maintenance.	31 Jan 2006	31 Jan 2006	1 Feb 2006 7:00	1 Feb 2006 15:30
33599	Tarraleah	Machine G4 (15MW) (18.75MVA-0.8PF) (11 kV) (428RPM)	Brushgear and minor mechanical maintenance.	5 Feb 2006	5 Feb 2006	2 Feb 2006 7:00	2 Feb 2006 15:30
30179	Trevallyn	Machine G1 (20mw) (25 MVA-0.8PF) (11 kV) (375 R.P.M.)	6 yearly maintenance and inspection tasks.	11 Jan 2006	8 Feb 2006	16 Jan 2006 7:00	13 Feb 2006 15:00
32799	Trevallyn	Machine G1 (20mw) (25 MVA-0.8PF) (11 kV) (375 R.P.M.)	Installation of upgraded bearing compartment oil mist containment system and refubish relief valve components and upgrade controls. more...	10 Apr 2006	28 Apr 2006	20 Nov 2006 7:00	8 Dec 2006 17:00
33204	Trevallyn	Machine G4 (26.5mw) (35.5 MVA-0.87PF) (11 kV) (375 R.P.M.)	Additional Work from Modernisation	11 Mar 2006	9 Apr 2006	19 Feb 2006 8:00	20 Mar 2006 16:30

The schedules proposed by ACO (^) and Hydro Tasmania (^^) (cont)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
33609	Trevallyn	Machine G4 (26.5mw) (35.5 MVA-0.87PF) (11 kV) (375 R.P.M.)	ENTEK Module Replacement.	27 Feb 2006	28 Feb 2006	23 Jan 2006 12:00	24 Jan 2006 15:00
33617	Trevallyn	Machine G3 (26.5mw) (35.5 MVA-0.87PF) (11 kV) (375 R.P.M.)	Investigate the cause of M/C trip on Governor Drive Low Oil Pressure more...	7 Feb 2006	8 Feb 2006	31 Jan 2006 6:39	1 Feb 2006 6:39
33624	Trevallyn	Machine G4 (26.5mw) (35.5 MVA-0.87PF) (11 kV) (375 R.P.M.)	Inspection of Lower Guide Bearing	12 Sep 2006	12 Sep 2006	2 Feb 2006 10:00	2 Feb 2006 12:00
30552	Tribute	Machine G1 13.8 kV - 92 MVA 0.9 PF (82.8 MW) 429 rpm	Machine Condition Assessment Outage.	8 Jan 2006	3 Feb 2006	22 Jan 2006 9:00	17 Feb 2006 16:00
33526	Tribute	All Equipment	To carry out Isolator maintenance and installation of earthing points.	4 Jan 2006	4 Jan 2006	24 Jan 2006 8:30	24 Jan 2006 17:00
33527	Tribute	All Equipment	Transline to be isolated for tailrace inspection due to earthing issues while carrying out this inspection.	4 Jan 2006	4 Jan 2006	28 Jan 2006 8:30	28 Jan 2006 17:00
30518	Tungatinah	Machine G1 (25MW) (31.25MVA-0.8PF) (11 kV) (600RPM)	Replacement of TX 1	5 Mar 2006	1 Apr 2006	4 Dec 2006 7:40	31 Dec 2006 16:30
30519	Tungatinah	T2 11/112.5 kV (Fixed tap) - 30 MVA ON	Replacement of TX 2	21 Aug 2006	20 Nov 2006	2 Oct 2006 7:40	31 Dec 2006 16:30
30523	Tungatinah	All Equipment	Replacement of machine cables more...	14 Feb 2006	29 Mar 2006	19 Mar 2006 7:00	1 May 2006 17:00
30527	Tungatinah	Machine G2 (25MW) (31.25MVA-0.8PF) (11 kV) (600RPM)	No 2 M/C Modernisation programme	2 Jan 2006	1 Apr 2006	3 Oct 2006 7:40	31 Dec 2006 16:30
30528	Tungatinah	Machine G1 (25MW) (31.25MVA-0.8PF) (11 kV) (600RPM)	No 1 M/C Modernisation programme	2 Oct 2006	31 Dec 2006	2 Oct 2006 7:40	31 Dec 2006 16:30
30961	Tungatinah	Machine G3 (25MW) (31.25MVA-0.8PF) (11 kV) (600RPM)	Replacement of TX 3	18 Feb 2006	23 Jun 2006	1 Jan 2006 7:40	12 May 2006 17:30

The schedules proposed by ACO (^) and Hydro Tasmania (^^) (cont)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
30962	Tungatinah	Machine G4 (25MW) (31.25MVA-0.8PF) (11 kV) (600RPM)	Replacement of TX 4	4 Jan 2006	5 Mar 2006	1 Jan 2006 7:40	8 Mar 2006 16:30
33079	Tungatinah	Machine G2 (25MW) (31.25MVA-0.8PF) (11 kV) (600RPM)	Class 2 functinal testing	20 Jul 2006	24 Jul 2006	4 Dec 2006 8:00	8 Dec 2006 16:30
33081	Tungatinah	Machine G5 (25MW) (31.25MVA-0.8PF) (11 kV) (600RPM)	Class 2 Functional Testing	3 Jun 2006	7 Jun 2006	7 Aug 2006 8:00	11 Aug 2006 16:30
33451	Tungatinah	Machine G5 (25MW) (31.25MVA-0.8PF) (11 kV) (600RPM)	Brushgear change-out and minor mechanical maintenance.	10 Jan 2006	10 Jan 2006	17 Jan 2006 8:15	17 Jan 2006 15:30
33452	Tungatinah	Machine G3 (25MW) (31.25MVA-0.8PF) (11 kV) (600RPM)	Brushgear change-out and minor mechanical maintenance.	1 Jan 2006	1 Jan 2006	16 Jan 2006 8:15	16 Jan 2006 15:30
33573	Tungatinah	Machine G1 (25MW) (31.25MVA-0.8PF) (11 kV) (600RPM)	Brushgear and minor mechanical maintenance.	4 Feb 2006	4 Feb 2006	3 Feb 2006 8:15	3 Feb 2006 15:30
33597	Tungatinah	Machine G4 (25MW) (31.25MVA-0.8PF) (11 kV) (600RPM)	Brushgear and minor mechanical maintenance.	6 Mar 2006	6 Mar 2006	27 Feb 2006 8:15	27 Feb 2006 15:30
30297	Wayatinah	Machine G2 (12.75 MW) 11 kV - 15 MVA 0.85 P.F. 250 r.p.m.	Class 2 Functional testing	21 Jan 2006	25 Jan 2006	3 Apr 2006 7:40	7 Apr 2006 16:30
30298	Wayatinah	Machine G3 (12.75 MW) 11 kV - 15 MVA 0.85 P.F. 250 r.p.m.	Class 2 Functional testing	21 May 2006	25 May 2006	10 Apr 2006 7:40	14 Apr 2006 16:30
33482	Wayatinah	Machine G2 (12.75 MW) 11 kV - 15 MVA 0.85 P.F. 250 r.p.m.	Brushgear and minor mechanical maintenance. more...	13 Jan 2006	13 Jan 2006	12 Jan 2006 7:15	12 Jan 2006 15:15
33522	Wayatinah	Machine G2 (12.75 MW) 11 kV - 15 MVA 0.85 P.F. 250 r.p.m.	more...	6 Jan 2006	6 Jan 2006	3 Jan 2006 10:00	3 Jan 2006 14:00
33564	Wayatinah	Machine G1 (12.75MW) (15MVA-0.85PF) (11kV) (250RPM)	Brushgear and minor mechanical maintenance.	5 Feb 2006	5 Feb 2006	8 Feb 2006 7:15	8 Feb 2006 15:15

The schedules proposed by ACO (^) and Hydro Tasmania (^^) (cont)

Req No	Station	Plant Name	Work Planned	^ACO Start	^ACO End	^^Hydro Start	^^Hydro End
33581	Wayatinah	Machine G3 (12.75 MW) 11 kV - 15 MVA 0.85 P.F. 250 r.p.m.	Brushgear and minor mechanical maintenance.	4 Mar 2006	4 Mar 2006	23 Feb 2006 7:15	23 Feb 2006 15:15
30133	Wilmot	Machine G1 (30.6 MW) (34MVA-0.9PF) (11KV) (600RPM)	Condition Assessment Maintenance Outage including repacking of Stator Joint & Wedging.	3 Jan 2006	19 Mar 2006	15 Jan 2006 8:30	31 Mar 2006 16:30
33569	Wilmot	All Equipment	To carry out testing on CT's & VT's	10 Apr 2006	10 Apr 2006	2 Feb 2006 7:00	2 Feb 2006 15:00

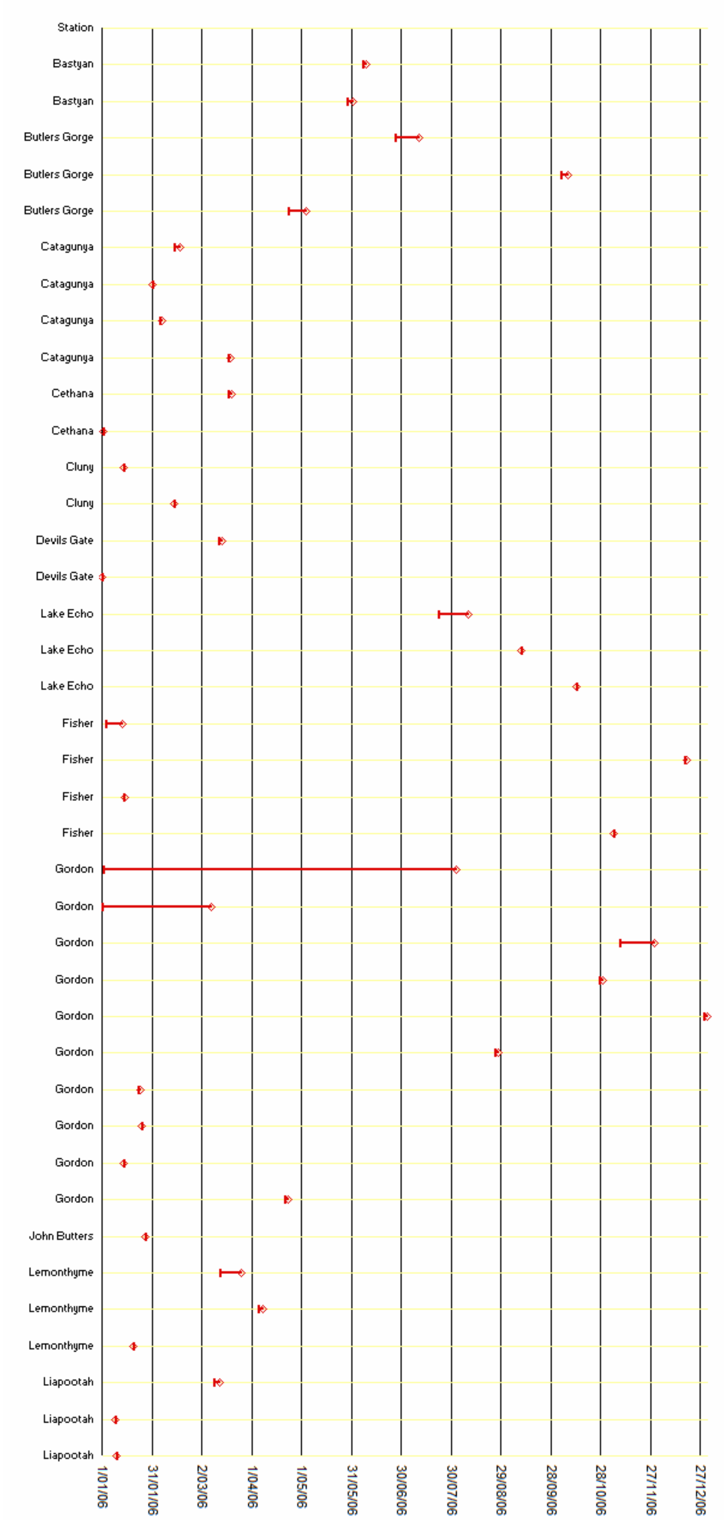


Figure 6.25: The ACO schedule-A (plotted)

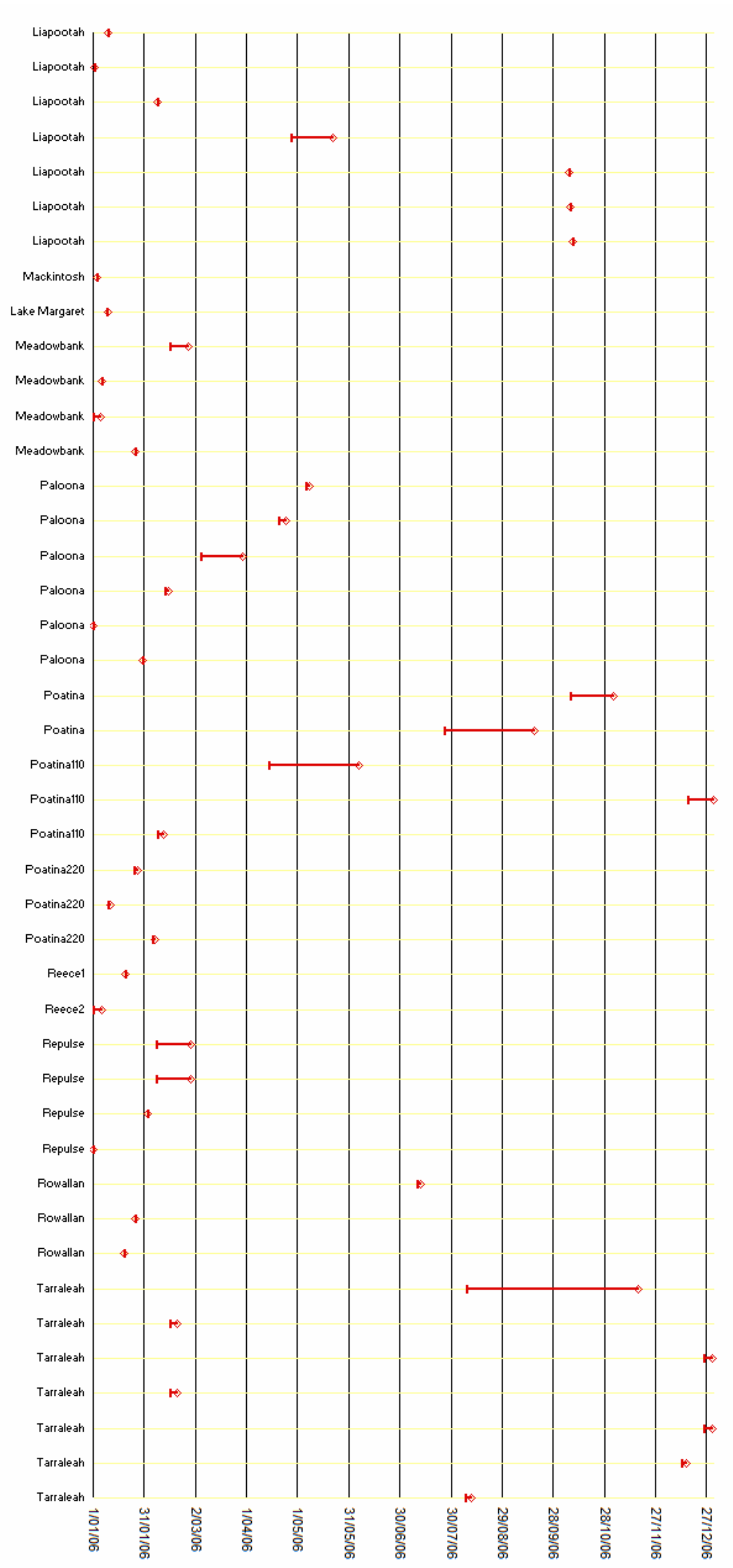


Figure 6.25 The ACO schedule-A (plotted) (cont)

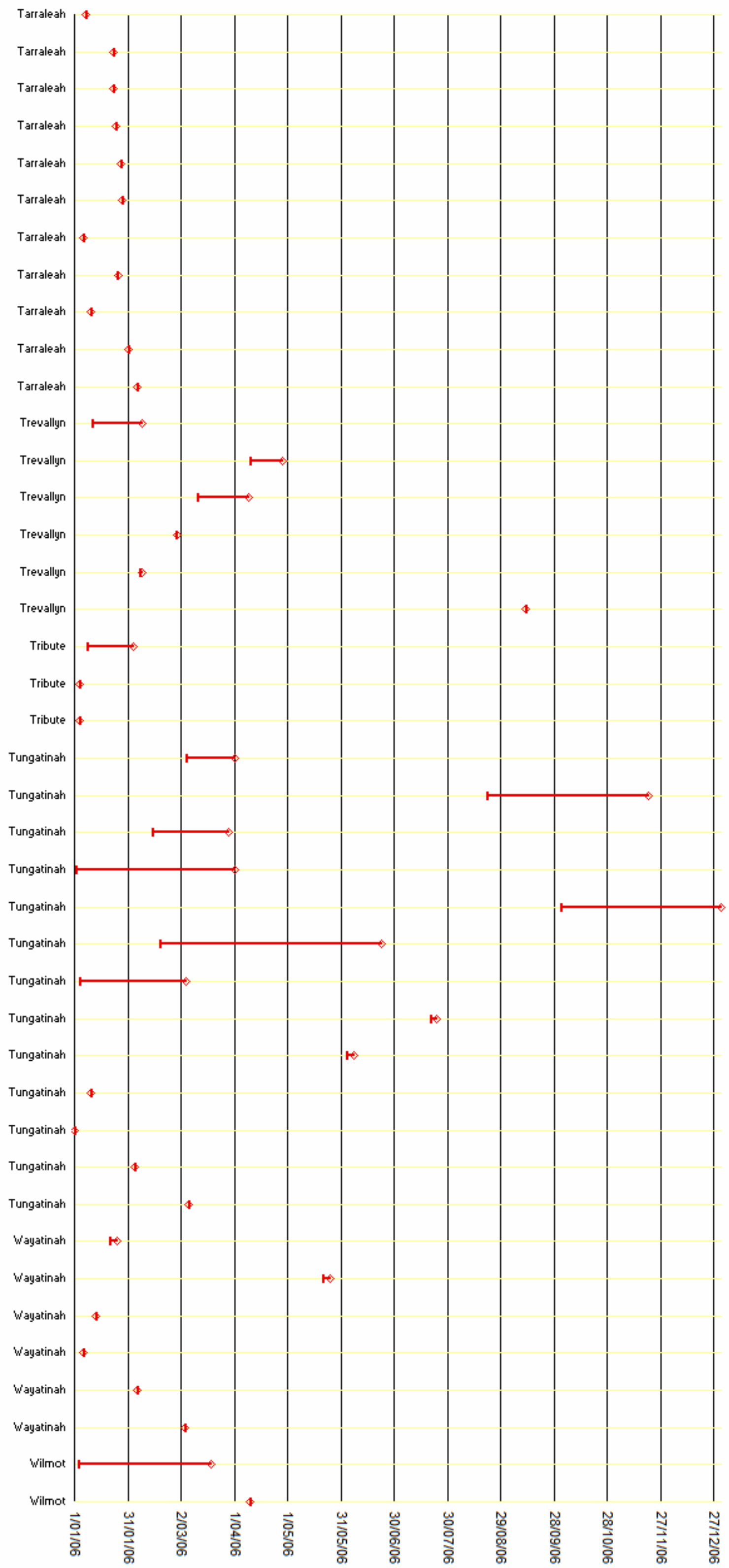


Figure 6.25 The ACO schedule-A (plotted) (cont)

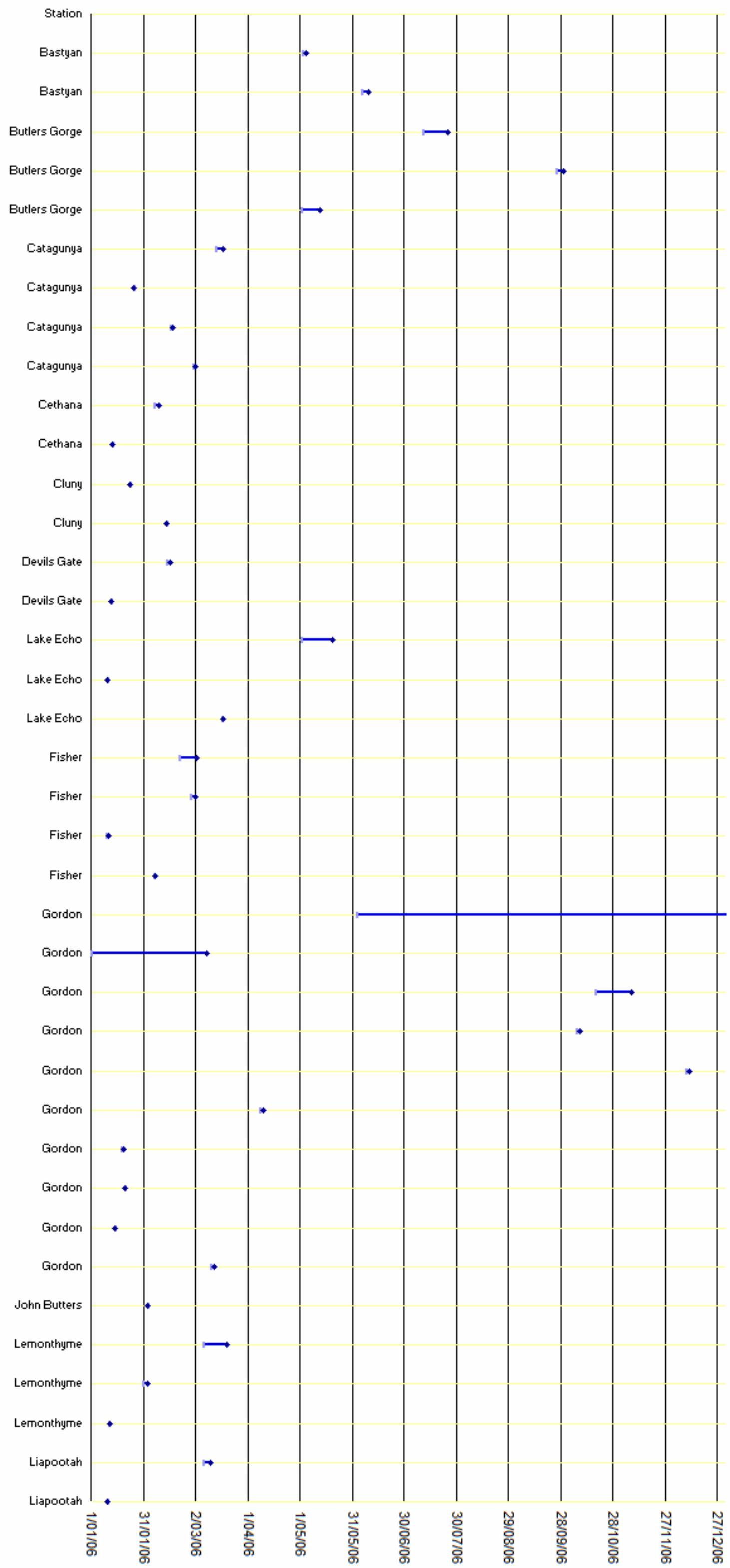


Figure 6.26: The Hydro schedule (plotted)

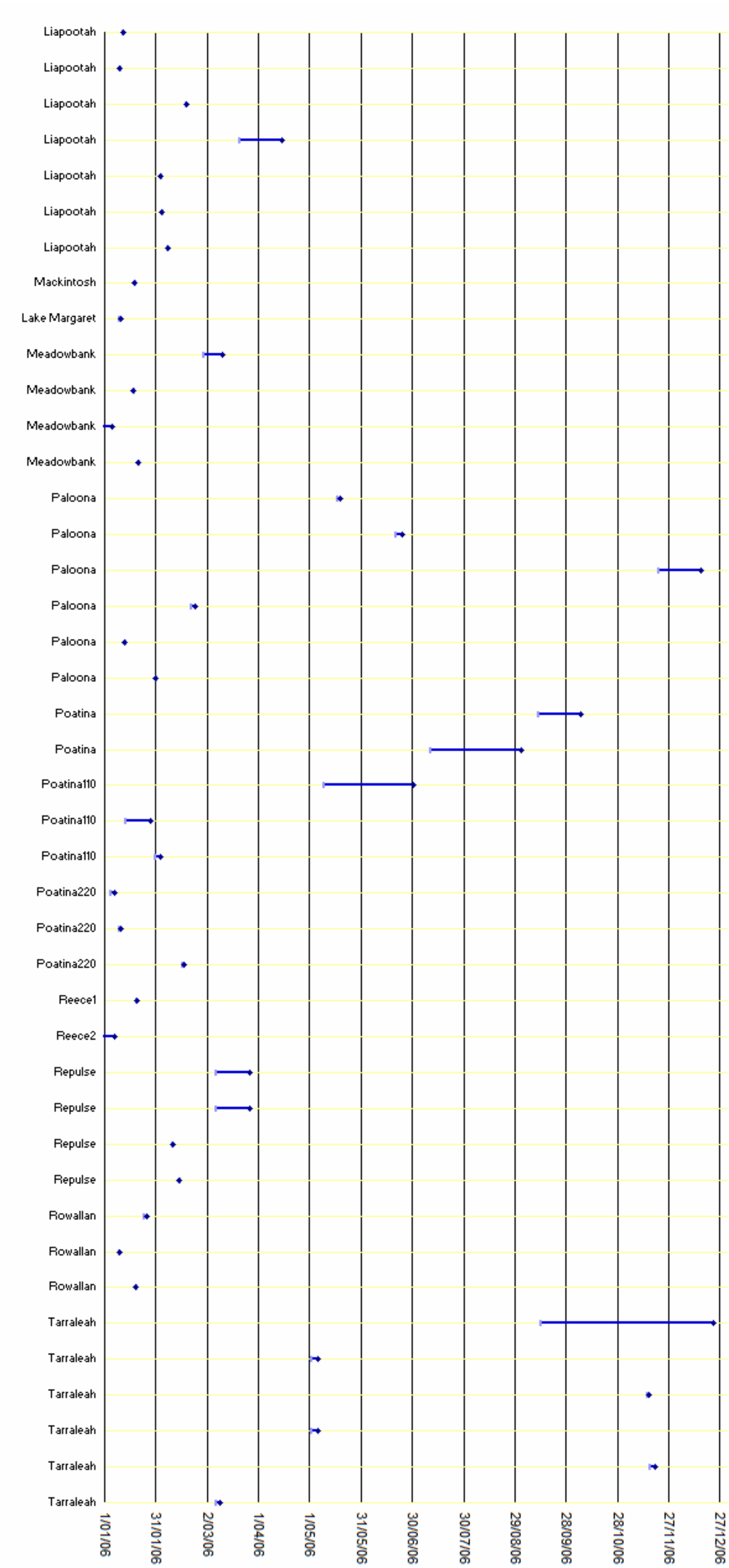


Figure 6.26 The Hydro schedule (plotted) (cont)

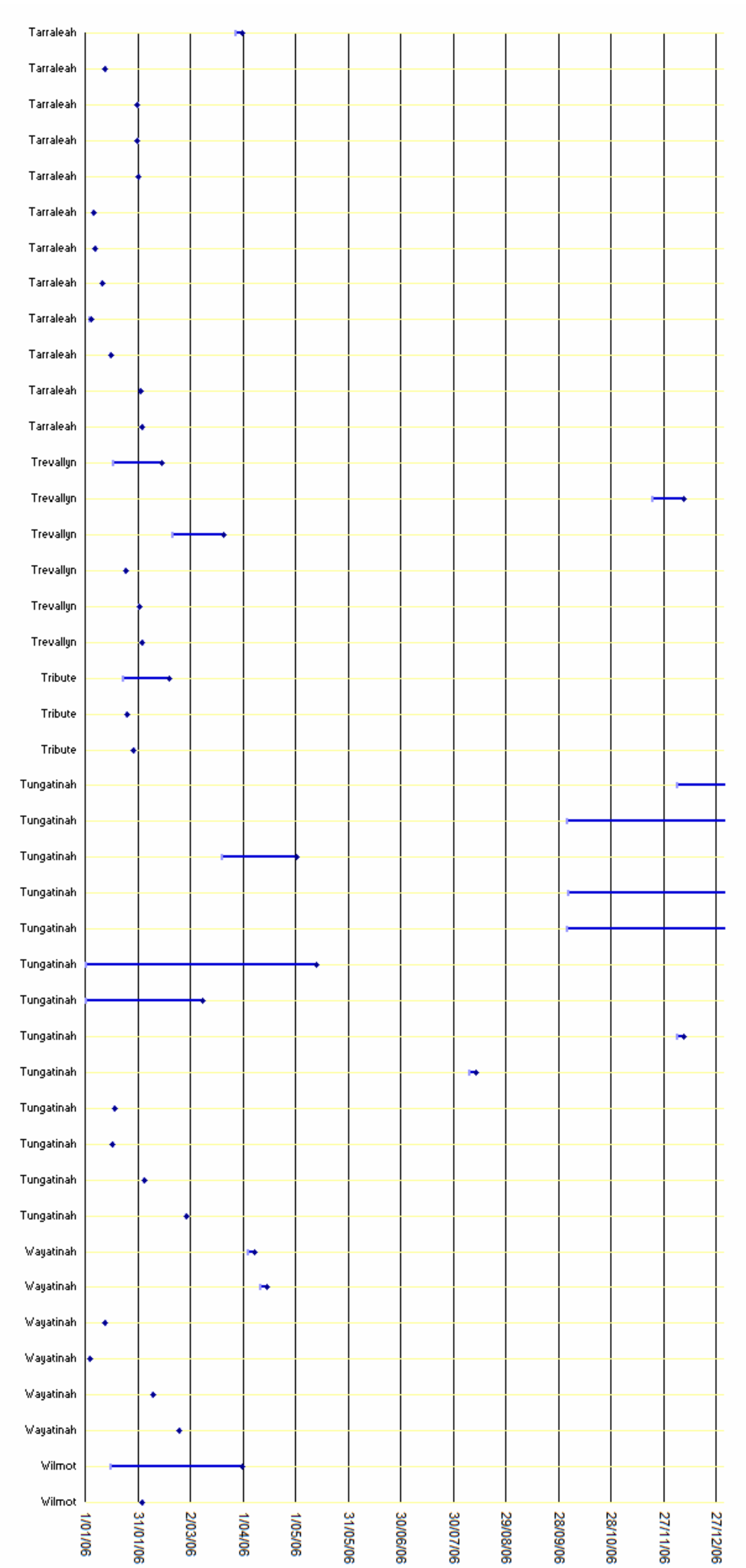


Figure 6.26 The Hydro schedule (plotted) (cont)

An assessment function (Eq. 6.6) normally utilized by Hydro Tasmania for routine maintenance scheduling is adopted to compare the Hydro and ACO schedules. The assessment function calculates a real-dollar cost based on the thermal generation profile over the planning horizon, as well as the expected total final energy in storage associated with a maintenance schedule, and is given by:

$$Cost(s) = \sum_{mth=Jan}^{Dec} THERM_{mth}(s) * 730 * \$50 - ETFEIS(s) * 8760 * \$25 \tag{6.6}$$

where $Cost(s)$ is a real-dollar cost ($) associated with maintenance schedule s; $THERM_{mth}(s)$ is the average thermal generation (MW) per month mth associated with maintenance schedule s, and $ETFEIS(s)$ is the expected total final energy in storage (MWy) of the power system associated with maintenance schedule s. In this case study, the cost of thermal generation is estimated to be $50 per MWh, while the value of energy in storage is $25 per MWh problem (Stolp, S., personal communication, 2006). 730 and 8760 are constants for the conversion of the energy term from Megawatt-month (MWm) to Megawatt-hour (MWh) and Megawatt-year (MWy) to MWh, respectively.

The objective function values and satisfaction of constraints associated with the ACO and Hydro schedules presented in Table 6.10 are repeated in Table 6.12. In addition, the costs associated with the schedules calculated using Eq. 6.6, as well as the monthly average thermal generation used for the cost calculations, are presented (Table 6.12). A negative cost given by Eq. 6.6 can be seen as a profit associated with a maintenance schedule. A comparison between the costs given by the two schedules reveals that a saving of over $500,000 could be achieved by using the ACO schedule over the Hydro schedule, based on the given information about the case study system. The encouraging results obtained by the ACO-PPMSO algorithm indicate its potential for being a useful maintenance scheduling tool.

Table 6.12: Objective values and constraint satisfaction associated with the ACO and the Hydro schedules

Criteria			Hydro schedule	ACO schedule-A
Objectives	Expected total final energy in storage (MWy)		742.39	745.17
	Total duration shortened/deferred (day)		0	0
	Monthly average thermal generation (MW)	*Jan*	189.8	193.1
		Feb	197.4	197.0
		Mar	194.5	194.1
		Apr	176.1	176.5
		May	143.0	145.3
		Jun	116.4	115.3
		Jul	99.2	99.5
		Aug	77.8	78.4
		Sep	75.0	76.0
		Oct	81.9	81.9
		Nov	106.1	105.1
		Dec	112.9	110.9
Constraints	Expected unserved energy (MWy)		0	0
	Violation of reserve constraints (day)		1	0
	Cost ($) using Eq. 6.6		**-1.053×10^8**	**-1.058×10^8**

Table 6.12 reveals that the $500,000 savings achieved when ACO schedule-A is used instead of the Hydro schedule can mainly be attributed to the higher total final energy in storage associated with the schedule obtained by ACO-PPMSO. In order to further investigate this, the levels of various major storages at the end of planning horizon associated with the ACO and Hydro schedules are shown in Table 6.13, in which storages are given in decreasing order of full supply levels. Full Supply Level (FSL) is the volume that can be stored between the crest

level of the spillway and the highest level of the invert of the lowest outlet used for power generation purposes (Hydro Tasmania). In other words, storages with higher FSLs can store larger amounts of water for the purpose of power generation. It can be seen from Table 6.13 that the final levels of the two largest storages of the system associated with the ACO schedule are higher (bolded) than those of the Hydro schedule. In particular, the difference of levels in Gordon storage between the two schedules is the most apparent (0.87% of 127186.0 cumec.days).

Table 6.13: Levels of major storages at the end of planning horizon associated with the ACO and Hydro schedules

Major Storages	FSL† (cumec.days)	ACO schedule-A (% FSL†)	Hydro schedule (% FSL†)
Lakes Gordon & Pedder	127186.0	**62.03**	61.10
Great Lake	35455.0	**34.71**	34.63
Lake King William	6179.6	80.37	80.60
Lake Echo	5912.9	72.85	74.35
Lake Burbury	4696.0	62.84	64.99
Lake Mackintosh	3163.9	70.68	74.44
Lake Rowallan	1396.3	**59.84**	59.46
Lake Murchison	724.6	37.36	38.85

† Full Supply Level (FSL) is the volume that can be stored between the crest level of the spillway and the highest level of the invert of the lowest outlet used for power generation purposes (Hydro Tasmania).

The three major Gordon outages take 67, 213 and 22 days, respectively. The two former outages involve machine M1 and M2, respectively, while the whole Gordon station is taken offline for the latter outage. The M1-machine outage is fixed to commence from Jan 1 to Mar 8, thus the outage periods of the other two maintenance tasks are critical in determining the final energy in storage of Lake Gordon. In the Hydro schedule, the M1-outage (213 days) takes place from June to December and mid-October to early November, which conincides with the timing of the station-outage, which occurs from mid-October to early November. While the timing of the station-outage period for Gordon associated with the ACO schedule is similar to that of the Hydro schedule, the M1-outage takes place from January to early August. The impact of the placement of the two tasks within the planning horizon on the total final energy in storage is investigated, taking into account the profiles of system demand (Figure

6.22) and storage inflow (Figure 6.23), as well as the rules implemented in the SYSOP simulation model.

The difference between the Lake Gordon energy in storage associated with the ACO schedule and that of the Hydro schedules at the end of month *mth* are given by:

$$\Delta EEIS\_Gord_{mth} = EEIS\_Gord_{mth}(\text{ACO}) - EEIS\_Gord_{mth}(\text{HYDRO}) \qquad (6.7)$$

where $EEIS\_Gord_{mth}$(ACO) is the expected energy of storage (%FSL) of Lake Gordon at the end of month *mth* associated with the ACO schedule; $EEIS\_Gord_{mth}$(HYDRO) is the expected energy of storage (%FSL) of Lake Gordon at the end of month *mth* associated with Hydro schedule.

The values of $\Delta EEIS\_Gord_{mth}$ for *mth* = Jan, Feb, …, Dec are shown in Figure 6.27. It can be seen that the two largest increments of $\Delta EEIS\_Gord_{mth}$ values occur in February and November, corresponding to the M2-machine and station outages that commence from January and August, respectively. It should be noted that the level of a storage increases when one or more generating machines undergo maintenance and are not being used for power generation. An investigation into the spillage conditions reveals that due to the Gordon M2-machine outage that takes place from January to August, run-of-river storages were drawn down for power generation, allowing more capacity for higher winter inflows. In this way, spillage occurs at the run-of-river storages, thus minimizing the total spills. Consequently, smaller portions of system demands are met by major storages, such as Lakes Gordon, Lake Pedder and Great Lake when the ACO schedule is used. In other words, the schedule obtained by ACO-PPMSO is optimized in a way such that the run-of-river generation is maximized in order to minimize the need for drawing down major storages, thus maximizing total final energy in storage.

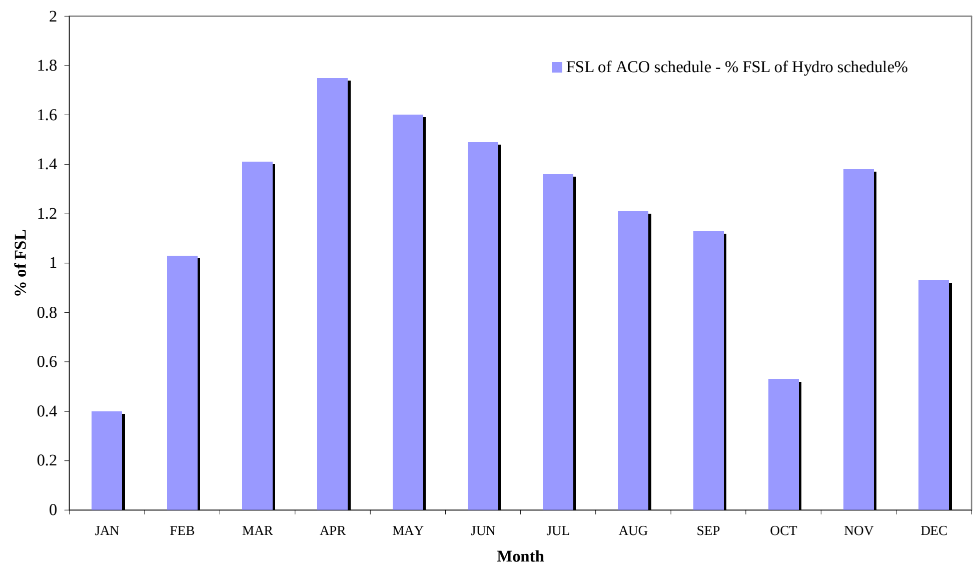


Figure 6.27: Difference between Lake Gordon end-of-month energy in storage (%FSL) associated with ACO and Hydro schedules

It was also found that the higher final energy in storage of Lake Gordon associated with the ACO schedule is due to its station maintenance in November. In the Hydro schedule, the station maintenance is scheduled to coincide with the M2-machine outage (Jun 2 to Dec 31). Consequently, the total outage duration of Gordon power station, including machine and station maintenance, associated with the ACO schedule is longer than that of the Hydro schedule. As storage level increases during maintenance, the ACO schedule is found to be more effective in maximizing energy in storage than the Hydro schedule.

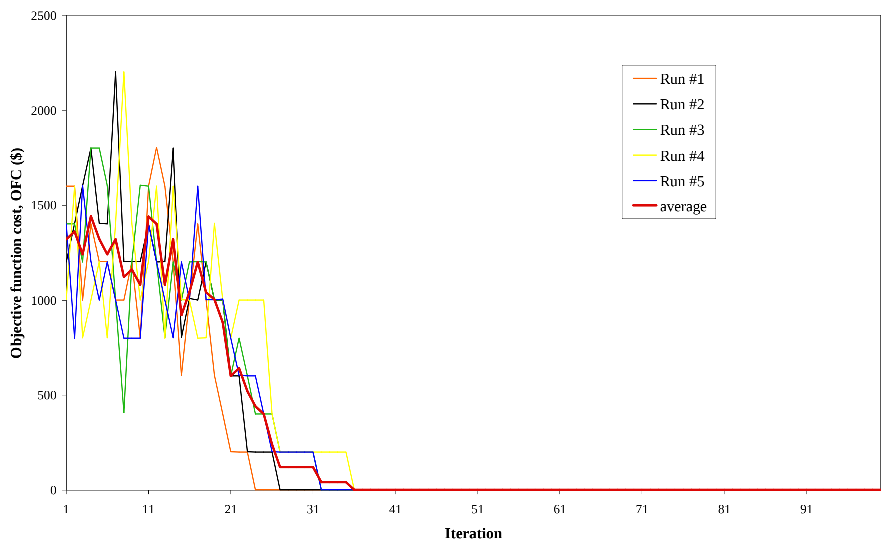
The behaviour of ACO-PPMSO in solving the Hydro Tasmania case study system can be understood from Figure 6.28a, which plots the iteration-best objective function cost (IB-OFC) recorded throughout the five ACO-PPMSO runs (using different random number seeds). Due to different starting positions in the problem search space, the IB-OFC curves given by the five optimisation runs are not identical. However, the overall trends of the curves are very similar, indicating a consistent performance of ACO-PPMSO, regardless of starting positions in the problem search space.

In order to gain a better understanding of the optimisation process, an IB-OFC_{avg} curve is obtained by averaging the best objective function cost given by the five optimisation runs in each iteration. The expected unserved energy (EUE), reserve constraints violation (ResVio), expected total final energy in storage (ETFEIS) and total reduction in maintenance duration due to shortening and deferral ($DurCut_{tot}$) associated with iteration-best schedules are plotted in Figures 6.29 to 6.32. The values presented in the plots are the average given by the five random number seed runs. They are referred to as the IB-EUE_{avg} (Figure 6.29), IB-$ResVio_{avg}$ (Figure 6.30), IB-$DurCut_{avg}$ (Figure 6.31) and IB-$ETFEIS_{avg}$ (Figure 6.32) curves, respectively.

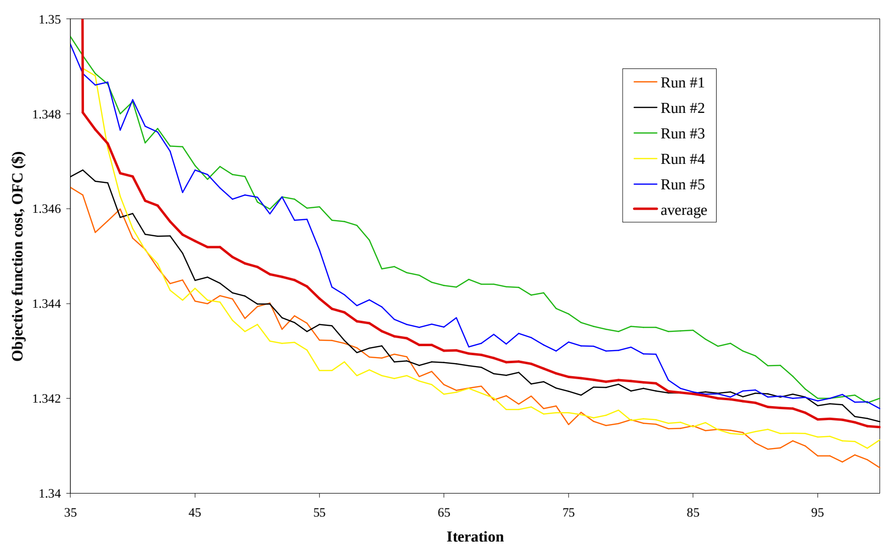
Figure 6.28a indicates that the IB-OFC_{avg} value of the first 20 iterations of the runs oscillate at significantly higher costs (greater than $1000) than those of later iterations. The oscillations of the IB-OFC values are attributed to the random search of the ACO algorithm at the earlier stages of the run before the promising solution components are marked with pheromone. The relatively high IB-OFC values during this stage are due to the high penalty costs associated with infeasible trial schedules. It can be seen that no feasible trial schedule with respect to all hard constraints (one that satisfies demand constraints, reliability constraints and are free from shortening and deferral decisions) is found until around iteration 35 (Figures 6.29 to 6.31). For the same reason, the IB-$ETFEIS_{avg}$ values of the first 20 iterations oscillate with a relatively large amplitude, and the maximization of ETFEIS becomes apparent from iteration 35 onwards, when feasible solutions have been found. It should also be noted that violation of completion constraints (trial schedules that contain shortening and deferral decisions) still exist, even though shortening and deferral options are not available in this case. Depending on the order in which maintenance tasks are considered, implementation of hard constraints using the construction-graph method (Section 4.4) can impede the scheduling of some maintenance tasks within the planning horizon.

Beyond iteration 35, the optimisation runs entered a stage where all iteration-best schedules found are feasible and the OFCs associated with these schedules are effectively the virtual costs derived from the ETFEIS associated with these schedules. Driven solely by the maximization of ETFEIS (Figure 6.32), the IB-OFC values during this stage improve at a much slower rate than those of earlier iterations (Figure 6.28b). However, it should be noted that not all trial schedules constructed at this stage are feasible. This is illustrated in Figure 6.33 where infeasibility ratios (see Eq. 5.20 for definition) are greater than zero throughout the runs.

On average, the computation time required for each run (10,000 evaluations) is approximately 48 hours on a Linux Symmetric Multi Processor Kernel (Memory: 1GB; CPU: AMD Athlon(tm) MP 2600+).



(a)



(b)

Figure 6.28: Objective function cost associated with iteration-best schedules (a) Iterations 1 to 100 and (b) Iterations 41 to 100

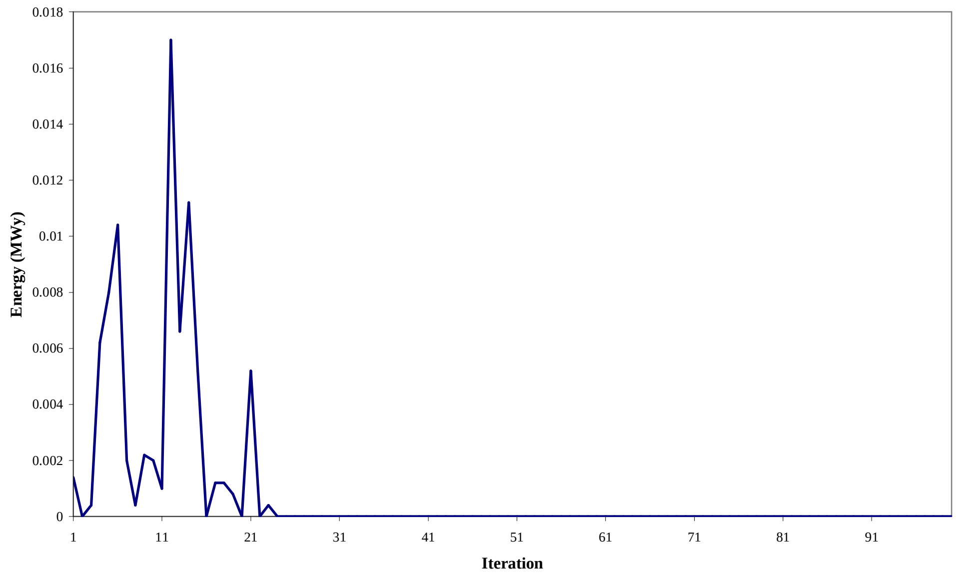


Figure 6.29: Averaged expected unserved energy (IB-EUEavg) associated with iteration-best schedules

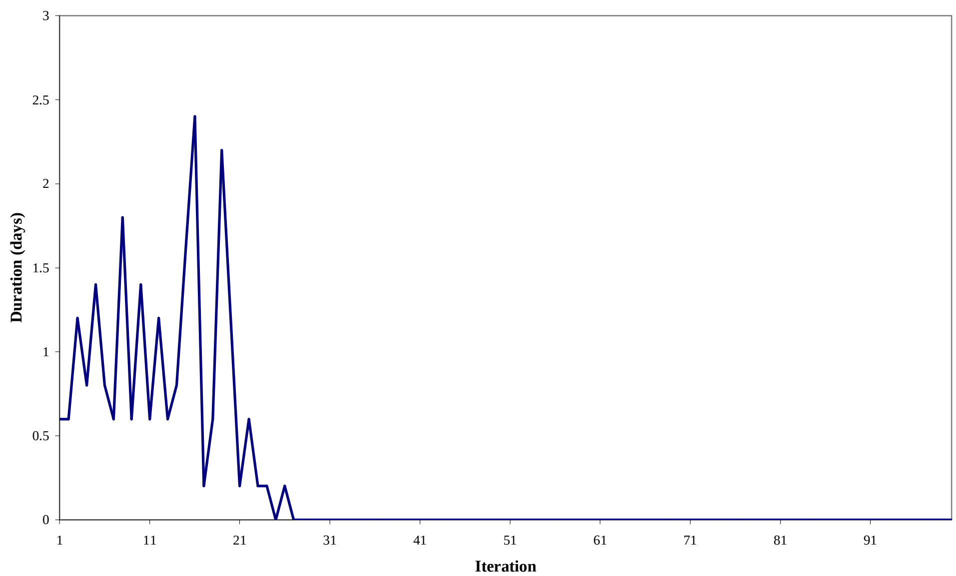


Figure 6.30: Averaged violation of reliability constraints (IB-ResVioavg) associated with iteration-best schedules

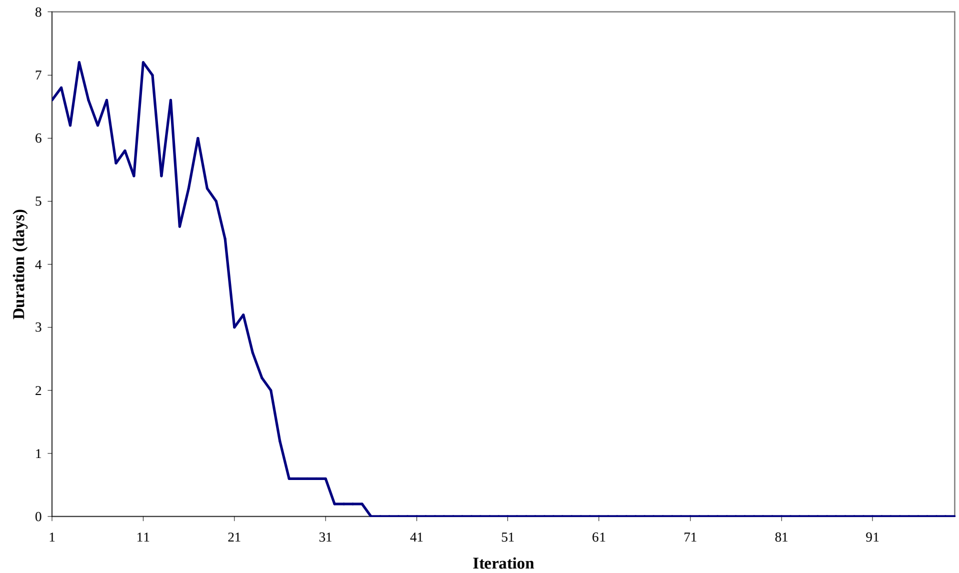


Figure 6.31: Averaged total reduction in maintenance duration due to shortening and deferral (IB-DurCutavg) associated with iteration-best schedules

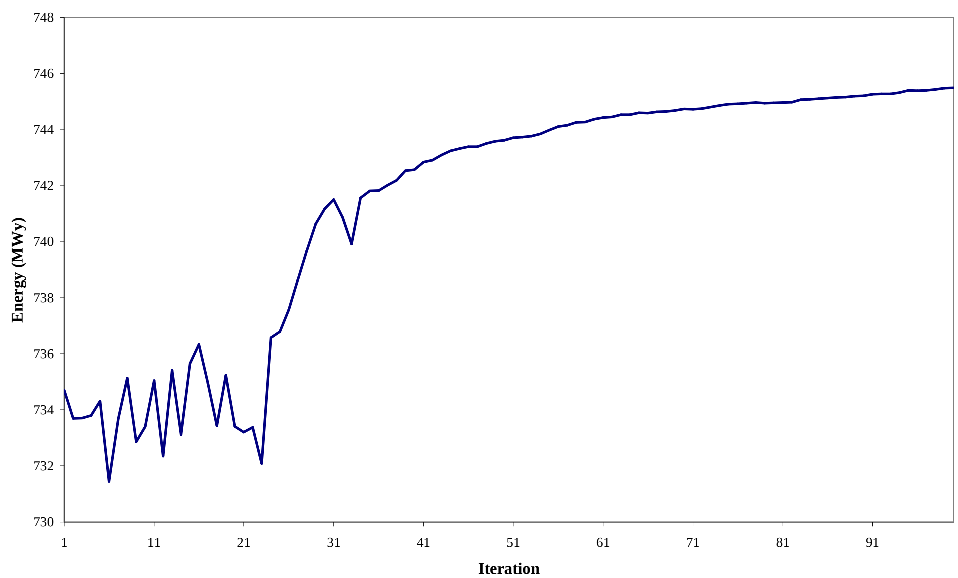


Figure 6.32: Averaged expected total final energy in storage (IB-ETFEISavg) associated with iteration-best schedules

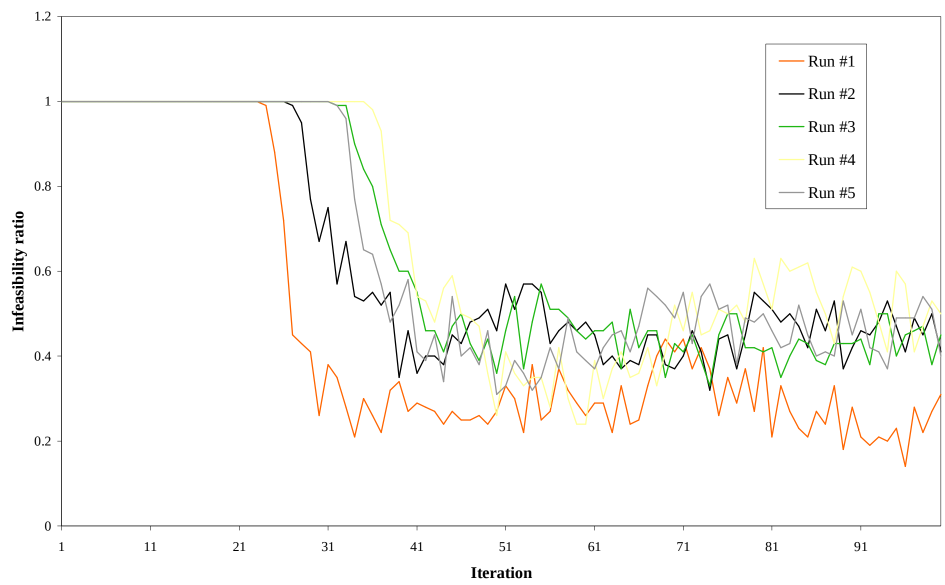


Figure 6.33: Infeasibility ratio for scenario-A runs

6.3.4.2 Scenario B

As a result of the increased demand, the best schedule found by ACO-PPMSO is infeasible (*EUE* > 0, *ResVio* = 5 days, Table 6.10) if all maintenance tasks must be completed within the planning horizon. On the other hand, if the options of shortening and deferral of maintenance tasks are considered, a feasible solution is obtained under the condition that a total of 99 days of outage is shortened/deferred (Table 6.10). It is clearly seen that the schedule obtained when shortening and deferral options are considered is more realistic, as demand shortfalls should be avoided in power systems, while maintenance tasks can be shortened by employing more staff or deferring the task to the next planning horizon. In the case of a 5% increase of demand, which is likely to occur as population grows, it is more practical to incorporate shortening and deferral options when scheduling for maintenance.

In addition, it can be seen that as demand increases, the expected total final energy in storage associated with the schedule obtained is lower, as a result of increased water usage for meeting the higher demand.

The behaviour of ACO-PPMSO in the scenario-B run when shortening and deferral were not considered is also examined. The IB-OFC, which improves in steps from around iterations 40 to 140 (Figure 6.34), acquires the form of the IB-ResVio curve (Figure 6.35), as the main driver of the optimisation run during that stage is the minimization of reserve constraint violation.

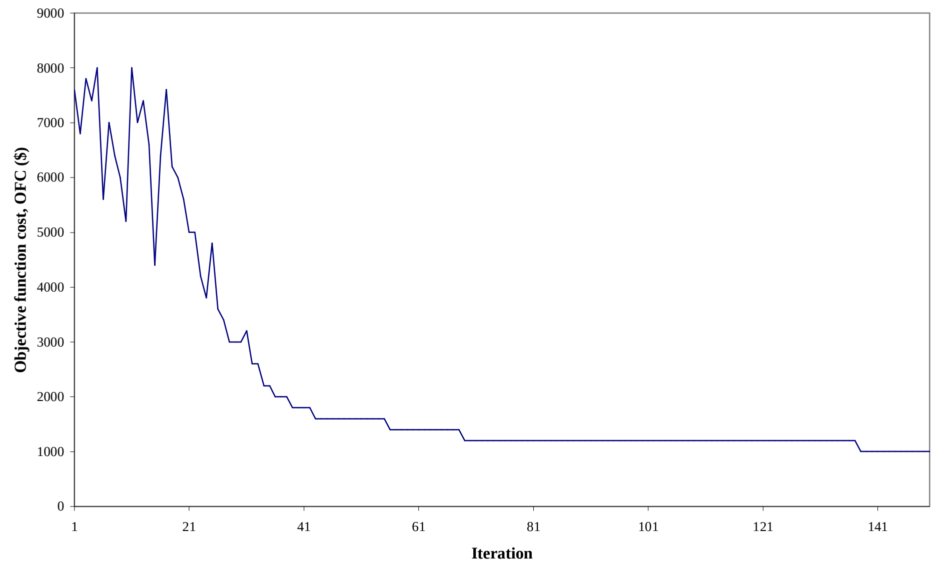


Figure 6.34: IB-OFC curve for scenario-B run

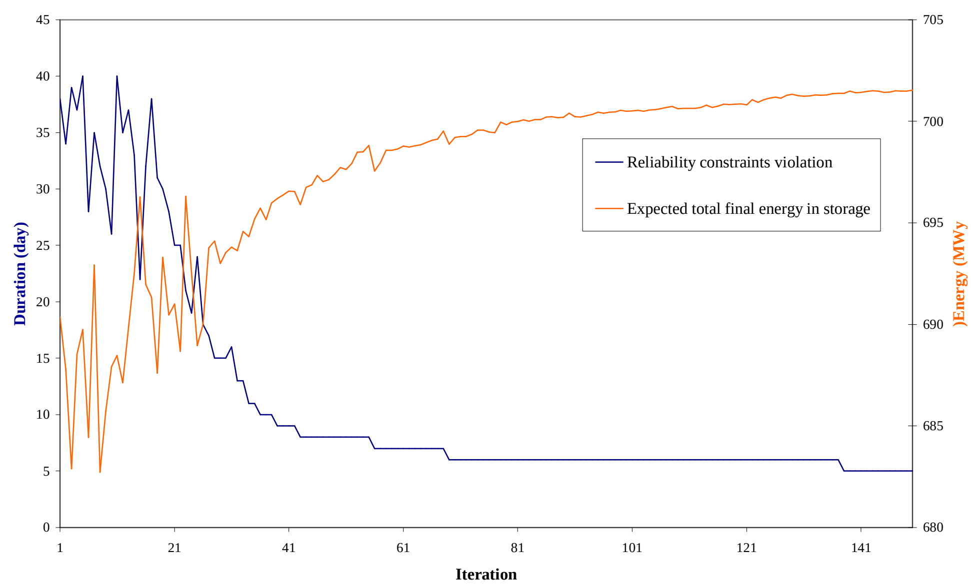


Figure 6.35: IB-ResVio and IB-ETFEIS for scenario-B run

When the optimisation run is performed again by considering the options of duration shortening and deferral (Figures 6.36 to 6.38), feasible solutions are found within the first 22 iterations (Figure 6.37). The algorithm then seems to minimize the IB-DurCut values at a constant rate for the next 40 iterations, followed by a 30-iteration stagnation before an abrupt improvement is made just before the 100th iteration (Figure 6.37). The abrupt improvement in the total reduction of maintenance shortening and deferral is found to take place due to a change from a deferral to a shortening decision (93 days) for a maintenance task, which normally takes 213 days to perform. Figure 6.38 depicts the maximization of IB-ETFEIS values from around iteration 60, when iteration-best schedules found are feasible and IB-DurCut values are hardly improving.

For the best schedule obtained in this run, a maintenance task that normally requires 213 days is shortened by 93 days and two maintenance tasks are deferred (Table 6.14). The objective values and satisfaction of various constraints associated with the best schedule are given in Table 6.10.

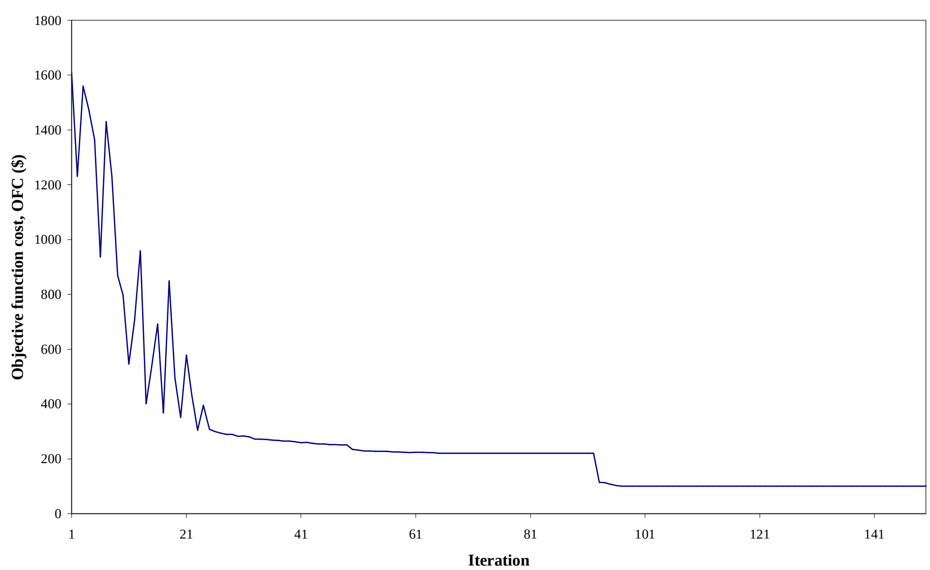


Figure 6.36: IB-OFC for scenario-B run (considering shortening and deferral)

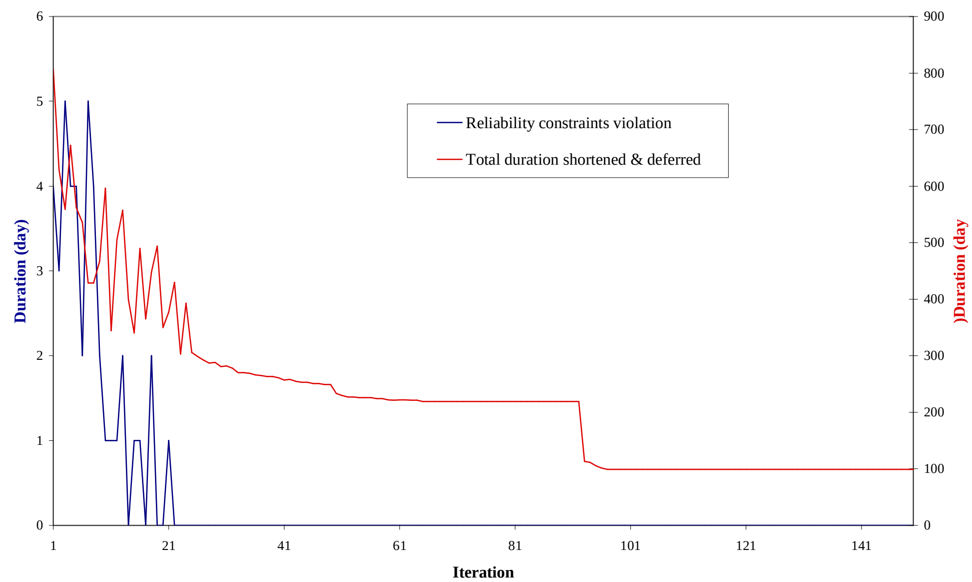


Figure 6.37: IB-ResVio and IB-DurCut for scenario-B run (considering shortening and deferral)

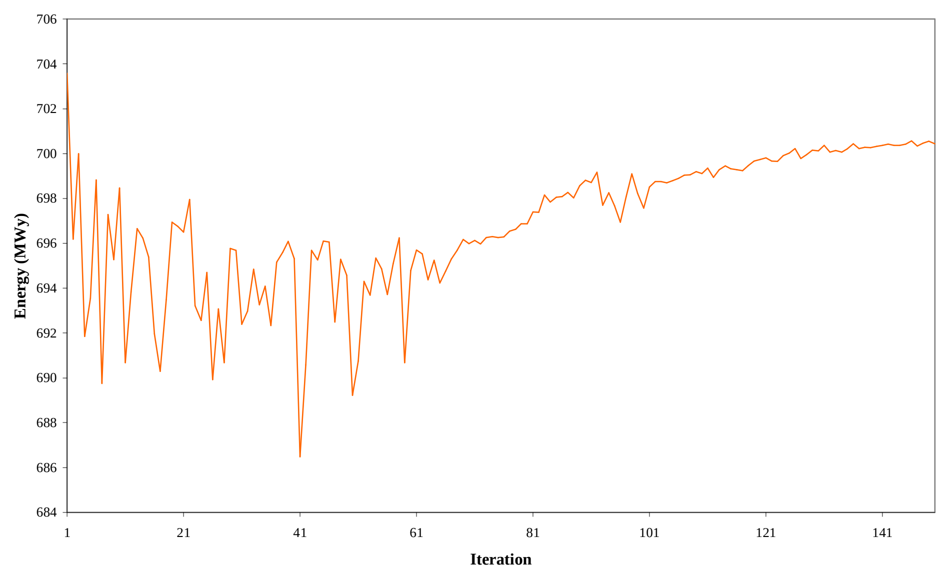


Figure 6.38: IB-ETFEIS for scenario-B run (considering shortening and deferral)

Table 6.14: Affected maintenance tasks as a result of increased system demand

Station	Task ID	Machine	Normal outage duration (days)	Decision	New outage duration (days)
Gordon	30072	M1	213	Shortened	120
Gordon	33601	M1, M2 & M3	3	Deferred	0
Gordon	31002	M2 & M3	3	Deferred	0

6.3.4.3 Scenario C

The schedule obtained by ACO-PPMSO for this scenario (referred to as ACO schedule–C hereafter) results in an average violation of reserve constraints for one day (Table 6.10), which is acceptable in terms of maintenance according to Hydro Tasmania's current practice. The utility of ACO schedule-A for maintenance despite the unavailability of the Gordon M1 machine (without re-optimisation using ACO-PPMSO) was checked. It was found that an average violation of reserve constraints for one day is associated with ACO schedule-A when used for scenario C. It can be seen from Table 6.15 that despite the similarity in the resulting reserve constraint violations, the total final energy in storage associated with ACO schedule-A used for scenario C is much lower than that of the schedule optimized for this scenario (ACO schedule-C). A higher final energy in storage is highly desirable as it increases the future reliability of the power system, which also provided additional security of the power system under the uncertainties of system demands and storage inflows in the next planning horizon. Therefore, re-optimisation should be done with ACO-PPMSO when there is a change in the condition of a power system to find the maintenance schedule that best suits the current scenario.

Table 6.15: Summary of the utility of two different schedules obtained by ACO-PPMSO for scenario C

	ACO schedule-A	ACO schedule-C (as in Table 6.10)
Expected total final energy in storage (MWy)	744.93	745.45
Expected unseved energy (MWy)	0.000	0.000
Violation of reserve constraints (days)	1	1

The results of the optimisation run carried out for this scenario indicate that feasible schedules exist despite the absence of the Gordon M1 machine (Table 6.10). The iteration-best schedules that satisfy reliability constraints evolved after about 30 iterations, when the maximization of ETFEIS takes place (Figure 6.39).

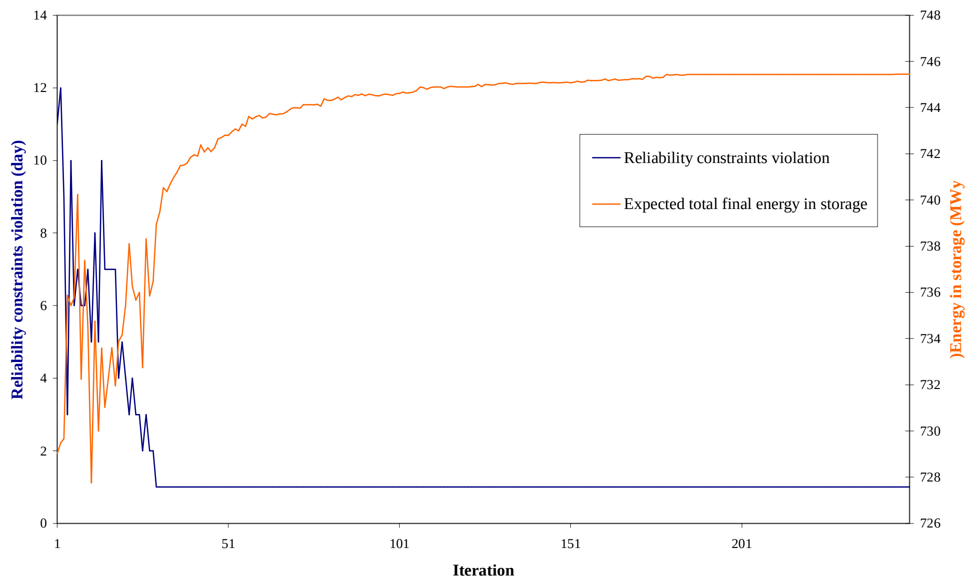


Figure 6.39: IB-ResVio and IB-ETFEIS for scenario-C run

6.3.4.4 Scenario D

The results of ACO-PPMSO obtained for scenario D with and without considering the minimum disruption to ACO schedule-A are summarized in Table 6.10. As a result of the review-optimisation run, in which minimum deviation from the original schedule is considered, the revised schedule obtained deviates from the original schedule by a total of 35 days (Table 6.10). In particular, the original schedule (ACO schedule-A, Table 6.11) is revised by changing the start time of three maintenance tasks, as given in Table 6.16. As a result of the change of start dates, the violation of reserve constraints is reduced to two days (compared to five days associated with the original schedule considering the late return of Tungatinah station).

Table 6.16: Revised start dates as a result of the late return of the Tungatinah station

Station	Task ID	Original start date	Revised start date	Deviation (day)
Poatina	32512	14 Apr 2006	15 Mar 2006	30
Paloona	30558	20 Apr 2006	22 Apr 2006	2
Poatina	31446	26 Jul 2006	27 Jul 2006	1
Tarraleah	32940	7 Aug 2006	8 Aug 2006	1
Devils Gate	33463	12 Mar 2006	11 Mar 2006	1

The optimisation outcome using ACO-PPMSO reveals that when the deviation from the original schedule is not considered, the best schedule obtained is feasible despite violation of reserve constraints by one day (Table 6.10). However, the new schedule obtained differs by a total of 2712 days from the original schedule. Although the schedule obtained from the new run is slightly better in terms of meeting reserve constraints and energy in storage, it is unfavourable from a practical point of view due to the high degree of disruption to the original schedule. The comparison clearly demonstrates the benefits of considering minimum deviation from the original schedule in circumstances where changes to an optimized schedule are inevitable due to an unexpected event.

The behaviour of ACO-PPMSO in finding a revised schedule considering minimum deviation from the original schedule is illustrated in Figures 6.40 to 6.42. It can be observed that the iteration-best OFC improves abruptly within the first 26 iterations (Figure 6.40) corresponding to reduced reliability constraint violations (Figure 6.41). The IB-dev curve oscillates despite the overall decreasing trend in the first 26 iterations and improves more consistently at a much lower rate in the following 50 iterations before stagnation occurs (Figure 6.42).

An interesting observation made from the optimisation run for this scenario is that in contrast to that for scenario A, the IB-ETFEIS plot did not increase (Figure 6.41) throughout this run. This could be due to one or both of the following reasons: (1) The original schedule used for this scenario is already maximized with respect to ETFEIS; (2) The minimization of deviation from the original schedule is weighted relatively higher than the maximization of ETFEIS in the objective function (Eq. 6.6). Therefore, the optimisation process is mainly driven by the minimization of total deviation from the original schedule.

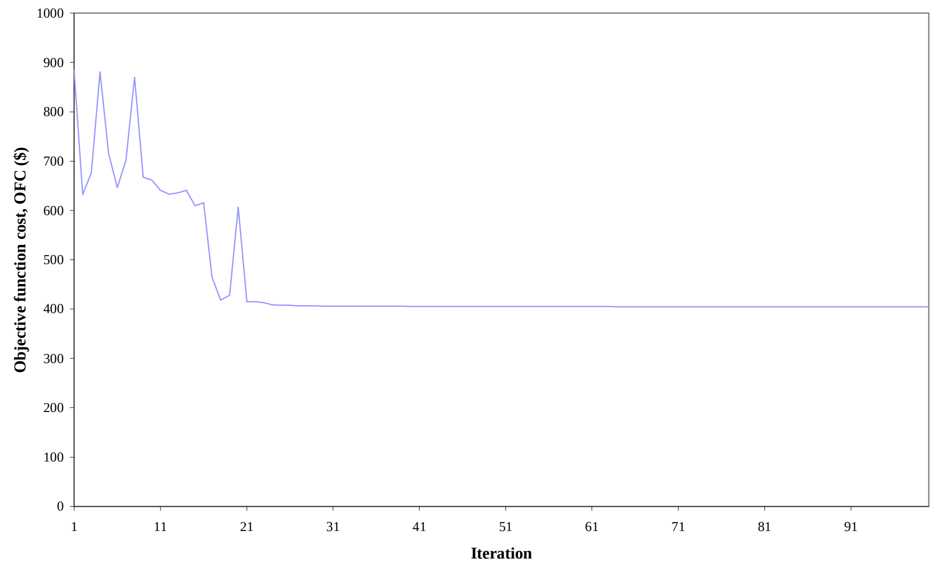


Figure 6.40: IB-OFC for scenario-D run

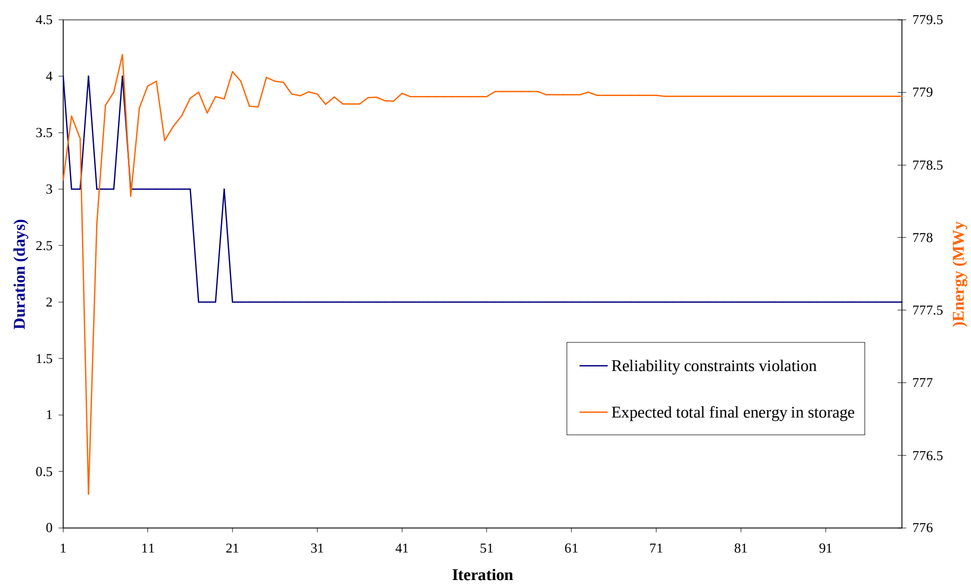


Figure 6.41: IB-ResVio and IB-ETFEIS for scenario-D run

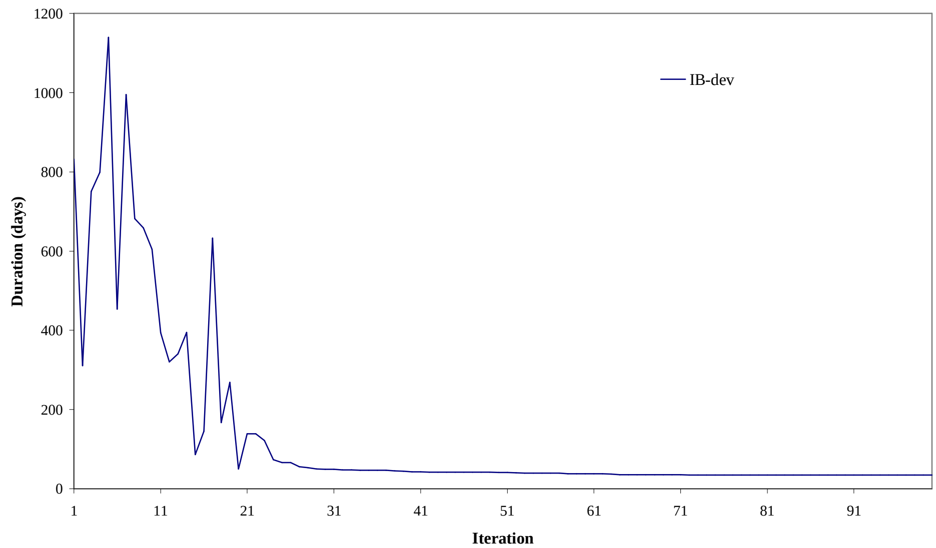


Figure 6.42: IB-dev for scenario-D run

6.4 Summary and conclusions

In this chapter, the new ACO-PPMSO formulation developed as part of this research has been implemented in solving two real-world maintenance scheduling case studies of different sizes and complexities. Firstly, a five-station hydropower case study system derived from the Hydro Tasmania system has been utilized to test the utility of ACO-PPMSO in solving real-world PPMSO problems. In particular, a test procedure was set up to check the utility of the shortening and deferral options, the impacts of the *Duration Extender* local search operators and the overall performance of ACO-PPMSO in solving a real-world maintenance scheduling problem. The test results indicated that the ability of ACO-PPMSO to incorporate shortening and deferral options is useful for producing practical maintenance schedules. In addition, the *Duration Extender* local search operator developed in this research was also shown to significantly improve the results obtained by ACO-PPMSO. The overall performance of ACO-PPMSO was shown to be promising when the results obtained were compared to those found by a maintenance scheduler and a random evaluation method.

The ACO-PPMSO was also applied to schedule maintenance for the full Hydro Tasmania system in 2006, involving the scheduling of 118 maintenance tasks within a planning horizon of 365 days. Four different scenarios that resemble those commonly encountered by a maintenance scheduler have been used in the case study, in which ACO-PPMSO was used to assist in making decisions. For a routine maintenance scenario, the schedule obtained by ACO-PPMSO was compared to a schedule used by Hydro Tasmania. It was found that the ACO schedule is superior in terms of both satisfaction of reserve constraints and total final energy in storage. A scenario, in which system demand is increased by 5%, was considered. The results obtained by ACO-PPMSO reveal that feasible schedules were found only if shortening and deferral options were considered. Shortening and deferral of maintenance tasks are common practice in the management of a power system during adverse conditions such as low storage inflows and higher-than-expected demand, for example. The incorporation of these options in the ACO-PPMSO formulation allows practical maintenance schedules to be obtained. In the third scenario, the unavailability of the Gordon M1 machine was assumed. Despite the loss of a major generating machine, a feasible maintenance schedule was obtained. In addition, the new ACO schedule obtained by re-running ACO-PPMSO considering the loss of the Gordon machine results in a better future system reliability (higher total final energy in storage). The results obtained for this scenario demonstrate the robustness of ACO-PPMSO in obtaining the best schedule under a changed system condition. In the fourth scenario, Tungatinah station was assumed to return late from maintenance. ACO-PPMSO was used to provide a maintenance schedule for the scenario with and without considering minimum disruption to an existing schedule, for which arrangements have already been made in relation to personpower and machinery. Despite a slight trade-off in reserve capacity, the schedule obtained deviates only by 35 days when minimization of disruption to the existing schedule was considered in the ACO-PPMSO run, which is much less than the total deviation of over 2000 days associated with the schedule obtained when minimum disruption was not used as an optimisation criterion. Scenario-D results clearly demonstrate the flexibility of the ACO-PPMSO formulation in catering for different optimisation objectives. The behaviour of ACO-PPMSO during the optimisation runs performed for the four scenarios have also been examined. It was found that in the early stages of the optimisation runs, the ACO-PPMSO searches randomly before the pheromone profile of the search space was built up, after which many infeasible solutions were found. After a number of iterations, when feasible solutions evolved, the algorithm searches more in the feasible regions of the search space for trial solutions that result in better objective function costs. The experimental results obtained for the four scenarios investigated show the robustness of the ACO-PPMSO formulation in handling different real-world circumstances, thus indicating the potential of the approach as an operational tool.

Chapter 7
Summary, Conclusions and Recommendations

In this chapter, a summary of the research and conclusions corresponding to the objectives specified in Section 1.2 are given, followed by recommendations for further work. A list of papers published and accepted as a result of this research is also presented.

7.1 Summary & Conclusions

The formulation of the power plant maintenance scheduling optimisation (PPMSO) problem has been generalized through interactions with maintenance scheduling practitioners. In addition to the conventional formulation of a PPMSO problem, where only the commencement times of maintenance tasks are determined, the duration of maintenance tasks are also considered in the generalized formulation. The availability of maintenance shortening and deferral allows maintenance scheduling to be solved in a more practical way.

A formulation that enables Ant Colony Optimisation (ACO) to be applied to the generalized power plant maintenance scheduling optimisation (PPMSO) problem has been developed. Several issues with regard to the practical utilization of the proposed formulation have been resolved. These include inclusion of heuristic information, a local search strategy and constraint-handling techniques.

Heuristic: As part of the new formulation, ACO-PPMSO, a new heuristic has been utilized to improve the performance of ACO by incorporating user's knowledge about the system. As an optional feature in the ACO-PPMSO algorithm, the heuristic formulation is used to construct, by higher probability, good-quality trial solutions at the early stage of an ACO run. In this way, optimum or near-optimal solutions can be determined in reduced computational time.

Local search: A new local search strategy, namely the *Duration Extender* has also been proposed for PPMSO problems, which allows shortening and deferral of maintenance activities. The local search strategy is designed to conduct a more refined search within the neighborhood of iteration-best maintenance schedules given by ACO, which are obtained using pheromone and heuristics.

Constraint handling: Constraints commonly encountered in PPMSO have been categorized based on whether they can be accounted for during the construction of a trial solution and whether they can be violated to achieve better objective values. Techniques for handling different constraint types have been proposed correspondingly. In particular, an advantage of using ACO for PPMSO is the possibility of incorporating some constraints during the construction of trial solutions, eliminating the need for complicated penalty functions in the formulation.

Written in the Fortran 90 language, the ACO-PPMSO formulation has been tested on four benchmark case studies (the 21- and 22-unit case studies, as well as the modified version of the two case studies). A three-stage testing procedure was implemented for this purpose. For all four case studies, the results obtained using ACO with the new heuristic formulation were found to be significantly better than those without heuristic. On the other hand, the impact of the *Duration Extender* local search strategy was found to be insignificant in some test runs, but significantly improved ACO performance in other runs. Overall, the results obtained by ACO-PPMSO were better than those obtained by other metaheuristics previously applied to the original 21- and 22-unit case studies.

The new ACO-PPMSO formulation has been applied to real-world maintenance scheduling problems, including a five-station hydropower system and the full Hydro Tasmania system. Various practical issues with respect to the application of ACO-PPMSO to real system, such as the implementation of constraints, have been addressed. When tested with the five-station hydropower system, the *Duration Extender* local search strategy was shown to significantly improve the results obtained by ACO. Test results with both case systems indicated that the ability of ACO-PPMSO to incorporate shortening and deferral options is useful for producing practical maintenance schedules. For both case systems, the overall performance of ACO-PPMSO was shown to be promising when the results obtained were compared with those found by a maintenance scheduler and a random evaluation method.

To summarize, an ACO formulation for power plant maintenance scheduling problems has been developed, tested and applied to real systems. Based on the results obtained, it can be concluded that the developed formulation:

- Determines good, if not the optimum, maintenance schedule of a case system within reasonable computational runtime.

- Provides a valuable tool for assisting maintenance schedulers, especially when complicated objective functions (e.g. profit margin) are involved in the scheduling process.

- Requires no prior experience in maintenance scheduling.

- Produces alternative maintenance schedules of similar quality (in terms of given criteria) for negotiation purposes.

- Handles changes in a power system (e.g. expansion) relatively easily.

7.2 Recommendations for future work

Despite the strengths of the new ACO-PPMSO formulation, there are possibilities that the formulation can be further improved by additional work, such as:

1. ***Real costs in objective function***

   The objective function used in the proposed ACO-PPMSO formulation is a function of objective and selected constraint terms (as detailed in Section 4.3.3). The objective function cost (OFC) associated with a maintenance schedule should ideally be the real cost incurred if the scheduled is implemented. Optimisation criteria such as maintenance costs and penalty costs due to demand shortfalls are usually known and hence are the real costs. On the other hand, total final energy in storage is a reflection of the storage status of a system, such that, a maintenance schedule is better if the total energy in storage at the end of planning horizon is higher. In other words, there is potential revenue derived, but not a cost incurred, due to the storage position at the end of a scheduling planning horizon. In the current ACO-PPMSO formulation, the 'cost' (in the objective function) associated with such criteria is represented by the reciprocal of the potential revenue derived from the total final energy in storage associated with the schedule. As a result, the real value of the total final energy in storage cannot be truly reflected in the optimisation process, although the real dollars derived from per unit storage maybe known. The current version of the ACO-PPMSO formulation can be further improved with regard to this shortcoming.

2. ***Multi-objective optimisation***

In the current ACO-PPMSO formulation, different weights are used in the objective function when multiple optimisation criteria are considered. This method works without difficulties only if the decision maker knows exactly the importance of (ideally the real cost associated with) each criterion. Otherwise, if such information is not available and the decision maker is interested in finding the Pareto-optimal solution set to the problem, the current formulation is deemed insufficient. ACO is a population-based metaheuritsic that generates multiple trial solutions (if desired) at each timestep (iteration) taken during the optimisation process. The metaheuristic is therefore capable of finding a set of Pareto-optimal solutions to a given multi-objective problem in a single run, without having to do multiple runs using different combinations of weights given to each objective. Related studies that have researched multi-objective optimisation using ACO include Mariano *et al*. (1999), Iredi *et al*. (2001) and Guntsh *et al*. (2003). The methods developed by these studies, or other studies in the future, maybe incorporated into the current version of the ACO-PPMSO formulation.

3. ***Uncertainty analysis***

The developed ACO-PPSMSO model utilizes histotrical system demands and inflow data for the optimisation process, eliminating the need for uncertainty anaylysis of the optimisation outcome. However, uncertainties associated with the optimized schedules produced as a result of the optimisation model should be investigated when forecasted data is used instead.

7.3 Published and accepted papers

The following publications have been produced as a result of this research:

Book chapter

Foong W., Maier H.R. and Simpson A.R. (To appear) Ant colony optimisation for power plant maintenance scheduling. *Swarm Intelligence: Focus on Ant and Particle Swarm Optimisation.*

Journal papers

Foong W., Simpson A.R. and Maier H.R. (To appear) Ant colony optimisation for power plant maintenance scheduling – A five-station hydropower system. *Annals of Operations Research.*

Foong W., Maier H.R. and Simpson A.R. (To appear) Ant colony optimisation for power plant maintenance scheduling – An improved formulation. *Engineering Optimisation.*

Conference papers (Refereed)

Foong W., Maier H.R. and Simpson A.R. (2005) Ant colony optimisation for power plant maintenance scheduling optimisation. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005)*. Washington D.C., USA. June 25 - 29, Vol. 1, pp.249-256.

Foong W., Maier H.R. and Simpson A.R. (2005) Ant colony optimisation for power plant maintenance scheduling optimisation. *Proceedings of the*

Graduate Student Workshop at the 2005 Genetic and Evolutionary Computation Conference (GECCO-2005). Washington D.C., USA. June 25 - 29.

Conference papers (Refereed by abstract)

Foong W., Maier H.R. and Simpson A.R. (2005) Ant colony optimisation for power plant maintenance scheduling optimisation: Simplified Hydro Tasmania system. *2nd Multidisciplinary International Conference on Scheduling: Theory & Application (MISTA-2005)*. New York, USA. July 18 - 21.

Chapter 8
References

Abdulwhab, A., R. Billinton, A. A. Eldamaty and S. O. Faried (2004). Maintenance scheduling optimisation using a genetic algorithm (GA) with a probabilistic fitness function. *Electric Power Components and Systems* 32: 1239-1254.

Afshar, M. H. (2006). Application of a max-min ant system to joint layout and size optimisation of pipe networks. *Engineering Optimisation* 38(3): 299-317.

Agostini, F. P., D. D. O. Soares-Pinto, M. A. Moret, C. Osthoff and P. G. Pascutti (2006). Generalized simulated annealing applied to protein folding studies. *Journal of Computational Chemistry* 27(11): 1142-1155.

Ahmad, A. and D. P. Kothari (2000). A practical model for generator maintenance scheduling with transmission constraints. *Electric Machines and Power Systems* 28: 501-513.

Al-Shihabi, S. (2004). Backtracking ant system for the Traveling Salesman Problem. *Ant Colony Optimisation and Swarm Intelligence, Proceedings*. 3172**:** 318-325.

Aldridge, C. J., K. P. Dahal and J. R. McDonald (1999). Genetic algorithms for scheduling generation and maintenance in power systems. *Modern Optimisation Techniques in Power Systems*. Y.-H. Song. Dordrecht ; Boston, Kluwer Academic Publishers**:** 63-89.

Arzamascev, D. A. and V. P. Oboskalov (1970). Determination of the Plan of Basic Repairs of Equipment of Power Stations Using Co-ordinate Optimisation. *Energetika* 8.

Basu, M. (2005). A simulated annealing-based goal-attainment method for economic emission load dispatch of fixed head hydrothermal power systems. *International Journal of Electrical Power and Energy Systems* 27(2): 147-153.

Baykasoglu, A. and T. Gocken (2006). A tabu search approach to fuzzy goal programs and an application to aggregate production planning. *Engineering Optimisation* 38(2): 155-177.

Beswick, R., L. Brown, P. Clark, S. Fama, S. Gamble, D. Jamrozik, A. Masters, D. Wheeler and C. Viney (2003). Hydro Tasmania - An overview. *The power of nature*.

Billinton, R. (1991). Criteria used by Canadian utilities in the planning and operation of generating capacity. *Applied Reliablity Assessment in Electric Power Sstems, New York: IEEE Press***:** 186-191.

Billinton, R. and M. Fotuhi-Firuzabad (1996). Reserve capacity assessment in small isolated electric power generating systems. *IEE Power Engineering Journal* 10(2): 73-80.

Blazewicz, J., P. Lukasiak and M. Milostan (2005). Application of tabu search strategy for finding low energy structure of protein. *Artificial Intelligence In Medicine* 35(1-2): 135-145.

Blum, C. (2002). ACO applied to group shop scheduling: A case study on intensification and diversification. *ANTS 2002 - From Ant Colonies to Artificial Ants: Third International Workshop on Ant ALgorithms*, Berlin, Springer Verlag.

Blum, C. and A. Roli (2003). Metaheuristics in combinatorial optimisation: Overview and conceptual comparison. *ACM Computing Surveys* 35(3): 268-308.

Blum, C., A. Roli and M. Dorigo (2001). HC-ACO: The hypercube framework for Ant Colony Optimisation. *Proceedings of MIC'2001 - Metaheuristics International Conference*.

Bullnheimer, B., R. F. Hartl and C. Strauss (1999). A new rank-based version of the Ant System: A computational study. *Central European Journal for Operations Research and Economics* 7(1): 25-38.

Burke, E. K., J. A. Clarke and A. J. Smith (1998). Four methods for maintenance scheduling. *Third International Conference on Artificial Neural Nets and Genetic Algorithms (ICANNGA '97)*, Norwich, Vienna, Springer Verlag.

Burke, E. K. and A. J. Smith (2000). Hybrid Evolutionary Techniques for the Maintenance Scheduling Problem. *IEEE Transactions on Power Systems* 15(1): 122-128.

Caputo, A. C., L. Fratocchi and P. M. Pelagagge (2006). A genetic approach for freight transportation planning. *Industrial Management & Data Systems* 106(5-6): 719-738.

Chang, H. and W. C. Hou (2006). Optimisation of membrane gas separation systems using genetic algorithm. *Chemical Engineering Science* 61(16): 5355-5368.

Chattopadhyay, D. (1998). A practical maintenance scheduling program: Mathematical model and case study. *IEEE Transactions on Power Systems* 13(4): 1475-1480.

Chattopadhyay, D., K. Bhattacharya and J. Parikh (1995). A systems approach to least-cost maintenance scheduling for an interconnected power system. *IEEE Transactions on Power Systems* 10(4): 2002-2007.

Chen, L. N. and J. Toyoda (1990). Maintenance scheduling based on two-level hierarchical structure to equalize incremental risk. *IEEE Transactions on Power Systems* PWRS-5: 1510-1516.

Christiaanse, W. R. and A. H. Palmer (1971). A technique for the automated scheduling of the maintenance of generating facilities. *PICA Conference Proceedings*.

Christiaanse, W. R. and A. H. Palmer (1972). A technique for the automated scheduling of the maintenance of generating facilities. *IEEE Transactions on Power Apparatus and Systems* PAS-91: 137-144.

Coello Coello, C. A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput. Methods Appl. Mech. Engrg*(191): 1245-1287.

Colorni, A., M. Dorigo, V. Maniezzo and M. Trubian (1994). Ant system for job-shop scheduling. *Belg. J. oper. Res. Stat. Comp. Sci.* 34(1): 39-53.

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

Dahal, K. P. and J. R. McDonald (1997). Genetic maintenance scheduling of electric power systems using genetic algorithms with integer representation. *2nd International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA '97)*.

Dahal, K. P. and J. R. McDonald (1998). Generational and steady state genetic algorithms for generator maintenance scheduling problems. *Third International Conference on Artificial Neural Nets and Genetic Algorithms (ICANNGA '97)*, Norwich, Vienna, Springer Verlag.

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

de Franca, F. O., F. J. Von Zuben and L. N. de Castro (2004). Definition of capacited p-medians by a modified max min ant system with local search. *Neural Information Processing*. 3316**:** 1094-1100.

den Besten, M., T. Stützle and M. Dorigo (2000). Ant Colony Optimisation for the total weighted tardiness problem. *Proceedings of Parallel Problem Solving from Nature (PPSN-VI)*, Springer Verlag.

Deneubourg, J.-L., J. M. Pasteels and J. C. Verhaeghe (1983). Probabilistic behaviour in ants: a strategy of errors? *Journal of Theoretical Biology* 105: 295-311.

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

Dorigo, M. (1992). Optimisation, Learning and Natural Algorithms. *Dipartimento di Elettronica*. Italy, Politechnico di Milano**:** 140.

Dorigo, M., C. Blum and M. Manfrin. (2004a, Aug 2004). Metaheuristic Network. Retrieved 26 Aug 2006, 2006.

Dorigo, M., G. Di Caro and L. M. Gambardella (1999). Ant algorithms for distributed discrete optimisation. *Artificial Life* 5: 137-172.

Dorigo, M. and L. M. Gambardella (1997a). Ant colonies for the travelling salesman problem. *BioSystems* 43: 73-81.

Dorigo, M. and L. M. Gambardella (1997b). Ant Colony System: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* 1: 53-66.

Dorigo, M., V. Maniezzo and A. Colorni (1991). Positive feedback as a search strategy. Milan, Dip. Elettronica, Politecnico di Milano, Italy.

Dorigo, M., V. Maniezzo and A. Colorni (1996). Ant system: optimisation by a colony of cooperating ants. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics* 26: 29-42.

Dorigo, M. and T. Stützle (2002). The ant colony optimisation metaheuristic: Algorithms, applications and advances. *Handbook of Metaheuristics*. F. Glover and G. Kochenberger. Norwell, MA, Kluwer Academic Publishers. 57**:** 251-285.

Dorigo, M. and T. Stützle (2004b). Ant Colony Optimisation. Cambridge, MA, MIT Press.

Downsland, K. A. (1993). Simulated annealing. *Modern heuristic techniques for combinatorial problems*. C. R. Reeves. Oxford, Blackwell Scientific Publications**:** 20-69.

Egan, G. T., T. S. Dillon and K. Morsztyn (1976). An Experimental Method of Determination of Optimal Maintenance Schedules in Power Systems Using the Branch-and-Bound Technique. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-6(8): 538-546.

Ekwue, A. O. (1999). Industrial Applications Of Artificial Intelligence Techniques. *Modern optimisation techniques in power systems*. Y.-H. Song. Dordrecht ; Boston, Kluwer Academic Publishers**:** 261-275.

El-Amin, I., S. Duffuaa and M. Abbas (2000). A tabu search algorithm for maintenance scheduling of generating units. *Electric Power Systems Research* 54: 91-99.

El-Zonkoly, A. (2006). Optimal meter placement using genetic algorithm to maintain network observability. *Expert Systems with Applications* 31(1): 193-198.

Escudero, L. F., J. W. Horton and J. E. Scheiderich (1980). On maintenance scheduling for energy generators. *IEEE Winter Power Meeting*, New York.

Foong, W. K., H. R. Maier and A. R. Simpson (2005a). Ant Colony Optimisation (ACO) for Power Plant Maintenance Scheduling Optimisation (PPMSO). *GECCO 2005: Proceedings of the Genetic and Evolutionary Computation Conference*, Washington D.C., USA.

Foong, W. K., K. Y. Phang, H. Y. Seah and C. L. Tan (2000). Ant Colony Optimisation for Pipe Network Distribution System: Final Year Research Report. Adelaide, School of Civil & Environmental Engineering, University of Adelaide.

Foong, W. K., A. R. Simpson and H. R. Maier (2005b). Ant Colony Optimisation for Power Plant Maintenance Scheduling Optimisation - A Five-Station Hydropower System. *Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA-2005)*, New York, USA.

Gambardella, L. M. and M. Dorigo (1995). Ant-Q: A reinforcement learning approach to the travelling salesman problem. *Proceeedings of LM-5, Twelfth International Conference on Machine Learning*, Morgan Kaufmann Publisher, San Francisco, CA, USA.

Garver, L. L. (1972). Adjusting maintenance schedules to levelize risk. *IEEE Transactions on Power Apparatus and Systems* 91(5): 2057-2063.

Gen, M., F. Altiparmak and L. Lin (2006). A genetic algorithm for two-stage transportation problem using priority-based encoding. *OR Spectrum* 28(3): 337-354.

Glover, F. (1989). Tabu Search, Part I. *ORSA Journal on Computing* 1(3): 190-206.

Glover, F. and M. Laguna (1997). Tabu Search. Boston, Kluwer Academic Publishers.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimisation and Machine Learning. Massachusetts, Addison-Wesley Publishing Company.

Goldberg, D. E. and C. H. Kuo (1987). Genetic algorithms in pipeline optimisation. *Journal of Computing in Civil Engineering, ASCE* 1(2): 128-141.

Goldman, F. E. and L. W. Mays (2000). The application of simulated annealing to the optimal operation of water systems. *29th Annual Water Resources Planning and Management Conference*, Tempe, Arizona, USA, ASCE.

Guntsh, M. and M. Middendorf (2003). Solving Multi-critera Optimisation Problems with Population-Based ACO. *Evolutionary Multi-Criterion Optimisation, Second International Conference (EMO'03)*, Faro, Portugal.

Hajji, O., S. Brisset and P. Brochet (2005). A new tabu search method for continuous parameter optimisation: Application to design problems in electromagnetic. *European Transactions on Electrical Power (Special Issue SI)* 15(6): 527-540.

Halhal, D., G. A. Walters, D. Ouazar and D. A. Savic (1997). Water network rehabilitation with structured messy genetic algorithms. *Journal of Water Resources Planning and Management, ASCE* 123(3): 137-146.

Hodge, B. M., F. Pettersson and N. Chakrabarti (2006). Re-evaluation of the optimal operating conditions for the primary end of an integrated steel plant using multi-objective genetic algorithms and Nash equilibrium. *Steel Research International* 77(7): 459-461.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems, University of Michigan Press.

Hoover, W. M., J. Toyoda and M. Chen (1976). A new operation economics approach applied to the realistic scheduling of large-thermal machine downtime. *Control of Power Systems Conference and Exposition*, Oklahoma City, Oklahama.

Huang, S.-J. (1998). A Genetic-Evolved Fuzzy System For Maintenance Scheduling Of Generating Units. *Electrical Power & Energy Systems* 20(3): 191-195.

Hydro Tasmania. Storage levels. Retrieved January 23,2007, 2007.

Iredi, S., D. Merkle and M. Middendorf (2001). Bi-criterion optimisation with multi colony ant algorithms. *Evolutionary Multi-Criterion Optimisation, First International Conference (EMO'01)*, Berlin, Heidelberg, Springer Verlag.

Jackson, N. O. (2005, 9 Jun 2005). Tasmania's population. Retrieved 22 Aug 2006, 2006.

Jubai, A., B. Jing and J. Yang (2006). Combining fuzzy theory and a genetic algorithm for satellite image edge detection. *International Journal of Remote Sensing* 27(14): 3013-3024.

Kazantzis, M. D., A. R. Simpson, D. Kwong and S. M. Tan (2002). A new methodology for optimizaing the daily operations of a pumping plant. *Proceedings of 2002 Conference on Water Resources Planning*, Roanoke, USA, ASCE.

Kim, H., Y. Hayashi and K. Nara (1997). An algorithm for thermal unit maintenance scheduling through combined use Of GA, SA And TS. *IEEE Transactions on Power Systems* 12(1): 329-335.

Kim, H. and K. Nara (1995). A method for maintenance scheduling using GA combined with SA. *Trans. IEE Japan* 115-B(11).

Kim, H., K. Nara and M. Gen (1994). A method for maintenance scheduling using GA combined with SA. *Computers and Industrial Engineering*(27): 477-480.

Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi (1983). Optimisation by Simulated Annealing. *Science*(220): 671-680.

Kumar, Y., B. Das and J. Sharma (2006). Service restoration in distribution system using non-dominated sorting genetic algorithm. *Electric Power Systems Research* 76(9-10): 768-777.

Lei, D. M. and Z. M. Wu (2006). Tabu search for multiple-criteria manufacturing cell design. *International Journal of Advanced Manufacturing Technology* 28(9): 950-956.

Li, B., L. P. Chen, Z. D. Huang and Y. F. Zhong (2006). Product configuration optimisation using a multiobjective genetic algorithm. *International Journal of Advanced Manufacturing Technology* 30(1-2): 20-29.

Liao, G. C. and T. P. Tsao (2006). Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting. *IEEE Transactions on Evolutionary Computation* 10(3): 330-340.

Lin, C. E., C. J. Huang, C. L. Huang, C. C. Liang and S. Y. Lee (1992). An Expert System for Generator Maintenance Scheduling Using Operation Index. *IEEE Transactions on Power Systems* 7(3): 1141-1148.

Liu, N., B. Huang and M. Chandramouli (2006). Optimal siting of fire stations using GIS and ANT algorithm. *Journal of Computing in Civil Engineering* 20(5): 361-369.

Lopez-Ibanez, M., T. D. Prasad and B. Paechter (2005). Optimal pump scheduling: Representation and multiple objectives. *Proceedings of the Eight International Conference on Computing and Control for the Water Industry (CCWI2005)*, University of Exeter, UK.

Lovat, G. and S. Celozzi (2006). A six-loop magnetic-field exposure system for extremely low-frequency applications. *IEEE Transactions on Magnetics* 42(8): 1982-1990.

Mackle, G., D. A. Savic and G. A. Walters (1995). Applications of genetic algorithms to pump scheduling for water supply. *Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA'95*, Sheffield, U.K., IEE Conference Publication.

Maier, H. R., A. R. Simpson, A. Zecchin, C., W. K. Foong, K. Y. Phang, H. Y. Seah and C. L. Tan (2003a). Ant colony optimisation for Design of Water Distribution Systems. *Journal of Water Resources Planning and Management* 129(3): 200-209.

Maier, H. R., A. R. Simpson, A. C. Zecchin, W. K. Foong, K. Y. Phang, H. Y. Seah and C. L. Tan (2003b). Ant colony optimisation distribution for design of water systems. *Journal of Water Resources Planning and Management-Asce* 129(3): 200-209.

Mantawy, A. H., S. A. Soliman and M. E. El-Hawary (2003). The lon-term hydro-scheduling problem - a new algorithm. *Electric Power Systems Research* 64(1): 67-72.

Mariano, C. E. and E. Morales (1999). MOAQ an ant-Q algorithm for multiple objective optimisation problems. *Proceedings of the Genetic and Evolutionary Computation Conference*, Orlando, Florida, USA, Kaufmann, Morgan.

Merkle, D. and M. Middendorf (2003). Ant colony optimisation with global pheromone evaluation for scheduling a single machine. *Applied Intelligence* 18(1): 105-111.

Merkle, D., M. Middendorf and H. Schmeck (2002). Ant colony optimisation for Resource-Constrained Project Scheduling. *IEEE Transactions on Evolutionary Computation* 6(4): 333-346.

Merz, P. (2000). Memetic algorithms for combinatorial optimisation problems: fitness landscapes and effective search strategies. *Department of Electrical Engineering and Computer Science*, University of Siegen: 221.

Moro, L. M. and A. Ramos (1999). Goal programming approach to maintenance scheduling of generating units in large scale power systems. *IEEE Transactions on Power Systems* 14(3): 1021-1028.

Mukerji, R., H. M. Merrill, B. W. Erickson and J. H. Parker (1991). Power Plant Maintenance Scheduling: Optimizing Economics And Reliability. *IEEE Transactions on Power Systems* 6(2): 476-483.

Nishimura, T. (1997). Random number generator.

Nowicki, E. and C. Smutnicki (1996). A fast taboo search algorithm for the job-shop problem. *Management Science* 42(2): 797-813.

Panigrahi, C. K., P. K. Chattopadhyay, R. N. Chakrabarti and M. Basu (2006). Simulated annealing technique for dynamic economic dispatch. *Electric Power Components and Systems* 34(5): 577-586.

Pfahringer, B. (1996). Multi-agent search for open shop scheduling: Adpating the Ant-Q formalism. Vienna, Austrian Research Institute for Artificial Intelligence.

Prasad, T. D., M. Lopez-Ibanez and B. Paechter (2006). Ant-colony optimisation for optimal pump scheduling. *8th Annual Water Distribution Systems Analysis Symposium*, Cincinnati, Ohio, USA.

Pregej, A., M. Begovic and A. Rohatgi (2006). Recloser allocation for improved reliablity of DG-enhanced distribution networks. *IEEE Transactions on Power Systems* 21(3): 1442-1449.

Rajendran, C. and H. Ziegler (2004). Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research* 155(2): 426-438.

Ramirez-Rosado, I. J. and J. A. Dominquez-Navarro (2006). New multiobjective tabu search algorithm for fuzzy optimal planning of power distribution systems. *IEEE Transactions on Power Systems* 21(1): 224-233.

Reeves, C. R. (1993). Genetic algorithms. *Modern heuristic techniques for combinatorial problems*. C. R. Reeves. Oxford, Blackwell Scientific Publications: 151-196.

Sarkar, D., S. Rohani and A. Jutan (2006). Multi-objective optimisation of seeded batch crystallization processes. *Chemical Engineering Science* 61(16): 5282-5295.

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

Savic, D. A., G. A. Walters and M. Schwab (1997). Multiobjective genetic algorithms for pump scheduling in water supply. *Evolutionary Computing Workshop, AISB'97*, Springer-Verlag Berlin.

Simpson, A. R. and D. E. Goldberg (1994). Pipeline optimisation via genetic algorithms: from theory to practice. *2nd International Conference on Pipeline Systems*, Edinburgh, Scotland.

Socha, K. (2003). The influence of run-time limits on choosing ant system parameters. *Genetic and Evolutionary Computation - Gecco 2003, Pt I, Proceedings*. 2723**:** 49-60.

Socha, K., M. Sampels and M. Manfrin (2003). Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. *Applications of Evolutionary Computing*. 2611**:** 334-345.

Solimanpur, M., P. Vrat and R. Shankar (2004). Ant colony optimisation algorithm to the inter-cell layout problem in cellular manufacturing. *European Journal of Operational Research* 157(3): 592-606.

Song, Y.-H. (1999). Modern optimisation techniques in power systems. Dordrecht ; Boston, Kluwer Academic Publishers.

Stremel, J. P. and R. T. Jenkins (1981). Maintenance scheduling under uncertainty. *IEEE Transactions on Power Apparatus and Systems* PAS-100(2): 460-465.

Stützle, T. and H. Hoos (1997). Improvements on the Ant System: Introducing MAX-MIN Ant System. *Proc. Int. Conf. Artificial Neural Networks and Genetic Algorithms*, Springer Verlag, Wien.

Stützle, T. and H. H. Hoos (2000). MAX-MIN Ant System. *Future Generation Computer Systems* 16: 889-914.

Suresh, R. K. and K. M. Mohanasundaram (2006). Pareto archived simulated annealing for job shop scheduling with multi objectives. *International Journal of Advanced Manufacturing Technology* 29(1-2): 184-196.

Ting, T. O., K. P. Wong and C. Y. Ching (2006). A hybrid genetic algorithm/particle swarm approach for evaluation of power flow in electric network. *Advances in Machine Learning and Cybernetics Lecture Notes in Artificial Intelligence* 3930: 908-917.

Todorovski, M. and D. Rajicic (2006). An initialization procedure in solving optimal power flow by genetic algorithm. *IEEE Transactions on Power Systems* 21(2): 480-487.

Tug, E., M. Sakiroglu and A. Arsian (2006). Automatic discovery of the sequential accesses from web log data files via a genetic algorithm. *Knowledge-based Systems* 19(3): 180-186.

Van Laarhoven, P. J. M. and E. H. L. Aarts (1987). Simulated Annealing: Theory and applications, D. Reidel Publ. Co.

Van Zyl, J. E., D. A. Savic and G. A. Walters (2004). Operational optimisation of water distribution systems using a hybrid genetic algorithm. *Journal of Water Resources Planning and Management, ASCE* 130(2): 160-170.

Victoire, T. A. A. and A. E. Jeyakumar (2005). Unit commitment by a tabu-search-based hybrid-optimisation technique. *IEE Proceedings-Generation Transmission and Distribution* 152(4): 563-574.

Wang, Y. and E. Handschin (2000). A new genetic algorithm for preventive unit maintenance scheduling of power systems. *Electrical Power & Energy Systems*(22): 343-348.

Wikipedia. (2006a, 9 Aug 2006). Electricity market. Retrieved 23 Aug 2006, 2006.

Wikipedia. (2006b, 21 Aug 2006). Tasmania. Retrieved 21 Aug 2006, 2006.

Xu, H. F., M. Sohoni, M. McCleery and T. G. Bailey (2006). A dynamic neighborhood based tabu search algorithm for real-world flight instructor scheduling problems. *European Journal of Operational Research* 169(3): 978-993.

Yamayee, Z., K. Sidenblad and M. Yoshimura (1983). A computational efficient optimal maintenance scheduling method. *IEEE Transactions on Power Apparatus and Systems* PAS-102(2): 330-338.

Yan, W., F. Liu, C. Y. Chung and K. P. Wong (2006). A hybrid genetic algorithm-interior point method for optimal reactive power flow. *IEEE Transactions on Power Systems* 21(3): 1163-1169.

Zecchin, A. C., A. R. Simpson, H. R. Maier, M. Leonard, A. J. Roberts and M. J. Berrisford (2006). Application of two ant colony optimisation algorithms to water distribution system optimisation. *Mathematical and Computer Modelling* 44(5-6): 451-468.

Zecchin, A. C., A. R. Simpson, H. R. Maier and J. B. Nixon (2005). Parametric study for an ant algorithm applied to water distribution system optimisation. *Ieee Transactions on Evolutionary Computation* 9(2): 175-191.

Zhao, F., I. Ubaka and A. Gan (2005). Transit network optimisation - Minimizing transfer and maximizing service coverage with an integrated simulated annealing and tabu search method. *Network Modelling 2005 Transportation Research Record* 1923: 180-188.

Zürn, H. H. (1975). Generation maintenance scheduling via successive approximation dynamic programming. *IEEE Transactions on Power Apparatus and Systems* PAS-94(2): 665-671.

Zürn, H. H. and V. H. Quintana (1977). Several objective criteria for optimal generator preventive maintenance scheduling. *IEEE Transactions on Power Apparatus and Systems* PAS-96(3): 984-992.