16

Power Plant Maintenance Scheduling Using Ant Colony Optimization

Wai Kuan Foong, Holger Robert Maier and Angus Ross Simpson
School of Civil & Environmental Engineering, University of Adelaide
Australia

1. Introduction

Under the pressure of rapid development around the globe, power demand has drastically increased 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 (Lin *et al.*, 1992). One major concern of such development is the optimization 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 (Endrenyi *et al.*, 2001). 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.

Over the past two decades, many studies have focused on the development of methods for optimizing maintenance schedules for power plants. Traditionally, mathematical programming approaches have been used, including dynamic programming (Yamayee *et al.*, 1983), integer programming (Dopazo & Merrill, 1975), mixed-integer programming (Ahmad & Kothari, 2000) and the implicit enumeration algorithm (Escudero *et al.*, 1980). More recently, metaheuristics have been favored, including genetic algorithms (GAs) (Aldridge *et al.*, 1999), simulated annealing (SA) (Satoh & Nara, 1991) and tabu search (TS) (El-Amin *et al.*, 2000). These methods have generally been shown to outperform mathematical programming methods and other conventional approaches in terms of the

Source: Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Book edited by: Felix T. S. Chan and Manoj Kumar Tiwari, ISBN 978-3-902613-09-7, pp. 532, December 2007, Itech Education and Publishing, Vienna, Austria

quality of the solutions found, as well as computational efficiency (Aldridge *et al.*, 1999; Satoh & Nara, 1991).

Ant Colony Optimization is a relatively new metaheuristic for combinatorial optimization problems that is based on the foraging behavior of ant colonies (Dorigo & Stützle, 2004). Compared to other optimization methods, such as GA, ACO has been found to produce better solutions in terms of computational efficiency and quality when applied to a number of combinatorial optimization problems, such as the Traveling Salesman Problem (TSP) (Dorigo & Gambardella, 1997a). Recently, ACO has also been successfully applied to scheduling, including the job-shop, flow-shop and resource-constrained project scheduling problems (Bauer *et al.*, 1999; Colorni *et al.*, 1994; Merkle *et al.*, 2002; Stützle, 1998). Recently, a formulation that enables ACO to be applied to the power plant maintenance scheduling optimization (PPMSO) problem has been introduced by the authors of this chapter (Foong *et al.*, 2005). The formulation was tested on a 21-unit case study and shown to outperform other metaheuristic methods previously applied to the same case study (Foong *et al.*, 2005). In Foong *et al.* (Accepted for publication), the formulation was further tested on a simplified version of a real hydro PPMSO problem, which was solved again using an improved version of the formulation (Foong *et al.*, 2008).

The overall aim of this chapter is to formalize the ACO-PPMSO formulation presented in Foong *et al.* (2005) and to extend the testing of the formulation by applying it to three additional case studies. In addition, the utility of a local search strategy and a heuristic formulation when adopting ACO-PPMSO are examined. In section 2, the general formulation of the PPMSO problem is introduced, are the proposed approach for using ACO to solve this problem (ACO-PPMSO) is introduced in section 3. The four problem instances on which the proposed approach has been tested are described in section 4 and the experimental procedures, results and discussion are presented in section 5. In section 6, a summary and conclusions are given.

2. Power Plant Maintenance Scheduling Optimization

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 $f(s^* \in S^*) \le f(s \in S)$ (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 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: $T_{n,chdur_n} = \{ t \in T_{plan}; chdur_n \in K_n: ear_n \le t \le lat_n - chdur_n + 1\}$ 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:

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

In addition, the following sets of variables are defined:

- $S_{n,t} = \{k \in T_{n,chdur_n}, chdur_n \in K_n: t - chdur_n + 1 \leq k \leq t\}$ 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 is the minimization of the total maintenance duration shortened/deferred (Foong *et al.*, 2008). 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 include 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. 2 to 6.

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:

$$ear_n \leq start_n \leq lat_n - chdur_n + 1 \quad \text{for all } d_n \in D. \tag{2}$$

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

Load constraints (Eq. 3) 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:

$$\sum_{d_n \in D} P_{n,t} - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \geq L_t \quad \text{for all } t \in T_{plan}. \tag{3}$$

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:

$$\sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} Res_{n,k}^r \leq ResAvai_t^r \quad \text{for all } t \in T_{plan}, r \in R. \tag{4}$$

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:

$$start_2 > start_1 + chdur_1 - 1. \tag{5}$$

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

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:

$$NormDur_n \geq chdur_n \geq MinDur_n, \text{ for all } d_n \in D. \tag{6}$$

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.

3. ACO for Power Plant Maintenance Scheduling Optimization (ACO-PPMSO)

Ant Colony Optimization (ACO) is a metaheuristic inspired by the foraging behavior of ant colonies (Dorigo & Stützle, 2004). By marking the paths they have followed with pheromone trails, ants are able to communicate indirectly and find the shortest distance between their nest and a food source when foraging for food. When adapting this search metaphor of ants to solve discrete combinatorial optimization problems, artificial ants are considered to explore the search space of all possible solutions. The ACO search begins with a random solution (possibly biased by heuristic information) within the decision space of the problem. As the search progresses over discrete time intervals, ants deposit pheromone on the components of promising solutions. In this way, the environment of a decision space is iteratively modified and the ACO search is gradually biased towards more desirable regions of the search space, where optimal or near-optimal solutions can be found. Readers are referred to Dorigo & Stützle (2004) for a detailed discussion of ACO metaheuristics and the benchmark combinatorial optimization problems to which ACO has been applied. Due to its robustness in solving these problems, ACO has recently been applied to, and obtained some

encouraging results for, real-world engineering problems, such as the design of optimal water distribution systems (Maier *et al.*, 2003) and in the area of power systems (Gomez *et al.*, 2004; Huang, 2001; Kannan *et al.*, 2005; Su *et al.*, 2005).
As is the case with other metaheuristics, ACO can be linked with existing simulation models of power systems, regardless of their complexity, when solving a PPMSO problem. In addition, the unique way in which ACO problems are represented by using a graph makes ACO inherently suitable for handling various constraints that are commonly encountered in PPMSO problems. In this section, the novel formulation that enables ACO to be applied to PPMSO problems (herein referred to as ACO-PPMSO) introduced by Foong *et al.* (2005) is formalized.

3.1 Problem representation

Before the PPMSO problem can be optimized using ACO, it has to be mapped onto a graph shown in Fig. 1, which is expressed in terms of a set of decision points consisting of the N maintenance tasks that need to be scheduled $D = \{d_1, d_2, d_3, \ldots, d_N\}$.

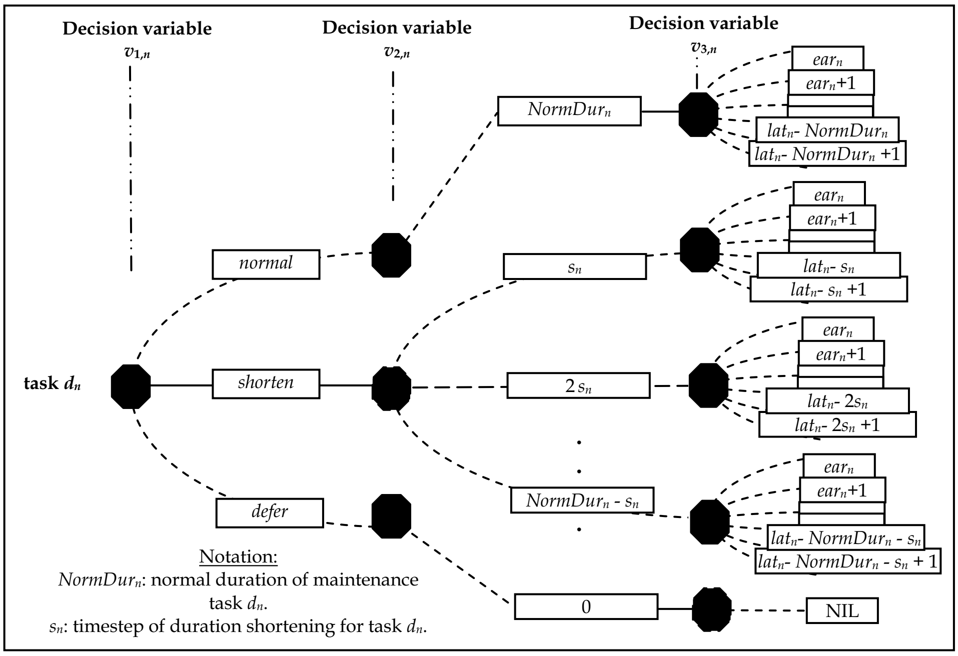

Figure 1. Proposed ACO-PPMSO graph

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 is 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 Fig. 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 (Fig. 1). This graph can then be utilized to construct trial solutions using the ACO-PPMSO algorithm introduced in section 3.2.2.

3.2 ACO-PPMSO Algorithm

The new formulation proposed for power plant maintenance scheduling using Ant Colony Optimisation is implemented via an ACO-PPMSO algorithm, represented by the flowchart given in Fig. 2. The mechanisms involved in each procedure of the proposed ACO-PPMSO algorithm are detailed in sections 3.2.1 to 3.2.6.

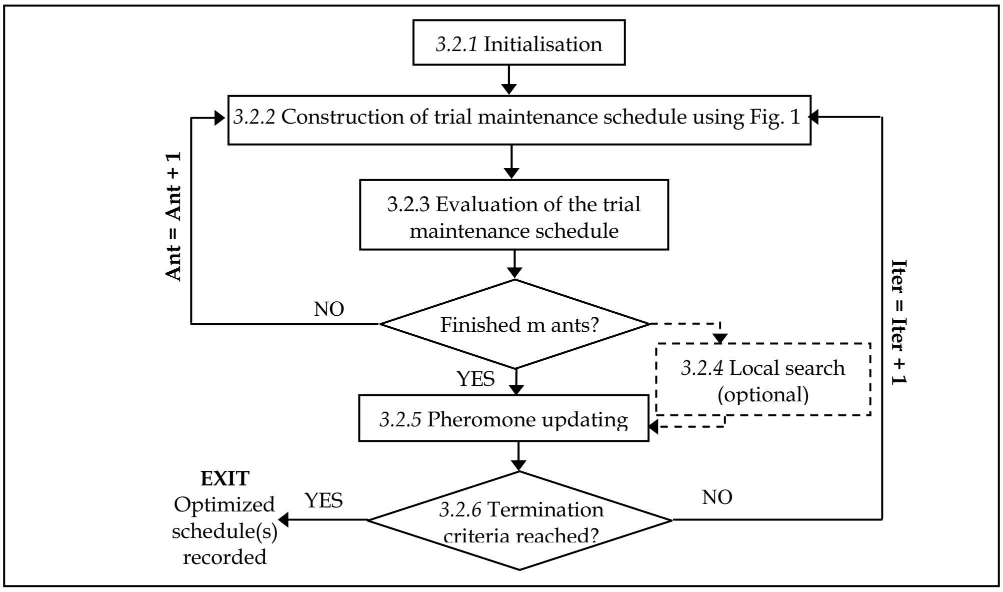

Figure 2. ACO-PPMSO algorithm

3.2.1 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 trail concentrations (τ_0), number of ants, pheromone evaporation rate etc.) need to be defined.

3.2.2 Construction of a trial maintenance schedule

A trial maintenance schedule is constructed using the ACO-PPMSO graph shown in Fig. 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:

$$p_{n,opt}(t) = \frac{\left[\tau_{n,opt}(t)\right]^{\alpha} \cdot \left[\eta_{n,opt}\right]^{\beta}}{\sum_{y \in DP_{c,n}} \left[\tau_{n,y}(t)\right]^{\alpha} \cdot \left[\eta_{n,y}\right]^{\beta}} . \quad (7)$$

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. 7 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 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. 7 for each stage.

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

Table 1. Adaptations for Eq. 7 in stages 1, 2 and 3 of the selection process

Stage 1: 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 Fig. 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. 7). 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. 8). 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. 7 to be used at the stage 1 selection process are summarized in Table 1. It is suggested that values of the heuristics should be selected such that:

$$\eta_{n,defer} < \eta_{n,shorten} < \eta_{n,normal} . \tag{8}$$

Stage 2: Once a decision has been made at stage 1, the selection process proceeds to stage 2 (decision variable $v_{2,n}$ in Fig. 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. 7 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 3.3 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. 9).

$$\eta_{n,dur} \propto dur . \tag{9}$$

The substitutions for the various terms in Eq. 7 when used in stage 2 are summarized in Table 1.

Stage 3: Once a maintenance duration has been selected, the solution construction process enters stage 3 (decision variable $v_{3,n}$ in Fig. 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,…, lat_n – $chdur_n$ + 1}, given a chosen duration of $chdur_n$. In order to utilize Eq. 7 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. 10 to 15.

$$\eta_{n,chdur_n,day} = \left(\eta^{Res}_{n,chdur_n,day}\right)^{w} \cdot \eta^{Load}_{n,chdur_n,day} . \tag{10}$$

$$\eta^{Res}_{n,chdur_n,day} = \frac{\sum_{k \in J_{n,chdur_n,day}} Y_{ResV(k)=0} \cdot R_{n,chdur_n,day}(k)}{\sum_{k \in J_{n,chdur_n,day}} (Y_{ResV(k)=0} - 1) \cdot R_{n,chdur_n,day}(k)} . \quad (11)$$

$$\eta^{Load}_{n,chdur_n,day} = \frac{\sum_{k \in J_{n,chdur_n,day}} Y_{LoadV(k)=0} \cdot C_{n,chdur_n,day}(k)}{\sum_{k \in J_{n,chdur_n,day}} (Y_{LoadV(k)=0} - 1) \cdot C_{n,chdur_n,day}(k)} . \quad (12)$$

$$Y_{ResV(k)=0} = \begin{cases} 1 & \text{if no violation of resource constraints in time period } k \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

$$Y_{LoadV(k)=0} = \begin{cases} 1 & \text{if no violation of load constraints in time period } k \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

$$w = \begin{cases} 1 & \text{if resource constraints are considered} \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

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); $J_{n,chdur_n,day} = \{day \in DP_{3,n,chdur_n} : day \leq k \leq day + chdur_n - 1\}$ 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. 10 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. 10 can be set to 0 (Eq. 15). 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.

3.2.3 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:

$$OFC(s) = f\left(obj_1(s), obj_2(s), \ldots, obj_{Z_T}(s), vio_1(s), vio_2(s), \ldots, vio_{C_T}(s)\right). \quad (16)$$

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. 16. 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. 16. 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 constraints can only be checked upon completion of a complete trial solution and therefore the violations of these constraints, if there are any, can only be reflected through penalty terms in the objective function (Eq. 16). Detailed categorizations of constraints commonly encountered in PPMSO problems, as well as the appropriate methods of handling them, are presented in section 3.3.

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 3.2.2 and 3.2.3, where m (the number of ants) is predefined in procedure 3.2.1, 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.

3.2.4 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 (den Besten *et al.*, 2000; Dorigo & Gambardella, 1997b), little success has been obtained in others (Merkle *et al.*, 2002). Local search has also been found useful for some problems (Foong *et al.*, 2008) where the formulation of heuristics is difficult (Dorigo & Stützle, 2004).

In this formulation, local search is coupled with ACO to solve the PPMSO problem. The local search operator proposed in this chapter is called PPMSO-2-opt, which is a modification of 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, ‘neighbor maintenance schedules’ are generated by exchanging the maintenance start times of a pair of randomly selected tasks of the ‘target maintenance schedule’. 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.

3.2.5 Pheromone updating

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.

Despite its original inspiration from the foraging behaviour of ant colonies, various ACO algorithms have evolved, such as Elitist-Ant System (EAS) (Dorigo (1992); Dorigo *et al.* (1996)) and Max-Min Ant System (MMAS) (Stützle & Hoos, 1997; Stützle & Hoos, 2000). 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:

$$\tau_*(t+1) = \rho \cdot \tau_*(t) + \Delta\tau_*(t). \tag{17}$$

$$\Delta\tau_*(t) = \sum_{s \in Sol_{update}} q = \begin{cases} \dfrac{Q}{OFC(s_{update})} & \text{if } * \in s_{update} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where t is the index of iteration; (1 - ρ) is the pheromone evaporation rate; the subscript 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_* at 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).

As EAS and MMAS are utilized in solving the PPMSO case study systems presented in section 4, the following additional specifications are made according to the general pheromone updating rules:

(A) Elitist-Ant System (EAS)

In EAS, only the least-OFC schedule(s) in every iteration is/are rewarded (Eq. 19).

$$Sol_{update}(t) = s_{iter-best}(t). \tag{19}$$

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

(B) Max-Min Ant System (MMAS)

Similarly to EAS, MMAS only rewards iteration-best trial solution(s) (Eq. 19). 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:

$$\tau_{max}(t+1) = \frac{1}{1-\rho} \cdot \frac{Q}{OFC_{iter-best}(t)}. \tag{20}$$

$$\tau_{c,min}(t+1) = \frac{\tau_{max}(t+1)(1-\sqrt[n]{p_{best}})}{(avg_c - 1)\sqrt[n]{p_{best}}}. \tag{21}$$

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 3.2.2; 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:

$$\tau_{c,min}(t) \leq \tau_{n,opt}(t) \leq \tau_{max}(t)\,;\, opt \in DP_{c,n} \;\; c = 1,2,3 \; for\ all\ t, n. \tag{22}$$

3.2.6 Termination of run

Procedures 3.2.2 to 3.2.5 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.

3.3 Constraints Handling

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. 2), precedence (Eq. 5) and minimum maintenance duration (Eq. 6) constraints can be addressed when trial solutions are being generated during ant cycles (procedure described in section 3.2.2). On the other hand, the violation of load (Eq. 3) and resource (Eq. 4) 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, minimum maintenance duration, precedence and load constraints, even the slightest violations are generally intolerable. On the other hand, constraints, such as resource 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 (Fig. 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.

Penalty-based technique: In ACO-PPMSO, penalty functions, which 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), 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.

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.

4. Problem Instances

In order to test the utility of the proposed ACO-PPMSO formulation, it is applied to 4 problem instances, including 21- and 22-unit benchmark case studies from the literature and modified versions of these case studies. The 21- and 22-unit case studies have been chosen as they enable comparisons to be made with results obtained in previous studies. However, as these case studies can be solved without the need for maintenance shortening and deferral, modifications to the case studies are introduced in this chapter to test this feature of the proposed formulation. Details of the four problem instances are given below.

4.1 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. Due to space constraints, system details are not presented here but can be found in Aldridge *et al.* (1999). 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 Aldridge *et al.* (1999).
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.

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:

$$Min\left\{ SSR = \sum_{t\in T_{plan}} \left(\sum_{n=1}^{N} P_n - \sum_{d_n\in D_t} \sum_{k\in S_{n,t}} X_{n,k} P_n - L_t \right)^2 \right\}. \tag{23}$$

where P_n is the generating capacity of unit d_n; L_t is the anticipated load for period t, *subject to* the maintenance window, load and personpower constraints, as given by:

$$ear_n \le start_n \le lat_n - NormDur_n + 1 \qquad \text{for all } d_n \in D. \tag{24}$$

$$\sum_{d_n\in D_t} \sum_{k\in S_{n,t}} X_{n,k} Res_{n,k} \le ResAvai_t \quad \text{for all } t \in T_{plan}. \tag{25}$$

$$\left(\sum_{n} P_n - \sum_{d_n\in D_t} \sum_{k\in S_{n,t}} X_{n,k} P_n \right) \ge L_t \quad \text{for all } t \in T_{plan}. \tag{26}$$

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 3.3), whereas both load and personpower constraints are indirect and are therefore taken into account by using penalty-based techniques (section 3.3).

When applying the ACO-PPMSO formulation to this case study, the heuristic developed as part of this research (Eqs. 10 to 15) 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. 10 was set to 1, as utilization of resource (personpower) constraints is considered 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. 27).

$$OFC(s) = SSR(s) \cdot \left(ManVio_{tot}(s) + 1 \right) \cdot \left(LoadVio_{tot}(s) + 1 \right). \tag{27}$$

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. 28 to 31. 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:

$$ManVio_{tot}(s) = \sum_{t \in T_{MV}} \left(\sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} Res_{n,k} - ResAvai_t \right). \quad (28)$$

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

$$T_{MV} = \left(t : \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} Res_{n,k} > ResAvai_t \right). \quad (29)$$

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.

$$LoadVio_{tot}(s) = \sum_{t \in T_{LV}} \left(\sum_{n} P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \right). \quad (30)$$

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

$$T_{LV} = (t : \sum_{n} P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n < L_t). \quad (31)$$

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

4.2 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 can be found in Escudero *et al.* (1980). 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 (see Escudero et al., 1980) 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.

Problem formulation

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 sum of the deviation of generation reserve from the average reserve over the planning horizon (Eqs. 32 to 34):

$$Min\left\{LVL = \sum_{t \in T_{plan}} \left| Res_{avg} - Res_t \right| \right\}. \tag{32}$$

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

$$Res_t = \sum_{n=1}^{N} P_n - \sum_{n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n - L_t. \tag{33}$$

$$Res_{avg} = \frac{\sum_{t \in T_{plan}} Res_t}{T}. \tag{34}$$

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:

$$ear_n \leq start_n \leq lat_n - NormDur_n + 1 \qquad \text{for all } d_n \in D. \tag{35}$$

$$\left(\sum_n P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \right) \geq L_t \quad \text{for all } t \in T_{plan}. \tag{36}$$

$$\left(\sum_n P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \right) \geq 1.2 L_t \quad \text{for all } t \in T_{plan}. \tag{37}$$

$$\begin{cases} start_3 > start_2 + NormDur_2 - 1 \\ start_6 > start_5 + NormDur_5 - 1 \end{cases} \tag{38}$$

$$\begin{cases} X_{15,k} = 0 \text{ for } k = \left[start_{16}, ..., start_{16} + NormDur_{16} - 1\right] \\ X_{16,k} = 0 \text{ for } k = \left[start_{15}, ..., start_{15} + NormDur_{15} - 1\right] \\ X_{21,k} = 0 \text{ for } k = \left[start_{22}, ..., start_{22} + NormDur_{22} - 1\right] \\ X_{22,k} = 0 \text{ for } k = \left[start_{21}, ..., start_{21} + NormDur_{21} - 1\right] \end{cases} \tag{39}$$

It is interesting to note that, given the same objective, the objective functiom 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. 10) 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 need to be 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. 40).

$$OFC(s) = LVL(s) \cdot (LoadResVio_{tot}(s) + 1). \quad (40)$$

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. 40 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. 41 and 42. 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:

$$LoadResVio_{tot}(s) = \sum_{t \in T_{LV}} \left(\sum_{n} P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n \right). \quad (41)$$

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

$$T_{LV} = (t : \sum_{n} P_n - \sum_{d_n \in D_t} \sum_{k \in S_{n,t}} X_{n,k} P_n < 1.2 L_t). \quad (42)$$

4.3 Modified 21-unit system

The 21-unit case study system described in section 4.1 was modified in the following ways in order to ensure that maintenance task shortening and/or deferral are required to satisfy load constraints:

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. While all maintenance tasks have the option of being deferred, some maintenance tasks can be carried out in durations shorter than the original outage duration (shown in Table 2). The personpower requirements for shortened durations are also detailed in Table 2.

Unit No., n	Optional Outage Duration, (weeks)	Personpower required for each week, $Res_{n,wk(wk=1,2,\dots, NormDur_n)}$ (person)
1	5	10, 10, 10, 8, 5
	3	15, 14, 14
2	3	15, 15, 10
5	3	17, 17, 16
8	4	13, 13, 13, 6
9	8	3, 3, 3, 2, 2, 3, 3, 3
	6	4, 4, 3, 3, 4, 4
	4	6, 5, 5, 6
	2	11, 11
10	2	15, 15
14	2	20, 20
20	2	20, 20

Table 2. Personpower utilization for the modified 21-unit case study system