ARTICLE TEMPLATE

Minimizing Equipment Shutdowns in Oil and Gas Campaign Maintenance

Z. Seif[a], E. Mardaneh[a], R. Loxton[a] and A. Lockwood[b]

[a]School of Electrical Engineering, Computing, and Mathematical Sciences, Curtin University, Perth, Australia; [b]Woodside Energy Ltd., Perth, Australia

ARTICLE HISTORY
Compiled February 17, 2020

ABSTRACT
This paper considers the problem of scheduling periodic maintenance items for oil and gas plants. Each maintenance item involves various maintenance tasks and may require temporary equipment shutdowns, which are costly and highly disruptive to production. The aim is to minimize equipment shutdowns by grouping maintenance items with similar shutdown requirements into short-term maintenance operations called campaigns. Real plants can involve tens of thousands of maintenance items and thus manually scheduling the campaigns is an extreme challenge. In this paper, we develop a mixed-integer linear programming model for optimally allocating maintenance items to campaigns so that total shutdown cost is minimized. The model incorporates constraints on maintenance deadlines, campaign times, maintenance item suppression and labour hours per campaign. We solve the model for realistic scenarios involving data for Karratha Gas Plant in Western Australia, which is the main processing plant for the massive North West Shelf oil and gas project.

KEYWORDS
Plant Maintenance; Asset Management; Oil and Gas; Optimization; Scheduling

1. Introduction

Maintenance is a major cost in any asset-intensive industry and oil and gas is no exception (Al-Turki et al. (2019)). Oil and gas plants contain thousands of individual pieces of equipment, all competing for limited maintenance resources, and different pieces of equipment have different priorities and require different maintenance strategies. Balancing these varying requirements to achieve an overall optimal outcome is a major challenge for maintenance managers, who must strike a delicate balance between over-maintenance (which is inefficient and costly) and under-maintenance (which can lead to unacceptable risks).

For the massive, isolated plants common in the resources industry, heavy maintenance work is often concentrated into intensive short-term maintenance campaigns where the plant is partially or completely shut down. This "campaign maintenance" approach is necessary when the work sites are situated in remote areas where it is not possible to base a large workforce. Consequently, local resources are limited and most maintenance personnel are employed on a "fly-in fly-out" basis.

CONTACT R. Loxton. Email: r.loxton@curtin.edu.au

With the growing need to minimize maintenance costs and unplanned outages, the last decade has seen tremendous growth in machine learning methods for predicting equipment faults—their timing, location, and likely causes (Fuqiong et al. (2013); Khorasgani et al. (2016); Sankararaman (2015); Shi and Zeng (2016)). Even with perfect predictions, however, designing an optimal maintenance schedule is far from trivial given the massive number of inter-related components in a real-world plant. For campaign maintenance, there are two key planning and scheduling problems:

(1) Single campaign planning – determine the optimal work sequence for a series of inter-related maintenance activities within a single campaign, with the aim of minimizing campaign duration and the resources required; and
(2) Multi-campaign planning – determine the best way of dividing maintenance work into campaigns and schedule the timing of these campaigns over a long-term time horizon (typically many years), with the aim of maximizing plant availability and minimizing cost.

The first problem is essentially a resource-constrained project scheduling problem (RCPS), a classic optimization problem that has been studied since the 1960s. The RCPS is NP-hard and requires selecting the timing of each activity in a project given precedence relationships and constraints on the number of resources available (Blazewicz et al. (1983)). The objective is typically to minimize schedule duration and the problem can be represented as a network in which the nodes represent activities and the arcs represent precedences. See Mika et al. (2015) for a comprehensive review of the RCPS and Hartmann and Briskorn (2010) for a discussion on some of its extensions.

Both exact and heuristic methods have been proposed for solving the RCPS. Most of the exact methods are based on branch-and-bound algorithms that exploit the network structure of the problem to determine efficient lower bounds, branching strategies, and subproblem selection strategies. The lower bounds help to reduce the size of the search tree and the most common lower bounds are the critical path lower bound (obtained by removing the resource constraints), the resource capacity lower bound (obtained by removing the precedence constraints), and the critical capacity lower bound (similar to the resource capacity lower bound but with the early start times from the critical path taken into account). Other, more advanced lower bounds used in the literature include the critical sequence lower bound, the parallel machine lower bound, and the incompatible pairs lower bound (Coelho and Vanhoucke (2018); Shim and Kim (2007)). The branching strategies dictate how the full RCPS is decomposed into subproblems, and the main strategies used include serial branching (Sprecher (2000)), parallel branching (Demeulemeester and Herroelen (1992)), and activity start branching (Dorndorf et al. (2000); Ranjbar et al. (2012)). Finally, the subproblem selection strategies dictate the order in which the subproblems in the search tree are solved, either through depth-first search, best bound search, or some combination.

Branch-and-bound methods for solving the RCPS—and, by extension, the single campaign planning problem (1) above—are exact algorithms that, while mathematically rigorous, are often time-consuming for real problems. Thus, various heuristic and meta-heuristic methods have also been proposed in the literature (Kolisch and Hartmann (2006)). Heuristic methods generate a feasible schedule or a set of schedules from scratch using either a serial approach (iterating activity by activity) or parallel approach (iterating time instant by time instant) (Boctor (1990); Kolisch and Hartmann (1999); Li and Willis (1992)). In both approaches, various priority rules are used to select the next activity to schedule, such as choosing the activity with the maximum number of successors, choosing the activity with the shortest duration, or choosing the activity with the minimum latest start time (Kolisch (1996)). These heuristic methods are specific to the RCPS, whereas meta-heuristics are general-purpose frameworks designed for a wide range of problems. For the RCPS, the most popular meta-heuristics include local search algorithms such as tabu search (Lambrechts et al. (2008)), population-based algorithms such as particle swarm optimization (Jarboui et al. (2008)), and learning algorithms such as neural networks (Agarwal et al. (2011)). Hybrid methods for the RCPS take this one step further by combining different meta-heuristics, such as using local search methods in population-based optimization (Ziarati et al. (2011)), or by combining meta-heuristics with exact methods such as branch-and-bound or constraint programming (Yoosefzadeh and Tareghian (2013)). See Pellerin et al. (2019) for a recent survey of hybrid meta-heuristic methods for the RCPS.

The references described above are for the general RCPS—of which the campaign planning problem is a special case—rather than being specifically targeted at maintenance. One of the only maintenance-focused papers in this area of the literature is Megow et al. (2011), which discusses heuristic methods for solving the RCPS in turnaround and shutdown maintenance projects. Two other papers (Pillac et al. (2013); Zamorano and Stolletz (2017)) focus on the problem of routing maintenance technicians in short-term maintenance projects. This routing aspect is related to the campaign planning problem (1), but is generally excluded from RCPS formulations.

Although our discussion thus far reveals that optimal scheduling for short-term maintenance operations or campaigns has been extensively studied, problem (2) on long-term multi-campaign planning has yet to be explored in the literature. Instead, the focus is on continuous maintenance strategies where the maintenance effort is evenly distributed across the time horizon (Ebrahimipour et al. (2015); Moinian et al. (2017)). Discrete maintenance campaigns—short bursts of intensive maintenance activity followed by long periods of minimal maintenance—are ignored.

This paper fills this gap in the literature by proposing a mixed-integer linear programming model for the multi-campaign planning problem introduced above. In this respect, the paper is similar in philosophy to the exact solution approaches for the RCPS, which also generally involve mixed-integer programming models, but the structure of our model and the corresponding mathematical results are completely different due to the different problem settings. Our model involves allocating periodic maintenance activities, called maintenance items, to a fixed number of maintenance campaigns and determining the timing of those campaigns over the remaining life of the plant. Since the maintenance items may require certain systems and equipment to be shut down, the aim is to group maintenance items with similar shutdown requirements into the same campaign to minimize re-work. This can be achieved by performing some items more frequently than required (over-maintenance) and other items less frequently than required (under-maintenance). For example, it may be advantageous to delay a maintenance item so that it coincides with other similar items requiring the same shutdowns. In each campaign, the amount of work is restricted due to resource limitations. Moreover, there are limits on the extent to which maintenance items can be delayed, and some safety-critical items cannot be delayed at all.

The remainder of this paper is organized as follows. Section 2 describes our mixed-integer linear programming model for the multi-campaign planning problem. Section 3 gives the mathematical justification for the model, and then Section 4 presents the results from various numerical simulations. Section 4 consists of two parts: Section 4.1 covers a real case study involving data for Karratha Gas Plant operated by Woodside

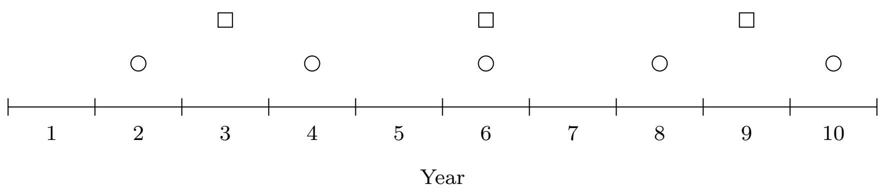


Figure 1. Maintenance schedule for a two-year maintenance item (circle) and a three-year maintenance item (square).

Energy Ltd., Australia's largest independent oil and gas company, and Section 4.2 discusses the results for randomly-generated test problems. Section 5 concludes the paper with a summary of our contributions and suggestions for future work.

2. Mathematical Model

2.1. Overview

The key entities in our maintenance scheduling problem are *maintenance items*, *maintenance plans* and *maintenance campaigns*. A maintenance item is a recurring service operation—consisting of various maintenance tasks—for specific pieces of equipment, and a maintenance plan is a group of related maintenance items. The maintenance items are performed as part of maintenance campaigns, each lasting several weeks during the year. These campaigns are intensive short-term maintenance operations that typically involve hundreds or thousands of specialist maintenance personnel. The number of campaigns per year and the number of years in the time horizon are input parameters for the optimization model.

Each maintenance item takes a certain *duration* and requires a certain number of *resources* to complete. Each maintenance item also has a *cycle time* or *frequency* that defines the ideal rate at which it should be performed, as shown in the example schedule in Figure 1. The frequency is only a guide and in practice some maintenance items can deviate from their recommended frequency. For example, a yearly maintenance item may be "brought forward" and repeated after only 10 months so that it aligns with other maintenance items requiring similar resources. In this paper we only consider low-frequency maintenance items whose cycle times are an integer number of years, which constitutes the bulk of maintenance items in an oil and gas plant. The assumption of integer frequencies ensures that each maintenance item in the optimization model occurs at most once per year.

A maintenance plan typically contains maintenance items corresponding to different service frequencies for the same pieces of equipment—for example, a two-year service and a three-year service as shown in Figure 1. Such items are grouped into a single plan so that they can be synchronized for efficiency. In the case of Figure 1, the two-year and three-year services should be synchronized so that when they coincide every sixth year, they occur together in the same campaign.

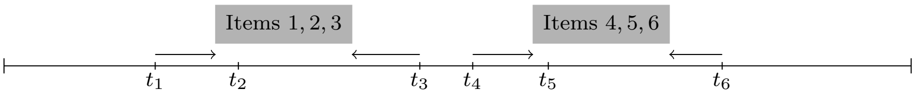
In some cases, when two maintenance items in the same plan coincide in a particular year, the tasks in one of the items will replace (suppress) the tasks in the other item. For example, a two-year maintenance service may already include all of the tasks in the one-year service, meaning that the one-year service is redundant every second year. The precise suppression rules are defined by a *hierarchy value* attached to each maintenance item, and higher hierarchy values suppress lower hierarchy values. In other words, when multiple maintenance items coincide, only the maintenance item (or items) with the highest hierarchy value is performed, and those with lower hierarchies are ignored. For example, consider a maintenance plan containing a one-year maintenance item with hierarchy value 1, and two-year and three-year maintenance items with hierarchy value 2. The one-year item has the lowest hierarchy value and is thus omitted every second and third year, when it coincides with either the two-year item or the three-year item. In every sixth year when all three maintenance items coincide, both the two-year and the three-year items must be performed because they share the same hierarchy value, but the one-year item is redundant.

The hierarchy values are user inputs that can be altered to reflect different maintenance strategies. Some maintenance plans may not use suppression, in which case all items in the plan have the same hierarchy. For maintenance plans using suppression, the hierarchy values will normally be proportional to the cycle time, since maintenance items on long cycles are typically more extensive than those on short cycles. For example, if the two-year and three-year maintenance items in Figure 1 use suppression, then the tasks in the three-year item will likely supercede those in the two-year item when they coincide every six years.

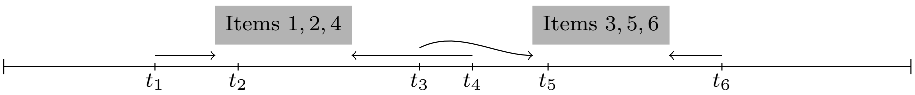
The optimization problem we consider involves scheduling repeated instances of maintenance plans over a multi-year time horizon. Each maintenance plan is allocated to a specific campaign and all items in the plan are performed as part of this campaign in the years they are due. Hence, if the first instance of a maintenance item is completed in a certain campaign, then all subsequent instances of that item are completed in the same campaign (but in different years). The number of items performed in a plan varies from year to year depending on the frequencies. The decision variables in the optimization problem define the timing of each campaign in each year, and the allocation of maintenance plans to campaigns. There are constraints on the minimum and maximum campaign durations and the number of labour hours in a campaign. Moreover, campaigns cannot overlap.

Some maintenance items require certain systems and equipment in the plant to be shut down temporarily. The shutdowns required for a maintenance item can range from the local shutdown of a single piece of equipment to a complete plant shutdown. The objective function in the optimization model measures the total cost of all required shutdowns over the time horizon. This cost can be reduced by grouping maintenance plans requiring the same shutdowns into the same campaign. To achieve an optimal grouping that minimizes overall shutdown cost, maintenance items can be delayed or advanced with respect to their due dates. The extent to which a maintenance item can be delayed or advanced is constrained based on the item's importance. In particular, maintenance items that are critical to safety or production typically cannot be delayed.

We conclude this section with a small example to illustrate the relationship between items, campaigns, and shutdowns. Suppose that there are two campaigns and six maintenance items to schedule during a particular year in the time horizon and the due dates for the items are in increasing order as shown in Figure 2. Suppose further that items 1, 3, 5, and 6 require no equipment shutdowns, and items 2 and 4 require a full plant shutdown. Assuming a balanced workload of three items per campaign, the obvious allocation would be to allocate items 1–3 to the first campaign and items 4–6 to the second campaign, as shown in Figure 2(a). This involves delaying items 1 and 4 and advancing items 3 and 6, and results in two plant shutdowns—one for item 2 in the first campaign, and one for item 4 in the second campaign. But by swapping items 3 and 4, we obtain the situation shown in Figure 2(b), where only the first



(a) Items 1 and 4 delayed; items 3 and 6 advanced; 2 plant shutdowns



(b) Items 1 and 3 delayed; items 4 and 6 advanced; 1 plant shutdown

Figure 2. A simple example illustrating the relationship between items, campaigns, and shutdowns. Here, the grey boxes denote the campaigns and t_i denotes the due time for maintenance item i.

campaign requires a shutdown. Here, items 1 and 3 are delayed and items 4 and 6 are advanced. This simple example shows how the number of shutdowns can be reduced by judiciously delaying and advancing maintenance items and changing the allocations of items to campaigns. In reality the problem is far more complex since we need to balance the requirements across multiple years, thousands of maintenance items, and constraints on the extent to which maintenance items can be delayed and advanced. The optimization model takes all of this into account.

2.2. Sets and Parameters

The model involves a set of maintenance plans P, a set of maintenance items I, a set of maintenance campaigns C, and a set of years Y. Each item in I belongs to a plan in P and the campaigns in C are repeated in each year. The plans, items, campaigns, and years are labelled by integers, with $\max Y$ denoting the final year in the planning horizon and $\max C$ denoting the last campaign in each year.

The key parameters defining each maintenance item $i \in I$ are:

- f_i = frequency (in whole years) of maintenance item i;
- y_i^* = first due year of maintenance item i;
- τ_i^* = due day of maintenance item i in year y_i^*;
- d_i = duration (in days) of maintenance item i;
- r_i = number of resources required to complete maintenance item i; and
- h_i = suppression hierarchy value (a positive integer) for maintenance item i.

According to this input data, maintenance item i is due on day τ_i^* in years y_i^*, $y_i^* + f_i$, $y_i^* + 2f_i$, and so on.

Let I_p denote the set of maintenance items in maintenance plan p. Furthermore, let I_{py} denote the set of items in plan p that are due in year y. This set is completely defined by the input data and it depends on the item frequencies (since the frequencies define the years in which the items are due) and the suppression hierarchy rules (since some items may be suppressed if they coincide with other items higher in the suppression hierarchy). The precise logic for defining I_{py} is given in Algorithm 1. The algorithm starts with $I_{py} = \emptyset$ for each y and then loops through each item and each year, progressively updating I_{py} to ensure that it only contains the active items with the highest hierarchy value.

Although maintenance item i is due on day τ_i^* in years y_i^*, $y_i^* + f_i$, $y_i^* + 2f_i$ and so on, this does not mean that it must occur exactly on day τ_i^*, as maintenance items

Algorithm 1 Returns the maintenance items due in each year for plan p

Initialize the set of maintenance items performed in plan p in each year: $I_{py} \leftarrow \varnothing, y \in Y$
Initialize the dominant hierarchy value in plan p in each year: $\bar{h}_{py} \leftarrow 1, y \in Y$
for all $(i \in I_p)$ **do**
  Initialize the year counter: $y \leftarrow y_i^*$
  while $(y \leq \max Y)$ **do**
    if $(h_i > \bar{h}_{py})$ **then**
      Item i suppresses all items currently in I_{py}: $I_{py} \leftarrow \{i\}$
      Update the dominant hierarchy value for plan p in year y: $\bar{h}_{py} \leftarrow h_i$
    else if $(h_i = \bar{h}_{py})$ **then**
      Item i has the same hierarchy value as the items currently in I_{py}: $I_{py} \leftarrow I_{py} \cup \{i\}$
    end if
    Update the year counter: $y \leftarrow y + f_i$
  end while
end for

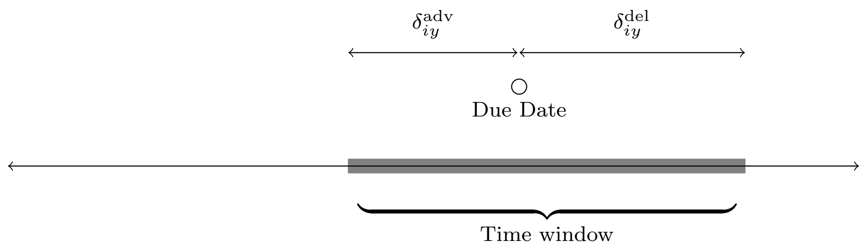


Figure 3. The time window for maintenance item i due in year y.

can be delayed or advanced when it is advantageous to do so. Let δ_{iy}^{adv} denote the maximum number of days that item i can be advanced in year y, and let δ_{iy}^{del} denote the maximum number of days that item i can be delayed in year y. These parameters define *time windows* around the maintenance due times; see Figure 3.

The key input parameters related to campaigns are:

- $\tilde{t}_{cy}^{\min}$ = lower bound for the start time of campaign c in year y;
- $\tilde{t}_{cy}^{\max}$ = upper bound for the end time of campaign c in year y;
- ϵ = minimum duration (in days) between consecutive campaigns;
- $\Delta_{\min}$ = lower bound (in days) for campaign duration;
- $\Delta_{\max}$ = upper bound (in days) for campaign duration;
- $R_{\max}$ = maximum number of resources available per campaign per year;
- α = minimum ratio of campaign duration to item duration (must be at least 1);
- $\rho_{\min}$ = ratio defining the minimum proportion of maintenance items that must be assigned to each campaign in each year (must be between 0 and 1); and
- $\rho_{\max}$ = ratio defining the maximum proportion of maintenance items that can be assigned to each campaign in each year (must be between 0 and 1).

Here, $\epsilon > 0$ ensures that campaigns do not overlap and the ratios $\rho_{\min} \in (0, 1)$ and $\rho_{\max} \in (0, 1)$ ensure that the campaign workloads are sufficiently balanced.

Finally, we now introduce the input data and notation related to the equipment shutdowns required by maintenance items. This is the central aspect of the optimization model because shutdowns are disruptive and the goal is to minimize them by grouping similar maintenance items into the same campaign. The shutdowns required for a maintenance item are defined with respect to a structure diagram called the *plant hierarchy tree*; see Figure 4 for an example.

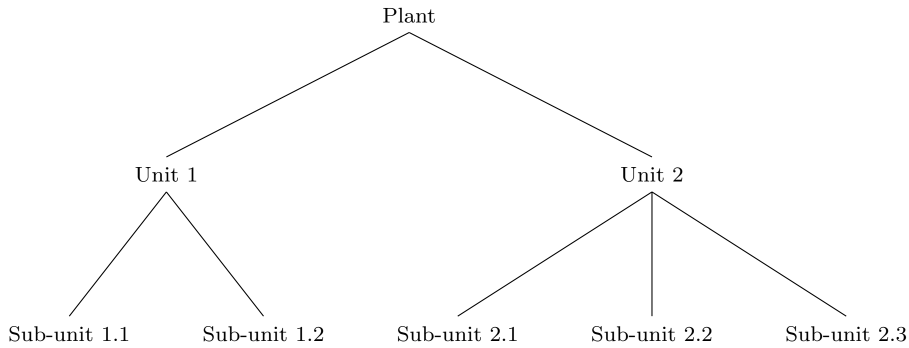


Figure 4. A three-level hierarchy tree for a plant decomposed into units and sub-units.

The plant hierarchy tree consists of multiple levels, with the top level containing a single node representing the entire plant, the intermediate levels containing nodes representing units and systems in the plant, and the bottom level containing nodes representing individual pieces of equipment. The edges in the plant hierarchy tree connect nodes in consecutive levels, indicating a parent-child relationship in the hierarchy.

Let N be the set of nodes in the plant hierarchy tree and let $\psi(n)$ denote the unique path from node n to the root node in the top level, where $\psi(n) = n$ if node n is itself the root node. For example, path $\psi(n)$ starting at sub-unit 2.1 in Figure 4 is

$$\text{Sub-unit } 2.1 \rightarrow \text{Unit } 2 \rightarrow \text{Plant}.$$

Each maintenance item may require a shutdown of zero, one, or multiple nodes in the plant hierarchy tree. This information is input data for the optimization model and is typically stored in maintenance databases. If a maintenance item requires node n to be shut down, then all of its child nodes must also be shut down while the maintenance item is performed. For example, in Figure 4, a shutdown of unit 1 would also necessitate shutdowns of sub-units 1.1 and 1.2. Given a group of maintenance items, the dominant shutdown units are the highest-level nodes in the plant hierarchy tree that must be shut down to perform the maintenance items in the group. The dominant shutdown units are independent and are not covered by any higher-level shutdowns. Mathematically, we say that node $n \in N$ is a *dominant shutdown unit* for a group of maintenance items I' if the following two conditions are satisfied:

(i) At least one maintenance item in I' requires a shutdown of node n; and
(ii) No maintenance item in I' requires a shutdown of any node in $\psi(n) \setminus \{n\}$ (that is, a node higher than node n in the plant hierarchy tree).

For example, referring to Figure 4, if I' contains one maintenance item requiring a shutdown of unit 1 and another maintenance item requiring a shutdown of sub-unit 1.1, then unit 1 is the dominant shutdown unit, since sub-unit 1.1 sits below unit 1 in the hierarchy tree and is thus automatically covered by unit 1. If I' also contains a maintenance item requiring a shutdown of sub-unit 2.2, then both unit 1 and sub-unit 2.2 are dominant shutdown units.

Let $w_n > 0$ denote the cost of a shutdown of node n and let s_{pyn} be a binary parameter indicating whether node n is a dominant shutdown unit for the maintenance items in I_{py}. Thus, $s_{pyn} = 1$ means that node n and all of its descendants in the

Set	Description
P	Set of maintenance plans
I	Set of maintenance items
C	Set of maintenance campaigns
Y	Set of years
I_p	Set of maintenance items in plan p
I_{py}	Set of maintenance items in plan p due in year y
N	Set of nodes in the plant hierarchy tree

Table 1. Sets in the mathematical model.

hierarchy tree must be shut down to perform plan p in year y. This information can be extracted directly from the maintenance databases.

Nodes higher in the plant hierarchy are more costly to shut down than nodes lower in the hierarchy, with a complete plant shutdown being the most expensive. Since the cost of shutting down a node is greater than the cost of individually shutting down each of its children in the hierarchy tree, we impose the following conditions on the cost parameters:

$$w_n > \sum_{j \in N:\, \psi(j) \setminus \psi(n) = j} w_j, \quad \forall n \in N, \tag{1}$$

where the summation on the right-hand side is taken over all immediate children of node n in the plant hierarchy. Condition (1) states that the total cost of shutting down each sub-unit of unit n is less than the cost of shutting down unit n itself.

The sets and parameters described above, which form the inputs for the mathematical optimization model, are listed in Tables 1 and 2.

2.3. Problem Statement

There are three classes of decision variables in the optimization model: decision variables to govern the allocation of plans to campaigns, decision variables to govern the timing of campaigns, and decision variables to link the plan allocations with the shutdown requirements. The decision variables in the first class are binary variables x_{pc} indicating whether maintenance plan p is allocated to campaign c. The decision variables in the second class are continuous-valued variables $t_{cy}^{\min}$ and $t_{cy}^{\max}$, which define the start and end times of campaign c in year y, respectively (measured as the number of days since the beginning of the year). The decision variables in the third class are binary variables ξ_{cyn} indicating whether node n is a dominant shutdown unit for campaign c in year y. These decision variables are summarized in Table 3.

The scheduling decisions for allocating maintenance items to campaigns are taken over multiple years because the frequency of each item is a whole number of years, and the frequency of some items can exceed 10 years. Within each year, the campaign times and the limits on delaying and advancing items are expressed in days. Our model does not determine the timing of each maintenance item; it only decides which items go in which campaigns and at what times the campaigns should be scheduled to ensure that all constraints related to the maintenance deadlines can be satisfied. The precise scheduling of maintenance items within a single campaign is a later, more detailed

Parameter	Description
f_i	Frequency of maintenance item i
r_i	Resources for maintenance item i
d_i	Duration of maintenance item i
y_i^*	First due year for maintenance item i
τ_i^*	Due day for maintenance item i in year y_i^*
h_i	Suppression hierarchy value for maintenance item i
δ_{iy}^{adv}, δ_{iy}^{del}	Advance/Delay limits for maintenance item i
s_{pyn}	Shutdown indicators for maintenance items in plan p in year y
w_n	Shutdown cost for node n
$\tilde{t}_{cy}^{\min}$, $\tilde{t}_{cy}^{\max}$	Lower/Upper bounds for start/end of campaign c in year y
ϵ	Min duration between consecutive campaigns
$\Delta_{\min}$, $\Delta_{\max}$	Min/Max campaign durations
α	User-defined weight for campaign durations
$R_{\max}$	Max resources per campaign per year
$\rho_{\min}$, $\rho_{\max}$	Min/Max ratios for number of items assigned to each campaign

Table 2. Parameters in the mathematical model.

Decision Variable	Type	Description
x_{pc}	Binary	Maintenance plan allocations
$t_{cy}^{\min}$, $t_{cy}^{\max}$	Continuous	Campaign start/end times
ξ_{cyn}	Binary	Dominant shutdown indicators

Table 3. Decision variables in the optimization model.

step that would involve solving a RCPS, as explained in the introduction.

The objective in our optimization model is to minimize the total shutdown cost over the entire time horizon:

$$\min_{x_{pc},\, t_{cy}^{\min},\, t_{cy}^{\max},\, \xi_{cyn}} \quad \sum_{c\in C} \sum_{y\in Y} \sum_{n\in N} w_n \xi_{cyn}. \tag{2}$$

We now list the various constraints in the model.

- Each maintenance plan is allocated to a single campaign:

$$\sum_{c\in C} x_{pc} = 1, \quad \forall p \in P. \tag{3}$$

- Bound constraints on the campaign start and end times:

$$\tilde{t}_{cy}^{\min} \le t_{cy}^{\min} \le t_{cy}^{\max} \le \tilde{t}_{cy}^{\max}, \quad \forall c \in C, \quad \forall y \in Y. \tag{4}$$

- Bound constraints on the campaign durations:

$$\Delta_{\min} \le t_{cy}^{\max} - t_{cy}^{\min} \le \Delta_{\max}, \quad \forall c \in C, \quad \forall y \in Y. \tag{5}$$

- Each campaign is at least as long as the items that it contains:

$$t_{cy}^{\max} - t_{cy}^{\min} \geq \alpha\, x_{pc} \max_{i \in I_{py}} d_i, \quad \forall c \in C, \quad \forall y \in Y, \quad \forall p \in P : I_{py} \neq \varnothing. \tag{6}$$

- Minimum duration between consecutive campaigns:

$$t_{c+1,y}^{\min} \geq t_{cy}^{\max} + \epsilon, \quad \forall c \in C \setminus \{\max C\}, \quad \forall y \in Y, \tag{7}$$
$$t_{1,y+1}^{\min} \geq t_{\max C,y}^{\max} - 365 + \epsilon, \quad \forall y \in Y \setminus \{\max Y\}. \tag{8}$$

- Limit on labour hours per campaign:

$$\sum_{p \in P} \sum_{i \in I_{py}} 24 d_i r_i x_{pc} \leq R_{\max}(t_{cy}^{\max} - t_{cy}^{\min}), \quad \forall c \in C, \quad \forall y \in Y. \tag{9}$$

- Workload balancing constraints:

$$\rho_{\min} \sum_{p \in P} |I_{py}| \leq \sum_{p \in P} |I_{py}| x_{pc} \leq \rho_{\max} \sum_{p \in P} |I_{py}|, \quad \forall c \in C, \quad \forall y \in Y. \tag{10}$$

- Time window constraints for items in the first due year:

$$t_{cy_i^*}^{\min} - \tau_i^* \leq \delta_{iy_i^*}^{\mathrm{del}} + M(1 - x_{pc}), \quad \forall c \in C, \quad \forall p \in P, \quad \forall i \in I_p, \tag{11}$$
$$\tau_i^* - t_{cy_i^*}^{\max} + d_i \leq \delta_{iy_i^*}^{\mathrm{adv}} + M(1 - x_{pc}), \quad \forall c \in C, \quad \forall p \in P, \quad \forall i \in I_p, \tag{12}$$

  where M is a sufficiently large real constant.
- Time window constraints for items in subsequent years:

$$t_{cy}^{\min} - t_{c(y-f_i)}^{\min} \leq \delta_{iy}^{\mathrm{del}} + M(1 - x_{pc}), \quad \forall c \in C, \quad \forall y \in Y : y > y_i^*, \quad \forall p \in P, \quad \forall i \in I_{py}, \tag{13}$$
$$t_{c(y-f_i)}^{\max} - t_{cy}^{\max} \leq \delta_{iy}^{\mathrm{adv}} + M(1 - x_{pc}), \quad \forall c \in C, \quad \forall y \in Y : y > y_i^*, \quad \forall p \in P, \quad \forall i \in I_{py}, \tag{14}$$

  where M is a sufficiently large real constant.
- Shutdown indicator constraints:

$$\xi_{cyn} \geq x_{pc} - \sum_{j \in \psi(n) :\, j \neq n} \xi_{cyj}, \quad \forall c \in C, \quad \forall y \in Y, \quad \forall n \in N, \quad \forall p \in P : \, s_{pyn} = 1, \tag{15}$$

  where the summation on the right-hand side is taken over all nodes except node n on path $\psi(n)$.

The optimization problem is to choose x_{pc}, $t_{cy}^{\min}$, $t_{cy}^{\max}$, and ξ_{cyn} to minimize the objective function (2) subject to constraints (3)-(15). This is a mixed-integer linear programming problem for which real-life instances can be very large—for example, the number of maintenance plans can easily exceed 1000 in practice. For simplicity, the model assumes that each year has 365 days and hence leap years are not considered.

Constraints (3) ensure that each maintenance plan is assigned to exactly one campaign, so that items in the same plan are performed in the same campaign in the years in which they are due. Thus, $x_{pc} = 1$ means that items in plan p are always performed as part of campaign c; they cannot change campaign from year to year.

Constraints (4)-(8) govern the timing of campaigns: (4) ensures that the campaign times are within the specified windows, (5) ensures that the campaign durations are within the specified minimum and maximum limits, (6) ensures that each campaign is sufficiently long to contain the maintenance items that it has been allocated (recall that $\alpha \geq 1$), and (7) and (8) ensure that campaigns do not overlap (recall that $\epsilon > 0$). Note also that constraint (4) forces each maintenance item i to occur during its planned years y_i^*, $y_i^* + f_i$, $y_i^* + 2f_i$, and so on—delaying or advancing an item into a non-planned year is prohibited, even if the maximum delay and advance parameters allow. Thus, for example, maintenance item i due on day 5 of year y cannot be advanced into the previous year even if $\delta_{iy}^{\text{adv}} > 5$.

Constraints (9) and (10) restrict the campaign workloads: the first set of constraints specifies a maximum limit on the labour hours per campaign, and the second set of constraints ensures that the number of items allocated to a campaign is within the minimum and maximum percentages of the total number of items due in that year.

The mathematical formulations of constraints (3)-(10) are self-explanatory. Constraints (11)-(15) require further explanation and this is provided in the next section. In particular, Proposition 1 in the next section shows that constraints (11) and (12) enforce the time window constraints for the first instance of each maintenance item, and similarly Proposition 2 shows that constraints (13) and (14) enforce the time window constraints for subsequent instances of each maintenance item. Propositions 3 and 4 show that constraints (15) ensure the decision variables ξ_{cyn} correctly define which nodes are the dominant shutdown units in each campaign in each year.

3. Time Window and Shutdown Constraints

In this section, we provide the mathematical justification for constraints (11)-(15).

3.1. Time Window Constraints for Maintenance Item Deadlines

Consider an arbitrary maintenance item i. In the first year item i is due (year y_i^*), the time window for commencing item i is $[\tau_i^* - \delta_{iy_i^*}^{\text{adv}}, \tau_i^* + \delta_{iy_i^*}^{\text{del}}]$. Thus, item i can occur in campaign c in year y_i^* if and only if the intersection of $[\tau_i^* - \delta_{iy_i^*}^{\text{adv}}, \tau_i^* + \delta_{iy_i^*}^{\text{del}}]$ and $[t_{cy_i^*}^{\min}, t_{cy_i^*}^{\max} - d_i]$ is non-empty as shown in Figure 5. The following result gives a set of constraints to characterize this condition.

Proposition 1. Maintenance item $i \in I$ can be scheduled in campaign $c \in C$ in year y_i^* if and only if

$$t_{cy_i^*}^{\min} - \tau_i^* \leq \delta_{iy_i^*}^{\text{del}}, \tag{16}$$

$$\tau_i^* - t_{cy_i^*}^{\max} + d_i \leq \delta_{iy_i^*}^{\text{adv}}, \tag{17}$$

$$t_{cy_i^*}^{\min} \leq t_{cy_i^*}^{\max} - d_i. \tag{18}$$

Proof. The intersection of $[t_{cy_i^*}^{\min}, t_{cy_i^*}^{\max} - d_i]$ and $[\tau_i^* - \delta_{iy_i^*}^{\text{adv}}, \tau_i^* + \delta_{iy_i^*}^{\text{del}}]$ is non-empty if

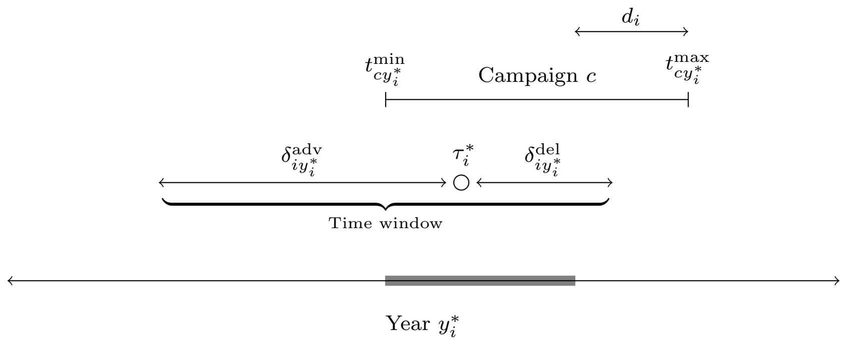


Figure 5. Allocating maintenance item i to campaign c in year y_i^*. The grey area shows the feasible start times for item i.

and only if

$$\max(t^{\min}_{cy_i^*}, \tau_i^* - \delta^{\text{adv}}_{iy_i^*}) \leq \min(t^{\max}_{cy_i^*} - d_i, \tau_i^* + \delta^{\text{del}}_{iy_i^*}). \tag{19}$$

Hence, it is sufficient to show that (16)-(18) are equivalent to (19). Suppose first that (19) holds. Then clearly,

$$\begin{aligned}
&t^{\min}_{cy_i^*} \leq \max(t^{\min}_{cy_i^*}, \tau_i^* - \delta^{\text{adv}}_{iy_i^*}) \leq \min(t^{\max}_{cy_i^*} - d_i, \tau_i^* + \delta^{\text{del}}_{iy_i^*}) \leq \tau_i^* + \delta^{\text{del}}_{iy_i^*},\\
&\tau_i^* - \delta^{\text{adv}}_{iy_i^*} \leq \max(t^{\min}_{cy_i^*}, \tau_i^* - \delta^{\text{adv}}_{iy_i^*}) \leq \min(t^{\max}_{cy_i^*} - d_i, \tau_i^* + \delta^{\text{del}}_{iy_i^*}) \leq t^{\max}_{cy_i^*} - d_i,\\
&t^{\min}_{cy_i^*} \leq \max(t^{\min}_{cy_i^*}, \tau_i^* - \delta^{\text{adv}}_{iy_i^*}) \leq \min(t^{\max}_{cy_i^*} - d_i, \tau_i^* + \delta^{\text{del}}_{iy_i^*}) \leq t^{\max}_{cy_i^*} - d_i,
\end{aligned}$$

from which (16)-(18) are easily obtained. Conversely, assume that (16)-(18) hold. Then from (16) and (18),

$$t^{\min}_{cy_i^*} \leq \min(t^{\max}_{cy_i^*} - d_i, \tau_i^* + \delta^{\text{del}}_{iy_i^*}). \tag{20}$$

Furthermore, from (17) and since both $\delta^{\text{adv}}_{iy_i^*}$ and $\delta^{\text{del}}_{iy_i^*}$ are non-negative,

$$\tau_i^* - \delta^{\text{adv}}_{iy_i^*} \leq \min(t^{\max}_{cy_i^*} - d_i, \tau_i^* + \delta^{\text{del}}_{iy_i^*}). \tag{21}$$

Combining (20) and (21) gives (19), as required. $\square$

For each maintenance plan p, we need to impose (16)-(18) for the unique c such that $x_{pc} = 1$ (the campaign to which p belongs) and all items $i \in I_p$ (the items in plan p). Inequality (18) is in fact already implied by (6). The other inequalities (16) and (17) correspond to constraints (11) and (12), which reduce to (16) and (17) when $x_{pc} = 1$. When $x_{pc} = 0$, these constraints are redundant for M sufficiently large.

Now consider an arbitrary maintenance item i in year $y > y_i^*$. Prior to year y, the previous instance of item i is executed in year $y - f_i$ in a certain campaign c. The precise start time of item i in year $y - f_i$ could be any point in the interval $[t^{\min}_{c(y-f_i)}, t^{\max}_{c(y-f_i)} - d_i]$ and this previous start day is also the due time for item i in year $y > y_i^*$. Hence, item i can occur in campaign c in year y if and only if, for each potential start time $t \in [t^{\min}_{c(y-f_i)}, t^{\max}_{c(y-f_i)} - d_i]$, the intersection of $[t - \delta^{\text{adv}}_{iy}, t + \delta^{\text{del}}_{iy}]$ and

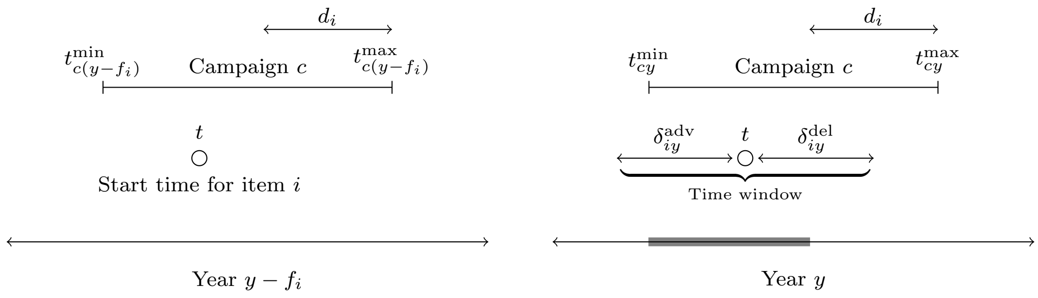


Figure 6. Allocating maintenance item i to campaign c in year $y > y^*_i$. The grey area shows the feasible start times for item i in year y.

$[t^{\min}_{cy}, t^{\max}_{cy} - d_i]$ is non-empty as shown in Figure 6. This then leads to the following result, which is the analogue of Proposition 1 for $y > y^*_i$.

Proposition 2. Maintenance item $i \in I$ can be scheduled in campaign $c \in C$ in year $y > y^*_i$ if and only if

$$t^{\min}_{cy} - t^{\min}_{c(y-f_i)} \le \delta^{\text{del}}_{iy}, \tag{22}$$
$$t^{\max}_{c(y-f_i)} - t^{\max}_{cy} \le \delta^{\text{adv}}_{iy}, \tag{23}$$
$$t^{\min}_{cy} \le t^{\max}_{cy} - d_i. \tag{24}$$

Proof. The intersection of $[t^{\min}_{cy}, t^{\max}_{cy} - d_i]$ and $[t - \delta^{\text{adv}}_{iy}, t + \delta^{\text{del}}_{iy}]$ is non-empty for each $t \in [t^{\min}_{c(y-f_i)}, t^{\max}_{c(y-f_i)} - d_i]$ if and only if

$$\max(t^{\min}_{cy}, t - \delta^{\text{adv}}_{iy}) \le \min(t^{\max}_{cy} - d_i, t + \delta^{\text{del}}_{iy}), \qquad \forall t \in [t^{\min}_{c(y-f_i)}, t^{\max}_{c(y-f_i)} - d_i]. \tag{25}$$

We complete the proof by showing that (22)-(24) are equivalent to (25).

First, for the reverse implication, assume that (25) holds. Then clearly (24) also holds. Choosing $t = t^{\min}_{c(y-f_i)}$ in (25) gives

$$t^{\min}_{cy} \le \max(t^{\min}_{cy}, t^{\min}_{c(y-f_i)} - \delta^{\text{adv}}_{iy}) \le \min(t^{\max}_{cy} - d_i, t^{\min}_{c(y-f_i)} + \delta^{\text{del}}_{iy}) \le t^{\min}_{c(y-f_i)} + \delta^{\text{del}}_{iy},$$

which proves (22). Furthermore, choosing $t = t^{\max}_{c(y-f_i)} - d_i$ in (25) gives

$$\begin{aligned} t^{\max}_{c(y-f_i)} - d_i - \delta^{\text{adv}}_{iy} &\le \max(t^{\min}_{cy}, t^{\max}_{c(y-f_i)} - d_i - \delta^{\text{adv}}_{iy}) \\ &\le \min(t^{\max}_{cy} - d_i, t^{\max}_{c(y-f_i)} - d_i + \delta^{\text{del}}_{iy}) \le t^{\max}_{cy} - d_i, \end{aligned}$$

proving (23).

Now, for the forward implication, assume (22)-(24) hold. Then, from (22) and (24), for all $t \ge t^{\min}_{c(y-f_i)}$,

$$t^{\min}_{cy} \le t^{\min}_{c(y-f_i)} + \delta^{\text{del}}_{iy} \le t + \delta^{\text{del}}_{iy}, \qquad t^{\min}_{cy} \le t^{\max}_{cy} - d_i.$$

Hence,

$$t_{cy}^{\min} \le \min(t_{cy}^{\max} - d_i, t + \delta_{iy}^{\mathrm{del}}). \tag{26}$$

Moreover, from (23) and since $\delta_{iy}^{\mathrm{adv}} \ge 0$ and $\delta_{iy}^{\mathrm{del}} \ge 0$, for all $t \le t_{c(y-f_i)}^{\max} - d_i$,

$$t - \delta_{iy}^{\mathrm{adv}} \le t_{c(y-f_i)}^{\max} - d_i - \delta_{iy}^{\mathrm{adv}} \le t_{cy}^{\max} - d_i, \qquad t - \delta_{iy}^{\mathrm{adv}} \le t + \delta_{iy}^{\mathrm{del}}.$$

Thus,

$$t - \delta_{iy}^{\mathrm{adv}} \le \min(t_{cy}^{\max} - d_i, t + \delta_{iy}^{\mathrm{del}}). \tag{27}$$

Combining (26) and (27) completes the proof. $\square$

As with (18) in Proposition 1, inequality (24) in Proposition 2 is implied by (6). For each maintenance plan p, the other two inequalities in Proposition 2 need to be imposed for the campaign c to which p belongs and for all items $i \in I_{py}$ belonging to plan p in each year $y > y_i^*$. This leads to constraints (13) and (14) in the optimization model. These constraints, along with (11) and (12), ensure that the maintenance deadlines are respected for each maintenance item.

Recall that constraints (11)-(14) involve a parameter M that must be set sufficiently large so that the constraints are redundant when $x_{pc} = 0$. For constraints (11) and (12), it is sufficient to choose M as

$$M = \max_{c \in C,\, i \in I} \left\{ \max\left\{ \tilde{t}_{cy_i^*}^{\max} - \tau_i^* - \delta_{iy_i^*}^{\mathrm{del}}, \tau_i^* - \tilde{t}_{cy_i^*}^{\min} + d_i - \delta_{iy_i^*}^{\mathrm{adv}} \right\} \right\} - \Delta_{\min},$$

since by constraints (4) and (5),

$$\begin{aligned} t_{cy_i^*}^{\min} - \tau_i^* - \delta_{iy_i^*}^{\mathrm{del}} &\le \tilde{t}_{cy_i^*}^{\max} - \Delta_{\min} - \tau_i^* - \delta_{iy_i^*}^{\mathrm{del}}, \\ \tau_i^* - t_{cy_i^*}^{\max} + d_i - \delta_{iy_i^*}^{\mathrm{adv}} &\le \tau_i^* - \tilde{t}_{cy_i^*}^{\min} - \Delta_{\min} + d_i - \delta_{iy_i^*}^{\mathrm{adv}}. \end{aligned}$$

Similarly, for constraints (13) and (14), it is sufficient to choose

$$M = \max_{\substack{c \in C,\, i \in I,\, y \in Y: \\ y > y_i^*,\, i \in \cup_p I_{py}}} \left\{ \max\left\{ \tilde{t}_{cy}^{\max} - \tilde{t}_{c(y-f_i)}^{\min} - \delta_{iy}^{\mathrm{del}}, \tilde{t}_{c(y-f_i)}^{\max} - \tilde{t}_{cy}^{\min} - \delta_{iy}^{\mathrm{adv}} \right\} \right\} - \Delta_{\min},$$

where the outer maximization is taken over all campaigns c and all items i that are due in year $y > y_i^*$. This expression for M follows immediately from

$$\begin{aligned} t_{cy}^{\min} - t_{c(y-f_i)}^{\min} - \delta_{iy}^{\mathrm{del}} &\le \tilde{t}_{cy}^{\max} - \Delta_{\min} - \tilde{t}_{c(y-f_i)}^{\min} - \delta_{iy}^{\mathrm{del}}, \\ t_{c(y-f_i)}^{\max} - t_{cy}^{\max} - \delta_{iy}^{\mathrm{adv}} &\le \tilde{t}_{c(y-f_i)}^{\max} - \tilde{t}_{cy}^{\min} - \Delta_{\min} - \delta_{iy}^{\mathrm{adv}}. \end{aligned}$$

3.2. Shutdown Indicator Constraints

We now turn our attention to the final set of constraints (15). The two main results proved in this section show that under (15), variables ξ_{cyn} correctly define which nodes are the dominant shutdown units in each campaign in each year.

Proposition 3. Suppose that the maintenance plans in P have been allocated to the campaigns in C through the decision variables x_{pc}, $p \in P$, $c \in C$. Based on this allocation and the plant hierarchy tree, for each $c \in C$, $y \in Y$, and $n \in N$, define

$$\xi_{cyn} = \begin{cases} 1, & \text{if node } n \text{ is a dominant shutdown unit for campaign } c \text{ in year } y, \\ 0, & \text{otherwise.} \end{cases}$$

Then ξ_{cyn}, $c \in C$, $y \in Y$, $n \in N$, as defined above are feasible for (15).

Proof. Let $c \in C$, $y \in Y$, and $n \in N$ be arbitrary but fixed. If $\xi_{cyn} = 1$, then

$$x_{pc} - \sum_{j \in \psi(n)\colon j \neq n} \xi_{cyj} \leq 1 - \sum_{j \in \psi(n)\colon j \neq n} \xi_{cyj} \leq 1 = \xi_{cyn}, \quad \forall p \in P : \, s_{pyn} = 1,$$

which shows (15) is satisfied as required. On the other hand, if $\xi_{cyn} = 0$, then for each $p \in P$ with $s_{pyn} = 1$, either $x_{pc} = 0$ (case 1) or $x_{pc} = 1$ (case 2). For case 1,

$$x_{pc} - \sum_{j \in \psi(n)\colon j \neq n} \xi_{cyj} = - \sum_{j \in \psi(n)\colon j \neq n} \xi_{cyj} \leq 0 = \xi_{cyn},$$

which shows (15) is satisfied as required. For case 2, plan p belongs to campaign c and node n is a dominant shutdown unit for plan p in year y, but not for campaign c in year y. Thus, there must exist a dominant shutdown unit j' above node n on path $\psi(n)$ (that is, $j' \in \psi(n) \setminus \{n\}$). This implies $\xi_{cyj'} = 1$ according to our definition and therefore

$$x_{pc} - \sum_{j \in \psi(n)\colon j \neq n} \xi_{cyj} = 1 - \sum_{j \in \psi(n)\colon j \neq n} \xi_{cyj} \leq 1 - \xi_{cyj'} = 0 = \xi_{cyn},$$

which shows that (15) is satisfied as required. □

Proposition 3 shows that using the binary variables ξ_{cyn} to indicate which nodes are the dominant shutdown units in each campaign in each year, as per our modelling approach, is feasible with respect to constraints (15). However, this is not necessarily the only feasible choice, and hence Proposition 3 alone is insufficient to prove that the proposed cost function (2) correctly measures the total shutdown cost over the entire time horizon. To prove that (2) is indeed correct, we require the following additional result, which relies on condition (1) stating that the cost of shutting down an entire unit is greater than the cost of shutting down all of its individual sub-units.

Proposition 4. Suppose that condition (1) holds. Given fixed values of x_{pc}, $p \in P$, $c \in C$, the optimal values of ξ_{cyn}, $c \in C$, $y \in Y$, $n \in N$, that minimize (2) subject to (15) are such that $\xi_{cyn} = 1$ if and only if node n is a dominant shutdown unit for campaign c in year y.

Proof. Since the cost function (2) is linear and the constraints (15) only link variables ξ_{cyn} for the same campaign and the same year, these variables can be optimized independently for each $c \in C$ and $y \in Y$. Hence, in this proof, we let $c \in C$ and $y \in Y$ be fixed but arbitrary and assume that ξ_{cyn}, $n \in N$, are optimal values that minimize $\sum_n w_n \xi_{cyn}$ (the cost terms corresponding to c and y) subject to (15).

We first consider the reverse implication and assume, to the contrary, that campaign c in year y has at least one dominant shutdown unit n with $\xi_{cyn} = 0$. Let N_1 denote the set of all such dominant shutdown units. For any $n \in N_1$, constraint (15) gives

$$x_{pc} - \sum_{j \in \psi(n):\, j \neq n} \xi_{cyj} \leq 0 = \xi_{cyn}, \quad \forall p \in P:\ s_{pyn} = 1. \tag{28}$$

Clearly there exists at least one p with $x_{pc} = 1$ in the above inequality, since otherwise $n \in N_1$ could not be a dominant shutdown unit for campaign c in year y. Let $\bar{p} := \bar{p}(n)$ denote one such p (arbitrarily chosen) with $x_{pc} = 1$ and $s_{pyn} = 1$. Then for each $n \in N_1$, substituting $p = \bar{p}(n)$ into (28) gives

$$x_{\bar{p}c} - \sum_{j \in \psi(n):\, j \neq n} \xi_{cyj} = 1 - \sum_{j \in \psi(n):\, j \neq n} \xi_{cyj} \leq 0.$$

This inequality can only be satisfied if $\xi_{cyj} = 1$ for some $j \in \psi(n) \setminus \{n\}$.

Let $\mu := \mu(n)$ denote one such node with $\mu(n) \in \psi(n) \setminus \{n\}$ and $\xi_{cy\mu} = 1$, and let N_2 denote the set of all such $\mu(n)$:

$$N_2 = \{\mu(n) : n \in N_1\}.$$

Clearly, $\mu(n)$ is above node n in the plant hierarchy and N_1 and N_2 are disjoint. Now, define $\bar{\xi}_{cyn}$, $n \in N$, as follows:

$$\bar{\xi}_{cyn} = \begin{cases} 1, & \text{if } n \in N_1, \\ 0, & \text{if } n \in N_2, \\ \xi_{cyn}, & \text{if } n \notin N_1 \cup N_2. \end{cases}$$

We will show that $\bar{\xi}_{cyn}$, $n \in N$, are feasible with respect to (15), which requires checking constraint (15) for each $n \in N$. There are three cases to consider:

1. $n \in N_1$;
2. $n \notin N_1$, but n is below a node in N_1 in the plant hierarchy; and
3. $n \notin N_1$, and n is not below a node in N_1 in the plant hierarchy.

For case 1,

$$x_{pc} - \sum_{j \in \psi(n):\, j \neq n} \bar{\xi}_{cyj} \leq 1 - \sum_{j \in \psi(n):\, j \neq n} \bar{\xi}_{cyj} \leq 1 = \bar{\xi}_{cyn}, \quad \forall p \in P:\ s_{pyn} = 1, \tag{29}$$

which shows that (15) for $n \in N_1$ is satisfied. For case 2, there exists $n' \in N_1$ such that $n' \in \psi(n) \setminus \{n\}$. Thus,

$$x_{pc} - \sum_{j \in \psi(n):\, j \neq n} \bar{\xi}_{cyj} \leq 1 - \sum_{j \in \psi(n):\, j \neq n} \bar{\xi}_{cyj} \leq 1 - \bar{\xi}_{cyn'} = 0 \leq \bar{\xi}_{cyn}, \quad \forall p \in P:\ s_{pyn} = 1, \tag{30}$$

which again shows that (15) is satisfied in this case. Finally, for case 3, when $x_{pc} = 0$, we have

$$x_{pc} - \sum_{j \in \psi(n)\colon j \neq n} \bar{\xi}_{cyj} = - \sum_{j \in \psi(n)\colon j \neq n} \bar{\xi}_{cyj} \leq 0 \leq \bar{\xi}_{cyn}, \quad \forall p \in P : x_{pc} = 0,\ s_{pyn} = 1. \tag{31}$$

On the other hand, when $x_{pc} = 1$ for plan p with $s_{pyn} = 1$, the dominant shutdown units for campaign c in year y must include a node $n'' \in \psi(n)$ (either node n or a node above node n). The scenario where $n'' \in N_1$ is covered in cases 1 and 2 and thus we may assume that $n'' \notin N_1$. Furthermore, $n'' \notin N_2$, since otherwise there would exist a dominant shutdown unit below n'' in the plant hierarchy, which is impossible because n'' is itself a dominant shutdown unit. Consequently, $\bar{\xi}_{cyn''} = \xi_{cyn''} = 1$ and for $n = n''$,

$$x_{pc} - \sum_{j \in \psi(n)\colon j \neq n} \bar{\xi}_{cyj} = 1 - \sum_{j \in \psi(n)\colon j \neq n} \bar{\xi}_{cyj} \leq 1 = \bar{\xi}_{cyn}, \qquad \forall p \in P : x_{pc} = 1,\ s_{pyn} = 1, \tag{32}$$

and for $n \neq n''$,

$$x_{pc} - \sum_{j \in \psi(n)\colon j \neq n} \bar{\xi}_{cyj} = 1 - \sum_{j \in \psi(n)\colon j \neq n} \bar{\xi}_{cyj} \leq 1 - \bar{\xi}_{cyn''} = 0 \leq \bar{\xi}_{cyn}, \qquad \forall p \in P : x_{pc} = 1,\ s_{pyn} = 1. \tag{33}$$

Inequalities (29)-(33) show that $\bar{\xi}_{cyn}$, $n \in N$, are feasible with respect to constraints (15).

We now show that under condition (1),

$$w_n > \sum_{j \in N_1 \colon \mu(j) = n} w_j, \quad \forall n \in N_2. \tag{34}$$

Indeed, this is proved by letting S denote the set of nodes mapped to node $n \in N_2$ under the mapping $\mu(\cdot)$, letting Γ_l denote the set of nodes exactly l levels below node n in the plant hierarchy tree, and then using induction to deduce the following inequality for each integer $k \geq 1$:

$$w_n > \sum_{l=1}^{k} \sum_{j \in \Gamma_l \cap S} w_j + \sum_{j \in \Gamma_k \colon \psi(j) \cap S = \varnothing} w_j, \tag{35}$$

where

$$S = \big\{ j \in N_1 : \mu(j) = n \big\}, \quad \Gamma_l = \big\{ j \in N : n \in \psi(j) \setminus \{j\},\ |\psi(j) \setminus \psi(n)| = l \big\}.$$

The first term on the right-hand side above is a summation over all nodes in S that are no more than k levels below node n in the hierarchy tree, and the second term is a summation over all nodes that are exactly k levels below node n and whose paths to the root do not contain a node in S. When k is sufficiently large (at least as large as the number of levels below node n), inequality (35) simplifies to (34) as shown below:

$$w_n > \sum_{l=1}^{k} \sum_{j\in\Gamma_l\cap S} w_j + \sum_{j\in\Gamma_k:\,\psi(j)\cap S=\varnothing} w_j \geq \sum_{l=1}^{k} \sum_{j\in\Gamma_l\cap S} w_j = \sum_{j\in N_1:\,\mu(j)=n} w_j.$$

The basis step ($k = 1$) for (34) follows immediately from splitting the summation in (1) into two parts: one for $j \in S$ and one for $j \notin S$. For the induction step, assuming (35) is true for k, applying (1) to each node in the second summation gives

$$\begin{aligned}
w_n &> \sum_{l=1}^{k} \sum_{j\in\Gamma_l\cap S} w_j + \sum_{j\in\Gamma_k:\,\psi(j)\cap S=\varnothing} w_j \\
&\geq \sum_{l=1}^{k} \sum_{j\in\Gamma_l\cap S} w_j + \sum_{j\in\Gamma_k:\,\psi(j)\cap S=\varnothing} \sum_{j'\in N:\,\psi(j')\setminus\psi(j)=j'} w_{j'} \\
&= \sum_{l=1}^{k} \sum_{j\in\Gamma_l\cap S} w_j + \sum_{j\in\Gamma_{k+1}:\,\psi(j)\cap S=\varnothing} w_j + \sum_{j\in\Gamma_{k+1}\cap S} w_j = \sum_{l=1}^{k+1} \sum_{j\in\Gamma_l\cap S} w_j + \sum_{j\in\Gamma_{k+1}:\,\psi(j)\cap S=\varnothing} w_j,
\end{aligned}$$

which proves the induction step as required.

Using (34), the cost of $\bar{\xi}_{cyn}$, $n \in N$, satisfies

$$\begin{aligned}
\sum_{n\in N} w_n\bar{\xi}_{cyn} &= \sum_{n\in N_1} w_n + \sum_{n\notin N_1\cup N_2} w_n\xi_{cyn} \\
&= \sum_{n\in N_2} \sum_{j\in N_1:\,\mu(j)=n} w_j + \sum_{n\notin N_1\cup N_2} w_n\xi_{cyn} \\
&< \sum_{n\in N_2} w_n + \sum_{n\notin N_1\cup N_2} w_n\xi_{cyn} = \sum_{n\in N} w_n\xi_{cyn},
\end{aligned}$$

where the second equality is a consequence of $\mu(\cdot)$ being a many-to-one and onto mapping from N_1 to N_2, and the inequality in the middle follows from (34). Thus, by swapping the values of ξ_{cyn} for $n \in N_1$ and $n \in N_2$, we have reduced the cost while maintaining feasibility, contradicting the optimality of the current solution. Therefore, if n is a dominant shutdown unit for campaign c in year y, then we must have $\xi_{cyn} = 1$. This proves the reverse implication for the proposition.

To prove the forward implication, suppose that $\xi_{cyn'} = 1$ but n' is not a dominant shutdown unit for campaign c in year y. Then since $w_{n'} > 0$ and each dominant shutdown unit n'' also has $\xi_{cyn''} = 1$ (as per the first part of the proof), the cost of ξ_{cyn}, $n \in N$, is greater than the cost of the feasible solution in Proposition 3, which contradicts optimality. This proves the forward implication as required. □

Propositions 3 and 4 together explain the need for constraints (15). Proposition 4 shows that, at any optimal solution of problem (2)-(15), the binary variables ξ_{cyn} correctly indicate the dominant shutdown units, and thus (2) correctly measures the total shutdown cost. Since for any feasible allocation of plans to campaigns, defining ξ_{cyn} to indicate the dominant shutdown units as per Proposition 3 is always feasible, it follows that problem (2)-(15) is guaranteed to give the optimal allocations and campaign times that minimize overall cost.

4. Numerical Simulations

4.1. Case Study: Karratha Gas Plant in Western Australia

4.1.1. Background

To test the optimization model in Section 2, we considered maintenance data for LNG Train 5 at Karratha Gas Plant. This plant is operated by Woodside Energy and is the main processing facility for the North West Shelf oil and gas project in Australia. The plant hierarchy tree for LNG Train 5 has five levels as listed below:

- Plant section (entire LNG Train 5);
- Function;
- System;
- Sub-system; and
- Operation package.

Thus, LNG Train 5 contains multiple functions, each function contains multiple systems, each system contains multiple sub-systems, and each sub-system contains multiple packages. Plant section shutdowns are the highest priority, followed by function shutdowns, system shutdowns, and so on.

Each maintenance item is assigned a code that indicates the level of shutdown required to perform the maintenance item. The possible values are:

- 0 – no isolation required;
- 1 – local isolation only;
- 2 – operation package shutdown;
- 3 – sub-system shutdown;
- 4 – system shutdown;
- 5 – function shutdown; and
- 6 – complete plant section shutdown.

For example, if a maintenance item is assigned code 3, then the sub-system corresponding to this maintenance item must be shut down when the maintenance item is performed. This information defines the shutdown indicator parameters s_{pyn} for each maintenance plan.

4.1.2. Data Set

The data set for LNG Train 5 involves 1,206 maintenance plans and 2,166 maintenance items. The numbers of functions, systems, sub-systems, and operation packages are given in Figure 7. For each maintenance item i, the data set defines the next due time (y_i^* and τ_i^*), the duration (d_i), the frequency (f_i), and the suppression hierarchy value (h_i). Parameters r_i and $R_{\max}$ are not given in the data set and thus we ignored constraint (9) in these simulations.

The item frequencies range from 1 year to 25 years. Interestingly, although the plant hierarchy tree has five levels, the dominant shutdown units are all in the top and bottom levels: 3.37% of the maintenance items require operation package shutdown, 35% require a complete LNG Train 5 shutdown, and the remainder require only local isolation or no isolation.

Different test problems were defined by changing the number of campaigns, the minimum/maximum ratio parameters $\rho_{\min}$ and $\rho_{\max}$, and the maximum advance/delay parameters δ_{iy}^{adv} and δ_{iy}^{del} for $y \neq y_i^*$. The maximum advance parameter for the first

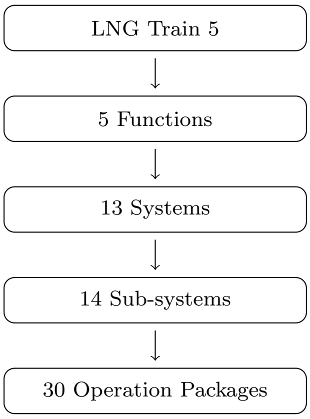


Figure 7. Number of nodes in each level of the hierarchy tree for LNG Train 5 in Karratha Gas Plant.

instance of each maintenance item was fixed to be 35% of the item's frequency:

$$\delta^{\mathrm{adv}}_{iy_i^*} = 0.35 \times f_i \times 365 = 127.75 f_i, \quad \forall i \in I.$$

Similarly, the maximum delay parameter for an item's first instance was also fixed to be 35% of the frequency, except for those items designated as technical integrity (TI) items for which delays are tightly controlled:

$$\delta^{\mathrm{del}}_{iy_i^*} = \begin{cases} 5 \text{ days}, & \text{for TI items}, \\ 0.35 \times f_i \times 365 = 127.75 f_i, & \text{for non-TI items}, \end{cases} \qquad \forall i \in I.$$

About 21% of the maintenance items in the data set are TI items. In the test scenarios, the first advance/delay limits $\delta^{\mathrm{adv}}_{iy_i^*}$ and $\delta^{\mathrm{del}}_{iy_i^*}$ (fixed as above) are always wider than the subsequent limits to give maximum opportunity for maintenance items to be scheduled in the correct campaign.

Each test problem has a time horizon of 20 years and the minimum and maximum campaign durations are

$$\Delta_{\mathrm{min}} = 20 \text{ days}, \quad \Delta_{\mathrm{max}} = 50 \text{ days}.$$

The other parameters in the constraints were chosen as

$$\epsilon = 60, \quad \alpha = 1, \quad \tilde{t}^{\mathrm{min}}_{cy} = 0, \quad \tilde{t}^{\mathrm{max}}_{cy} = 365.$$

Finally, the weights in the objective function are

$$w_n = \begin{cases} 1, & \text{if } n \text{ corresponds to an operation package}, \\ \frac{3}{2} \sum_{j \in N:\, \psi(j) \setminus \psi(n) = j} w_j, & \text{otherwise}, \end{cases}$$

where $\psi(j)$ and $\psi(n)$ are as defined in Section 2. These cost values penalize the overall shutdown of a node by 50% more than the individual shutdown of all of its child nodes. Actual cost data was not available, hence the need for the proxy costs defined above.

Recall that the maximum item frequency is 25 years, but the time horizon is only 20 years. This does not mean, however, that such "long-cycle" items should be excluded

Problem	Campaigns	$\rho_{\min}$	$\rho_{\max}$	Delay/Advance ($y \neq y_i^*$) $\delta_{iy}^{\text{adv}} = \delta_{iy}^{\text{del}}$ (non-TI for delay)
1	2	0.3	0.7	$0.20 f_i \times 365$
2	2	0.2	0.8	$0.05 f_i \times 365$
3	3	0.3	0.7	$0.20 f_i \times 365$
4	3	0.2	0.8	$0.05 f_i \times 365$
5	4	0.2	0.8	$0.05 f_i \times 365$

Table 4. Specifications for the five test problems in Section 4.1.

from the model; depending on when they were last executed, these items may still be due during the 20-year time horizon and hence they must be considered.

4.1.3. Results and Discussion

We implemented the linear programming model in the AIMMS optimization package, with CPLEX version 12.8 used as the solver. All simulations were performed on a standard desktop computer with an Intel Core i5-7400 processor and 8GB of RAM.

Five test problems were considered; see Table 4 for the full specifications. Problems 1 and 2 are for two campaigns, problems 3 and 4 are for three campaigns, and problem 5 is for four campaigns. Problems 1 and 3 have the tightest constraints on campaign workload balancing with $(\rho_{\min}, \rho_{\max}) = (0.3, 0.7)$, compared with $(\rho_{\min}, \rho_{\max}) = (0.2, 0.8)$ for the other problems. Problems 2, 4, and 5 have the tightest constraints on item advancing and delaying in years $y \neq y_i^*$, with the maximum advance/delay limits set at 5% of the item frequency (excluding the delay times for TI items, where the maximum delay is always 5 days). This 5% limit in problems 2, 4, and 5 compares with 20% for the other problems.

The dimensions of the test problems depend exclusively on the number of campaigns and not on the values of $\rho_{\min}$, $\rho_{\max}$, and $\delta_{iy}^{\text{adv}} = \delta_{iy}^{\text{del}}$. The test problems with two campaigns (problems 1 and 2) contain 4,932 integer variables, 81 continuous variables, and 57,056 constraints; the test problems with three campaigns (problems 3 and 4) contain 7,398 integer variables, 121 continuous variables, and 84,981 constraints; and the test problem with four campaigns (problem 5) contains 9,864 integer variables, 161 continuous variables, and 112,906 constraints.

CPLEX could solve each test problem in just a few seconds and the results are reported in Table 5. The table contains the following metrics for each optimal schedule:

- Number of plans allocated to each campaign;
- Number of plant and package shutdowns over the time horizon;
- Number of items advanced and delayed; and
- Average advance and delay times in days.

Here, a maintenance item i is considered to be delayed if its first due time τ_i^* is before the mid-point of the campaign to which it has been assigned; otherwise, the maintenance item is considered to be advanced. This can be expressed mathematically by

$$\tau_i^* - \sum_{c \in C} \sum_{p \in P:\, i \in I_p} \tfrac{1}{2}(t_{cy_i^*}^{\min} + t_{cy_i^*}^{\max}) x_{pc} \begin{cases} < 0 & \rightarrow \quad \text{item } i \text{ delayed,} \\ \geq 0 & \rightarrow \quad \text{item } i \text{ advanced.} \end{cases} \tag{36}$$

Problem	Plans Allocated				Shutdowns		Item Advances		Item Delays	
	C1	C2	C3	C4	Plant	Package	Total	Ave. (days)	Total	Ave. (days)
1	896	310	-	-	30	30	2109	140	57	24
2	1044	162	-	-	30	30	2086	147	80	21
3	566	306	334	-	30	45	1805	123	361	26
4	836	174	196	-	30	30	1906	140	260	28
5	539	221	234	212	30	50	1482	123	684	67

Table 5. Numerical results for Section 4.1.

The metric on the left of (36) is used to quantify the advance and delay times for the averages in Table 5. Note that this is only an approximate characterization of whether an item is delayed or advanced because the model does not determine the precise times at which maintenance items occur. In practice, the detailed scheduling of maintenance items within a certain campaign would occur later after the long-term schedule has been determined.

For both test problems with two campaigns, the optimal solution involves 30 plant shutdowns and 30 package shutdowns, and the number of shutdowns could not be reduced by further relaxing the constraints. The most flexible case we considered (not reported in the table) was $(\rho_{\min}, \rho_{\max}) = (0.2, 0.8)$ with maximum advance/delay limits set at 20% of the item frequency and this still yielded 30 plant shutdowns and 30 package shutdowns for two campaigns. For three campaigns, tightening the workload balancing constraints in problem 3 forces the optimal solution to have more package shutdowns than problem 4. For four campaigns, $(\rho_{\min}, \rho_{\max}) = (0.3, 0.7)$ is always infeasible since the lower bound $\rho_{\min} = 0.3$ is impossible to achieve.

The optimal campaign schedules for test problems 2, 4, and 5 are shown in Figure 8. Due to the different number of campaigns, the optimal schedules differ substantially, even though the schedules for test problems 2 and 4 involve the same number of shutdowns. Thus, even if the number of shutdowns does not change between problem scenarios, the campaign timings and durations needed to achieve the optimum number of shutdowns may be very different. In Figure 8 we also observe variability in the campaign timings from year to year due to the different item frequencies.

Test problems 1–5 model the situation where the timing of the first instance of each maintenance item can be optimized freely (up to 35% of the item frequency), but subsequent instances are more tightly constrained. For items with frequencies of three years and above, the maximum advance/delay limits for the first instance exceed 365 days, and thus these maintenance items can occur anywhere across the year. This is equivalent to ignoring the previous completion time and effectively re-setting the maintenance strategy from the start of the first year. This provides maximum flexibility to perform similar maintenance items together as part of the same campaign, thus reducing overall shutdown cost.

For each test problem, the dominant shutdown units in each campaign are either at the plant section level (the entire LNG Train 5) or the operation package level (the lowest level). This is because none of the maintenance items require separate system, sub-system, or function shutdowns, even though these are possible according to the plant hierarchy. The optimal solution for each test problem involved 30 LNG Train 5 shutdowns and either 30, 45, or 50 separate package shutdowns. There is naturally a high number of plant section shutdowns in the optimal schedules because a large proportion of maintenance items (around 35%) require a full plant shutdown. Although

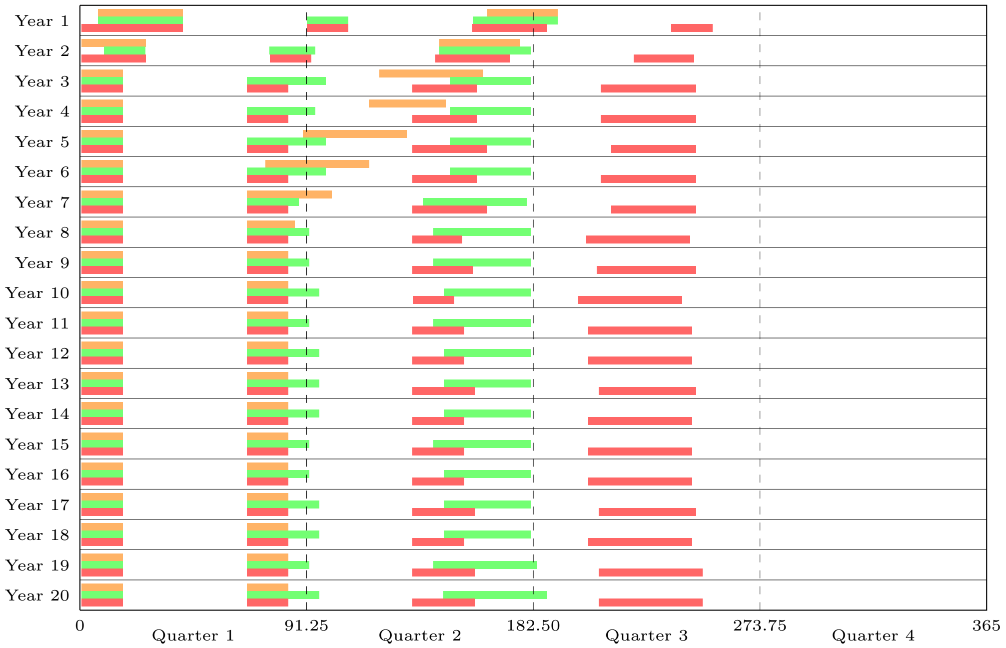


Figure 8. Optimal campaign schedules for problems 2, 4, and 5 in Section 4.1: the orange bars denote the campaigns for problem 2, the green bars denote the campaigns for problem 4, and the red bars denote the campaigns for problem 5.

the number of plant shutdowns is always the same, the schedules needed to achieve this are different, as seen in Figure 8 and the plan allocations in Table 5. There is also considerable variability in the average campaign duration, which ranges from 24.9 days for problem 2 to 36.1 days for problem 1.

In the optimal campaign allocations corresponding to the results in Table 5, more maintenance items are advanced than delayed. This is expected because for TI items the constraints on under-maintenance are much tighter than the constraints on over-maintenance. This is also the reason why the campaigns tend to be scheduled earlier in the year; see Figure 8. As the number of campaigns is reduced, the total number of package shutdowns also tends to reduce—again, this is expected because more items are consolidated into fewer campaigns. The trade-off is that the campaigns become larger and thus more costly, but this cost is not factored into the model.

Finally, we re-iterate that the results presented here do not take into account labour hours, and hence the optimal schedules may not be feasible in practice if there are tight restrictions on worker availability. Nevertheless, the workload balancing constraints defined by $\rho_{\min}$ and $\rho_{\max}$ ensure that the campaigns are not too lopsided in terms of item numbers.

4.2. Random Scenarios

4.2.1. Data Set

One of the limitations with the data set in Section 4.1 is that the item resource requirements are not included and thus constraint (9) is meaningless. Another limitation is that all of the dominant shutdown units are in the top and bottom levels of the plant hierarchy tree—the plant section and operation packages, respectively. This means that the binary variables ξ_{cyn} for the intermediate hierarchy levels—the plant sub-systems, systems, and functions—are essentially redundant.

Thus, to further test the model in Section 2, we randomly generated a set of 20 additional test problems in which all plant hierarchy levels are used and the maintenance items include resource data. Like the problems in Section 4.1, these random test problems involve five hierarchy levels and use shutdown codes in the range $\{0, \ldots, 6\}$ to define the shutdown level required. The plant hierarchy tree for these problems has 76 nodes: five children for the top node, and two children each for every other node. Hence, there is one node in the top level (representing the entire plant), 5 nodes in the second level, 10 nodes in the third level, 20 nodes in the fourth level, and 40 nodes in the bottom level. This is slightly larger than the plant hierarchy tree in Section 4.1.

Each test problem involves 3 campaigns, 3,000 maintenance items, and 1,000 potential maintenance plans. The problems were generated by randomly selecting the following data for each maintenance item uniformly from the given range:

- Maintenance plan containing the item – random integer in $\{1, \ldots, 1000\}$;
- System condition code – random integer in $\{0, \ldots, 6\}$;
- Frequency in years – random integer in $\{1, \ldots, 10\}$;
- Duration in days – random number in $[0, 4]$;
- Suppression hierarchy value – random integer in $\{1, \ldots, 5\}$;
- Resources – random integer in $\{1, \ldots, 5\}$;
- Location in the plant hierarchy tree – random integer in $\{37, \ldots, 76\}$ (the set of nodes representing operation packages in the hierarchy tree); and
- First execution year and day – defined by a random date between the start of the time horizon and the start date plus the frequency.

In addition, each maintenance item was designated as TI according to a Bernoulli distribution with probability 20%.

For the constraints, we chose $\rho_{\min} = 0.2$, $\rho_{\max} = 0.8$, and

$$\delta_{iy}^{\mathrm{adv}} = \begin{cases} 0.35 \times f_i \times 365 = 127.75 f_i, & \text{for } y = y_i^*, \\ 0.10 \times f_i \times 365 = 36.50 f_i, & \text{for } y \neq y_i^*, \end{cases}$$

$$\delta_{iy}^{\mathrm{del}} = \begin{cases} 10 \text{ days}, & \text{for TI items}, \\ 0.35 \times f_i \times 365 = 127.75 f_i, & \text{for non-TI items and } y = y_i^*, \\ 0.10 \times f_i \times 365 = 36.50 f_i, & \text{for non-TI items and } y \neq y_i^*. \end{cases}$$

The other parameters in the constraints were assigned the same values as in Section 4.1 (except for $R_{\max}$, which is not considered in Section 4.1):

$$\Delta_{\min} = 20, \quad \Delta_{\max} = 50, \quad \epsilon = 60, \quad \alpha = 1, \quad R_{\max} = 240, \quad \tilde{t}_{cy}^{\min} = 0, \quad \tilde{t}_{cy}^{\max} = 365.$$

The cost values w_n were also defined as in Section 4.1. The complete data set for the 20 test problems is available online (see Loxton and Mardaneh (2020)).

4.2.2. Results and Discussion

The random test problems are slightly larger than the problems in Section 4.1 because they involve more maintenance items and a larger plant hierarchy tree, but the increase in dimensions is not massive. Nevertheless, the random problems proved far more

Problem	Constraints	Variables		Best Cost	Optimality Gap
		Integer	Continuous		
1	138,729	7,431	121	3,747	3.40%
2	135,976	7,434	121	3,743	1.66%
3	142,878	7,422	121	3,710	2.40%
4	140,231	7,437	121	3,690	0.69%
5	141,573	7,413	121	3,759	3.82%
6	138,344	7,401	121	3,711	0.38%
7	142,671	7,440	121	3,751	2.71%
8	139,607	7,410	121	3,720	3.96%
9	140,064	7,386	121	3,764	3.12%
10	141,144	7,413	121	3,710	2.70%
11	142,163	7,401	121	3,711	3.28%
12	137,513	7,401	121	3,730	3.48%
13	139,433	7,401	121	3,818	4.24%
14	137,642	7,437	121	3,762	3.76%
15	138,667	7,398	121	3,831	3.60%
16	139,720	7,398	121	3,689	3.47%
17	140,445	7,422	121	3,726	2.85%
18	137,563	7,371	121	3,707	2.04%
19	136,894	7,398	121	3,796	3.09%
20	140,203	7,407	121	3,685	1.88%

Table 6. Numerical results for Section 4.2.

difficult to solve. This is because, as we have already mentioned, many of the binary variables and constraints in the previous test problems are redundant and can be quickly eliminated by CPLEX. Despite the increase in complexity, we could still achieve excellent results using CPLEX: after 2 hours of run time for each problem, the average optimality gap was 2.83%. Running CPLEX for longer would likely result in a tighter optimality gap, and the additional run time would certainly be justified in practice given the long-term nature of this planning problem.

Table 6 shows the individual results for each test problem after applying CPLEX version 12.8 for two hours (using the same computer as for the simulations in Section 4.1). Although these results do not have any practical meaning, they do show that the optimization model can be solved to near-optimality in a reasonable amount of time, even for scenarios that are far more complex than the real scenarios in Section 4.1. Note that unlike in Section 4.1, the number of binary variables and constraints here changes from problem to problem because of the random variation in the number of items per plan. The number of continuous variables, which only depends on the years and campaigns, is the same for every problem.

5. Conclusion

In this paper we have developed an integer programming approach for long-term maintenance scheduling in oil and gas plants. The problem is to choose the start and end times for a series of discrete maintenance operations, as well as the work allocated to each of these maintenance operations, over a multi-year time horizon. Although our focus has been on oil and gas plant maintenance, the proposed approach is also applicable to other situations and industries where heavy maintenance work is concentrated into several large-scale, discrete maintenance operations, such as plant maintenance at remote mine sites.

Despite the large dimensions involved, the optimization model in Section 2 could be solved effectively using the commercial solver CPLEX. The test problems in Section 4, which are based on real data from Australian oil and gas company Woodside Energy, took only a few seconds to solve completely on a basic desktop computer, while for the more complex randomly-generated problems in Section 4.2, a tight optimality gap could be achieved in around two hours of computer run time (the average gap for these problems was 2.83%). This is a relatively small computational cost given that the problem itself is formulated over decades and involves numerous large-scale maintenance campaigns and thousands of recurring maintenance activities.

Most of the existing literature on maintenance scheduling is focused on short-term planning for a single maintenance operation (such as the RCPS), and not on the long-term planning of multiple maintenance operations. The present paper is an initial step towards filling this gap in the literature. Looking ahead, this foundational work could potentially be extended in several important directions.

- The current model does not incorporate the costs of over- and under-maintenance and the costs of larger and more expensive campaigns, which is the downside of concentrating more work into fewer campaigns to reduce shutdowns.
- The model could be adapted to allow the number of campaigns to vary from year to year based on the maintenance requirements, leading to a new set of decision variables and more opportunities for optimization.
- Apart from shutdowns, the current model does not consider other synergies between maintenance items, such as two items being in the same location or requiring the same type of resources. Obviously, it would be preferable to schedule such items together in the same campaign where possible, just as it is preferable to schedule items with the same shutdowns together.
- The current model could also be extended by incorporating high-frequency items with frequencies of under one year, and allowing low-frequency items (say those with frequencies of ten years or above) to be delayed and advanced into other years before and after the due year.

Disclosure statement

The views expressed in this paper are the authors' personal views and are not attributable to Woodside Energy Ltd. The paper is intended to be a general discussion and does not constitute advice. Woodside disclaims all liability for any loss or damage incurred by any person or organization as a result of using, acting on, relying on, or disclosing any of the information contained in this paper or otherwise in connection with this paper.

References

Agarwal, A., Colak, S., & Erenguc, S. (2011). A neurogenetic approach for the resource-constrained project scheduling problem. *Computers and Operations Research*, 38(1), 44–50.

Al-Turki, U., Duffuaa, S., & Bendaya, M. (2019). Trends in turnaround maintenance planning: literature review. *Journal of Quality in Maintenance Engineering*, 25(2), 253–271.

Blazewicz, J., Lenstra, J. K., & Kan, A. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5(1), 11–24.
Boctor, F. F. (1990). Some efficient multi-heuristic procedures for resource-constrained project scheduling. *European Journal of Operational Research*, 49(1), 3–13.
Coelho, J., & Vanhoucke, M. (2018). An exact composite lower bound strategy for the resource-constrained project scheduling problem. *Computers and Operations Research*, 93, 135–150.
Demeulemeester, E., & Herroelen, W. (1992). A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science*, 38(12), 1803–1818.
Dorndorf, U., Pesch, E., & Phan-Huy, T. (2000). A branch-and-bound algorithm for the resource-constrained project scheduling problem. *Mathematical Methods of Operations Research*, 52(3), 413–439.
Ebrahimipour, V., Najjarbashi, A., & Sheikhalishahi, M. (2015). Multi-objective modeling for preventive maintenance scheduling in a multiple production line. *Journal of Intelligent Manufacturing*, 26(1), 111–122.
Fuqiong, Z., Zhigang, T., & Yong, Z. (2013). Uncertainty quantification in gear remaining useful life prediction through an integrated prognostics method. *IEEE Transactions on Reliability*, 62(1), 146–159.
Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1), 1–14.
Jarboui, B., Damak, N., Siarry, P., & Rebai, A. (2008). A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation*, 195(1), 299–308.
Khorasgani, H., Biswas, G., & Sankararaman, S. (2016). Methodologies for system-level remaining useful life prediction. *Reliability Engineering and System Safety*, 154, 8–18.
Kolisch, R. (1996). Efficient priority rules for the resource-constrained project scheduling problem. *Journal of Operations Management*, 14(3), 179–192.
Kolisch, R., & Hartmann, S. (1999). Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis. In *Project Scheduling* (pp. 147–178). Springer, Boston, USA.
Kolisch, R., & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, 174(1), 23–37.
Lambrechts, O., Demeulemeester, E., & Herroelen, W. (2008). A tabu search procedure for developing robust predictive project schedules. *International Journal of Production Economics*, 111(2), 493–508.
Li, K. Y., & Willis, R. J. (1992). An iterative scheduling technique for resource-constrained project scheduling. *European Journal of Operational Research*, 56(3), 370–379.
Loxton, R. & Mardaneh, E. (2020). Random Data for Testing a Maintenance Optimization Algorithm. Online webpage. URL: http://dx.doi.org/10.25917/5e1feb1386c72
Megow, N., Mohring, R. H., & Schulz, J. (2011). Decision support and optimization in shutdown and turnaround scheduling. *INFORMS Journal on Computing*, 23(2), 189–204.
Mika, M., Waligóra, G., & Węglarz, J. (2015). Overview and state of the art. In *Handbook on Project Management and Scheduling*, Vol. 1 (pp. 445–490), Springer.
Moinian, F., Sabouhi, H., Hushmand, J., Hallaj, A., Khaledi, H., & Mohammadpour, M. (2017). Gas turbine preventive maintenance optimization using genetic algorithm. *International Journal of System Assurance Engineering and Management*, 8(3), 594–601.
Pellerin, R., Perrier, N., & Berthaut, F. (2019). A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, in press.
Pillac, V., Guret, C., & Medaglia, A. (2013). A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, 7(7), 1525–1535.
Ranjbar, M., Khalilzadeh, M., Kianfar, F., & Etminani, K. (2012). An optimal procedure for minimizing total weighted resource tardiness penalty costs in the resource-constrained

project scheduling problem. *Computers and Industrial Engineering*, 62(1), 264–270.

Sankararaman, S. (2015). Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction. *Mechanical Systems and Signal Processing*, 52–53(1), 228–247.

Shi, H., & Zeng, J. (2016). Real-time prediction of remaining useful life and preventive opportunistic maintenance strategy for multi-component systems considering stochastic dependence. *Computers and Industrial Engineering*, 93, 192–204.

Shim, S. O., & Kim, Y. D. (2007). Scheduling on parallel identical machines to minimize total tardiness. *European Journal of Operational Research*, 177(1), 135–146.

Sprecher, A. (2000). Scheduling resource-constrained projects competitively at modest memory requirements. *Management Science*, 46(5), 710–723.

Yoosefzadeh, H. R., & Tareghian, H. R. (2013). Hybrid solution method for resource-constrained project scheduling problem using a new schedule generator. *The International Journal of Advanced Manufacturing Technology*, 66(5–8), 1171–1180.

Zamorano, E., & Stolletz, R. (2017). Branch-and-price approaches for the multiperiod technician routing and scheduling problem. *European Journal of Operational Research*, 257(1), 55–68.

Ziarati, K., Akbari, R., & Zeighami, V. (2011). On the performance of bee algorithms for resource-constrained project scheduling problem. *Applied Soft Computing*, 11(4), 3720–3733.