Approximating the Pareto frontier for bi-objective preventive maintenance and workshop scheduling

A Lagrangean lower bounding methodology for evaluating contracting forms

Gabrijela Obradović · Ann-Brith Strömberg · Felix Held · Kristian Lundberg

Abstract Effective planning of preventive maintenance plays an important role in maximizing the operational readiness of any industrial system. We consider an operating system and a maintenance workshop governed by two stakeholders who collaborate based on a mutual contract: components of the operating system that need maintenance are sent to the maintenance workshop, where necessary maintenance activities are performed and after which the maintained components are returned to the operating systems and ready to be used again. While the maintenance activities must obey the workshop capacity, the components should be returned to the operating system within a contracted time frame. For this problem, we developed in a previous work a mixed-integer linear optimizaiton model incorporating stocks of damaged as well as repaired components, workshop scheduling, and preventive maintenance planning for the operating system. We then investigated an availability contract between the stakeholders and which is in the paper at hand compared with a turn–around–time contract type, which is more often used in reality. Since, for real instance sizes, the latter leads to a computationally demanding

Gabrijela Obradović
Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, SE-412 96 Gothenburg, Sweden
E-mail: gabobr@chalmers.se

Ann-Brith Strömberg
Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, SE-412 96 Gothenburg, Sweden
E-mail: anstr@chalmers.se

Felix Held
Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, SE-412 96 Gothenburg, Sweden
E-mail: felix.held@chalmers.se

Kristian Lundberg
Saab AB, Linköping & Chalmers University of Technology, Gothenburg, Sweden
E-mail: kristian.lundberg@saabgroup.com

bi-objective optimization problem, we use Lagrangean relaxation and subgradient optimization to compute local lower bounds on the set of non-dominated points, complemented with math-heuristics to identify good feasible solutions (i.e., local upper bounds). Our suggested method thus provides a bounding of the set of non-dominated points for a turn–around time contract.

Keywords System Maintenance · Workshop Scheduling · Mixed-Integer Linear Optimization Model · Optimization of Contracting Forms · Simultaneous Scheduling · Bi-Objective Optimization · Subgradient Optimization · Lagrangean Relaxation

1 Introduction

Maintenance is performed in order for a system to remain in/get restored to its operational state (Swanson, 2001). Maintaining a system typically means repairing, replacing, overhauling, inspecting, servicing, adjusting, or testing the system and/or its components, so that there are no interruptions of the system's planned operations. The outcome of an effective maintenance planning is a reduced risk of failure (Papakostas et al., 2010) and an optimal use of the system's life (and of lives of its components). *Preventive maintenance* (PM) is planned and performed after a specified period of time, or when a specified system has been used for a certain period of time, in order to reduce the probability of system failure. *Corrective maintenance* (CM), on the other hand, is performed after a failure has occurred as a corrective measure to restore the system into an operational state. CM typically comes with a higher cost, since it is often associated with unplanned operation disruptions. We consider PM scheduling, while CM is implicitly considered through an additional cost which increases with the time between PM occasions. The increasing cost reflects the increased risk of having to perform CM. See Yu and Strömberg (2021) for a model that uses failure time distributions to model such additional costs.

We consider a setting with two stakeholders, one being the *system operator* and the other being the *maintenance workshop*. The system operator performs the operations (e.g., within train traffic the system operator would operate the trains according to a given timetable), while the maintenance workshop performs the repair of components sent from the system operation to the workshop. The two stakeholders can work independently or they can share their information and cooperate towards a common goal. We study the latter case, in which the stakeholders are integrated, each of them having one (or several) objective(s) they wish to optimize. The collaboration between the stakeholders is typically governed by a contract. We model, study, and compare two types of contracts: an 'availability of repaired components' contract type and a 'turn–around time' contract type, both of which are modelled as bi-objective optimization problems.

Although it is theoretically possible to identify the complete set of non-dominated points for a multi-objective optimization problem, finding an exact description of this set is often computationally too expensive (Shao and

Ehrgott, 2008). For a survey on approximation of the non-dominated set, see (Ruzika and Wiecek, 2005). Prins et al. (2006), propose a two-phase heuristic method for the bi-objective set covering problem, using a primal-dual Lagrangian relaxation to solve single objective set covering problems. Quttineh et al. (2022) approximate the Pareto frontier for a bi-objective set covering problem using an ε-constraint reformulation, a heuristic for set covering problems utilizing Lagrangean multipliers, and subgradient optimization.

For the preventive maintenance and workshop scheduling problem, we have previously presented a model with individual components' flow (Obradović, 2021) and another model with an aggregated flow of components for each component type (Obradović et al., 2022); the goal being to investigate how different contracting forms affect the efficiency of maintenance activities and the flow of components between the systems, as well as the availability of the operating systems over time. Subsequently, we introduced the modeling of jobs[1] along with a model for non-preemptive scheduling of components' repair in the workshop and a new formulation of an 'availability of repaired components' contract between the stakeholders (Obradović et al., 2023).

In the paper at hand we present a partly new bi-objective optimization modelling of the preventive maintenance and workshop scheduling problem, including preemptive and non-preemptive scheduling of components' repair, as well as two different contracting forms: one turn–around time contract and one availability contract. The scheduling model is summarized in Section 2 while the bi-objective problems, based on the contracting forms, are described in Section 3. The contracting forms are investigated and compared with respect to the resulting costs for maintenance as functions of lower limits on stock levels and due dates for component repair, respectively.

The main contribution of this work is an algorithm for bounding the area of uncertainty of a Pareto front for one turn–around time and one maintenance cost objective, and which constitutes a computationally heavy bi-objective optimization problem. In order to manage the computations in a reasonable time for real instance sizes, we use Lagrangean relaxation and subgradient optimization to compute local lower bounds on the set of non-dominated points, complemented with math-heuristics to identify good feasible solutions (i.e., local upper bounds) As an additional result, we present a framework for comparison of the two contracting forms.

The remainder of the article is organized as follows. In Section 2—as a starting point for this paper—we summarize the model (developed by Obradović et al. (2023)), concerning the multi-system preventive maintenance scheduling problem with interval costs (MS-PMSPIC), the structure of the maintenance workshop, the stock dynamics, and their integration with the operational demand on the systems. In Section 3, we define the objectives associated with the two stakeholders—one on the system maintenance side and one on the maintenance workshop side—and present the bi-objective modeling. Tests and results

[1] Every action taken in the maintenance workshop is considered as a job

are presented in Section 6, and in Section 7 we draw conclusions and present ideas for future research.

2 Mathematical model

The scheduling problem, presented in detail in (Obradović et al., 2023), is described as follows. A number of systems are operating to fulfill a common production demand; their operating schedules are assumed to be predefined, resulting in certain time-windows during which maintenance of the systems' components may be performed. While the systems operate their components degrade, which lead to a requirement for maintenance (i.e., service, replacement, or repair of the components). At a maintenance occasion one or several components are taken out of the system, sent to the maintenance workshop for repair, and returned back to the stock of repaired components, ready to be used again (by any of the systems). The components that are sent for repair are instantly replaced by components from the stock of repaired components (be the stock not empty). Thus, there is a circulating flow of individual components, being used and degraded, replaced, repaired or serviced, and then being made available for usage again by a system.

This system–of–systems is modeled such that the operating systems are preserved operational (if possible), and such that the capacity of the maintenance workshop is respected. After a component of type $i \in \mathcal{I} := \{1, \ldots, I\}$ is demounted from a system $k \in \mathcal{K}$ and once it is to be processed in a machine $l \in \mathcal{L}$ in the maintenance workshop, it is assigned a new 'job id', indexed by $n \in \mathcal{N}_i$. To enable a so-called time-indexed modeling (van den Akker et al., 2000) the time is discretized into a set $\mathcal{T} := \{1, \ldots, T\}$ of time steps. Depending on the length of the planning period, the components will undergo repair different many times. Further, J_i is defined as the number of individual components of type $i \in \mathcal{I}$.

The decision variables are defined as

x^{ik}_{st} = 1 if a component type $i \in \mathcal{I}$ in system $k \in \mathcal{K}$ receives PM at times $s \in \{0, \ldots, t-1\}$ and $t \in \{1, \ldots, T+1\}$, but not in-between; 0 otherwise

z^k_t = 1 if maintenance of system $k \in \mathcal{K}$ occurs at time $t \in \mathcal{T}$; 0 otherwise;

u^{inl}_t = 1 if a component type $i \in \mathcal{I}$ starts maintenance at time $t \in \mathcal{T}$ as job $n \in \mathcal{N}_i$ in machine $l \in \mathcal{L}$; 0 otherwise;

$a^i_t(b^i_t)$ = number of individuals of component type $i \in \mathcal{I}$ on the stock of damaged (repaired) components at time $t \in \mathcal{T} \cup \{0\}$;

α^{ink}_t = 1 if an individual of component type $i \in \mathcal{I}$ is taken out of a system $k \in \mathcal{K}$ at time $t \in \mathcal{T}$ and allocated to job $n \in \mathcal{N}_i$; 0 otherwise;

β^i_t = number of individuals of component type $i \in \mathcal{I}$ placed in any of the systems $k \in \mathcal{K}$ at time $t \in \mathcal{T}$.

The feasible set for the integrated problem is modeled by the constraints

$$\sum_{r=1}^{T+1} x_{0r}^{ik} = 1, \qquad i \in \mathcal{I},\ k \in \mathcal{K}, \tag{1a}$$

$$\sum_{s=0}^{t-1} x_{st}^{ik} = \sum_{r=t+1}^{T+1} x_{tr}^{ik}, \qquad i \in \mathcal{I},\ t \in \mathcal{T},\ k \in \mathcal{K}, \tag{1b}$$

$$\sum_{s=0}^{t-1} x_{st}^{ik} \le z_t^k, \qquad i \in \mathcal{I},\ t \in \mathcal{T},\ k \in \mathcal{K}, \tag{1c}$$

$$z_t^k \le \overline{z}_t^k, \qquad t \in \mathcal{T},\ k \in \mathcal{K}, \tag{1d}$$

$$x_{st}^{ik} = 0, \qquad \bar{t}_i \le s + \bar{t}_i < t \le T+1,\ i \in \mathcal{I},\ k \in \mathcal{K}. \tag{1e}$$

$$\sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}_i} \sum_{s=t-p^i+1}^{t} u_s^{inl} \le 1, \qquad t \in \mathcal{T},\ l \in \mathcal{L}, \tag{1f}$$

$$\sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} u_t^{inl} \le 1, \qquad n \in \mathcal{N}_i,\ i \in \mathcal{I}, \tag{1g}$$

$$\sum_{l \in \mathcal{L}} \sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}_i} \sum_{s=t-p^i+1}^{t} u_s^{inl} = \ell_t, \qquad t \in \mathcal{T}, \tag{1h}$$

$$\sum_{n \in \mathcal{N}_i} \alpha_t^{ink} - \sum_{s=0}^{t-1} x_{st}^{ik} = 0, \qquad i \in \mathcal{I},\ k \in \mathcal{K},\ t \in \mathcal{T}, \tag{1i}$$

$$\sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \alpha_t^{ink} \le 1, \qquad n \in \mathcal{N}_i,\ i \in \mathcal{I}, \tag{1j}$$

$$\sum_{n \in \mathcal{N}_i} \left(\sum_{k \in \mathcal{K}} \alpha_t^{ink} - \sum_{l \in \mathcal{L}} u_{t+\delta_a^i}^{inl} \right) + a_{t-1}^i = a_t^i, \quad t \in \{1-\delta_a^i, \dots, T+1\},\ i \in \mathcal{I}, \tag{1k}$$

$$a_t^i \ge 0, \quad t \in \{-\delta_a^i, \dots, T+1\},\ i \in \mathcal{I}, \tag{1l}$$

$$\beta_t^i = \sum_{k \in \mathcal{K}} \sum_{r=t+1}^{T+1} x_{tr}^{ik}, \qquad t \in \mathcal{T},\ i \in \mathcal{I}, \tag{1m}$$

$$b_t^i = b_{t-1}^i - \beta_t^i + \sum_{n \in \mathcal{N}_i} \sum_{l \in \mathcal{L}} u_{t-\delta_b^i-p^i}^{inl}, \qquad t \in \mathcal{T} \cup \{T+1\},\ i \in \mathcal{I}, \tag{1n}$$

$$b_t^i \ge \underline{b}^i, \qquad t \in \mathcal{T},\ i \in \mathcal{I}, \tag{1o}$$

$$J_i = \sum_{k \in \mathcal{K}} \sum_{r \in \overline{T}} x_{0r}^{ik} + \bar{a}_0^i + \bar{b}_0^i + \sum_{r=-\delta_b^i-p^i+1}^{0} \sum_{l \in \mathcal{L}} \sum_{n \in \mathcal{N}_i} \bar{u}_r^{inl}, \qquad i \in \mathcal{I}. \tag{1p}$$

In summary, the set of feasible solutions to our maintenance scheduling problem is modeled by[2] (1) with binary requirements on the variables x_{st}^{ik}, z_t^k, u_t^{inl}, and α_t^{ink}, and non-negativity and integer requirements on the variables a_t^i, b_t^i, β_t^i, and ℓ_t, for all relevant values of the indices.

3 Optimization objectives and contracting forms

We next define an availability of components contract (Obradović et al., 2023) and a turn–around time contract between the stakeholders, who face different costs and penalites. Penalties imposed on the maintenance workshop are illustrated in Fig. 1.

Cost of preventive maintenance. A maintenance occasion at a time step t for a system k generates a set-up cost $d_t > 0$, and every maintenance interval (s, t) for a component type i in any system yields an interval cost $c_{st}^i > 0$. The PM cost for the systems $k \in \mathcal{K}$ over the time steps $t \in \mathcal{T}$ imposed on the system operator is thus defined as $C^{\mathrm{PM}}(x, z) := \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} d_t z_t^k + \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} \sum_{t=1}^{T+1} \sum_{s=0}^{t-1} c_{st}^i x_{st}^{ik}$.

Minimize risk of lacking spare parts. To maintain a seamless operational schedule or at least minimize disruptions, it is vital to have a sufficient stock of spare components. This way, in case of an unforeseen breakdown, the faulty component can be replaced without disturbing the scheduled operations. Further, for an effective PM planning, it is crucial to consistently have a sufficient supply of spare parts in inventory. Hence, to avoid a lack of spare parts, a penalty

[2] For a detailed explanation of the model (1), see (Obradović et al., 2023).

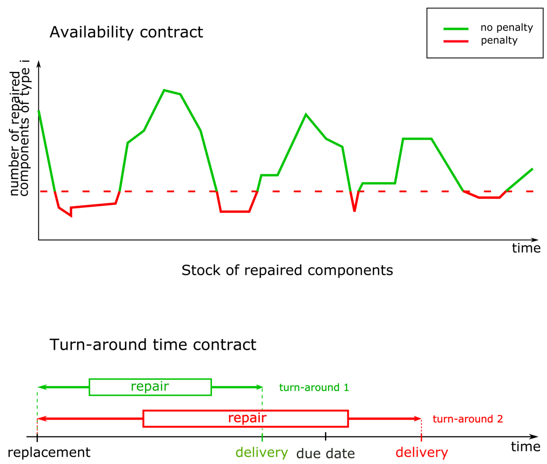


Fig. 1: Penalties imposed on the maintenance workshop under an availability and a turn–around time contract; the red (green) color represents the situation with (without) penalty.

$c_i^{\mathrm{AV}} > 0$ is imposed on the maintenance workshop per unit y_t^i that the inventory b_t^i falls below a certain limit $\underline{b}^i \geq 0$, defining an availability penalty function as $C^{\mathrm{AV}}(y) := \sum_{i \in \mathcal{I}} c_i^{\mathrm{AV}} \sum_{t \in \mathcal{T}} y_t^i$, (Obradović et al., 2023). The availability contract is then modeled as the bi-objective optimization problem to

$$\begin{aligned}
&\underset{x,z,u,\alpha,a,b,\beta,\ell}{\text{minimize}} && \big[C^{\mathrm{PM}}(x,z),\ C^{\mathrm{AV}}(y)\big], && (2a)\\
&\text{subject to} && \text{(1a)–(1n), (1p) hold}, && (2b)\\
&&& y_t^i \geq \underline{b}^i - b_t^i, \quad i \in \mathcal{I},\ t \in \mathcal{T}, && (2c)\\
&&& y_t^i,\ b_t^i \geq 0, \quad i \in \mathcal{I},\ t \in \mathcal{T}. && (2d)
\end{aligned}$$

Minimize risk of exceeding the contracted turn–around times for component repair. The 'turn–around time', v_{tat}^{in}, of a component type i is defined as the time interval from when it is taken out of one of the systems $k \in \mathcal{K}$ and assigned a job id n (i.e., a time t such that $\alpha_t^{ink} = 1$) until it is repaired and available to use again in one of the systems (i.e., a time t such that $u_{t-p^i-\delta_b^i}^{inl} = 1$). If a repaired component is delivered after its contracted due date, a penalty $c_{\mathrm{delay}}^i > 0$ per time unit v_{delay}^{in} that the corresponding job $n \in \mathcal{N}_i$ is delayed, is imposed on the maintenance workshop. A delay penalty function is then defined as $C^{\mathrm{DL}}(v_{\mathrm{delay}}) := \sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}_i} c_{\mathrm{delay}}^i v_{\mathrm{delay}}^{in}$, and the turn–around time[3] contract is modeled as the bi-objective optimization problem to

$$\begin{aligned}
&\text{minimize} && \big[C^{\mathrm{PM}}(x,z), C^{\mathrm{DL}}(v_{\mathrm{delay}})\big], && (3a)\\
&\text{subject to} && \text{(1a)–(1p) hold}, && (3b)\\
&\sum_{t=0}^{T^{\mathrm{ext}}} (t + p^i + \delta_b^i) \sum_{l \in \mathcal{L}} u_t^{inl} - \sum_{t=-\delta_a^i}^{T+1} t \sum_{k \in \mathcal{K}} \alpha_t^{ink} = v_{\mathrm{tat}}^{in}, && n \in \mathcal{N}_i,\ i \in \mathcal{I}, && (3c)\\
&v_{\mathrm{tat}}^{in} - q_{\mathrm{due}}^i \sum_{t=-\delta_a^i}^{T+1} \sum_{k \in \mathcal{K}} \alpha_t^{ink} \leq v_{\mathrm{delay}}^{in}, && n \in \mathcal{N}_i,\ i \in \mathcal{I}, && (3d)\\
&v_{\mathrm{delay}}^{in} \geq 0, && n \in \mathcal{N}_i,\ i \in \mathcal{I}. && (3e)
\end{aligned}$$

A compact version of the model (3), in which the turn–around times, defined in (3c), are implicit, is given by

$$\begin{aligned}
&\text{minimize} && \big[C^{\mathrm{PM}}(x,z), C^{\mathrm{DL}}(v_{\mathrm{delay}})\big], && (4a)\\
&\text{subject to} && \text{(1a)–(1p) hold}, && (4b)\\
&\sum_{t=0}^{T^{\mathrm{ext}}} \big(t + p^i + \delta_b^i\big) \sum_{l \in \mathcal{L}} u_t^{inl} - \sum_{t=-\delta_a^i}^{T+1} \big(t + q_{\mathrm{due}}^i\big) \sum_{k \in \mathcal{K}} \alpha_t^{ink} \leq v_{\mathrm{delay}}^{in}, && n \in \mathcal{N}_i, i \in \mathcal{I}, && (4c)\\
&v_{\mathrm{delay}}^{in} \geq 0, && n \in \mathcal{N}_i, i \in \mathcal{I}. && (4d)
\end{aligned}$$

[3] In order to compute turn–around times for all components that are sent for repair during the planning period, we extend $T + 1$ to T^{ext} as well as take into account the components that are initialized (i.e., for $t < 0$), whence the different sets for summation over t in (3c).

3.1 Complexity analysis

We next prove that both of the single-objective minimization problems to minimize $C^{\mathrm{PM}}(x,z)$ subject to (1a)–(1p), and to minimize $C^{\mathrm{DL}}(v_{\mathrm{delay}})$ subject to (1a)–(1p) and (4c)–(4d), can be reduced to the PMSPIC (Gustavsson et al., 2014), which is an NP-hard problem. Thereby, the bi-objective problem (4), after (any) scalarization (Ehrgott, 2005, Sec. 8.3), is an NP-hard problem.

That the availability bi-objective optimization problem (2) is NP-hard is shown in (Obradović et al., 2023).

Theorem 1 (Complexity) *The complete model of the system–of–systems* (1a)–(1p), (4c), (4d), *minimizing either of the costs* $C^{\mathrm{PM}}(x,z)$ *or* $C^{\mathrm{DL}}(v_{\mathrm{delay}})$, *binary requirements on the variables* x^{ik}_{st}, z^k_t, u^{inl}_t, *and* α^{ink}_t, *and non-negativity and integer requirements on the variables* v^{in}_{delay}, a^i_t, b^i_t, β^i_t, *and* ℓ_t, *for all relevant values of the indices, is NP-hard.*

Proof Define $L := |\mathcal{L}|$, $N_i := |\mathcal{N}_i|$, $i \in \mathcal{I}$, $N := \sum_{i\in\mathcal{I}} N_i$, and $K := |\mathcal{K}|$, and let $\underline{t}_i \geq 1$ denote the minimum number of time steps that a component of type i is used in a system k before it has to be maintained. Then, the lowest number of time steps needed for a component of type i to make one lap in the circulating flow equals $\underline{t}_i + \delta^i_a + p^i + \delta^i_b$. We define $M_i := \left\lceil \frac{T}{\underline{t}_i + \delta^i_a + p^i + \delta^i_b} \right\rceil$ and $R_i := \left\lceil \frac{T}{\underline{t}_i} \right\rceil$. Then, M_i denotes the maximum number of repairs each individual component of type i can undergo during the planning period $\mathcal{T}$, while R_i denotes the maximum number of replacements of component type i in one of the systems $k \in \mathcal{K}$ during the planning period $\mathcal{T}$.

Consider an instance of the model (1a)–(1p) with relevant binary, integer, and non-negativity requirements on the variables. For each $i \in \mathcal{I}$, assume that $\bar{a}^i_0 = 0$, $\bar{b}^i_0 = KR_i + \underline{b}^i$, and $\bar{u}^{inl}_t = 0$, $t \in \{1-\delta^i_b - p^i, \ldots, 0\}$ (at any time step $t \leq 0$, there are no components in either of the stock of damaged components and the workshop, while there are $KR_i + \underline{b}^i$ components in the stock of repaired components and one component in each of the systems $k \in \mathcal{K}$); together with the constraints (1p) this yields that $J_i = K(1+R_i) + \underline{b}^i$ (the total number of components of type i in the system–of–systems ensures that a feasible solution exists). Further, assume that for each $i \in \mathcal{I}$, $N_i = J_i M_i$ (the number of job indices equals the maximum total number of repairs of components of type i) and $L = N$ (the number of repair lines equals the total number of job indices).

Since $L = N$, there is one repair line for each possible repair job, such that any damaged component can be instantly repaired at any time, i.e., the constraints (1f)–(1g) are trivially fulfilled. Since $N_i = J_i M_i$, also the constraints (1j) are trivially fulfilled. Hence, w.l.o.g. the turn–around time for any damaged component of type i at any time $t \in \mathcal{T}$ equals the shortest possible turn–around time, i.e., $\delta^i_a + p^i + \delta^i_b$, which together with the constraints (1k)–(1l) yields that the value of each left-hand-side in (4c) is non-positive. Note that the constraints (1h), defining the variables ℓ_t, do not imply any restrictions. Further, since the number of job indices equals the maximum possible number of replacements, the constraints (1i) are trivially fulfilled, and since

$\bar{b}_0^i = KR_i + \underline{b}^i$ (such that whenever a replacement of a component of type i is required in any of the systems, there is a repaired component of type i available in stock), the constraints (1m)–(1o) are also trivially fulfilled.

For this instance, the constraints (1a)–(1p) are thus reduced to the constraints (1a)–(1e), which separate over the indices $k \in \mathcal{K}$, i.e., over the operating systems. The minimization of $C^{\mathrm{PM}}(x, z)$ subject to (1a)–(1p) is thus equivalent to K instances of the PMSPIC, which is an NP-hard problem (see Gustavsson et al. (2014) and Arkin et al. (1989)). The minimization of $C^{\mathrm{DL}}(v_{\mathrm{delay}})$ subject to (1a)–(1p), (4c)–(4d) separates into a feasibility problem defined by the constraints (1a)–(1e) (which in turn separates into K instances of the PMSPIC, each with a zero objective) and L one-variable linear minimization problems given by, for each $n \in \mathcal{N}_i$ and $i \in \mathcal{I}$, $\min\{c_{\mathrm{delay}}^i v_{\mathrm{delay}}^{in} : v_{\mathrm{delay}}^{in} \geq 0\} \equiv 0$, since the left-hand-side of (4c) is non-positive. We conclude that minimizing $C^{\mathrm{DL}}(v_{\mathrm{delay}})$ subject to (1a)–(1p), (4c)–(4d) is an NP-hard problem.

The theorem follows. □

4 Lagrangean relaxation and subgradient algorithm

Due to the constraints (4c)—corresponding to the turn–around time contract type—the bi-objective optimization problem (4) is computationally expensive. Hence, to enable the computation of an approximate Pareto front, we develop an algorithm based on Lagrangean duality. In order to easen the presentation, we introduce simplified notations in Table 1. The ε-constraint scalarized problem (Mavrotas, 2009) reformulation of (4) is, in these notations, given by

$$
\begin{aligned}
z^{\varepsilon} := \quad & \text{minimum} \quad && c^{\top} x, && \text{(5a)}\\
& \text{subject to} && (x, u) \in \mathcal{X}, && \text{(5b)}\\
& && Du \leq v, && \text{(5c)}\\
& && v \geq 0, && \text{(5d)}\\
& && e^{\top} v \leq \varepsilon, && \text{(5e)}
\end{aligned}
$$

where each of the expressions in (5b)–(5d) represents the respective expression in (4b)–(4d), while the objective (5a) corresponds to the minimization of $C^{\mathrm{PM}}(x, z)$ in (4a) and the left–hand–side of the constraint (5e) corresponds to the objective function $C^{\mathrm{DL}}(v_{\mathrm{delay}})$ in (4a). Since all of the constraints in (4c)–(4d) separate over the indices $n \in \mathcal{N}_i$ and $i \in \mathcal{I}$, the constraints (5c)–(5d) are also separable (i.e., the matrix D is assumed to be block-diagonal).

From the definition of (5) it follows that the inequality $z^{\varepsilon} \leq z^{\tilde{\varepsilon}}$ holds whenever $\varepsilon \geq \tilde{\varepsilon} \geq 0$. In the analysis to follow, we will utilize the single-objective optimization problem

$$
\begin{aligned}
z^{*} := \quad & \text{minimum} \quad && c^{\top} x, && \text{(6a)}\\
& \text{subject to} && (x, u) \in \mathcal{X}, && \text{(6b)}
\end{aligned}
$$

annotation vector/scalar/set	corresponding variable/parameter/set	indices
v	v^{in}_{delay}	$n \in \mathcal{N}_i,\ i \in \mathcal{I}$
x	(x, z)	
u	(u, α, β, a, b)	
N	$\sum_{i\in\mathcal{I}} N_i$	
$e > 0$	c^i_{delay}	$i \in \mathcal{I}$
$c \geq 0$	(d_t, c^i_{st})	$i \in \mathcal{I},\ 0 \leq s < t \leq T+1$
$\mathcal{X}$	set defined by the constraints (1a)–(1p)	
D	the constraint matrix corresponding to (4c)	

Table 1: Simplified notation used for describing the Lagrangean dual method.

with optimal set $\mathcal{X}^*$, and an optimal solution denoted as $(x^*, u^*) \in \mathcal{X}^*$. It holds that $z^* \leq z^\varepsilon$ for all $\varepsilon \geq 0$. Note that $z^* = z^\varepsilon$ holds for a large enough value of $\varepsilon > 0$, such that the inequalities $\varepsilon \geq e^\top Du > 0$ hold for all $u : (x, u) \in \mathcal{X}$.

In order to solve the model (5) more efficiently, we form a Lagrangean dual problem, where the constraints (5c) are Lagrangean relaxed, denoting the multipliers by $\mu_{in} \in \mathbb{R}_+$, $n \in \mathcal{N}_i$, $i \in \mathcal{I}$. For each value of $\varepsilon \geq 0$ in the ε-constraint method, the Lagrangean dual problem is then defined as

$$h^\varepsilon_* := \max_{\mu_{in}\geq 0,\, n\in\mathcal{N}_i,\, i\in\mathcal{I}} \{h^\varepsilon(\mu)\}, \tag{7}$$

where the Lagrangean dual function $h^\varepsilon : \mathbb{R}^N \mapsto \mathbb{R}$ is defined by

$$\begin{aligned} h^\varepsilon(\mu) &:= \operatorname*{minimum}_{x,u,v} \left\{ c^\top x + \mu^\top(Du - v) \,\middle|\, (x, u) \in \mathcal{X},\ v \geq 0,\ e^\top v \leq \varepsilon \right\}, \qquad (8)\\ &= \min_{x,u} \left\{ c^\top x + \mu^\top(Du) \,\middle|\, (x, u) \in \mathcal{X} \right\} - \max_{v} \left\{ \mu^\top v \,\middle|\, v \geq 0;\ e^\top v \leq \varepsilon \right\}. \end{aligned}$$

For any fixed value of the parameter $\varepsilon \geq 0$ and the multipliers $\mu \in \mathbb{R}^N$, we denote optimal solutions to the Lagrangean subproblems by

$$[x(\mu), u(\mu)] \in \operatorname*{argmin}_{x,u} \left\{ c^\top x + \mu^\top Du \,\middle|\, (x, u) \in \mathcal{X} \right\}; \tag{9a}$$

$$v^\varepsilon(\mu) \in \operatorname*{argmax}_{v} \left\{ \mu^\top v \,\middle|\, v \geq 0,\ e^\top v \leq \varepsilon \right\}, \tag{9b}$$

and for any $\mu \geq 0$, we define the index set

$$\mathcal{M}(\mu) := \operatorname*{argmax}_{(in): n\in\mathcal{N}_i, i\in\mathcal{I}} \left\{ \frac{\mu_{in}}{e_i} \right\}.$$

Then, for any $\varepsilon \geq 0$, it holds that $v^\varepsilon_{in}(\mu) = \frac{1}{|\mathcal{M}(\mu)|}\frac{\varepsilon}{e_i}$ for $(in) \in \mathcal{M}(\mu)$, and $v^\varepsilon_{in}(\mu) = 0$ for $(in) \notin \mathcal{M}(\mu)$. Then, the value of the maximization subproblem[4] in (9b) equals $\frac{\mu_{in}}{e_i}$, for some $(in) \in \mathcal{M}(\mu)$. In summary, the optimal value of the subproblem for any $\varepsilon \geq 0$ and any $\mu \in \mathbb{R}^N_+$ can be expressed as

$$h^\varepsilon(\mu) := h^0(\mu) - \varepsilon \cdot \max_{(in): n\in\mathcal{N}_i, i\in\mathcal{I}} \left\{ \frac{\mu_{in}}{e_i} \right\} \leq z^\varepsilon,$$

[4] The problem (9b) is a continuous knapsack problem

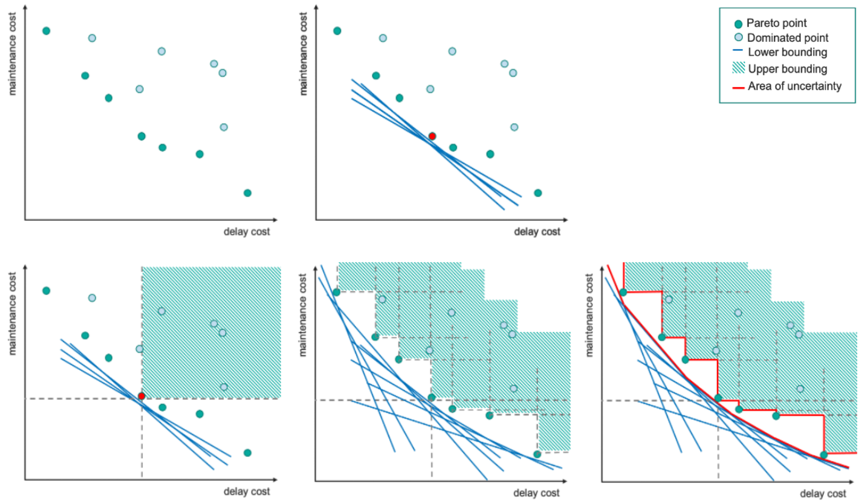


Fig. 2: Illustration of the bounding in the objective space. In the last figure, the polygon limited by a thick solid (red) line represents the area in which non-dominated points can be found

where $h^0(\mu) := c^\top x(\mu) + \mu^\top Du(\mu)$ and z^ε is the optimal objective value for the problem (5). Note that $h^\varepsilon(0) = h^0(0) - 0 \cdot \varepsilon = z^*$ for all $\varepsilon \geq 0$.

Given a subproblem solution $[x(\mu), u(\mu)] \in \mathcal{X}$, it holds (see (6)) that

$$c^\top x(\mu) \geq c^\top x^* = z^*.$$

Specifically, this upper bound inequality holds for the rightmost (the minimum of $c^\top x$ over $(x, u) \in \mathcal{X}$) point on the Pareto front (set of non-dominated points in the objective space). In order to find upper bounds for any non-dominated point, we suggest the following heuristic procedure: Set $\overline{v}(\mu) := [Du(\mu)]_+$ (where the projection $[\cdot]_+$ onto the non-negative orthant is defined component-wise), and let $\overline{\varepsilon}(\mu) := e^\top \overline{v}(\mu)$. It holds that the point $[x(\mu), u(\mu), \overline{v}(\mu)]$ is feasible in (5) for all $\varepsilon \geq \overline{\varepsilon}(\mu)$. Therefore, it holds that

$$c^\top x(\mu) \geq z^{\overline{\varepsilon}(\mu)}.$$

The bounding functions in the objective space that can be computed using the entities developed above are illustrated in Figure 2.

Since we *cannot* conclude any relation between $\overline{\varepsilon}(\mu) = e^\top \overline{v}(\mu)$ and $e^\top v^\varepsilon(\mu)$, the value of ε that is solved for is altered to a heuristic value, which may be higher or lower. A pair of upper bounds (on some non-dominated point in the objective space) is thus given by $\big[\overline{\varepsilon}(\mu)\,,\, c^\top x(\mu)\big]$, where $\overline{\varepsilon}(\mu) = e^\top \overline{v}(\mu) = e^\top [Du(\mu)]_+$. This means that any point $[\varepsilon, z]$ such that the inequalities $\varepsilon \geq e^\top [Du(\mu)]_+$, $z \geq c^\top x(\mu)$, and $\varepsilon + z > e^\top [Du(\mu)]_+ + c^\top x(\mu)$ hold, is dominated by the point $\big[\overline{\varepsilon}(\mu)\,,\, c^\top x(\mu)\big]$. See an illustration in Figure 3.

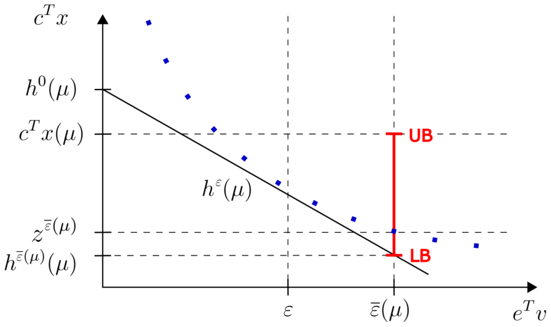


Fig. 3: Upper and lower bounds (red) from a given pair $[\varepsilon, \mu]$, a lower bounding function (black), and a set of points on the Pareto front (blue).

In summary, from every dual point $\mu \in \mathbb{R}^N_+$, a pair of upper bounds is obtained as $\left[e^\top [Du(\mu)]_+ \, , \, c^\top x(\mu)\right]$.

To optimize the Lagrangean dual problem (7) we use a subgradient optimization procedure—summarized in Algorithm 1—in which different strategies can be used to update the step length parameter $\theta^{(m)}$ and the estimated upper bounds $[\bar{\epsilon}(\mu^{(m)}), c^\top x(\mu^{(m)})]$ in each subgradient iteration m. We employ an adaptive step length update, that in practice has been shown to yield fast convergence to an optimal solution (Caprara et al., 1999). Starting from iteration 20, the step length parameter $\theta^{(m)}$ is updated in each subgradient iteration. In summary, the best and worst lower bounds $h^\varepsilon(\mu^{(m)})$ found during the latest 20 iterations are compared. If the absolute value of their difference is more (less) than 10% (1%) of the absolute value of the worst lower bound—implying that too large (small) steps are taken by the algorithm—then $\theta^{(m)}$ is multiplied by $\frac{1}{2}$ $(\frac{3}{2})$; if neither case applies, then $\theta^{(m)}$ is kept unchanged. The upper bound $(\widehat{h}^\varepsilon)^{(m)}$ used in the computation of the step length is set to $h^0(\mu^{(m)})$ and a decreasing term $\frac{\text{const}}{m}$ is added to the step length to stimulate progress during early iterations.

Algorithm 1 Subgradient Optimization Procedure

1: Choose a value for $\varepsilon \geq 0$, let $m := 0$, and initialize $\mu^{(0)} \geq 0$ and $\theta^{(0)} \in (0, 2)$
2: **repeat**
3: Solve the Lagrangean subproblems in (9) for $\mu = \mu^{(m)}$ and ε
4: Compute lower and upper bounds, $h^\varepsilon(\mu^{(m)})$ and $(\widehat{h}^\varepsilon)^{(m)}$, on the optimal value h^ε_*
5: Calculate a subgradient direction $\gamma^{(m)} := Du(\mu^{(m)}) - v^\varepsilon(\mu^{(m)})$
6: Compute the Polyak step length: $\phi^{(m)} := \theta^{(m)} \frac{\left((\widehat{h}^\varepsilon)^{(m)} - h^\varepsilon(\mu^{(m)}) + \text{const}/m\right)}{\|\gamma^{(m)}\|^2}$
7: Compute the next dual iterate: $\mu^{(m+1)}_{in} := \left[\mu^{(m)}_{in} + \phi^{(m)} \gamma^{(m)}_{in}\right]_+$
8: Let $m := m + 1$ and calculate a step length parameter $\theta^{(m)} \in (0, 2)$
9: **until** a termination criterion is fulfilled

5 Aggregation over the jobs

Since the bi-objective optimization problem (4) showed to be computationally expensive, we next modify the algorithm for solving it. After investigating (see App. A.1–A.4) different measures to efficiently utilize Algorithm 1, we found the computationally most successful as to aggregate the constraints (4c)–(4d), as well as the variables v_{delay}^{in}, α_t^{ink}, and u_t^{inl} over the indices $n \in \mathcal{N}_i$, such that $v_{\text{delay}}^{i} = \sum_{n\in\mathcal{N}_i} v_{\text{delay}}^{in}$, $\alpha_t^{ik} = \sum_{n\in\mathcal{N}_i} \alpha_t^{ink}$, $u_t^{il} = \sum_{n\in\mathcal{N}_i} u_t^{inl}$, and

$$\sum_{t=0}^{T^{\text{ext}}} \left(t + p^i + \delta_b^i\right) \sum_{l\in\mathcal{L}} u_t^{il} - \sum_{t=-\delta_a^i}^{T+1} \left(t + q_{\text{due}}^i\right) \sum_{k\in\mathcal{K}} \alpha_t^{ik} \le v_{\text{delay}}^i \ge 0, \quad i \in \mathcal{I}, \quad (10)$$

Besides that, as the requirement of non-preemption adds to the complexity—in general as well as in our modeling—we employ a preemptive flow of jobs through the maintenance workshop, replacing the constraints (1f) and (1h) by

$$0 \le \ell_t = \ell_{t-1} + \sum_{i\in\mathcal{I}} \sum_{l\in\mathcal{L}} \left(u_t^{il} - u_{t-p^i}^{il}\right) \le L, \qquad t \in \mathcal{T}.$$

The aggregation of constraints over jobs is employed also for (1g) and (1j), while in (1i), (1k), (1n), and (1p), a summation over the jobs is already present such that only the variable substitution needs to be employed.

The Lagrangean dual variables are altered to $\mu_i \in \mathbb{R}_+$, $i \in \mathcal{I}$. For each value of $\varepsilon \ge 0$ in the ε-constraint method, the Lagrangean dual problem is then defined as

$$h_*^\varepsilon := \max_{\mu_i \ge 0,\, i\in\mathcal{I}} \{h^\varepsilon(\mu)\},$$

where the Lagrangean dual function $h^\varepsilon : \mathbb{R}^I \mapsto \mathbb{R}$ is given by (cf. (8))

$$h^\varepsilon(\mu) = \min_{x,u} \left\{ c^\top x + \mu^\top (Du) \,\middle|\, (x,u) \in \mathcal{X} \right\} - \max_v \left\{ \mu^\top v \,\middle|\, v \ge 0;\ e^\top v \le \varepsilon \right\},$$

the vector v now representing the aggregated variables v_{delay}^i and D represents the constraint matrix in the aggregated version (10) of (4c).

Two heuristics are used to find upper bounds for the aggregated model. (I) As in Section 4, $\overline{v}(\mu)$ and $\overline{\varepsilon}(\mu)$ are computed such that $[x(\mu), u(\mu), \overline{v}(\mu)]$ is feasible in the aggregated version of (5) for all $\varepsilon \ge \overline{\varepsilon}(\mu)$. This results in an upper bound for the aggregated model with value $c^\top x \ge z^{\overline{\varepsilon}(\mu)}$. (II) An upper bound for the non-aggregated model is found by splitting the aggregated variables back into jobs, using a 'first in–first out' approach. Since up to time $t-1$, $\sum_{s=-\delta_a^i}^{t-1} \alpha_s^{ik}$ jobs have started, while α_t^{ik} jobs start at time t, it is possible to split the aggregated variables sequentially into jobs (i.e., reversing the aggregation) by (re)defining the binary variable values as

$$\alpha_t^{ink} = \begin{cases} 1, & \text{if } \sum_{s=-\delta_a^i}^{t-1} \alpha_s^{ik} < n \le \sum_{s=-\delta_a^i}^{t} \alpha_s^{ik}, \\ 0, & \text{otherwise}, \end{cases} \quad (11)$$

for all relevant indices i, n, k and t. Values for u_t^{inl} are analogously recovered from u_t^{il}. The splitting results in an ordering such that components arriving at the workshop first will start repairs first. Following the procedure developed in Section 4, using the non-aggregated variables, values for $\tilde{v}(\mu)$ and $\tilde{\varepsilon}(\mu)$ are computed from the non-aggregated turn–around times, which result in $[x(\mu), u(\mu), \tilde{v}(\mu)]$ being feasible in the non-aggregated version of (5) for all $\varepsilon \geq \tilde{\varepsilon}(\mu)$. Finally, an upper bound for the non-aggregated model with value $c^\top x \geq z^{\tilde{\varepsilon}(\mu)}$ is found.

The advantage of this approach is that is makes the subproblems significantly easier to solve and that both the primal and the dual spaces are reduced. Moreover, it is a proof of concept for the method suggested in Section 4. A disadvantage, however, is that the turn–around times, hence the delays, are aggregated over jobs n for each component type i; thus we do not compute the exact delay per individual job n. Further, replacing the constraints (4c) with the aggregated ones (10) implies a relaxation of the original model, leading to (for any $\varepsilon \geq 0$) optimistic estimates of the optimal values sought. Consequently, the aggregation alone, without the suggested heuristic, would be insufficient to approximate the Pareto front for the non-aggregated problem.

6 Application: Implementation, tests, and results

We present an application from the aerospace industry, within a collaboration with the Swedish aerospace and defence company Saab AB. For contract assessment purposes, the instance sizes are considered to be reasonable from a practical application point of view and the data sets used are based on knowledge mediated from the industrial partner; all data are normalized.

Our implementation is made using Julia (2012) and JuMP (Dunning et al., 2017), and the computations are performed by Gurobi (2020) on a cluster. Our code was run on Intel Xeon Gold 6130 processors and each solution was computed on 4 cores and 16 GB of RAM. The maximum number of threads used by Gurobi per run is limited to 4. We found that increasing the number of cores did not improve performance.

6.1 The main test instances and multi-objective settings

As main test cases, we consider $K = 10$ systems, each having $I = 5$ component types and the number of individual components of type $i \in \mathcal{I}$ being either $J_i = 35$ or $J_i = 25$. The operational and maintenance related differences of the component types are reflected by their respective repair times in the maintenance workshop and their respective due dates, which are chosen randomly within the same order of magnitude. The different component types are also assigned differently structured interval costs, all increasing with the time between maintenance occasions, reflecting the increasing risk of having to perform CM. The planning horizon is $T = 40$ time steps and the workshop capacity is chosen as either $L = 25$ or $L = 40$ parallel machines. The maximum number of jobs[5] needed for component type i during a planning period $\mathcal{T}$ is $N_i = J_i M_i$, where $M_i = \left\lceil \frac{T}{\underline{t}_i + \delta_a^i + p^i + \delta_b^i} \right\rceil$ denotes the maximum number of repairs each individual component of type i can undergo during this period. For computational efficiency reasons, however, the number of jobs is limited to $N_i = 80$ for each component type $i \in \mathcal{I}$. The processing times take values $p^i \in \{3, 4, 5\}$ and the transport times between the stocks and the maintenance workshop are $\delta_a^i = 2$ and $\delta_b^i = 1$, $i \in \mathcal{I}$. The largest allowed maintenance interval length (i.e., life span $\bar{t}_i$ of a component type i) is sampled from $\{10, \ldots, 15\}$, for $i \in \mathcal{I}$. The maintenance costs are $d_t = 5$, $t \in \mathcal{T}$, while c_{st}^i is varied: the smallest value (for the maintenance interval length $t - s = 1$) is 5, while the cost for the longest allowed maintenance interval varies in the range $[10, 100]$ over component types $i \in \mathcal{I}$. Availability penalty costs and penalties for late deliveries take values $c_i^{\mathrm{AV}} \in \{5, \ldots, 10\}$ and $c_i^{\mathrm{DL}} \in \{5, 6, 7\}$, respectively.

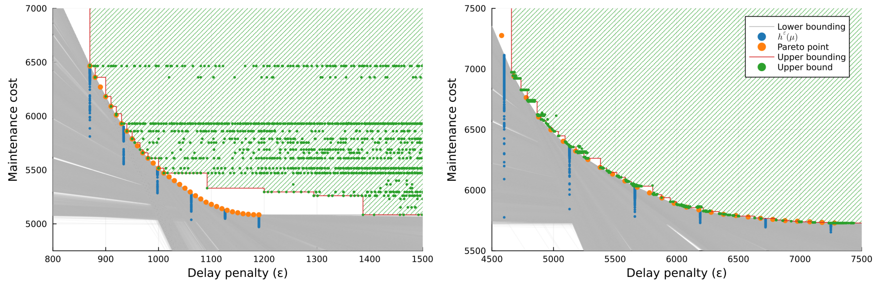
6.2 Computational tests and results

Figure 4 shows the performance of Alg. 1 applied to (4) with aggregated variables and constraints over jobs $n \in \mathcal{N}_i$ according to (10), for two instances. It is fairly easy to compute (some of) the points on the Pareto front for the aggregated problem, and they are utilized to validate that the Pareto front lies in the uncertainty area provided by the algorithm. Moreover, the algorithm approximates the area quite well. The instance illustrated in Fig. 4(a) has larger numbers of components and repair lines as compared to the one in Fig. 4(b), hence a much larger degree of freedom in terms of feasible solutions. In (a) the resulting maintenance costs are in the interval $[5000, 6500]$, while in (b) they are overall higher, in the interval $[5700, 7300]$. A similar observation is made regarding the resulting delay penalties; in (a) they are in the interval $[850, 1200]$, while in (b) the interval is increased to $[4500, 7300]$. For both cases, reducing the resulting delay penalties leads to increased maintenance costs, which is larger when the resources are more scarse (as in (b)). The average subproblem solution times[6] for (a) and (b) are 1.16 and 0.82 seconds, respectively; for each value of ε we run the subgradient algorithm for 10 000 iterations.[7] The lower bounding functions span the area below the Pareto front quite well while the upper bounds leave some gap when $\varepsilon \geq 1100$ in (a) and $\varepsilon \leq 4600$ in (b); this would possibly be resolved by allowing for more subgradient iterations. Since the aggregation of the problem (4) is a relaxation, the resulting lower bounding functions obtained for the aggregated model are valid also for (4). To approximate the area of uncertainty for the Pareto front of (4), we utilize the heuristic for computing upper bounds on non-dominated

[5] The number of variables is (approximately) proportional to the maximum total number of jobs $N = \sum_{i \in \mathcal{I}} N_i$

[6] For a comparison with subproblem solution times for the non-aggregated model, see A.4

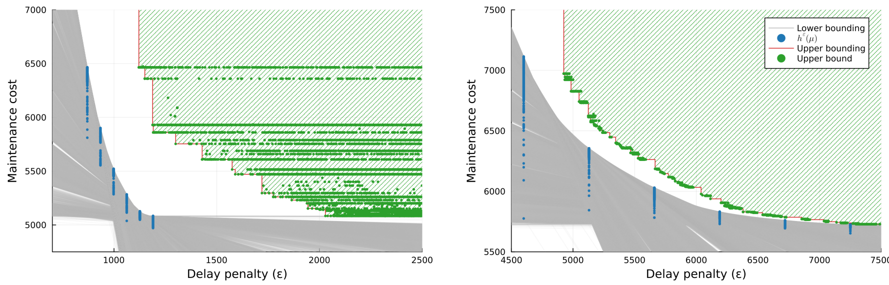
[7] In Figure 4 (b), the number of subgradient iterations for $\varepsilon = 4600$ was expanded to 60000 iterations as it took longer to come closer to that part of the Pareto front



(a) $(I, J_i, K, L, \underline{b}^i) = (5, 35, 10, 40, 0)$ (b) $(I, J_i, K, L, \underline{b}^i) = (5, 25, 10, 25, 0)$

Fig. 4: Results from Alg. 1 applied to the aggregation of constraints (4c)–(4d) over jobs $n \in \mathcal{N}_i$ for two different instances; proof of concept.

points described in Sec. 5. The resulting approximation of the area in which the Pareto front of (4) lies is illustrated in Fig. 5, for the same instances as in Fig. 4. It is noticeable that there is a better performance (i.e., smaller gaps) in (b). As in Fig. 4, instance (a) comes with more freedom in the variable and objective space, resulting in more freedom for the (heuristic) upper bounds. This can be observed in both Fig's. 4 and 5.

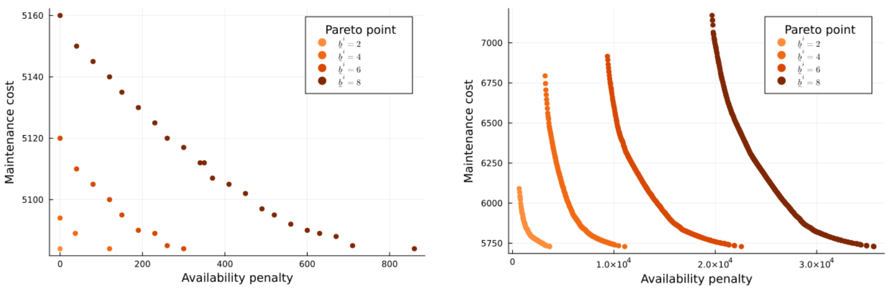


(a) $(I, J_i, K, L, \underline{b}^i) = (5, 35, 10, 40, 0)$. (b) $(I, J_i, K, L, \underline{b}^i) = (5, 25, 10, 25, 0)$.

Fig. 5: The approximated Pareto front and the area of uncertainty for the bi-objective optimization problem (4); instances the same as in Fig. 4.

The availability penalty was analyzed for different values of the lower limit $\underline{b}^i, i \in \mathcal{I}$, as presented in Fig. 6. In (b), there is in total 125 components in the whole system–of–systems, and out of which—at all times—50 are in one of the systems, implying that 75 components are either in the repair workshop or in one of the stocks. In (a), the number of components is larger; specifically there are ten more components of each type. Moreover, the capacity in the maintenance workshop is increased from $L = 25$ repair lines in (b) to $L = 40$

in (a). If, for example, $\underline{b}^i = 6$, $i \in \mathcal{I}$, then there is a penalty whenever the total number of available components, over all types $i \in \mathcal{I}$, on the stock of repaired components goes below 30. Hence, for each value $\underline{b}^i$, the availability penalties in (b) are higher. Moreover, the maintenance cost are higher in (b), which is expected as the resources are scarser, whence the maintenance intervals become longer due to a lack of repaired components necessary for replacement.



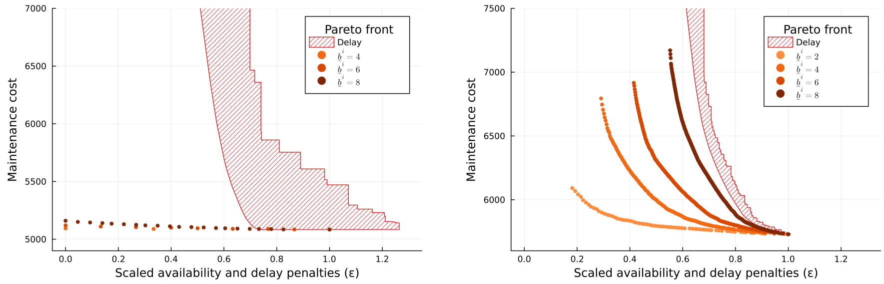
(a) $(I, J_i, K, L) = (5, 35, 10, 40)$. (b) $(I, J_i, K, L) = (5, 25, 10, 25)$.

Fig. 6: The computed points on the Pareto front for the availability contract with four different lower levels $\underline{b}^i, i \in \mathcal{I}$ on the stock of repaired components.

Studying the two contracts and their respective Pareto fronts (approximated or exact), it is only fair to compare the maintenance costs, as the penalties for late deliveries and levels below a lower limit on the stock of repaired components are both artificial costs utilized for modeling purposes. They are motivated by the contracting forms, and could be used to negotiate between the stakeholders. The penalties represent measures of contract violation. Figure 7 compares the shapes as well as levels of maintenance cost for the (computed and approximated) Pareto fronts corresponding to the two contracting forms. In (a) the maintenance costs remain almost at the same level for the availability contract. In comparison, the maintenance costs for the approximated turn–around time Pareto front are significantly higher, and the slope of the frontier is much steeper. In (b) the availability Pareto fronts approach the approximated turn–around time front for higher values of $\underline{b}^i$ and the shapes of the fronts are similar. The difference between the maintenance costs for the two contracting forms is highly affected by the value of $\underline{b}^i$.

7 Conclusions and Future Research

We present a method for bounding the Pareto frontier for a bi-objective optimization problem modeling a turn–around time contract for maintenance of



(a) $(I, J_i, K, L) = (5, 35, 10, 40)$. (b) $(I, J_i, K, L) = (5, 25, 10, 25)$.

Fig. 7: Comparison of the shapes of the computed Pareto fronts of the availability contract and the uncertainty area for the Pareto front of the turn–around time contract. Both penalties are normalized to the interval $[0, 1]$; for the delay (availability) penalty, the value 1 corresponds to the midpoint of the uncertainty interval (the right-most point on the Pareto front) for the lowest maintenance cost depicted in Fig. 5.

components, between two stakeholders. The problem is practically impossible to solve within a reasonable time frame using an off–the–shelf software. By employing a Lagrangean relaxation and a subgradient algorithm, together with a suitable problem relaxation (an aggregation over maintenance jobs), we obtain lower bounds on the Pareto front, while computing upper bounds on non-dominated points using a heuristic approach. The main bottleneck in our solution approach is the individual tracking of jobs that is necessary for computing the turn–around times. After aggregation over jobs, a heuristic for splitting the aggregated variables back into jobs is suggested.

Our results indicate that an availability contract performs better than a turn–around time contract in terms of cost and penalties (which do not necessarily represent the actual costs, but rather a measure of contract violation).

The suggested framework may be utilized further for contract comparison and evaluation. The method suggested for approximating the area of a Pareto front for a turn–around time bi-objective optimization problem proves to work quite well when applied to the aggregated model. However, for the non-aggregated model, there is room for improvement of the gap between the bounds. We suggest two measures for reducing the gap. One is to tighten the lower bounds, that is, to find higher lower bounds than the ones provided by the aggregated model. The other is to create a better performing heuristic for computing the upper bounds and finding feasible solutions. Another possible improvement of the suggested solution approach would be a different definition and/or implementation of the symmetry breaking constraints (see App. A.1).

A Appendix

A possible bottleneck for Alg. 1 and the method presented in Sec. 4 to be efficient is the subproblem complexity. Besides that, the suggested model does not ensure the correspondence between the variables α and u, regarding the indexing of jobs. This means that a pair (i, n) defining job n for a component of type i possibly does not represent the same job after the relaxation, or in a new subgradient iteration. To validate the computed turn–around times and delays, this information needs to be retrieved. We present an overview of the ideas and methods explored in order to make Alg. 1 more effective.

A.1 Symmetry breaking constraints

There are many equally good, symmetric, solutions to the problem (1) with respect to jobs. Presence of symmetries in combinatorial problems increase the size of the search space and therefore, time is wasted in visiting new solutions which are symmetric to the already visited solutions. The size of the search space can be reduced by the introduction of symmetry breaking constraints; e.g. (Kiziltan, 2004; Cherri et al., 2018). In order to retain the correspondence between the variables α_t^{ink} and u_t^{inl} throughout the relaxation, we let job id n have priority over job id $n+1$, for $n \in \mathcal{N}_i \setminus \{N_i\}$. Since an inherent property of our problem is that a replacement occurs prior to the corresponding repair, by construction there will be an equal number of jobs and repairs over the extended (for both $t < 0$ and $t > T + 1$) time horizon. Hence, for all $n \in \mathcal{N}_i \setminus \{N_i\}$, $i \in \mathcal{I}$, we include the constraints

$$\sum_{s=1-p^i-\delta_b^i}^{t} \sum_{l \in \mathcal{L}} \left(u_s^{inl} - u_s^{i,n+1,l} \right) \geq 0, \qquad t \in \{1 - p^i - \delta_b^i, \ldots, T_{\text{ext}}\},$$

$$\sum_{s=-\delta_a^i}^{t} \sum_{k \in \mathcal{K}} \left(\alpha_s^{ink} - \alpha_s^{i,n+1,k} \right) \geq 0, \qquad t \in \{-\delta_a^i, \ldots, T + 1\},$$

which, however, lead to a significant increase of computing times.

A.2 Heuristic matching of the replacement and repair variables

Another attempt of retrieval of the α–u correspondence was to, after solving the Lagrangean subproblem, find a matching of the variables α_t^{ink} and u_t^{inl} over $t \in \mathcal{T}^{\text{ext}}$ and $n \in \mathcal{N}_i$. In each subgradient iteration (see Algorithm 1), we (a) solve the subproblem, (b) do the matching, and (c) update the Lagrangean multipliers. The matching is done according to the *first in–first out* approach. The dual variables are updated based on the matched α–u values, whence in the next subgradient iteration, the multipliers will, most likely, be underestimated, possibly leading to non-subgradient step directions.

A.3 Optimizing for a matching of jobs for replacement and repair

In Section 5, a heuristic was introduced to split the aggregated integer variables α_t^{ik} and u_t^{il} into binary variables α_t^{ink} and u_t^{inl}. In particular, the chosen approach ensures that α_t^{ink} and u_t^{inl} are matched such that the first component arriving at the workshop is the first component to be repaired. These variables can then be used to produce a feasible upper bound to the non-aggregated version of (5). Note that $z_{\text{agg}}^{\varepsilon} \leq z^{\varepsilon}$ for the optimal objective value of the aggregated problem $z_{\text{agg}}^{\varepsilon}$ and the optimal objective value of the non-aggregated problem z^{ε}. Therefore, the linear lower bounds $h_{\text{agg}}^{\varepsilon}(\mu)$ for the aggregated problem remain valid lower bounds for the non-aggregated problem, i.e., $h_{\text{agg}}^{\varepsilon}(\mu) \leq z_{\text{agg}}^{\varepsilon} \leq z^{\varepsilon}$. Note that

α_t^{ink} and u_t^{inl} obtained via splitting of the aggregated variables cannot be used to find an improved linear bound at ε since we only have access to Lagrangian dual variables $\mu_i, i \in \mathcal{I}$ and not to $\mu_{in}, i \in \mathcal{I}, n \in \mathcal{N}_i$.

To find improved linear bounds with matched values of α_t^{ink} and u_t^{inl} at ε, job-splitting was incorporated into the otherwise aggregated model as a means of symmetry-breaking among jobs. The model in (1) is not impacted by aggregation over jobs and therefore it is sufficient to use integer variables α_t^{ik} and u_t^{il} as described in Section 5. In the following we introduce cumulative variables $\alpha_t^i := \sum_{s=-\delta_a^i}^t \sum_{k\in\mathcal{K}} \alpha_s^{ik}$, $t \in \{-\delta_a^i, \dots, T+1\}$, and $u_t^i := \sum_{s=0}^t \sum_{l\in\mathcal{L}} u_s^{il}$, $t \in \{0, \dots, T^{\text{ext}}\}$ for all $i \in \mathcal{I}$.

As shown in (11), jobs can be recovered from the aggregated variables by comparing cumulative variables with job indices, which can be achieved in model by using big-M constraints. For $t \in \{-\delta_a^i, \dots, T+1\}$ and $i \in \mathcal{I}$, α_t^i are split into $\alpha_t^{in} := \sum_{k\in\mathcal{K}} \alpha_t^{ink}$ for $n \in \mathcal{N}_i$ by introducing binary helper variables ρ_t^{in} and ψ_t^{in} and constraints[8]

$$1 - N_i\rho_t^{in} \quad \le \quad \alpha_{t-1}^i - (n-1) \quad \le \quad N_i(1-\rho_t^{in}); \tag{13a}$$

$$N_i(\psi_t^{in} - 1) \quad \le \quad \alpha_t^i - n \quad \le \quad N_i\psi_t^{in} - 1. \tag{13b}$$

The helper variables can then be used to recover $\alpha_t^{in} = \rho_t^{in} + \psi_t^{in} - 1$. The variables u_t^i are similarly split into $u_t^{in} := \sum_{l\in\mathcal{L}} u_t^{inl}$ by an analogous construction. The resulting values $\left[\alpha_t^{in}, u_t^{in}\right]_{n\in\mathcal{N}_i}$ are then used in (4) to compute the non-aggregated turn–around times as well as delays.

While appealing in theory, this formulation introduces $2N(T+3+\delta_a^i+T^{\text{ext}})$ auxiliary binary variables and $4N(T+3+\delta_a^i+T^{\text{ext}})$ auxiliary constraints, increasing the problem size substantially. This resulted in significantly increased computation times, making the model unsuited for use within the subgradient algorithm.

A.4 Computing times for different models and model properties

The motivation for why we chose to utilize the aggregation over jobs (see Sec. 5) as compared to the symmetry breaking constraints (see App. A.1) or optimizing for a matching of jobs for replacement and repair (see App. A.3) can be seen in Table 2. It is easily noticeable that the computing speed is much lower for the aggregated model, hence it is more useful and preferred over the two alternatives.

Table 2: Computing times per iteration of the subgradient algorithm for a (relative) duality gap tolerance of 10^{-4}. + : preemptive workshop scheduling employed. − : non-preemptive workshop scheduling employed. * : no feasible solution found.

$(I, J_i, K, L, \underline{b}^i)$	Model	Preemption	Solution time [s]
$(5,35,10,40,0)$	Sec. 5	+	1.16
$(5,25,10,25,0)$	Sec. 5	+	0.82
$(5,35,10,40,0)$	Sec. 5	−	12.00
$(5,25,10,25,0)$	Sec. 5	−	17.09
$(5,35,10,40,0)$	App. A.3	−	> 535.00
$(5,25,10,25,0)$	App. A.3	−	> 231.00
$(5,35,10,40,0)$	App. A.1	−	> 10000*
$(5,25,10,25,0)$	App. A.1	−	> 10000*

References

Arkin E, Joneja D, Roundy R (1989) Computational complexity of uncapacitated multi-echelon production planning problems. Operations Research Letters 8(2):61–66

[8] The constraints (13) ensure that $\rho_t^{in} = 1$ when $\alpha_{t-1}^i \le n-1$, $\rho_t^{in} = 0$ when $\alpha_{t-1}^i \ge n$, $\psi_t^{in} = 1$ when $\alpha_t^i \ge n$, and $\psi_t^{in} = 0$ when $\alpha_t^i \le n-1$.

Caprara A, Fischetti M, Toth P (1999) A heuristic method for the set covering problem. Operations Research 47(5):730–743, DOI 10.1287/opre.47.5.730

Cherri LH, Cherri A, Soler E (2018) Mixed integer quadratically-constrained programming model to solve the irregular strip packing problem with continuous rotations. Journal of Global Optimization 72:1–19, DOI 10.1007/s10898-018-0638-x

Dunning I, Huchette J, Lubin M (2017) JuMP: A Modeling Language for Mathematical Optimization. SIAM Review 59(2):295–320, DOI 10.1137/15M1020575

Ehrgott M (2005) Multicriteria Optimization, 2nd edn. Springer, Berlin, DOI 10.1007/3-540-27659-9

Gurobi (2020) Gurobi Optimizer Reference Manual. http://www.gurobi.com

Gustavsson E, Patriksson M, Strömberg A-B, Wojciechowski A, Önnheim M (2014) The preventive maintenance scheduling problem with interval costs. Computers & Industrial Engineering 76:390–400

Julia (2012) version 1.5. Julia: A fast dynamic language for technical computing, https://docs.julialang.org/en/v1/

Kiziltan Z (2004) Symmetry breaking ordering constraints. PhD thesis, Uppsala University, Sweden,
https://www.diva-portal.org/smash/get/diva2:164039/FULLTEXT01.pdf

Mavrotas G (2009) Effective implementation of the ϵ-constraint method in multi-objective mathematical programming problems. Applied Mathematics and Computation 213:455–465

Obradović G (2021) Mathematical modeling, optimization and scheduling of aircraft's components maintenance and of the maintenance workshop. Licentiate thesis, Chalmers University of Technology, Sweden, https://research.chalmers.se/en/publication/524023

Obradović G, Strömberg A, Lundberg K (2022) Simultaneous scheduling of replacement and repair of common components in operating systems. Annals of Operations Research DOI 10.1007/s10479-022-04739-8

Obradović G, Strömberg AB, Lundberg K (2023) An enhanced mathematical model for optimal simultaneous preventive maintenance scheduling and workshop planning. Optimization Online URL https://optimization-online.org/?p=24459

Papakostas N, Papachatzakis P, Kanthakis V, Mourtzis D, Chryssolouris G (2010) An approach to operational aircraft maintenance planning. Decision Support Systems 48:604–612

Prins C, Prodhon C, Calvo RW (2006) Two-phase method and Lagrangian relaxation to solve the bi-objective set covering problem. Annals of Operations Research 147:23–41, DOI 10.1007/s10479-006-0060-5

Quttineh NH, Ngulo U, Larsson T (2022) Approximating the Pareto frontier for a challenging real-world bi-objective covering problem. INFOR: Information Systems and Operational Research 60(3):342–358, DOI 10.1080/03155986.2022.2040274

Ruzika S, Wiecek MM (2005) Approximation methods in multiobjective programming. Journal of Optimization Theory and Applications 126:473–501, DOI 10.1007/s10957-005-5494-4

Shao L, Ehrgott M (2008) Approximately solving multiobjective linear programmes in objective space and an application in radiotherapy treatment planning. Mathematical Methods of Operations Research 68:257–276, DOI 10.1007/s00186-008-0220-2

Swanson L (2001) Linking maintenance strategies to performance. Journal of Production Economics 70:237–244, DOI 10.1016/S0925-5273(00)00067-0

van den Akker JM, Hurkens CAJ, Savelsberg MWP (2000) Time-indexed formulations for machine scheduling problems: Column generation. INFORMS Journal on Computing 12:111–124

Yu Q, Strömberg A-B (2021) Mathematical optimization models for long-term maintenance scheduling of wind power systems.
https://arxiv.org/abs/2105.06666

Statements and Declarations

The research leading to the results presented in this paper was financially supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA; project number 2017-04879), Chalmers University of Technology, and Saab AB. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at National Supercomputer Centre (NSC), Linköping University, partially funded by the Swedish Research Council through grant agreement no. 2022-06725. The authors have no relevant financial or non-financial interests to disclose.