Reliability Engineering and System Safety 219 (2022) 108197

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Critical project planning and spare parts inventory management in shutdown maintenance

Sha Zhu [a,*], Willem van Jaarsveld [b], Rommert Dekker [c]

[a] Department of Information & Organizations Management, ESCP Paris, 3 Rue Armand Moisant, 75015 Paris, France
[b] Industrial Engineering & Innovation Sciences, Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[c] Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR, Rotterdam, The Netherlands

ARTICLE INFO

Keywords:
Shutdown maintenance
Project planning
Spare parts management
On-condition maintenance

ABSTRACT

In the process industry, plant production is completely halted every so many years for large-scale maintenance comprising inspection, repair and overhaul. It is paramount that this *shutdown maintenance* is completed according to the planning, which requires timely availability of replacement parts. In this setting, we present a spare parts optimization model and algorithm that includes the cost-time trade-off and precedence constraints between maintenance activities. The objective is to balance between spare parts ordering cost and the expected *project delay* cost due to waiting for spare parts. Using two-stage stochastic programming, spare parts ordering policies are determined in the first stage and a detailed project schedule is developed in the second stage. We propose a sample average approximation with importance sampling and pruning of dominated activities to solve the problem, and demonstrate that this method solves large instances quickly. We also consider heuristics, e.g. the standard project management approach based on the widely used critical path method and two heuristics that draw from spare part inventory control literature. These heuristics give poor solutions.

1. Introduction

Maintenance activities that cannot be carried out under normal production conditions and that can only be performed if the equipment is not in use are referred to as *shutdown maintenance*; the terms *outage* and *turnaround* are roughly equivalent. Shutdown maintenance consists of disassembly, comprehensive inspection, repairing and replacement of parts, and overhaul. Though it is the most expensive of all types of maintenance, it is necessary and cost-effective, especially in industrial sectors such as refineries, chemical and petrochemical plants and power generation plants which usually have continuous production cycles [cf. 1]. Shutdowns are crucial for system safety and necessary for plants to ensure their reliability. Plants will suffer consequence or a great loss if the shutdown is poorly managed.

Shutdown maintenance is characterized by intensive work in a short shutdown interval, e.g., in the five-year refinery shutdowns the maintenance workforce is often *doubled*. A shutdown project consists of the phases initiation, preparation, execution and termination [cf. 1]: The initiation and preparation phases comprise determining the work scope and detailing it into tasks and activities while determining key resource requirements, for example mechanics, equipment, and spare parts. Execution includes conducting the maintenance and termination includes verification, evaluation, and documentation.

Suitable spare parts ordering is crucial to avoid project delays during execution while reducing the build-up of unneeded inventory [cf. 2]. Some of the spare parts that are inspected during execution are very likely to need replacement, but for other parts the probability that they need replacement may be small. These latter parts are typically replaced depending on their *condition*, which only becomes clear during project execution. Such *on-condition* maintenance tasks are prevalent in shutdown maintenance because they are cost-effective [cf. 3]. A shortage of the required spare parts results in costly emergency orders and may lead to project delays, and most shutdown planning methods therefore advocate that long-leadtime spare parts should be stocked. Yet exactly what constitutes a *long* leadtime is unclear, and perhaps more importantly: should parts that need replacement with a very low probability also be included?

We consider the spare parts ordering policy against this background of shutdown maintenance project planning. We focus on decision making during project initiation and preparation, while costs are mainly incurred during project execution. For modeling ease, we assume that each maintenance activity is associated with a single spare part (otherwise, duplicate activities can be created).

Parts with a large replacement probability are typically stocked to avoid the risk, and our method focuses on parts for which the

* Corresponding author.
E-mail addresses: shzhu@escp.eu (S. Zhu), W.L.v.Jaarsveld@tue.nl (W. van Jaarsveld), rdekker@ese.eur.nl (R. Dekker).

https://doi.org/10.1016/j.ress.2021.108197
Received 11 January 2021; Received in revised form 5 November 2021; Accepted 8 November 2021
Available online 26 November 2021
0951-8320/

replacement probability is small. Because spare parts are expensive, the stocking decision for such parts is nontrivial: Parts that are stocked but not needed cause high holding costs and may even become obsolete, while parts that are needed but not stocked lead to costly emergency orders and longer maintenance activity durations. In turn, a longer maintenance activity duration *may* lead to an extremely costly delay in the shutdown project, depending on precedence relations between maintenance activities and delays in other activities. We investigate the resulting trade-off between the ordering cost of spare parts and the delay cost of the shutdown project.

In our study, we merge two streams of literature, viz. *project planning* and *spare parts inventory control*. In standard spare parts inventory control theory one assumes that decisions can be made for each part separately based on the consequences of that part going out-of-stock. In our setting however, there is no clear out-of-stock costs and the consequence of one part being out of stock may also depend on the availability of other parts. As a consequence, spare parts inventory decisions also become interdependent, which brings new challenges unaddressed in prior literature.

We analytically characterize the interdependence of the stocking policy for two-activity networks. The focus of our paper is then on inventory control for general project networks. We propose and test two simple algorithms based on standard inventory theory, and one algorithm adapted from project management theory (viz. the Project Evaluation and Review Technique, PERT). We find that the former algorithms perform poorly because they fail to recognize the interdependence between activities. The latter algorithm is based on approximating the project delay distribution using a normal distribution, and we show that the normal approximation is unsuitable in our setting.

These findings motivate us to develop a two-stage stochastic programming formulation for spare parts inventory management in shutdown maintenance. Uncertainty in demand in each activity is represented by discrete scenarios. In the first stage ordering policies are determined. In the second stage a detailed schedule is derived as a function of the proposed spare parts orders. We derive conditions under which certain activities can be pruned, which reduces the problem scale. Subsequently, we demonstrate that a sample average approximation with importance sampling (SAA-IS) can be used to solve large instances relatively quickly. In numerical tests on two test beds, we demonstrate the superior performance of this algorithm. The insights and algorithms developed in this paper underline the unique characteristics of inventory planning for shutdown maintenance, and give practitioners a concrete set of tools to deal with these characteristics.

In the next section we give a brief review of the relevant literature. In Section 3 we present the two-stage stochastic model. We present analytical results for an example with only two activities to gain some high-level insights, and we discuss the impact of dominated paths on the optimal solution. We also present the sample-average approximation with importance sampling. Heuristics are proposed in Section 4, and computational results are presented in Section 5. We conclude in Section 6.

2. Literature review

Critical Path Method (CPM) and PERT are two widely used techniques in shutdown maintenance scheduling of time based network before 2007 [4]. CPM is mostly used in the deterministic case, while PERT focuses on uncertain activity times. PERT assumes that activity times are random variable with given density function which usually is beta, uniform and normal [5,6]. Several variations of PERT have been proposed [e.g. 7–9]. PERT related methods are widely studied and focus on estimating activity durations and project completion time. There is also work on project scheduling with stochastic activity durations, see e.g. [10]. In this stream, Bevilacqua et al. [4] and Megow et al. [11] are particularly related because of their focus on shutdown planning: Bevilacqua et al. [4] apply the theory of constraints and risk assessment, and carry out a case study at a IGCC plant. Megow et al. [11] consider resource leveling for shutdown planning. However, neither of these studies consider spare parts inventory provisioning. From a methodological perspective, Valls et al. [12] and Laguna et al. [13] are related. They consider *stochastic activity interruptions*: Each activity is composed of a certain duration time before interruption and a uncertain remaining time, which is similar to the activity durations in our paper which consist of a deterministic duration and an extra duration time due to the possible delay caused by spare parts shortages. However, the focus of these papers is on (re)scheduling the project activities while dealing with unexpected project delays, and those project delays are exogenous to the proposed model. In contrast, we focus on costly preventive actions (i.e. purchasing spare parts during project preparation) that reduce the occurrence of unexpected delays, while the schedule is considered exogenous in our study. Finally, while sample average approximation has been proposed in the setting of project planning, we are not aware of prior work in this setting that includes importance sampling techniques. van Jaarsveld and Scheller-Wolf [14] use a similar approach while studying component replenishment and allocation for assemble-to-order systems.

Determining the optimal spare parts provision policy for a shutdown is also related to the literature on maintenance planning. In a routine maintenance system of capital goods, good spare parts inventory decisions are often derived from a queuing model [15–19]. Rahimi-Ghahroodi et al. [20] design a cooperative contract between emergency supplier and service provider in asset management subject to random failures. Sleptchenko and van der Heijden [21] jointly optimize the component redundancy level with spare parts inventories in routine maintenance system with state-dependent failures. Godoy et al. [22] consider condition reliability in the decision making of critical spare parts. Condition monitoring allows routine maintenance to take actions more effectively and make a good use of the lifespan of spare parts by taking conditions of equipment into account [23,24]. Contrary to what is considered in these papers, shutdowns occur infrequently, and the maintenance scope is much larger than that of routine maintenance. The scheduling of activities in the shutdown has to be considered due to the limited resources such as workers, equipment and tools, making the spare parts arrangement more complex. In addition, shutdown maintenance is system oriented. The ordering decision for a maintenance activity not only depends on itself, but also on the structure of the project network. Queuing models are therefore not effective in determining the spare part provision policy for shutdowns, and there is limited work in the field of spare parts inventory control for shutdown maintenance; e.g. Vaughan [15] makes no difference between shutdown and preventive maintenance, while the former has a larger scope and the project network cannot be ignored. Our work thus enriches the literature on maintenance planning and spare parts inventory control by considering the project network as the information for spare parts ordering: We fill the gap between spare parts ordering and project planning by considering spare parts decisions that affect the occurrence of unexpected delays of activities in a project network.

3. Model

3.1. Overview

We consider the spare parts ordering problem during shutdown project planning for industrial plants. The project planning is given by a set of activities and precedence relations between those activities. Parts are replaced depending on their deterioration condition, which is only observed in the execution phase of the shutdown project when each maintenance activity is conducted. We assume that each maintenance activity has a single associated part that may need replacement. Spare parts ordering decisions are made in the initiation/preparation phase. Subsequently, during project execution, the part associated with each maintenance activity needs replacement with some given probability.

Table 1
Notation used.

Notation	Description
I	Set of activities.
$\mathscr{P}$	The set of precedence relations
p_i	Demand probability for activity i
d_i	Basic duration for activity i
l_i	Leadtime for the spare part i
c_i	Ordering cost for spare part i
e_i	Emergency ordering cost for spare part i
a_i	$a_i = e_i p_i$
R_i	Order decision for spare part i. R_i is binary.
Ω_i	Random variable which indicates whether spare part i is needed or not
t_i	Earliest starting time for activity i
t_{obj}	The given target completion time
q	Penalty cost for each time unit of delay
ω_s	Scenario s
π_s	Weight of scenario ω_s
Δ	Delay of the project

Since project preparation starts months or even years before shutdown execution, we assume that spare parts ordered during preparation are available on time for process execution, and hence there will be no shortage of such parts.

Spare parts shortages occur only when a spare part that was not ordered during process preparation turns out to be needed during process execution; in those cases, a costly emergency order is placed and the maintenance activity is suspended until spare parts arrive. Thus the duration of the corresponding activity is increased by the emergency lead time. We consider an emergency ordering cost for orders placed during shutdown, which reflects additional charges for speedy production and delivery as well as the lack of economies of scale when placing orders at the last moment. Spare parts shortages may cause vast delays in activity durations. To keep the study focused, such shortages are the only source of uncertainty that affect activity durations in our model. As a consequence, activities durations can take on two values, *regular* or *delayed*, depending on whether we stock the spare part or not.

There is a given target completion time for the shutdown project: A penalty is incurred for each time unit of *project delay*, i.e. each time unit that the project exceeds the target completion time. This is in line with the practice that after the shutdown completes, production is planned to resume and failing to do so results in lost production and a failure to deliver products to customers [cf 1]. I.e. delays cause a production loss that shall be monetized in order to compute the delay penalty. Power plants have a larger and petrochemical industries have a smaller project duration in general: the average duration of shutdowns for oil refineries are as high as 42 days. The decision maker aims to make the most economic decision on how many spare parts should be ordered for the shutdown project in the initial phase before the execution of the maintenance so that the completion time is satisfied with a relatively low ordering cost. The shortage of spare parts prolongs the duration of individual activities and leads to higher ordering cost as emergency orders are placed, while overstocking spare parts causes high holding costs and possibly obsolescence. The goal is to optimally make spare parts stocking decisions by trading off between the expected emergency order and the delay cost of the project and the costs of stocking many spare parts during initiation/preparation.

3.2. Detailed model formulation

Table 1 lists the notation used. We consider a project with a set of activities $i \in I$, where i denotes the activity index as well as the index of the corresponding spare part (This is without loss of generality: if an activity would need two or more parts it can be split up into subactivities requiring one part only.) Each activity is also a node in a directed, acyclic graph. It will be convenient to additionally include a *sink* node in the graph, such that the nodes in the graph are $I \cup \{\text{sink}\}$.

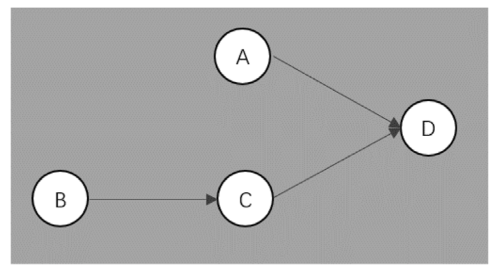

Fig. 1. A simple project with a sink node.

An example is shown in Fig. 1 where the sink node is D. Precedence relations are represented by arcs in that graph: An arc (i, j) pointing from node $i \in I \cup \{\text{sink}\}$ to node $j \in I \cup \{\text{sink}\}$ denotes that j can only start after i finishes. We denote the set of precedence relations by $\mathscr{P}$.

The binary decision variable R_i denotes whether part i is ordered in the initiation/preparation phase. Let $\mathbf{R}$ denote the corresponding column vector.

The uncertainty regarding spare parts requirements is represented in discrete scenarios: Each activity $i \in I$ has an associated Bernoulli distributed random variable Ω_i that indicates whether the associated spare part is needed or not. (Remember that we assume that either the spare part is needed, or it is not, hence a Bernoulli random variable is an appropriate model here.) Let $\boldsymbol{\Omega}$ denote the column vector of Ω_i for all $i \in I$, and let p_i denote the demand probability of spare parts in activity $i \in I$. The duration of activity i is resolved given the spare parts requirements and corresponding stocks: The duration is $d_i + l_i$ when there is a shortage or d_i otherwise, where d_i denotes the basic duration and l_i denotes the emergency leadtime for the part.

The delay cost and the emergency ordering cost of the project is determined in the second stage under the given order policy. Additionally, let h_i denote the total holding, obsolescence and scrapping costs if part i remains unused when the project is finalized, i.e. if it is procured but not used during execution [see e.g. 25, for more extensive discussions], and let c'_i denote the regular procurement costs for part i. Let c_i denote the *expected* part i costs when deciding to procure part i, i.e. $c_i = (1 - p_i)h_i + c'_i$. Let $\mathbf{c}$ denote the corresponding column vector. Emergency ordering costs are denoted e_i for spare part i.

Given R_i and Ω_i, let $E_i = \max(0, \Omega_i - R_i)$ indicate whether an emergency order is needed for activity $i \in I$. Also, given actual activity durations, t_i denotes the earliest starting time for node $i \in I \cup \{\text{sink}\}$: For $i \in I$, t_i corresponds to the activity starting time, while the precedence relations $\mathscr{P}$ ensure that t_{sink} corresponds to the project completion time. The delay of the project, denoted by Δ, is defined as $\Delta = max(t_{sink} - t_{obj}, 0)$, where t_{obj} denotes the given target completion time. Let q denote the penalty cost for each time unit of delay.

We propose a two-stage stochastic programming model for the problem [cf. 26]. In the first stage, spare parts ordering decisions are taken based on known probabilistic information. In the second stage, completing times of activities are derived as a function of the proposed spare parts orders and part requirements. Hence the problem can be represented by the following stochastic program:

$$\min_{\mathbf{R}}\{\mathbf{c}^{\mathbf{T}}\mathbf{R} + \mathbb{E}_{\boldsymbol{\Omega}} v(\mathbf{R}, \boldsymbol{\Omega})\} \tag{1}$$

where $v(\mathbf{R}, \boldsymbol{\Omega})$ is given by

$$v(\boldsymbol{R}, \boldsymbol{\Omega}) = \min_{\Delta, E_i} q\Delta + \sum_{i\in I} e_i E_i \tag{2}$$

s.t.

$$t_j - t_i \geq d_i \quad \forall (i,j) \in \mathscr{P} \tag{3a}$$

$$t_j - t_i + R_i l_i \geq d_i + l_i \Omega_i \quad \forall (i,j) \in \mathscr{P} \tag{3b}$$

$$E_i \geq \max\{\Omega_i - R_i, 0\} \tag{3c}$$

$$\Delta \geq \max\{t_{sink} - t_{obj}, 0\} \tag{3d}$$

$$R_i \in \{0,1\} \quad \forall i \in I \tag{3e}$$

$$t_i \geq 0 \quad i \in I \cup \{\text{sink}\} \tag{3f}$$

Here, $v(\boldsymbol{R}, \boldsymbol{\Omega})$ denotes the delay and emergency ordering costs when the part ordering decision is $\boldsymbol{R}$ given that the spare parts demand is $\boldsymbol{\Omega}$. This can be inferred by inspecting the linear program (LP) (2)–(3f): The objective of this LP is minimizing the delay costs $q\Delta$ and emergency ordering costs $\sum_{i\in I} e_i E_i$. We take into account precedence constraints between activities (3a) and activity durations (3b).

Additionally, (3c) reflects that an emergency order is placed when the part is needed but was not stocked during project preparation. The delay of the shutdown project Δ is the positive difference between the realized completion time and the target completion time, as reflected in constraint (3d).

3.3. Intuitive insights

We next derive some properties of the proposed model. First, to gain intuitive understanding of the interplay between inventory decisions in our model, we shall investigate in detail optimal inventory control for a stylized model:

Proposition 1. *Consider a project consisting of two consecutive activities, under the assumption that target completion time is shorter or equal to the sum of the basic duration of the two activities. We consider costs $c_1 > c_2$, and assume that $e_1 = c_1$ and $e_2 = c_2$: there are no additional costs for emergency shipment. For this model, the dependence of the optimal ordering policy on demand probabilities p_1 and p_2 can be analytically characterized based on the quantities $E = d_1 + d_2 - t_{obj}$, $F = d_1 + l_1 + d_2 - t_{obj}$, $G = d_1 + d_2 + l_2 - t_{obj}$, $K = d_1 + l_1 + d_2 + l_2 - t_{obj}$, which are all non-negative under our assumption. The characterization is summarized in Fig. 2.*

A detailed derivation of this result appears in Appendix A. The vertical line is derived by comparing the policy of stock item 1 and stock both items. As both policies stock item 1, the failure probability of item 1 (p_1) has no impact on the decision anymore. Therefore, we observe the vertical line. The horizontal line can be explained similarly.

Proposition 1 shows that the spare part ordering decision for one activity not only depends on the failure probability of the part itself, but also on that of the other activity. This sets the model apart from typical models in spare parts inventory literature. It also makes clear that the problem is not trivial and that it calls for tailored algorithms and heuristics that we shall develop next.

3.4. Model reduction

Depending on the structure of the project, some paths may have a lot of slack. In some circumstances, this allows us to immediately derive the optimal inventory decision for some activities. To formalize this, a *path* will denote a sequence $(i_1, i_2, \ldots, i_n)$ of activities with a precedence relation between every pair of subsequent activities, i.e., such that $(i_j, i_{j+1}) \in \mathscr{P}$ for $j = 1, \ldots, n-1$. Paths can be dominated by other paths:

Definition 1. Consider two paths $\mathscr{I} = (i_1, i_2, \ldots, i_n)$ and $\mathscr{J} = (j_1, j_2, \ldots, j_m)$. Denote the activities that are in $\mathscr{I}$ but not in $\mathscr{J}$ by $\mathscr{I} \setminus \mathscr{J}$, and define $\mathscr{J} \setminus \mathscr{I}$ similarly. We say that $\mathscr{I}$ is *dominated* by $\mathscr{J}$ if the two paths have the same start and end node ($i_1 = j_1$, $i_n = j_m$), and if $\sum_{i\in\mathscr{I}\setminus\mathscr{J}} d_i + l_i < \sum_{j\in\mathscr{J}\setminus\mathscr{I}} d_j$.

A path that is dominated can never become the critical path. This leads to the following result:

Proposition 2. *Consider an activity $i \in I$, and suppose that every path $\mathscr{I}$ that contains i is dominated (by some path $\mathscr{J}$). Then activity i can never be on a critical path, and hence it is optimal to place an order for part i ($R_i = 1$) if and only if $c_i < p_i e_i$.*

While the statement is quite intuitive, we give a formal proof next for completeness.

Proof. To see that activity i can never be on a critical path, we proceed by contradiction: Suppose that there is some first-stage decision $\mathbf{R}$ and some parts requirements realization $\boldsymbol{\omega}$ such that the critical path $\mathscr{I}$ contains i. By assumption, $\mathscr{I}$ is dominated by some path $\mathscr{J}$, and hence we find that the slack for path $\mathscr{I}$ is bounded below by $\sum_{j\in\mathscr{J}\setminus\mathscr{I}} d_j - \sum_{i\in\mathscr{I}\setminus\mathscr{J}} (d_i + \Omega_i l_i)$. The slack must thus be larger than zero, indicating that emergency orders placed for activity i are always in time to ensure that path $\mathscr{I}$ will not become critical. This contradicts that the path $\mathscr{I}$ is critical, which proves our first claim.

Now, if i cannot be on a critical path, then for all realizations $\boldsymbol{\omega}$ of $\boldsymbol{\Omega}$ the associated delay Δ is not affected by R_i. Hence, the inventory decision R_i for activity i impacts only the first-stage ordering costs, and the second stage emergency ordering costs, leading to the second claim. □

Activities that can never be on the critical path are said to be dominated. Before we use SAA or any of the other algorithms that will be proposed for solving the problem, we always first use Proposition 2 to reduce the number of activities in the approximating problem (4): In a pre-processing stem the decisions for dominated activities are fixed to their respective optima, which in some cases substantially simplifies solving of the remaining problem.

3.5. Sample average approximation

Since Ω_i takes on discrete values, $\boldsymbol{\Omega}$ takes on values from a discrete, finite set. This set has $2^{|I|}$ elements. Because $2^{|I|}$ grows quickly in $|I|$, the deterministic equivalent of (1) may be intractable. We therefore propose to use sample average approximation [cf. 27], which approximates the expectation in (1) by drawing a *sample* that consists of scenarios for $\boldsymbol{\Omega}$ with corresponding weights.

The standard approach to construct a sample S involves drawing for $s = 1, \ldots, |S|$ an independent replication $\boldsymbol{\omega}_s$ of $\boldsymbol{\Omega}$. Then S is constructed as $\{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_{|S|}\}$, and the *weight* of each scenario $\boldsymbol{\omega}_s$ equals $\pi_s = \frac{1}{|S|}$. We refer to such a sample as a *standard sample* of size $|S|$. We shall discuss an alternative approach for constructing samples in Section 3.6, which gives rise to heterogeneous scenario weights.

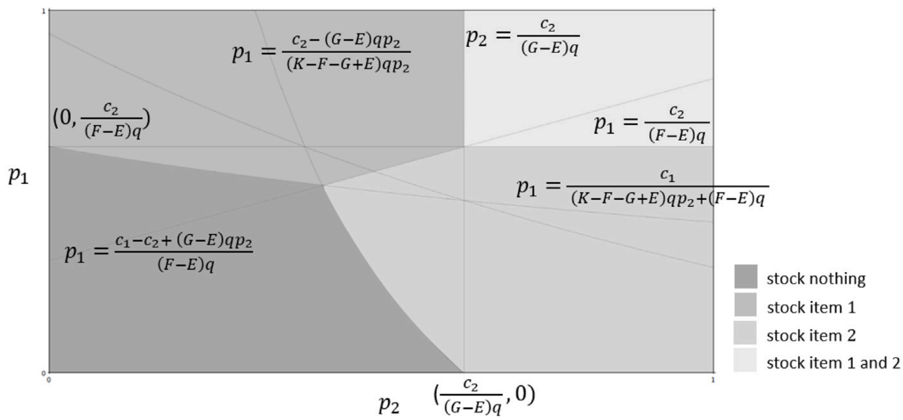

Fig. 2. The optimal ordering policy with regard to demand probability for a project with two activities, under the assumptions detailed in Proposition 1.

Since $\boldsymbol{\omega}_{s,i} \in \{0,1\}$ indicates whether part i is required in scenario $\boldsymbol{\omega}_s$, and since R_i indicates whether part i is procured, the approximating problem for sample S is given by:

$$\min_{\mathbf{R}}\{\mathbf{c}^{\mathbf{T}}\mathbf{R} + \sum_{s=1}^{|S|} \pi_s v(\mathbf{R}, \boldsymbol{\omega}_s)\} \tag{4}$$

where $v(\mathbf{R}, \boldsymbol{\omega}_s) = \min_{\Delta_s, E_{i,s}} q\Delta_s + \sum_{i\in I} e_i E_{i,s}$
s.t.

$$\begin{aligned} t_{j,s} - t_{i,s} &\geq d_i \quad \forall (i,j) \in \mathscr{P} \\ t_{j,s} - t_{i,s} + R_i l_i &\geq d_i + l_i \boldsymbol{\omega}_{s,i} \quad \forall (i,j) \in \mathscr{P} \\ E_{i,s} &\geq \max\{\boldsymbol{\omega}_{s,i} - R_i, 0\} \\ \Delta_s &\geq \max\{t_{sink,s} - t_{obj}, 0\} \\ R_j, \omega_{j,s} &\in \{0,1\} \quad \forall (i) \in \mathscr{P}, \forall (\boldsymbol{\omega}_s) \in \Omega_S \\ t_{i,s} &\geq 0 \quad i \in I \cup \{\text{sink}\} \end{aligned}$$

Here, $t_{j,s}$ represents the earliest finish time of activity j for sample s, while Δ_s represents the project delay for scenario s. Finally, $E_{i,s}$ indicates whether an emergency order is placed for part i in scenario s.

To apply the sample average approximation, we obtain a solution by solving the approximating problem for a sample S. The standard Monte Carlo upper bound is obtained by evaluating the candidate solution using a sample that is independently obtained from S. The Monte Carlo lower bound is the expectation of the approximating problem (4), and is estimated by repeatedly solving the problem for several independently drawn samples [cf. 27].

3.6. Importance sampling

We next discuss an alternative approach for sample generation. Since parts with high replacement probabilities typically need to be stocked, this paper focuses on cases where the replacement probability is small. As a consequence, most scenarios that are generated independently are unlikely to result in project delays, and hence have no contribution to delay costs. Still, sufficient scenarios which do result in project delays are needed in order to accurately approximate the delay cost function. We use importance sampling approach to enable accurate delay cost approximations while keeping the total number of scenarios in the samples within bounds. The latter is important because solving (4) for samples with many scenarios is impractical.

To evaluate whether scenarios contribute to a better approximation of the delay costs, we evaluate each generated scenario based on an initial solution and classify it as *valid* if it leads to project delay and invalid otherwise. All valid scenarios are included in the mixed-integer programming (MIP) formulation (4) to obtain the ordering policy, while most invalid scenarios are discarded. The weight of the remaining invalid scenarios is adapted such that discarding the remaining scenarios does not introduce a bias. As a consequence, we ensure that the lower bounds derived in [27] continue to be valid. As the demand probability for each activity is small, a large proportion of scenarios are invalid scenarios, and discarding most of those scenarios substantially reduces the computational burden without seriously harming the approximative power. The importance sampling approach for obtaining a sample involves the following steps:

- **Step 1:** Generate a standard sample of size N_0, where N_0 is small enough such that the corresponding MIP (4) is tractable. Solve the MIP and obtain an initial solution that indicates which parts to stock.
- **Step 2:** Repeat Step 1 five times to obtain five solutions, and construct a reference solution $\mathbf{R}'$ as follows: $\mathbf{R}'_i = 1$ for those activities i which were stocked in all the five solutions, while $\mathbf{R}'_i = 0$ otherwise.
- **Step 3:** Generate N_1 scenarios, where N_1 is large enough to include sufficiently many unlikely but costly scenarios. Evaluate N_1 scenarios using the initial candidate solution $\mathbf{R}'$, and classify into valid scenarios or invalid scenarios based on whether there is project delay for that solution. Let N_v denote the number of valid scenarios and let N_{inv} denote the number of invalid scenarios N_1.
- **Step 4:** To construct S, we combine all N_v valid scenarios (with weight $1/N_1$) with a randomly selected subset of the invalid scenarios. This random subset consists of $n_{inv} < N_{inv}$ invalid scenarios, and the selected scenarios have a bigger weight $\frac{N_{inv}}{N_1 \times n_{inv}}$ so that no bias is introduced by discarding the remaining scenarios.

The samples constructed with these steps are referred to as skewed samples. When generating multiple skewed samples for a problem, the reference solution of Step 2 is reused, i.e., only steps 3 and 4 are repeated.

4. Heuristics

An LP approach might not be able to solve a large scale problem, and some companies might not have an LP solver. Therefore, we propose five heuristics as benchmarks. Three of them are scenario based heuristics. We also present a heuristic based on a standard critical path method.

4.1. Scenario based heuristics

We will test three scenario based heuristics that search locally in the solution space: the removal heuristic, the stock heuristic and the combined heuristic. The stock heuristic is the opposite of the removal heuristic and the combined heuristic chooses the better solution of the

Table 2
Parameters of the example for the removal heuristic.

Activity	Cost	Failure probability	Basic duration	Emergency lead time
A	30	0.2	40	20
B	10	0.1	20	40
C	40	0.2	10	30
D	20	0.3	30	30

two. All local search heuristics start with a generated standard sample consisting of N_1 scenarios that is used to estimate costs of solutions throughout the search procedure.

The **removal heuristic** is initiated from the intuition that we should not stock the spare part for which the purchase cost minus its reduction on delay costs is largest: We start with all spare parts being ordered. Next, iterating over all spare parts, we use the sample to estimate the cost savings of instead not stocking that spare part. We select the solution corresponding to the largest cost saving. This process is repeated until no cost saving is observed by removing any remaining spare part.

We use a simple example to illustrate the removal heuristic. The example network is small enough such that we can evaluate solutions exactly, so we do not make use of evaluations based on samples in the example. The project network is shown in Fig. 1 and the parameters are in Table 2. The target completion time for the project is 100 and the delay cost are 1000 per time unit. Assume that $e_i = 110\% c_i$ for each activity. We start from the initial solution $S_0^{rh} = (1,1,1,1)$ that all spare parts being ordered. The corresponding average cost is $C_0^{rh} = 100$. Next we consider solution $(1,1,1,0)$ where there is no stock for activity D; the corresponding cost is estimated to be 86. For order policy $(1,1,0,1)$, we have cost of 68. For order policy $(1,0,1,1)$, the corresponding cost is 91. The cost is 76 under order policy $(0,1,1,1)$. We observe that the smallest cost is obtained under policy $(1,1,0,1)$, so we update the solution from $(1,1,1,1)$ to $(1,1,0,1)$.

Then we start from solution $(1,1,0,1)$ and repeat the process. We obtain the average cost 1254 under policy $(1,1,0,0)$, 659 for $(1,0,0,1)$ and 44 for $(0,1,0,1)$. The smallest cost is obtained under policy $(0,1,0,1)$. So the updated solution is $(0,1,0,1)$ and the new average cost is 44.

Again starting from solution $(0,1,0,1)$, the smallest local improvement is obtained under $(0,0,0,1)$ with average cost 635. We found that this step cannot reduce the cost anymore so we stop here. The final order policy is then $S_{final}^{rh} = (0,1,0,1)$ with cost $C_{final}^{rh} = 44$ and no project delays.

The **stock heuristic** is the opposite of the removal heuristic: It starts from no stock for any activity and then stocks for the activity with the largest cost saving. This is repeated until no cost saving is observed. The **combined heuristic** obtains a solution from the stock heuristic and another solution from the removal heuristic, and selects the solution with the lowest costs as estimated using a separate sample.

For a project network with n activities, the removal and stock heuristic have a time complexity of $\mathcal{O}(n^2)$ cost evaluations.

4.2. Heuristics — standard critical path method with normal distributed project time assumption

Motivated by the widely used standard critical path method ([5]), we apply an approach based on the standard critical path method with normal distributed project time assumption. We consider the mean duration of each activity to find the initial order decision and critical path, and improve the order decision locally. Let a_i denote the expected emergency ordering cost where $a_i = e_i p_i$ and $\mathbf{a}$ is the column vector of a. $\mathbf{v}$ denotes a column vector with all elements equal to 1. Given the mean duration $d_i + l_i p_i$ for each activity $i \in I$ where p_i denotes the demand probability, we have the deterministic problem as follows.

$$\min_{\mathbf{R}} \{ \mathbf{c^{TR}} + \mathbf{a^T}(\mathbf{v} - \mathbf{R}) + q\Delta \} \tag{5}$$

s.t.

$$t_j - t_i \geq d_j + l_j p_j (1 - R_j) \quad \forall (i,j) \in \mathscr{P}$$
$$\Delta \geq \max\{t_{sink} - t_{obj}, 0\}$$
$$t_i \geq 0 \quad i \in I \cup \{\text{sink}\}$$
$$R_i \in \{0,1\} \quad \forall i \in I$$

The details of the approach are as follows:

- **Step 1:** Solve the problem (5) and obtain the initial order decision $R_i, i \in I$. Consider the mean activity duration $d_i + (1 - R_i) l_i p_i$ and find the critical path. Let I_P denote the set of critical activities.
- **Step 2:** The mean of the total project time is $\mu = \sum_{i \in I_P} d_i + (1 - R_i) l_i p_i$. The variance of the total project time is $\sigma^2 = \sum_{i \in I_P} l_i^2 p_i (1 - p_i)(1 - R_i)$.
- **Step 3:** The total project time is assumed to be normally distributed with μ and σ^2. The expected project delay is $\Delta = \int_{t_{obj}}^{\infty} (t - t_{obj}) \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2} dt = -t_{net}[1 - \Phi(\frac{t_{net}}{\sigma})] + \sigma\phi(\frac{t_{net}}{\sigma})$, where $t_{net} = t_{obj} - \mu$, and $\Phi(.)$, $\phi(.)$ are the Cdf, pdf of a standard normal distribution.
- **Step 4:** The total project cost is given by $\sum_{i \in I} c_i R_i + e_i p_i (1 - R_i) + q\Delta$.
- **Step 5:** For a critical activity j with $R_j = 0$ in the initial order decision, let $R_j = 1$ and calculate the corresponding total project cost with the normal approximation described in step 1–4. Change the order decision if the total project cost is reduced.
- **Step 6:** Repeat Step 5 until no cost reduction is observed.

4.3. Heuristics - Based on standard inventory control theory

We will also consider two simple heuristics based on standard inventory theory. These heuristics each consider the spare parts individually, without taking into account the project activity network. The first heuristic is based on setting a (cycle) service level target, i.e. a target probability of avoiding shortage for each part. Since parts are used either in quantity 1, or in quantity 0, using a service level target of $0 \leq \alpha \leq 1$ implies that all parts which have a failure probability that is larger than $1 - \alpha$ will be stocked. This *threshold heuristic 1* thus only considers the failure probability in decision making rather than comprehensive information such as the stock cost and the structure of the project network. It represents a simple decision rule that one might adopt in practice: Stock parts that are sufficiently likely to be replaced. In *threshold heuristic 2*, we stock item i only if $p_i \times d_i / c_i$ is higher than a given threshold. Similar to the threshold heuristic 1, this heuristic represents a simple decision rule that we stock parts that have the highest contribution considering the failure probability, leadtime and the cost.

5. Numerical experiments

To evaluate the proposed model and solution approaches, we consider project networks from two sets of instances. First we consider instances from the online dataset MMLIB, which can be downloaded from http://www.projectmanagement.UGent.be. Though this artificial dataset is designed for the multi-mode resource-constrained project scheduling problem, we use the network structure and generate parameters such as failure probability in our problem. The library has 540 single-mode, 50-activity instances with a start and a sink node. We conduct experiments on the first 10 instances.

The experiment based on the instances from MMLIB has some limitations. Though the instances have 50 activities, all paths in the project networks are relatively short, making the problems small scale: the longest path of the networks ranges between 5 and 10 activities and is 7 activities on average. We therefore also consider large scale instances with random directed acyclic project networks which have long paths. In the next two sections we discuss the data and the setup of the experiments. Section 5.3 gives the results.

5.1. Experiment setup — instances from MMLIB

We generate parameters including basic processing time, lead time, spare parts cost and demand probability for each activity of the project networks. The basic processing time d_i is the deterministic duration under mode 2 given in the instance. Empirically, it is well-known that most parts are relatively standard and cheap, while some parts are more complex and therefore expensive. To closely model this reality, most parts (90%) are relatively common, resulting in shorter emergency leadtimes and moderate spare part costs. Other parts (10%) are technologically complex and have longer emergency leadtimes and very high costs. For the common parts, the demand probability p_i is generated following a uniform distribution between $[0, 0.2)$, spare costs c_i following a uniform distribution on $[100, 300)$ and leadtimes l_i following a uniform distribution between $[\frac{d_i}{2}, d_i)$. The complex parts have a failure probability p_i generated uniformly on $[0, 0.002)$, spare parts costs generated uniformly on $[10000, 30000)$, and a lead time l_i generated on $[15d_i, 30d_i)$. For all parts, the emergency ordering cost e_i is set to be 110% of the costs c_i. Note that the demand probabilities are overall smaller than those for the MMLib instances and the deadline is tighter. The delay cost q of the project is 100,000 per time unit. The target project completion time is set as follows: The target completion time t_{obj} is set to be the project finishing time in which the expected duration of each activity $d_i + p_i l_i$ is considered.

As the instances from library-MMLIB are small scale, we do not use importance sampling and adopt *standard* samples as discussed in Section 3.5. We obtain a solution by solving the sample approximation (4), by the various local search heuristics, and by our threshold heuristics. In SAA and the heuristics, we use samples of different sizes (1000, 5000 and 10,000 scenarios) and assess the impact on solution quality. The lower bound of SAA is estimated by solving the stochastic model 100 times, and using the resulting 100 objective values to create an asymptotic estimator. For evaluating the various solutions, we take the average of 100 standard samples, each with 500,000 scenarios. For the threshold heuristic 1, the threshold failure probability is empirically chosen to be 0.01, which is widely used in practice; similarly, the value correspondingly to threshold heuristic 2 is 0.005.

5.2. Experiment setup — instances with random directed acyclic project networks

We generate 20 random directed acyclic project networks, of which ten with 50 activities and ten with 100 activities, using the approach discussed in Appendix C. Over the ten 50-activity instances, the average longest path has 39 activities. The corresponding average for the ten 100-activity instances is 54. We use slightly different parameter settings to test whether they influence the performance of the methods. The basic processing time for each activity is a uniform random number $d_i \in [10, 20]$. For 26% of the activities, the demand probability p_i is generated following a uniform distribution on $(0, 0.5]$; for 40% of the activities the corresponding range of p_i is $(0, 0.05]$; for 30% of the activities the corresponding range of p_i is $(0, 0.005]$; and for 4% of activities the corresponding range of p_i is $[0, 0.0005)$. For the first three categories that together comprise 96% of parts, leadtimes l_i are generated uniformly on $[\frac{d_i}{2}, d_i]$, and spare parts costs are generated uniformly on $[100, 300]$. For the last category comprising the remaining 4% of parts, part cost c_i are generated uniformly on $[10000, 30000)$ while leadtimes are generated uniformly on $l_i \in [30d, 60d_i)$. For all parts, the emergency ordering costs e_i are set to be 110% of the costs c_i.

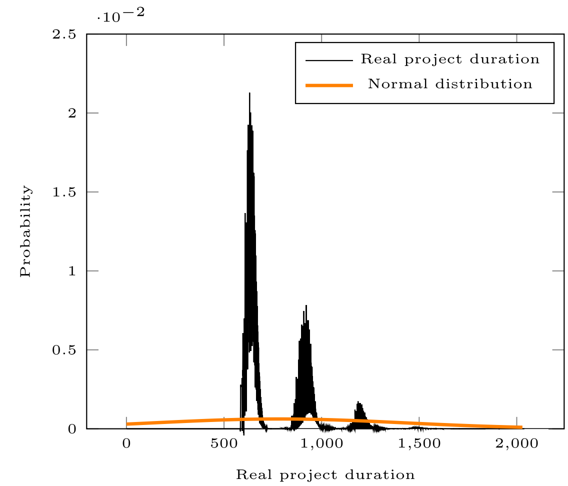

Fig. 3. Example of distribution of real project duration based on the solution of normal approximation (The mean and std of the normal distribution in orange is obtained from the step 2 in Section 4.2 (normal approximation heuristic)).

The target project completion time t_{obj} is set to be the lifespan in which the expected duration of each activity $d_i + p_i l_i$ is considered. The delay cost of the project is 100,000 per time unit. We apply solution approaches SAA (both with skewed and standard samples) and heuristics to solve the two sets of instances. For the threshold heuristic 1, the threshold failure probability is empirically chosen to be 0.01, which is widely used in practice; the value correspondingly to threshold heuristic 2 is 0.005.

For the various algorithms, we generate standard samples for which we consider $N = 8,000$, $N = 200,000$ and $N = 500,000$ scenarios per sample; we found that N must be large as some parts have small failure probability but high consequence due to stock out. These samples are used to generate solutions for the local search heuristics, but the SAA problem (4) is computationally intractable for samples with $N \geq 200,000$ scenarios. Therefore we also adopt the importance sampling approach of Section 3.6 with $N_1 = N$, i.e. we generate N samples initially of which some will be discarded. In particular, we include 3% of the invalid scenarios, i.e. $n_{inv} = 0.03 \times N_{inv}$, and the remaining invalid scenarios are discarded. To obtain the reference solution $\mathbf{R}'$ for classifying scenarios as valid or invalid, we use $N_0 = 50,000$ for $N > 200,000$, and $N_0 = 2,000$ when $N = 8,000$.

We refer to the sample average approximation (SAA) algorithm which uses the skewed samples obtained in this fashion as the SAA algorithm with importance sample (SAA-IS). Both for the standard SAA algorithm and the SAA-IS algorithm we apply the approach discussed in Section 5.1 to obtain estimators of lower and upper bounds. To solve the various MIPs that arise in these algorithms, we use CPLEX 12.8.0. All experiments are performed on an Intel Core i7 2.30 GHz processor laptop with 16.0 GB RAM.

5.3. Results

We compare the average cost and computation time of instances under each experiment setting. The results of instances from library-MMLIB are shown in Tables 3–5. The results of instances with random directed acyclic project networks are shown in Tables 6–8. Presented results are the average values over the instances in the two benchmark sets.

Tables 4 and 7 show the Monte Carlo upper bound and lower bound estimators obtained using SAA and SAA-IS, respectively.

Table 3
Cost comparison — instances from library-MMLIB.

Nr. activ.	Scen- arios	SAA	Remov. Heu.	Stock Heu.	Comb. Heu.
	1000	117081.3	120846.5 (3.2%)	343819.6 (193.7%)	120846.5 (3.2%)
50	5000	87805.7	100975.0 (15.0%)	322618.1 (267.4%)	100975.0 (15.0%)
	10000	88306.8	104223.1 (18.0%)	256546.5 (190.5%)	104223.1 (18.0%)
Nr. activ.	Scen- arios	SAA	Normal Heu.	Threshold Heu.1	Threshold Heu.2
	1000	117081.3	736639.8 (529.2%)	178641.7 (52.6%)	162674.7 (38.9%)
50	5000	87805.7	736415.2 (738.7%)	178641.7 (103.5%)	162674.7 (85.3%)
	10000	88306.8	736691.0 (734.2%)	178641.7 (102.3%)	162674.7 (84.2%)

Table 4
Lower bound and upper bound of the problem — instances from library-MMLIB.

Nr. activ.	Scenarios	Lower bound	Upper bound	Avg. cost gap
	8000	59743.1	116114.9	94.4%
50	200000	81321.8	93637.7	15.1%
	500000	82524.9	86516.4	4.8%

Table 5
Computation time comparison (in sec.) — instances from library-MMLIB.

Nr. activ.	Scen- arios	SAA	Remov. Heu.	Stock Heu.	Comb. Heu.	Normal. Heu.	Threshold. Heu.1	Threshold. Heu.2
	1000	0.6	5.5	1.2	6.7	0.2	$< 10^{-3}$	$< 10^{-3}$
50	5000	1.1	9.0	3.2	12.2	0.5	$< 10^{-3}$	$< 10^{-3}$
	10000	3.6	13.1	9.6	22.7	0.4	$< 10^{-3}$	$< 10^{-3}$

Problems based on the instances from MMLIB have relatively short paths, and therefore it is relatively easy to find scenarios which may lead to extreme makespans. For that reason, SAA is able to solve the problem, which is why the SAA-IS approach is not included. The SAA outperforms the other methods both in cost and computation time. The removal heuristic is worse than the SAA, with 3.2%, 15.0% and 18.0% higher in cost for scenarios $N_1 = 1000, 5000$ and 10000 respectively. For stock heuristic the cost gap becomes 193.7%, 267.4% and 190.5%. The combined heuristic selects the better solution between the removal and stock heuristics for each instance. We found that in these cases, the stock heuristic performs much worse than the removal. Therefore, the combined heuristic selects the result of the removal heuristic in each case. The standard critical path method with normal distributed project time assumption leads to large project delays Δ. The cost is around 7 times higher than those from the SAA and the removal heuristic. The average cost by the threshold heuristic 1, which only stocks for parts with probability higher than 0.01, is around 1.5 times larger than the Monte Carlo upper bound. We can see that though the threshold heuristic is easy to implement in practice, it is far away from the optimal. Similar results can be found for the threshold heuristic 2 which stores the highest contribution by considering the failure probability, lead time and the cost. Finally note that the SAA sometimes gives higher costs than the upperbound, as the latter is an estimator only.

For the instances with random directed acyclic project networks, the average cost based on the candidate solution of each method is given in Table 6. The Monte Carlo upper bound and lower bound are shown in Table 7. In general, SAA-IS gives the best performance both in the objective value and computation time. The performance of SAA-IS increases with the number of scenarios. For instances with 50 activities, the cost by the SAA-IS with $N_1 = 8000$ scenarios is 41.8% less than the removal heuristic. With scenarios increased to $N_1 = 500000$, the cost of the SAA-IS is 38.2% less. The cost of the stock heuristic is 3.5% higher than the SAA-IS. Fig. 4 presents the breakdown cost of each method. The removal heuristic and the normal approximation has much higher emergency cost than the SAA-IS. The SAA is only solvable for a small number of scenarios. Though the removal and stock heuristics outperform SAA and SAA-IS given a very small number of scenarios, they substantially exceed the upper bound on the optimal objective function of the stochastic problem obtained using SAA-IS when the scenario number is large. The smallest cost and computation time is observed under approach SAA-IS given a large scenario number. As shown in Table 7, with the increase of the number of scenarios from $N_1 = 8000$ to 500000, the average cost gap between lower bound and upper bound over 10 instances shrinks from 18.6% to 2.4%.

The stock heuristic has a lower average cost than the removal heuristic. This is observed in most experiment sets. As the demand probability for each activity is very small, the final decision only orders spare parts for few activities of the shutdown project. In each decision round, the stock heuristic updates the decision for one activity with the largest cost saving. Similar to the removal heuristic, the stock heuristic also cannot guarantee close-to-optimal solutions as it might stock for one activity at the cost of skipping a couple of activities with each has a smaller cost saving but a larger total cost saving. The longest path of project networks from MMLIB has 7 activities in average, while the longest path of the 50-activity random directed acyclic networks has 39 activities in average and 54 activities for 100-activity instances. Comparing Table 3 to Table 6, the removal heuristic performs better than the stock heuristic in all instances in the MMLIB experiment. In the random directed acyclic project networks with 50 activities, the stock heuristic beats the removal heuristic in most instances. For 100-activity project networks, the stock heuristic performs better than the removal heuristic in all instances. We may conclude that the cost advantage of the stock heuristic compared to the removal heuristic is more significant in project network with longer paths.

In both instance sets, we find that the critical path based heuristic with normal distributed project time assumption leads to solutions far away from the optimum. Fig. 3 shows the plot of a real project duration under the solution of such method and the corresponding normal distribution. This figure is based on one of the network with 50 activities in our experiment. We observe that the real project duration distribution is multimodal and normal distribution is not a good approximation in such case. An example is presented in Appendix B, which shows normal approximation fails to stock for activity 9 and 13. This behavior results in very poor solutions: e.g. the cost is more than 7 times higher than those of the SAA-IS method in the experiments with random directed acyclic project networks where the delay cost q is 100,000. These findings clearly underline the need for scenario-based approaches such as those developed in this paper. We emphasize that the critical path

Table 6
Cost comparison — instances with random directed acyclic project networks.

Nr. activ.	Scen- arios	SAA	SAA- IS	Remov. Heu.	Stock Heu.
50	8000	28586.8	29981.4	51544.3	35336.5
	200000	–	29090.9	48407.5	35408.3
	500000	–	28089.2	45508.3	33322.9
	Scen- arios	Comb. Heu.	Normal. Heu.	Threshold. Heu 1.	Threshold Heu.2
	8000	33462.2	148899.3	37624.5	36125.8
	200000	32881.4	148948.9	37624.5	36125.8
	500000	30692	129002.2	37624.5	36125.8
Nr. activ.	Scen- arios	SAA	SAA- IS	Remov. Heu.	Stock Heu.
100	8000	43980.8	44670.1	140560.7	46033.9
	200000	–	43356.1	127705.2	44584.5
	500000	–	42352.2	130149.1	43820.6
	Scen- arios	Comb. Heu.	Normal. Heu.	Threshold. Heu 1.	Threshold Heu.2
	8000	46033.9	198999.8	55106.3	54286.1
	500000	44584.5	199814.3	55106.3	54286.1
	8000	43820.6	183855.8	55106.3	54286.1

Table 7
Lower bound and upper bound of the problem — instances with random directed acyclic project networks.

Nr. activ.	Scenarios	Lower bound	Upper bound	Avg. cost gap
50	8000	24489.5	29050.8	18.6%
	200000	27226.7	28829.0	5.9%
	500000	27615.4	28268.5	2.4%
100	8000	34604.5	44050.0	27.3%
	200000	39671.9	42805.9	7.9%
	500000	40078.9	42262.8	5.4%

Table 8
Computation time comparison (in sec.) — instances with random directed acyclic project networks.

Nr. activ.	Scen- arios	SAA	SAA- IS	Remov. Heu.	Stock Heu.	Comb. Heu.	Normal. Heu.	Threshold. Heu.	Threshold Heu.2
50	8000	7.7	0.5	1.9	0.7	2.5	0.3	$< 10^{-3}$	$< 10^{-3}$
	20000	–	7.2	48	15.7	63.7	0.3	$< 10^{-3}$	$< 10^{-3}$
	50000	–	21.5	119.5	38.8	158.4	0.3	$< 10^{-3}$	$< 10^{-3}$
100	8000	23.1	1.8	18.5	4.3	22.8	0.2	$< 10^{-3}$	$< 10^{-3}$
	20000	–	17.6	487.1	130.7	617.8	0.2	$< 10^{-3}$	$< 10^{-3}$
	50000	–	75.1	1177.7	317.5	1495.2	0.4	$< 10^{-3}$	$< 10^{-3}$

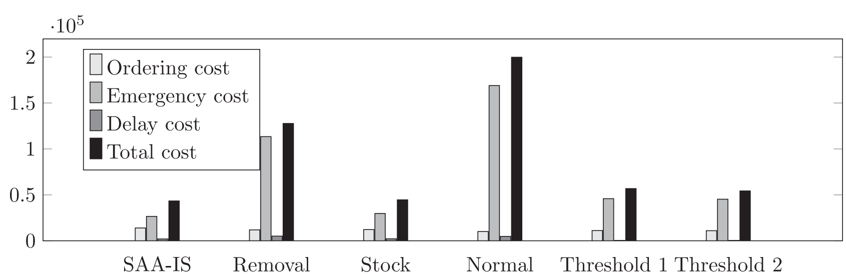

Fig. 4. Breakdown cost of each method (random directed acyclic project networks of 100 activities with 500,000 scenarios).

method is the textbook approach for project management in stochastic environments and may be adopted by practitioners that are unaware of the risks of using quasi-deterministic methods in highly stochastic environments.

The threshold heuristic 1 stocks spare parts only when the parts' failure probabilities are higher than the threshold. It is easy to implement. However, our experiments show that its cost is higher than the scenario based approach, e.g. SAA, SAA-IS, the stock or removal heuristics. The cost gap is around 30%, yet it costs less than the approach based on the normal distribution. As the threshold heuristic only considers the failure probability in decision making, it excludes the information such as the position of the activity in the project network, the impact of stock out on the finishing time and the cost of each spare part. The example in Appendix B demonstrates that parts with very low failure probabilities (0.005) still may need stocking. The threshold heuristic 2 stocks spare parts which contribute the most based on the failure probability, leadtime and cost. Our experiments show that though the threshold heuristic 2 has lower cost than threshold heuristic 1, it leads to large cost compared to the scenario based approach.

In general, the use of Proposition 2 can substantially reduce the scale of the problem. This is especially beneficial for the SAA, SAA-IS, removal and stock heuristics with a large number of scenarios.

For the randomly generated project networks, 41% of activities are dominated and their inventory decision is fixed a priori. For instances from library-MMLIB, 12% of all activities are dominated.

6. Discussion and conclusion

A good planning for shutdown maintenance is rewarding as delays are costly. We investigated the order policy of spare parts in the initial/preparation phase of the shutdown maintenance. In the shutdown project, the delay due to the shortage of spare part in each activity propagates along with the path in the project network, which may lead to delays Δ in the project, and hence costly production losses. We investigate the case where the demand probability of each part is small, which complicates the order decision. For a two-activity project, we completely characterized the optimal solution. For general networks, we developed a two-stage integer linear stochastic program to obtain the optimal order size. We found that for activities that are never on the critical path, the optimal solution can be expressed in closed form. In solving the problem, we proposed sample average approximation with importance sampling for the activities that can be on the critical path. We also proposed removal and stock heuristics for the problem, yet they are not faster nor more accurate. We found that scenario based heuristics give an acceptable approximation. The solution of the heuristic based on standard critical path with normal distributed project time assumption is far away from the optimum.

In the present paper, the duration of each activity is delayed by a fixed number of days if a spare part is needed and not readily available; this delay correspond to an emergency shipment. Emergency shipment leadtimes could also be stochastic, and our stochastic programming approach could incorporate that. Also, a variable base duration of activities could be straightforwardly included. Our scenario-based approach may also be useful for project planning under low-probability, high consequence risks in general, e.g. for infrastructure projects.

CRediT authorship contribution statement

Sha Zhu: Acquisition of data, Analysis and/or interpretation of data, Writing – original draft. **Willem van Jaarsveld:** Conception and design of study, Writing – original draft, Writing – review & editing. **Rommert Dekker:** Conception and design of study, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Approval of the version of the manuscript to be published (the names of all authors must be listed).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ress.2021.108197.

References

[1] Al-Turki Umar, Duffuaa Salih, Bendaya M. Trends in turnaround maintenance planning: literature review. J Qual Maint Eng 2019.
[2] Valmet. Ensure a successful shutdown through proper planning. 2021, URL https://www.valmet.com/Medias/north-america/paper-and-board-enews/The-importance-of-planning/ensure-a-successful-shutdown-through-proper-planning/.
[3] Zhu Sha, van Jaarsveld Willem, Dekker Rommert. Spare parts inventory control based on maintenance planning. Reliab Eng Syst Saf 2020;193:106600.
[4] Bevilacqua M, Ciarapica FE, Giacchetta G. Critical chain and risk analysis applied to high-risk industry maintenance: A case study. Int J Proj Manage 2009;27(4):419–32.
[5] Nahmias S. Production and operation analysis sixth edition. New York: McGraw Hill Education; 2013.
[6] Mummolo Giovanni. Measuring uncertainty and criticality in network planning by PERT-path technique. Int J Proj Manage 1997;15(6):377–87.
[7] Lau Amy Hing-Ling, Lau Hon-Shiang, Zhang Yue. A simple and logical alternative for making PERT time estimates. IIE Trans 1996;28(3):183–92.
[8] Pontrandolfo Pierpaolo. Project duration in stochastic networks by the PERT-path technique. Int J Proj Manage 2000;18(3):215–22.
[9] Abbasi Ghaleb Y, Mukattash Adnan M. Crashing PERT networks using mathematical programming. Int J Proj Manage 2001;19(3):181–8.
[10] Herroelen Willy, Leus Roel. Project scheduling under uncertainty: Survey and research potentials. European J Oper Res 2005;165(2):289–306.
[11] Megow Nicole, Möhring Rolf H, Schulz Jens. Decision support and optimization in shutdown and turnaround scheduling. INFORMS J Comput 2011;23(2):189–204.
[12] Valls Vicente, Laguna Manuel, Lino Pilar, Pérez Angeles, Quintanilla Sacramento. Project scheduling with stochastic activity interruptions. In: Project scheduling. Springer; 1999, p. 333–53.
[13] Laguna Manuel, Lino Pilar, Pérez Angeles, Quintanilla Sacramento, Valls Vicente. Minimizing weighted tardiness of jobs with stochastic interruptions in parallel machines. European J Oper Res 2000;127(2):444–57.
[14] van Jaarsveld Willem, Scheller-Wolf Alan. Optimization of industrial-scale assemble-to-order systems. INFORMS J Comput 2015;27(3):544–60.
[15] Vaughan Timothy S. Failure replacement and preventive maintenance spare parts ordering policy. European J Oper Res 2005;161(1):183–90.
[16] Kurz Julian. Capacity planning for a maintenance service provider with advanced information. European J Oper Res 2016;251(2):466–77.
[17] Sleptchenko Andrei, van der Heijden Matthieu C, van Harten Aart. Using repair priorities to reduce stock investment in spare part networks. European J Oper Res 2005;163(3):733–50.
[18] Van Jaarsveld Willem, Dekker Rommert. Spare parts stock control for redundant systems using reliability centered maintenance data. Reliab Eng Syst Saf 2011;96(11):1576–86.
[19] van der Heijden Mattieu C, Alvarez EM, Schutten Johannes MJ. Inventory reduction in spare part networks by selective throughput time reduction. Int J Prod Econ 2013;143(2):509–17.
[20] Rahimi-Ghahroodi Sajjad, Al Hanbali Ahmad, Zijm Willem HM, Timmer Judith B. Emergency supply contracts for a service provider with limited local resources. Reliab Eng Syst Saf 2019;189:445–60.
[21] Sleptchenko Andrei, van der Heijden Matthieu. Joint optimization of redundancy level and spare part inventories. Reliab Eng Syst Saf 2016;153:64–74.
[22] Godoy David R, Pascual Rodrigo, Knights Peter. Critical spare parts ordering decisions using conditional reliability and stochastic lead time. Reliab Eng Syst Saf 2013;119:199–206.
[23] de Jonge Bram, Teunter Ruud, Tinga Tiedo. The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance. Reliab Eng Syst Saf 2017;158:21–30.
[24] Poppe Joeri, Basten Rob JI, Boute Robert N, Lambrecht Marc R. Numerical study of inventory management under various maintenance policies. Reliab Eng Syst Saf 2017;168:262–73.
[25] Van Jaarsveld Willem, Dekker Rommert. Estimating obsolescence risk from demand data to enhance inventory control—A case study. Int J Prod Econ 2011;133(1):423–31.
[26] Birge John R, Louveaux Francois. Introduction to stochastic programming. Springer Science & Business Media; 2011.
[27] Mak Wai-Kei, Morton David P, Wood R Kevin. Monte Carlo bounding techniques for determining solution quality in stochastic programs. Oper Res Lett 1999;24(1–2):47–56.