

Kosztván, Zsolt T.; Pribojszki-Németh, Anikó; Szalkai, István

**Article**

## Hybrid multimode resource-constrained maintenance project scheduling problem

Operations Research Perspectives

**Provided in Cooperation with:**

Elsevier

*Suggested Citation:* Kosztván, Zsolt T.; Pribojszki-Németh, Anikó; Szalkai, István (2019) : Hybrid multimode resource-constrained maintenance project scheduling problem, Operations Research Perspectives, ISSN 2214-7160, Elsevier, Amsterdam, Vol. 6, pp. 1-13, <https://doi.org/10.1016/j.orp.2019.100129>

This Version is available at:

<https://hdl.handle.net/10419/246404>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by/4.0/>



ELSEVIER

Contents lists available at ScienceDirect

## Operations Research Perspectives

journal homepage: [www.elsevier.com/locate/orp](http://www.elsevier.com/locate/orp)

# Hybrid multimode resource-constrained maintenance project scheduling problem

Zsolt T. Kosztyán<sup>\*,a</sup>, Anikó Pribojszki-Németh<sup>a</sup>, István Szalkai<sup>b</sup>

<sup>a</sup> Department of Quantitative Methods, University of Pannonia, Egyetem str. 10. Hungary

<sup>b</sup> Department of Mathematics, University of Pannonia, Egyetem str. 10. Hungary

## ARTICLE INFO

## Keywords:

Maintenance management  
Project management  
Scheduling  
Agile/lean approach  
Flexibility

## ABSTRACT

Organizing maintenance tasks is a specific and important field of project and production management. Well-planned, properly scheduled, and effectively communicated maintenance tasks result in the accomplishment of more work that is performed more efficiently at a lower cost. In maintenance and production management, agile and lean approaches are becoming more frequently used, where preventive/predictive maintenance (PPM) requires flexible project plans to decide which system components (or equipment) should be maintained to achieve the target system reliability. Maintenance projects have been treated as fixed sequences of corrective/preventive tasks, while the agile/lean approach allows for and usually requires prioritizing tasks and reorganizing the project structure. The fixed structure of maintenance projects and traditional maintenance project scheduling algorithms are not considered representative of these agile properties. The paper models the preventive resource-constrained maintenance project scheduling problem and aims at optimizing its maintainability. A matrix-based algorithm is proposed to apply to both system and project structures that include both traditional and agile/lean project management approaches.

## 1. Introduction

Advanced maintenance management strategies such as reliability-centered maintenance (RCM) (see, e.g., Ref. [1]), risk-based maintenance management (RBM) [2] and total productive maintenance (TPM) [3] focus on system reliability or the risks of the system, instead of focusing only on the reliability of separate system components. Therefore, the scheduling of preventive maintenance tasks requires the consideration of overall system reliability. The latest maintenance management trends attempt to include both agile and lean approaches [4]. Therefore, the corrective/preventive maintenance tasks can be prioritized to maximize system reliability [5].

The challenge of managing and organizing maintenance tasks as maintenance projects can be summarized as follows. (1) Although maintenance tasks can be assigned to equipment components based on previous maintenance experience and previous maintenance task lists from machine books, and therefore, time, cost, and resource demands can be linked to corrective maintenance tasks, increased reliability can only be linked to the system components. Therefore, *when specifying corrective/preventive maintenance tasks, the increase in the reliability of the system component(s) as a result of corrective/preventive tasks should be estimated* [6]. (2) The structure of a system specifies the

system reliability, and it can be fundamentally different from the structure of the maintenance projects, which include (corrective/preventive) maintenance tasks. Therefore, *when optimizing a maintenance project, two types of structures, i.e., the system structure and the project structure, should be linked*. [7]. (3) *The lean and agile approach allows for the consideration of flexible maintenance projects* [5] to focus on maximizing system reliability and prioritizing tasks based on those tasks that increase system reliability the most when considering the time, cost, and resource constraints.

In this case, the traditional time-cost tradeoff and/or multimode methods should be extended to address the growth in component reliability and should be combined with the agile and lean approaches to enable prioritization of the corrective/maintenance tasks to maximize system reliability. To date, no exact or hybrid algorithms are available to support the agile/lean approach in maintenance management. Moreover, agile/lean approaches are more effective than traditional approaches [4,5]. Therefore, the algorithmic support of the agile/lean maintenance management approaches requires further study. The proposed hybrid algorithm specifies a maintenance project (the set of decided, to-be-realized maintenance tasks that form a predefined task list and propose a sequence of completion (henceforward, project structure)) with an exact method within expectedly polynomial

\* Corresponding author.

E-mail address: [kzst@gtk.uni-pannon.hu](mailto:kzst@gtk.uni-pannon.hu) (Z.T. Kosztyán).

<https://doi.org/10.1016/j.orp.2019.100129>

Received 27 March 2019; Received in revised form 26 June 2019; Accepted 13 November 2019

Available online 14 November 2019

2214-7160/ © 2019 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/4.0/>).

computation time. The proposed exact method for specifying the project structure is combined with a metaheuristic method to select an optimal completion mode.

The proposed method can not only be treated as a hybrid method but can also be considered a hybrid of the traditional (discrete) multimode resource constrained project scheduling problem (MRCPSP) and the agile/lean approach. The algorithmic background of agile and hybrid approaches is a rarely studied field, while the use of agile methods is common in the field of software projects. These methods are becoming increasingly popular in other areas, such as maintenance [4,5]. Kosztyán and Szalkai [8] showed how to combine two approaches, such as traditional continuous tradeoff methods, with flexible agile approaches; however, in the case of maintenance, we should select a completion mode from discrete technologies. Therefore, instead of using continuous trade-off methods, we should use the MRCPSP in combination with the agile approach. Since the MRCPSP is a generalization of the discrete tradeoff methods, which are usually NP-hard problems [9,10], an exact algorithm alone cannot be used to solve the problem. Nevertheless, because the selection of tasks and dependencies can be performed within expectedly polynomial computation time [11], we can propose a faster and better algorithm than a purely metaheuristic method, as detailed in Ref. [7]. In addition, the proposed method avoids the tradeoff assumption; therefore, different types of technology can be considered, where the time, cost and resource demands can vary.

To analyze the proposed method, first, it is compared to the existing metaheuristic method of Kosztyán [7], where tradeoffs between time-cost-resources and the increased reliability of the system component(s) are assumed. Simultaneously with the simulation techniques, discrete, multimode resource allocations are analyzed for maintaining different numbers of system components and for comparing traditional and agile/lean techniques.

The paper is organized as follows. Section 2 presents the main background of the study. In Section 3, the mathematical background is described. In Section 4, we compare the proposed method with previous methods, and in Section 5, we demonstrate the proposed method in practice. In Section 6, we interpret the simulation results, and in Section 7, we summarize the contributions to the existing literature, and we discuss the limitations of the proposed algorithm, suggesting future directions.

## 2. Literature review

The proposed method involves three areas of research. It combines two fields of scheduling approaches, namely, traditional and agile scheduling approaches, and also uses the preventive maintenance management techniques. Fig. 1 shows the Venn diagram of the classification of maintenance and scheduling problems.

In the case of maintenance management, two kinds of approaches, such as project management (to schedule corrective/preventive maintenance tasks) and (preventive) maintenance management (to prioritize tasks when considering the system and component reliability) should be integrated to improve the maintainability. Therefore, we focus on the intersection of scheduling and maintenance techniques (see (5–7), (9–12) fields in Fig. 1). Other fields are very shortly reviewed.

- (1) The first project scheduling method was published six decades ago [12]. This network-based technique assumes a fixed logic plan. Despite later techniques [13] that allowed for alternative subnetworks, these subnetworks required being specified before scheduling the tasks; therefore, these techniques are very scarcely used in flexible project management, such as agile and lean projects.
- (2) The importance of the time-cost tradeoff problem (TCTP) was recognized more than five decades ago, almost simultaneously with the development of project planning techniques [14,15]. However, while the continuous version of tradeoff problems

can be solved within polynomial computation time [16], in the general case of the discrete tradeoff problems, the resulting conditional time and cost minimization problems, except for several special network structures (see, e.g., [9]), are strongly NP-hard problems [10]. Tradeoff problems assume that shorter duration or better quality both require additional cost [17]; this assumption is true for all the discrete, continuous and stochastic versions of the tradeoff problem.

- (3) In contrast, in the case of multimode problems, an arbitrary set of technologies can be specified [18,19]. It is a better fit to the construction and maintenance projects, where different kinds of technology (i.e., completion modes) require different kinds of time/cost/resource demands.
- (4) Despite the large number of publications in traditional project management, the flexible project management approaches, such as agile and extreme project management, are very rarely studied [20]. The agile approaches currently prioritize the tasks and allow for flexible tasks. Instead of specifying trade-offs between time, quality and cost or multimode completion modes, the agile project scheduling algorithms, when necessary, postpone low-priority tasks to the next project (so-called sprint) and/or reorganize the project according the flexible dependencies [11].
- (5) The hybrid project management approach can combine agile and traditional techniques [21]. In contrast to the lack of algorithmic support, this approach is popular not only in software project management but also in maintenance management [4].
- (6) To our best knowledge, there is only one publication (see Ref. [8]) that combines traditional tradeoff methods and the flexible project scheduling.
- (7) At the same time, to our best knowledge, there is no published method that can combine both the MRCPSP and the agile project scheduling approaches.
- (8) The primary goal of preventive maintenance is to avoid or mitigate the consequences of equipment failure. This goal may be achieved by preventing failure before it actually occurs [22,23]. A piece of equipment can require one or more preventive tasks [24]. When calculating the increase in system reliability, we should model system reliability using a reliability block diagram (RBD) and consider the increase in equipment reliability due to maintenance tasks. Instead of component reliability, the risk effects, their dependencies [25] and their propagation [26] can also be modeled using a multiobjective hierarchical method [27].
- (9) At the same time, if more maintenance tasks need to be completed, they should be organized into a maintenance project [28] to control the time/resource/cost demands.
- (10–11) Traditional tradeoff methods [see, e.g., Refs. 29,30] and the MRCPSP [31] are already used in maintenance projects. Nevertheless, numerous scholars from the 1960s (see, e.g., Ref. [32]) to recent times (see, e.g., Refs. [28,33]) have assumed that maintenance projects are fixed, similar to construction projects. Moreover, in these studies, the different structures of the system from which the risk originates are not separated from the projects.
- (12–13) Recently applied agile and lean approaches allow for the prioritization of maintenance tasks [4]. Furthermore, determining how to specify an exact algorithm to specify which maintenance tasks and which type of technology (the so-called completion mode) should be selected to increase system reliability while maintaining deadlines and budgets remains an open question. Additionally, ascertaining how to identify the shortest/least expensive project possible to increase system reliability to the desired degree is also unclear. These questions are the basis of system maintainability.

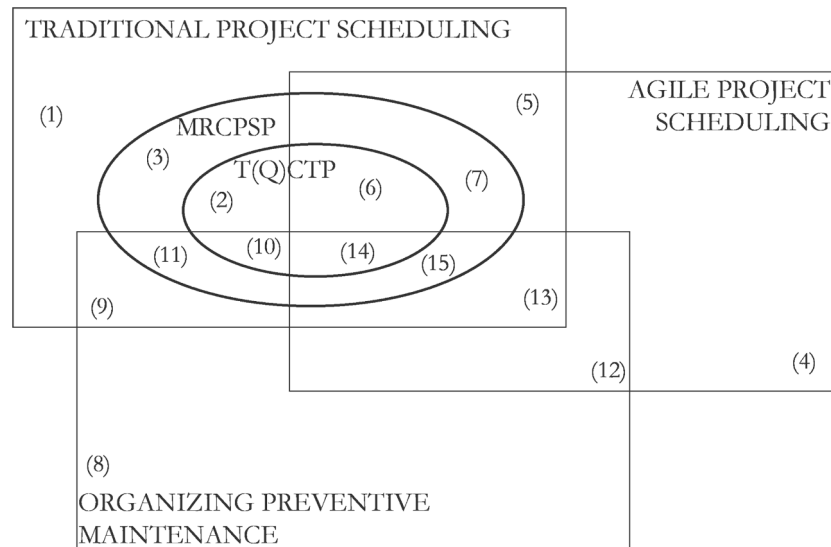


Fig. 1. Categorization of maintenance management and project scheduling techniques.

- (14) While a system structure can be considered static, an agile project structure can be flexible [11]. Component reliability should be linked to the system components, and other parameters of corrective/preventive maintenance, such as duration, cost and resource demands, should be assigned to the completion mode of the corrective/preventive maintenance task. Kosztyán [7] proposed this distinction but assumed tradeoffs instead of multimodes, which can be independent from each other. Furthermore, Kosztyán [7] lacks a principle foundation and algorithmic treatment. *The aim of this paper is to fill this gap.*
- (15) Since there is no publication that combines traditional MRCPSP and agile scheduling (see item (6)), there is consequently no publication that also takes into account the different structures of projects and the system when using hybrid (traditional and agile) project scheduling techniques. In contrast to Ref. [7], the proposed model does not assume any tradeoff function between time and cost, time and resources, or cost and resources. These completion modes, or in other words, technologies, can be different from each other. In contrast to traditional tradeoff and multimode methods, the proposed model does not assume a fixed maintenance project plan. However, corrective/preventive tasks must be performed if the reliability of a system component falls below a certain value.

The aim in specifying the proposed method is to improve the system maintainability, where *maintainability* has two main components: *serviceability* (ease of conducting scheduled inspections and servicing) and *repairability* (ease of restoring service after a failure) [34]. In the case of repairability, the aim is to increase system reliability as much as possible within time/cost constraints (referred to as problem 1 hereafter), while in the case of serviceability, the question is determining how to increase system reliability to the desired degree as fast as possible or as inexpensively as possible (referred to as problem 2a or problem 2b, respectively). To the best of our knowledge, no exact or hybrid method exists that can model and solve both repairability and serviceability and combine both the traditional multimode and recent agile/lean approaches.

In the proposed algorithm, several objective functions can be specified, and the corresponding objectives are as follows: (1) maximize system reliability within time and cost constraints (see repairability), and (2) achieve a desired level of system reliability (2a) with a minimal

budget or (2b) a minimal project duration (see serviceability).

We propose combining the MRCPSP method with agile project scheduling because the case of maintenance and alternative technologies can be considered, and in the general case, these relations cannot be assumed. Therefore, we should select from alternative technologies that can maximize system reliability by increasing component reliability while maintaining deadlines and budgets. This problem is an extension of the discrete time-cost tradeoff problem and is also NP-hard; therefore, the discrete TCTP should also be solved by heuristic or metaheuristic methods [35,36].

The proposed algorithm includes three *phases*. The first two phases propose exact selection algorithms, but at the end of the second phase, we have to solve a multimode problem, that is, a resource/cost/time constraint problem. Since the number of modes of maintenance tasks is usually very limited, the proposed algorithm uses ant colony optimization (ACO) [37].

ACO is a metaheuristic for solving difficult computational problems and was first introduced by Marco Dorigo in 1992 [38]. This approach mimics the behavior of ants in nature while searching for the shortest path between the ant nest and a source of food by applying pheromones to exchange messages. In the basic ACO algorithm (Ant System - AS), the ants choose the path where the pheromone concentration is the highest with probability  $p$ . Due to the behavior of the ants, the shortest path will have the highest pheromone concentration.

ACO is one of the fastest ways to solve discrete tradeoff [39] and multimode problems [37]. This method proved to be the fastest during our implementation. In contrast to Kosztyán's [7] method, the first two phases of the algorithm are exact algorithms, and the proposed ACO implementation produced better performance than the genetic algorithm or pattern searches (see Section 4). Therefore, we integrated the ACO approach with the proposed  $M^5$  algorithm. Additionally, we compared the genetic algorithms proposed by Kosztyán [7]. In contrast to Kosztyán's [7] method, we used metaheuristic methods only in the final phase. We show that our method can find the optimal solution if the final phase can produce a feasible resource allocation.

### 3. Mathematical formalization of the problem and the solution

Since the hybrid multimode resource constrained maintenance project scheduling problem (HMCMPSP) has not previously been specified, the *mathematical formalization* of the proposed problem statement, the matrix-based management model and the detailed steps of the algorithm are defined in the following subsections.

### 3.1. Definitions

**Definition 1.** Denote by  $K: = \{k_1, k_2, \dots, k_z\}$  the finite set of system components,  $r: K \rightarrow [0, 1]$ ,  $r(k_i) \in [0, 1]$  the reliability value of them, and  $TSR(K) \in [0, 1]$  is the (total) system reliability.

The maintenance tasks related to system component  $k_i (i = 1, \dots, z)$  are mandatory if  $r(k_i) < c_{r_i}$ , where  $c_{r_i}$  is a given minimal reliability (or critical) value for  $k_i$ , and other tasks are supplementary.

**Definition 2.** We call any finite set  $A = \{a_1, \dots, a_n\}$  the set of possible activities or tasks of the project. The subset of uncertain or supplementary tasks is  $\tilde{A} = \{\tilde{a}_1, \dots, \tilde{a}_\sigma\} \subseteq A$ , where  $\tilde{A}$  is any fixed subset of  $A$ . Then,  $\bar{A} = A \setminus \tilde{A}$  is the subset of mandatory tasks.

The set of (project) scenarios is  $\Xi(A): = \{S \subseteq A: \bar{A} \subseteq S\}$ .

Mandatory tasks must be performed; supplementary tasks can be omitted. In the decision process about uncertain task realization, we always have two options: include or exclude.  $S$  denotes the set of tasks that will be fulfilled by the algorithm.

**Definition 3.** The relation triplet  $(<, \sim, \bowtie)$  on  $A \times A$  is a relation representation of a hybrid project plan if for any  $a_i, a_j \in A$ ,  $i \neq j$ , we have the following:

- (i)  $a_i < a_j$  represents strict or required dependency between  $a_i$  and  $a_j$ , i.e.,  $a_j$  must not be started unless  $a_i$  has been completed,
- (ii)  $a_i \sim a_j$  represents no dependency between  $a_i$  and  $a_j$ , i.e., the starting time of  $a_j$  is not affected by  $a_i$ , and
- (iii)  $a_i \bowtie a_j$  represents uncertain or flexible dependency between  $a_i$  and  $a_j$ .

The flexible dependencies ( $\bowtie$ ) must be resolved (realized) by the algorithm: we can decide that these tasks will be completed either in a sequential ( $<$ ) or in a parallel ( $\sim$ ) manner.

**Remark 1.** Clearly,  $<$  is a partial order that excludes cycles such as  $a_1 < a_2 < a_3 < \dots < a_1$ . Therefore, by the standard topological ordering algorithm, we may assume that  $a_i < a_j \Rightarrow i < j$ .

**Definition 4.** A project structure or net is a deterministic logic plan, i.e., every flexible relation is realized. The triplet  $X = (S, <, \sim)$  represents the final project structure of a given project scenario  $S \subseteq A$ .

The easiest way to input and modify all the data above is the use of a special  $n \times n$  matrix  $\mathbf{M}$ , which we call a project plan. ( $\mathbf{M}_{i,j}$  denotes the entry in row  $i$  and column  $j$ .) The entries of  $\mathbf{M}$  are  $\emptyset, X, ?$  such that

- (i) for any  $i \leq n$ ,  $\mathbf{M}_{i,i} = "X"$  for  $a_i \in \bar{A}$  and  $\mathbf{M}_{i,i} = "?"$  for  $a_i \in \tilde{A}$ ,
- (ii) for any  $i, j \leq n$ ,  $i < j$ ,  $\mathbf{M}_{i,j} = "X" \Leftrightarrow a_i < a_j$ ,

$\mathbf{M}_{i,j} = "\emptyset" \Leftrightarrow a_i \sim a_j$  and  $\mathbf{M}_{i,j} = "?" \Leftrightarrow a_i \bowtie a_j$ .

By Remark 1, we know that in terms of the entries "X",  $\mathbf{M}$  is an upper triangular matrix both in the input and in all further steps of the algorithm.

**Definition 5.** Any finite set of quadruplets  $W = \{(t_i, c_i, \mathbf{R}_i, \Delta r_i): i = 1, \dots, m\}$  of nonnegative real numbers and the vector  $\mathbf{R}_i$  is called a multimode protocol (MMP).

Denote by  $t_{\min} = \min_i t_i$ ,  $t_{\max} = \max_i t_i$ ,  $\Delta r_{\min} = \min_i \Delta r_i$ ,  $\Delta r_{\max} = \max_i \Delta r_i$ ,  $c_{\min} = \min_i c_i$ ,  $c_{\max} = \max_i c_i$  the minimal/maximal values of the demands of the tasks. If for each  $a \in A$ , we are given a protocol  $W_a \subseteq W$ , then we call the set  $\mathcal{W} = \{W_a: a \in A\}$  a multimode maintenance management problem ( $\mathbf{M}^4\mathbf{P}$ ). For any  $S \in \Xi(A)$  and  $\mathcal{W}$ , a project schedule is a set

$$\vec{w} = \{(t^a, c^a, \mathbf{R}^a, \Delta r^a): a \in S\} \quad (1)$$

where  $(t^a, c^a, \mathbf{R}^a, \Delta r^a) = w_a \in W_a \in \mathcal{W}$  for each  $a \in S$ .

We interpret  $(t, c, \mathbf{R}, \Delta r) \in W_a$  as paying the cost  $c$  for solving the task  $a \in A$  in time  $t$  with the set of resources  $\mathbf{R}$  and with an increase in

(desired) reliability  $\Delta r$  using the mode assigned to  $(t, c, \mathbf{R}, \Delta r)$ .

The desired increase in the reliability of a system component depends on the maintenance tasks that are assigned to it. One or more tasks can be assigned to a system component, but a task can be assigned to only one system component.

**Definition 6.** Denote by  $\Delta r_k^a$  the maintenance activity, i.e., the increase in reliability of system component  $k \in K$  if  $a \in A$  is completed.  $\Delta r_k^a := \Delta r^a$  if  $a$  is assigned to  $k$ ; otherwise,  $\Delta r_k^a := 0$ . The total increase in reliability of the system component  $k \in K$  is  $\Delta r_k = \sum_{a \in S} \Delta r_k^a$ .

Now, we are ready to provide the upper and lower bounds for time, cost, resources and the increase in component reliability for each stage (in any PHASE) of the algorithm.

**Proposition 1. (i)** For any  $\mathbf{M}$  and  $\mathcal{W}$ , the minimal cost bound is

$$C_{\min}(\mathbf{M}, \mathcal{W}) = \sum_{\mathbf{M}_{i,i} = "X"} c_{\min}^a. \quad (2)$$

(ii) For any project schedule  $\vec{w}$ , the total project cost of  $\vec{w}$  is

$$c(\vec{w}) = \sum_{(t^a, c^a, \mathbf{R}^a, \Delta r^a) \in w_a} c^a. \quad (3)$$

For time bounds, we must consider the  $<$  dependencies.

**Definition 7.**

- (i)  $\vec{P} = (a_{i_1}, a_{i_2}, \dots, a_{i_\ell})$  is called a real path of length  $\ell \in \mathbb{N}$  ( $\ell \leq n$ ) if  $\mathbf{M}_{i_j, i_{j+1}} = "X"$  for  $1 \leq i_j < n$ ,  $j \leq \ell$  and  $a_{i_1} < a_{i_2} < \dots < a_{i_\ell}$ .

(ii)

$$T_{\min}(\vec{P}, \mathcal{W}) = \sum_{a \in \vec{P}} t_{\min}^a. \quad (5)$$

- (iii)  $\vec{P}$  is a longest min-path of  $\mathbf{M}$  if  $T_{\min}(\vec{P}, \mathcal{W})$  is maximal, assuming that  $\vec{P}$  contains only mandatory tasks (i.e.,  $\mathbf{M}_{i,i} = "X"$  whenever  $a_i \in \vec{P}$ ). Any such  $\vec{P}$  is called a critical path, and the set  $\{a_{i_1}, a_{i_2}, \dots, a_{i_\ell}\}$  is called a set of critical activities. We denote this maximum by the following:

$$T_{\min}(\mathbf{M}, \mathcal{W}) = \max_{\vec{P} \text{ real path}} T_{\min}(\vec{P}, \mathcal{W}). \quad (6)$$

A longest min-path in any  $\mathbf{M}$  can be found by a standard algorithm within  $O(n + d)$  time, where  $n$  and  $d$  are the numbers of tasks and dependencies, respectively.

**Proposition 2. (i)** For any  $\mathbf{M}, \mathcal{W}$  and  $K$ , the lower and upper bounds for the increase in reliability are

$$\Delta r_{\min}(\mathbf{M}, \mathcal{W}, K) = \sum_{a \in A, \mathbf{M}_{i,i} = "X"} \sum_{k \in K} \Delta r_k^a_{\min}, \quad (7)$$

$$\Delta r_{\max}(\mathbf{M}, \mathcal{W}, K) = \sum_{a \in A, \mathbf{M}_{i,i} \neq "0"} \sum_{k \in K} \Delta r_k^a_{\max}. \quad (8)$$

(ii) For any project schedule, the total increase in system reliability is

$$\Delta r(\vec{w}, K) = \sum_{(t^a, c^a, \mathbf{R}^a, \Delta r^a) \in w_a} \sum_{a \in S} \Delta r_k^a. \quad (9)$$

**Definition 8.** For any  $\mathbf{M}, \mathbf{N} \in \{X, \emptyset, ?\}^{n \times n}$ , we say that

- (i)  $\mathbf{N}$  is an in-/out-diagonal extension of  $\mathbf{M}$  if all the "X" and " $\emptyset$ "

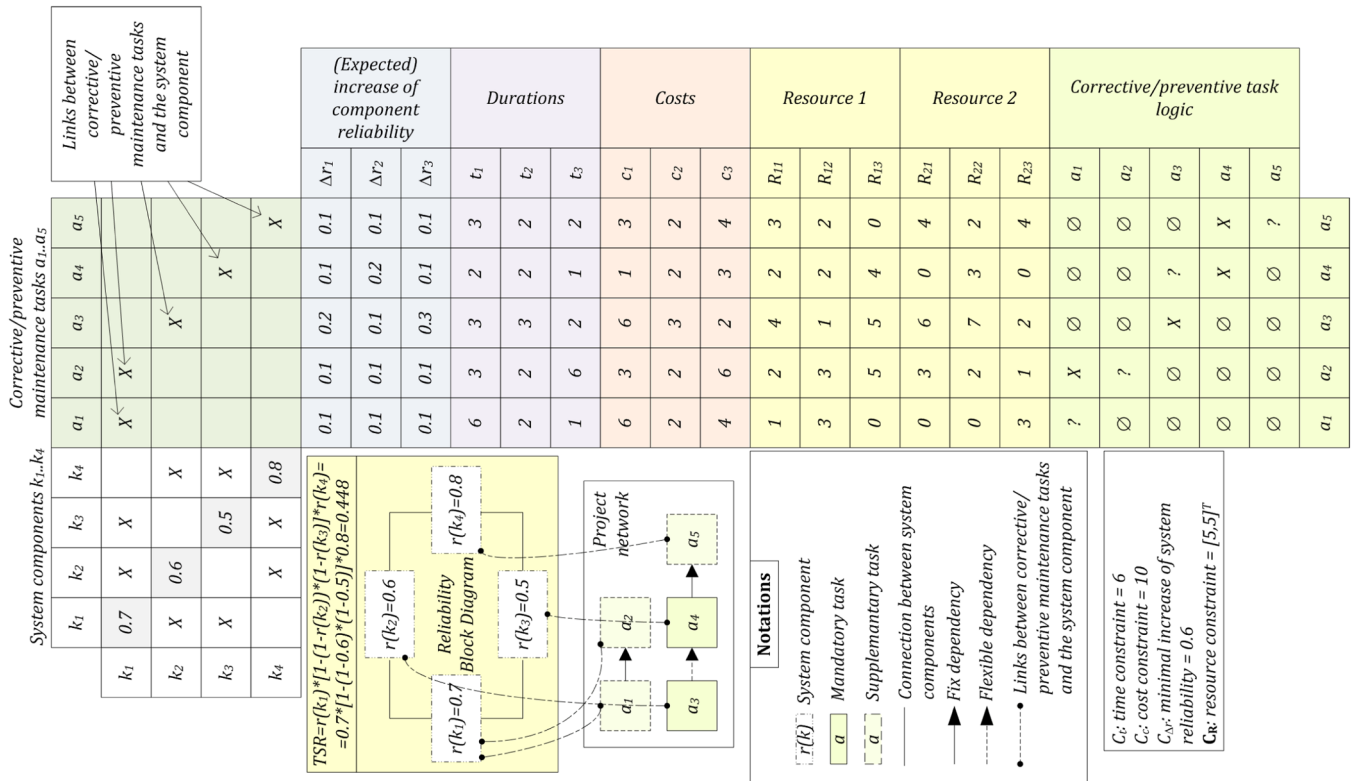


Fig. 2. Matrix representation of the hybrid multimode resource constrained maintenance project scheduling problem.

symbols in  $\mathbf{M}$  remain unchanged in  $\mathbf{N}$  and some (possibly none) “?” symbols on/off the diagonal of  $\mathbf{M}$  are changed to either “X” or “ $\emptyset$ ”. In this case,  $\mathbf{M}$  is a **restriction** of  $\mathbf{N}$ .

- (ii) For  $i, j \leq n$  and  $\mathbf{M}_{i,j} = \text{“?”}$ , we denote by  $\mathbf{M}[i, j = X]$  and  $\mathbf{M}[i, j = \emptyset]$  the matrices if only  $\mathbf{M}_{i,j}$  has been changed to either “X” or “ $\emptyset$ ”.
- (iii)  $\mathbf{N}$  is an **in-/out-diagonal closure** of  $\mathbf{M}$  if  $\mathbf{N}$  contains no “?” symbols on/off of the diagonal and  $\mathbf{N}$  in/out extends  $\mathbf{M}$ .

The next theorem easily follows from the above definitions.

- Theorem 1.** For any  $\mathbf{M}$ , any in- or out-closure  $\mathbf{N}$  of  $\mathbf{M}$  and  $\vec{w}$
- (10)  $C_{\min}(\mathbf{M}, \mathcal{W}) \leq C_{\min}(\mathbf{N}, \mathcal{W})$
  - (11)  $T_{\min}(\mathbf{M}, \mathcal{W}) \leq T_{\min}(\mathbf{N}, \mathcal{W})$
  - (12)  $\Delta r_{\max}(\mathbf{M}, \mathcal{W}, K) \geq \Delta r_{\max}(\mathbf{N}, \mathcal{W}, K)$
  - (13)  $C_{\min}(\mathbf{N}, \mathcal{W}) \leq \mathbf{c}(\vec{w})$
  - (14)  $T_{\min}(\mathbf{N}, \mathcal{W}) \leq \mathbf{t}(\vec{w})$
  - (15)  $\Delta r_{\max}(\mathbf{N}, \mathcal{W}, K) \geq \Delta r(\vec{w}, K)$
  - (16)  $T_{\min}(\mathbf{M}, \mathcal{W}) \leq T_{\max}(\mathbf{M}, \mathcal{W})$
  - (17)  $C_{\min}(\mathbf{M}, \mathcal{W}) \leq C_{\max}(\mathbf{M}, \mathcal{W})$
  - (18)  $\Delta r_{\min}(\mathbf{M}, \mathcal{W}, K) \leq \Delta r_{\max}(\mathbf{M}, \mathcal{W}, K)$

**Definition 9.** For  $\mathbf{M} \in \{X, \emptyset\}^{n \times n}$  and  $\vec{w}$ , we use the following notations for **total project cost** and **time**:  $TPC(\mathbf{M}, \vec{w}) = \mathbf{c}(\vec{w})$  and  $TPT(\mathbf{M}, \vec{w}) = \mathbf{t}(\vec{w})$ .

The **total increase in system reliability** is as follows:

$$\Delta TSR(\mathbf{M}, \vec{w}, K) = \Delta r(\vec{w}, K). \tag{19}$$

**Definition 10.** Denote by  $\mathbf{A}(\mathbf{M}, \mathcal{W}, \tau) = A(\vec{w}, \tau)$  the set of scheduled tasks that are in progress at time  $\tau$ , and let  $\mathbf{TPR}_k(\vec{w}, \tau) = \sum_{j \in A(\vec{w}, \tau)} R_{j,k}$

be the (total project) resource demands at time  $\tau$  for resource  $R_k$ . Denote the (total project) resource demands at time  $\tau$  by  $\mathbf{TPR}(\mathbf{M}, \mathcal{W}, \tau) = \mathbf{TPR}(\vec{w}, \tau)$  and the

$$\mathbf{TPR}(\mathbf{M}, \mathcal{W}) = \mathbf{TPR}(\vec{w}) = \max_{\tau} \mathbf{TPR}(\mathbf{M}, \mathcal{W}, \tau) = \mathbf{TPR}(\vec{w}, \tau)$$

total project time by

$$\mathbf{TPR}(\mathbf{M}, \mathcal{W}) = \mathbf{TPR}(\vec{w}) = \max_{\tau} \mathbf{TPR}_1(\vec{w}, \tau), \max_{\tau} \mathbf{TPR}_2(\vec{w}, \tau), \dots, \max_{\tau} \mathbf{TPR}_\rho(\vec{w}, \tau)$$

3.2. Problem statements

Now, we define the **hybrid multimode resource-constrained maintenance project scheduling problem** (HMCMPSP). In PHASE ONE, we have to find a feasible project scenario ( $S$ ). Since problems 1/1, 1/2a and 1/2b differ only in terms of the target functions, the considered problems’ formulations in PHASE ONE are combined. (In PHASE ONE we choose only one of these problems and solve it with our algorithm.)

**Problem 1.** PHASE ONE (1/1, 1/2a, 1/2b): let  $A$  be a finite set of activities (represented by  $\mathbf{M}$ ), let  $K$  be a finite set of system components, and let  $\mathcal{W}$  be the set of modes. Further, let  $C_c, C_t, C_{\Delta r} \in \mathbb{R}^+$  be given such that  $C_{\min}(\mathbf{M}, \mathcal{W}) \leq C_c, T_{\min}(\mathbf{M}, \mathcal{W}) \leq C_t$  and  $\Delta r_{\max}(\mathbf{M}, \mathcal{W}, K) \geq C_{\Delta r}$ .

We have to find an in-closure  $\mathbf{M}'$  of  $\mathbf{M}$  ( $S \subseteq A$ ) such that in problem 1/1

$$\Delta r(\mathbf{M}', \mathcal{W}, K) \rightarrow \max \tag{20}$$

in problem 1/2a

$$C(\mathbf{M}', \mathcal{W}) \rightarrow \min \tag{21}$$

in problem 1/2b

$$T(\mathbf{M}', \mathcal{W}) \rightarrow \min \tag{22}$$

assuming

$$C_{\min}(\mathbf{M}', \mathcal{W}) \leq C_c \tag{23}$$

$$T_{\min}(\mathbf{M}', \mathcal{W}) \leq C_t \tag{24}$$

$$\Delta r_{\max}(\mathbf{M}', \mathcal{W}, K) \geq C_{\Delta r}. \quad (25)$$

As mentioned in Section 2, several objective functions can be specified, and the corresponding problems are as follows:

(1) *Repairability*: maximize the system reliability within the time and cost constraints (see Fig. 3(a), which is a result of Fig. 2, if the target function is Eq. (20)), and (2) *Serviceability*: achieve the desired level of system reliability within (a) a *minimal budget* (see Fig. 3(b), which is a result of Fig. 2, if the target function is Eq. (21)) or (b) a *minimal project duration* (see Fig. 3(c), which is a result of Fig. 2, if the target function is Eq. (22)).

In PHASE TWO, we have to find a feasible project *structure*  $\mathcal{X} = (S, <, \sim)$ . Since the dependencies are selected to be excluded or included, the selection does not influence the budget and the increase in system reliability, such that PHASE TWO must be specified only for problem 2/2b.

**Problem 2.** PHASE TWO (2/2b): let  $\mathbf{M}'$ , i.e.,  $S \subseteq A$  be a solution of PHASE ONE. Further, let  $C_t \in \mathbb{R}^+$  be given such that  $T_{\min}(\mathbf{M}', \mathcal{W}) \leq C_t$ . Now, find a *structure*, i.e., an *off-closure*  $\mathbf{M}''$  of  $\mathbf{M}'$  such that

$$T(\mathbf{M}'', \mathcal{W}) \rightarrow \min \quad (26)$$

assuming

$$T_{\min}(\mathbf{M}'', \mathcal{W}) \leq C_t. \quad (27)$$

In PHASE THREE, we have to find a feasible *project schedule* ( $\vec{w}$ ). In this case, we have to select from the completion modes, which influence the budget, project duration, and increase in system reliability. In PHASE ONE and PHASE TWO, *resource constraints* were ignored because the resource allocation required to specify the minimal/maximal resource demands unnecessarily slows down the proposed algorithm. In PHASE THREE, the *resource demands* are considered. Since in PHASE THREE, only the target functions differ for problems 3/1, 3/2a and 3/2b, the formal descriptions for them are combined.

**Problem 3.** PHASE THREE (3/1, 3/2a, 3/2b): let  $\mathbf{M}'$  be a solution of PHASE ONE, and let  $\mathbf{M}''$  (i.e.  $\mathcal{X} = (S, <, \sim)$ ) be a solution of PHASE TWO (2b). Let  $K$  be a finite set of system components, and let  $\mathcal{W}$  be the set of modes. Further, let  $C_c, C_t, C_{R_1}, C_{R_2}, \dots, C_{R_p}, C_{\Delta r} \in \mathbb{R}^+$ ,  $\mathbf{C}_R = [C_{R_1}, C_{R_2}, \dots, C_{R_p}]^T$  be given such that  $C_{\min}(\mathbf{M}', \mathcal{W}) \leq C_c$ ,  $T_{\min}(\mathbf{M}'', \mathcal{W}) \leq C_t$ ,  $\mathbf{TPR}_{\min}(\mathbf{M}'', \mathcal{W}) \leq \mathbf{C}_R$  and  $\Delta R_{\max}(\mathbf{M}', \mathcal{W}, K) \geq C_{\Delta r}$ . Now, find  $\vec{w}$  such that in problem 3/1

$$\Delta r(\vec{w}, K) \rightarrow \max \quad (28)$$

in problem 3/2a

$$\mathbf{c}(\vec{w}) \rightarrow \min \quad (29)$$

in problem 3/2b

$$\mathbf{t}(\vec{w}) \rightarrow \min \quad (30)$$

assuming

$$\mathbf{t}(\vec{w}) \leq C_t \quad (31)$$

$$\mathbf{c}(\vec{w}) \leq C_c \quad (32)$$

$$\mathbf{TPR}(\vec{w}) \leq \mathbf{C}_R \quad (33)$$

$$\Delta r(\vec{w}, K) \geq C_{\Delta r}. \quad (34)$$

### 3.3. The matrix representation

Fig. 2 shows an example for the matrix representation of the

problem. We used a matrix-based method instead of network-planning techniques because this method allows us to model both the system and the project structure in one so-called multidomain matrix model [40]. An example of the matrix model is presented in Fig. 2, and the result is presented in Fig. 3. The first important part of the proposed matrix model is the system structure, which is modeled with an RBD. The second main part is the project domain matrix (PDM) [11], which describes a flexible project (see Fig. 2) and is detailed in Ref. [7].

**Example 1.** Consider 4 components of a piece of equipment, where the component reliabilities are as follows:  $r(k_1) = 0.7$ ,  $r(k_2) = 0.6$ ,  $r(k_3) = 0.5$ , and  $r(k_4) = 0.8$ . For the corrective/preventive tasks,  $a_1$  and  $a_2$  are assigned to system component  $k_1$ , and  $a_3, \dots, a_5$  are assigned to  $k_2, \dots, k_4$ , respectively. The increase in component reliabilities, time, cost, and resource demands is presented in domains in Fig. 2. The deadline is 6 days ( $C_t$ ), the budget is \$10k ( $C_c$ ), and the resource constraint is 5 repairpeople and 5 maintenance tools ( $C_R$ ). The minimal final system reliability is 0.6 ( $C_{\Delta r}$ ). The critical reliability ( $cr$ ) value of the system component is specified as 0.6.

The question is how to specify the maintenance projects, including corrective/preventive maintenance tasks and completion modes, if the system reliability has to be maximized or the budget/project durations have to be minimized, according to the constraints.

The first domain of the matrix representation describes the reliability diagram of the system, and this domain is symmetric. The diagonal represents the component reliability (or, in another model, the component availability). In the next domain, maintenance **tasks** are assigned to the system components. The estimation of the increase in component reliability is based on the completion mode of the tasks.

### 3.4. The algorithm

The algorithm changes the “?” symbols in the logic domain represented by matrix  $\mathbf{M}$  to either “X” or “ $\emptyset$ ” step-by-step in the diagonal in PHASE ONE, and in the off-diagonal in PHASE TWO. The matrix representation of a (final) project *structure* contains no “?” symbols.

#### 3.4.1. Phase one

Given the matrix  $\mathbf{M}_0 \in \{X, \emptyset, ?\}^{n \times n}$ , fix an ordering of the symbols “?” in the first  $\sigma$  rows (columns). The algorithm changes these symbols to either “X” or “ $\emptyset$ ” one-by-one in *this* order, and such a change is called a **step**. These changes are not final, and the original matrix  $\mathbf{M}_0$  is saved. We look for the optimum similar to the “back-and-forth” methods, saving information about the possible paths we might investigate later in the **buffer B** (a set). After replacing as many elements of  $\mathbf{M}$  as we can (satisfying Eq. (23) through (25)), we return to the cases in  $B$  that have<sup>1</sup> a greater increase in system reliability  $\Delta r(\mathbf{M}, \mathcal{W}, K) \#$  in lower budget  $C(\mathbf{M}, \mathcal{W}) \#$  in lower project duration  $T(\mathbf{M}, \mathcal{W})$  than  $\mathbf{M}$ .

Denote by  $\mathbf{M}$  the actual matrix *before* the next replacement; thus,  $\mathbf{M}_{i,j} = ? \Leftrightarrow i \leq j \leq \sigma$  and let  $\mathbf{M}[i, i = Y]$  be the matrix *after* replacing  $\mathbf{M}_{i,i}$  with  $Y$ , where  $Y \in \{X, \emptyset\}$ . *Before* replacing  $\mathbf{M}_{i,i}$ , we save the *other* possibility that we do not follow in the present step in  $B$ . The elements of  $B$  have the form

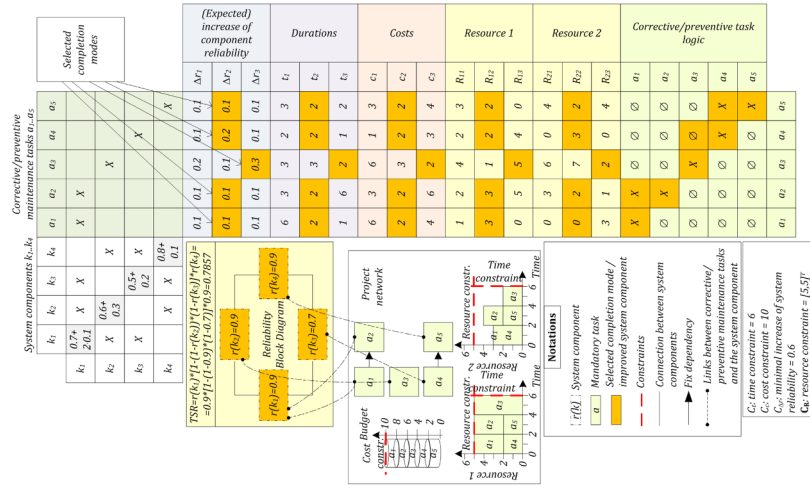
$$\mathbf{b} = (i, \mathbf{M}_{1..n,1..n} Y, \Delta r_{\max}(\mathbf{M}[i, i = Y], \mathcal{W}, K)) = (i, \vec{m}, Y, \Delta r_{\mathbf{b}}) \# \quad (35)$$

$$\mathbf{b} = (i, \mathbf{M}_{1..n,1..n} Y, C_{\min}(\mathbf{M}[i, i = Y], \mathcal{W})) = (i, \vec{m}, Y, C_{\mathbf{b}}), \# \quad (36)$$

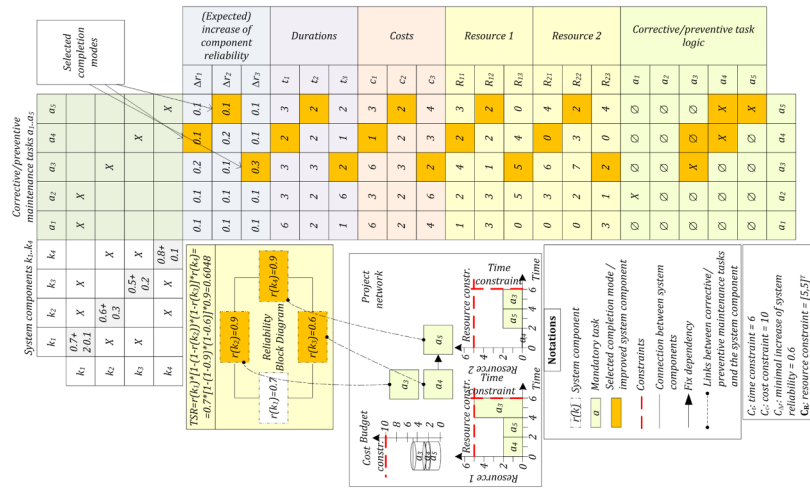
$$\mathbf{b} = (i, \mathbf{M}_{1..n,1..n} Y, T_{\min}(\mathbf{M}[i, i = Y], \mathcal{W})) = (i, \vec{m}, Y, T_{\mathbf{b}}). \quad (37)$$

**Remark 2.**  $B$  contains only those  $\mathbf{M}[i, i = Y]$  extensions that have not yet been investigated but fulfill the bounds in Eqs. (23) through (25):

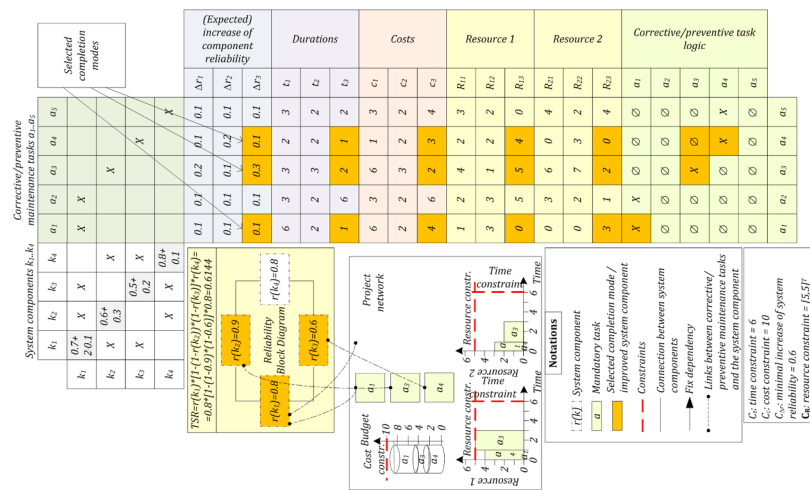
<sup>1</sup> From now, the phrases “aaa # bbb # ccc” abbreviate the texts “aaa (in Problem 1), bbb (in Problem 2a) and ccc (in Problem 2b)”.



(a) Result of problem 1-3/1 – repairability



(b) Result of problem 1-3/2a – Serviceability



(c) Result of problem 1-3/2b – Serviceability

Fig. 3. Results of maintenance projects.



$$C_{\min}(\mathbf{M}[i, i = Y], \mathcal{W}) \leq C_c \quad (38)$$

$$T_{\min}(\mathbf{M}[i, i = Y], \mathcal{W}) \leq C_t \quad (39)$$

$$C_{\Delta r} \leq \Delta r(\mathbf{M}[i, i = Y], \mathcal{W}, K). \quad (40)$$

**Remark 3.** Before starting step  $i$  for  $1 < i$ , at the end of step  $i - 1$ , we place  $Y$  into the  $i - 1$ -th entry of  $\mathbf{M}$ . Since we are now focusing on further extending this configuration,  $B$  does *not* contain the corresponding record  $\# \mathbf{b} = (i - 1, \vec{m}, Y, C_b) \# \mathbf{b} = (i - 1, \vec{m}, Y, T_b)$ .

The algorithm starts a **new cycle** whenever it returns to an element of  $B$  and begins to replace the "?" symbols from the  $i = i_0 + 1$ -th entry of the diagonal. PHASE ONE may have a solution if, in at least one cycle, we are able to step  $i$  up to  $\sigma$  (satisfying Eq. (23) through Eq. (25)). We store all the in-closures  $\mathbf{M}'$  of  $\mathbf{M}_0$  that are found and may be optimal solutions. If  $B$  contains an element  $\mathbf{b}$  with a greater increase in system reliability  $\#$  lower budget  $\#$  lower project duration than  $\mathbf{M}'$  has (i.e.,  $\Delta r_b > \Delta r(\mathbf{M}', \mathcal{W}, K) \# C_b < C(\mathbf{M}', \mathcal{W}) \# T_b < T(\mathbf{M}', \mathcal{W})$ ), then we start a new cycle from  $\mathbf{b}$ . During this cycle, *either*  $i$  cannot be increased to  $\sigma$  or  $\Delta r_b = \Delta r_{\max}(\mathbf{M}[i, i = Y], \mathcal{W}, K) \# C_b = C_{\min}(\mathbf{M}[i, i = Y], \mathcal{W}) \# T_b = T_{\min}(\mathbf{M}[i, i = Y], \mathcal{W})$  may decrease below  $\Delta r(\mathbf{M}, \mathcal{W}, K) \#$  may increase above  $C(\mathbf{M}, \mathcal{W}) \#$  may increase above  $T(\mathbf{M}, \mathcal{W})$  or we might obtain a solution better than  $\mathbf{M}'$ .

The *pseudocode* of the algorithm is as follows.

**START** Let  $i_0 := 0, i = 1, B := \emptyset$ .

**GENERAL STEP** ( $1 \leq i \leq \sigma, \mathbf{M}$  is the actual matrix,  $Y \in \{X, \emptyset\}$ ). Let

$$\mathbf{b}_Y^i := (i, \mathbf{M}_{1\dots n, 1\dots n}, Y, \Delta r_{\max}(\mathbf{M}[i, i = Y]), \mathcal{W}, K), \# \quad (41)$$

$$\mathbf{b}_Y^i := (i, \mathbf{M}_{1\dots n, 1\dots n}, Y, C_{\min}(\mathbf{M}[i, i = Y]), \mathcal{W}), \# \quad (42)$$

$$\mathbf{b}_Y^i := (i, \mathbf{M}_{1\dots n, 1\dots n}, Y, T_{\min}(\mathbf{M}[i, i = Y]), \mathcal{W}). \quad (43)$$

**CASE i)** None of  $\mathbf{b}_\emptyset^i$  and  $\mathbf{b}_X^i$  fulfill Eqs. (38)-(40) and  $B = \emptyset$ . Then, STOP since Problem 1 has no solution.

**CASE ii)** None of  $\mathbf{b}_\emptyset^i$  and  $\mathbf{b}_X^i$  fulfill Eqs. (38)-(40) but  $B \neq \emptyset$ . Now, choose any element  $\mathbf{b} \in B$  such that  $\Delta r_b$  is maximal (in  $B$ )  $\#$   $C_b$  is minimal (in  $B$ )  $\#$   $T_b$  is minimal (in  $B$ ). Then, reset the diagonal of  $\mathbf{M}$  according to  $\vec{m}$ , set  $i := j$ , delete  $\mathbf{b}$  from  $B$ , and go to the General Step.

**CASE iii)** Exactly *one* of  $\mathbf{b}_X^i$  and  $\mathbf{b}_\emptyset^i$  fulfills Eqs. (38)-(40), for example,  $\mathbf{b}_Y^i$ . Let  $\mathbf{M}_{i,i} := "Y"$ , and go to Step Increasing  $i$ .

**CASE iv)** Both  $\mathbf{b}_X^i$  and  $\mathbf{b}_\emptyset^i$  fulfill Eqs. (38)-(40).

If  $\Delta r_{\max}(\mathbf{M}[i, i = X], \mathcal{W}, K) \leq \Delta r_{\max}(\mathbf{M}[i, i = \emptyset], \mathcal{W}, K) \# C_{\min}(\mathbf{M}[i, i = X], \mathcal{W}) \geq C_{\min}(\mathbf{M}[i, i = \emptyset], \mathcal{W}) \# T_{\min}(\mathbf{M}[i, i = X], \mathcal{W}) \geq T_{\min}(\mathbf{M}[i, i = \emptyset], \mathcal{W})$  then let  $\mathbf{M}_{i,i} := "X"$ ,  $B := B \cup \{\mathbf{b}_X^i\}$ , and go to Step Increasing  $i$ .

If  $\Delta r_{\max}(\mathbf{M}[i, i = X], \mathcal{W}, K) > \Delta r_{\max}(\mathbf{M}[i, i = \emptyset], \mathcal{W}, K) \# C_{\min}(\mathbf{M}[i, i = X], \mathcal{W}) < C_{\min}(\mathbf{M}[i, i = \emptyset], \mathcal{W}) \# T_{\min}(\mathbf{M}[i, i = X], \mathcal{W}) < T_{\min}(\mathbf{M}[i, i = \emptyset], \mathcal{W})$  then let  $\mathbf{M}_{i,i} := "X"$ ,  $B := B \cup \{\mathbf{b}_X^i\}$ , and go to Step Increasing  $i$ .

**STEP INCREASING** (i) If  $i < \sigma$ , then let  $i := i + 1$ , and go to General Step. In the case  $i = \sigma$ , go to Check Step.

**CHECK STEP** ( $i = \sigma$ ) First, save the recent  $\mathbf{M}$  with its  $\Delta r(\mathbf{M}, \mathcal{W}, K) \# C(\mathbf{M}, \mathcal{W}) \# T(\mathbf{M}, \mathcal{W})$ . If  $B$  contains an element  $\mathbf{b} = (i, \vec{m}, Y, T_b)$  such that

$$\Delta r_b > \Delta r(\mathbf{M}, \mathcal{W}, K) \# \quad (44)$$

$$C_b < C(\mathbf{M}, \mathcal{W}) \# \quad (45)$$

$$T_b < T(\mathbf{M}, \mathcal{W}) \quad (46)$$

then return to  $\mathbf{b}$ , and start a new cycle, i.e., reset the diagonal of  $\mathbf{M}$  according to  $\vec{m}$ , set  $i := j$ , delete  $\mathbf{b}$  from  $B$ , and go to General Step.

**END of the Algorithm.**

**Theorem 2.** The saved matrices (in the Check Step) of the above algorithm are exactly the optimal solutions of Problem 1. No saved matrices exist if and only if Eqs. (23) through (25) in Problem 1 PHASE ONE have no solution.

**Proof.** In each step, the algorithm chooses the better of (at most) two

possibilities and buffers the other for possible further investigation. The value  $T(\mathbf{M}, \mathcal{W})$  for each  $\mathbf{M}$  is a sharp *upper bound* for each further continuation of  $\mathbf{M}$ . Therefore, all the buffered possibilities with smaller  $T$  than that of the finished (and saved) matrices (in Check Step) can be deleted from the buffer. Since the algorithm checks *each* of the remaining elements of the buffer (see Check Step), at the end, we must obtain each optimal solution.  $\square$

### 3.4.2. Phase two

Since PHASE TWO has no effect on the project budget and the system reliability, it reduces the project duration. In the general case, when the constraint of the parallel completion affects resource availability, the algorithm and most of the notation for PHASE TWO are the same as those for PHASE ONE. We have to consider the "?" symbols *out* of the diagonal of  $\mathbf{M}$  in a fixed (but arbitrary) order. *Before* each replacement, we save the other possibility in a buffer, similarly to Eq. (37), and check the conditions corresponding to Eq. (27), such as Eq. (38)-(40) corresponding to Eq. (23)-(25). In each step, we have to refresh  $T_{\min}(\mathbf{M}, \mathcal{W})$ . The properties of the algorithm in PHASE TWO can be proven similarly to Theorem 2.

We consider resources and resource constraints in PHASE THREE. Therefore, if we obtain a feasible project scenario, we obtain the best one if all flexible dependencies are omitted.

### 3.4.3. Phase three

The final project structure specifies a discrete multimode resource constrained maintenance project scheduling problem (MRCMPSP). If no feasible solution of the given MRCMPSP algorithm exists, we return to PHASE ONE and select the next project scenario (and project structure) from the buffer.

The proposed method uses Li's [37] ACO method in this phase to find a feasible resource allocation. We compared the results with other metaheuristics, such as the hybrid genetic algorithm [41], but ACO found feasible solutions faster. Since the original target functions were either to maximize system reliability (in problem 1) or minimize the budget (in problem 2a) or project duration (in problem 2b), the feasible resource allocation is optimal for the original target functions.

**Example 2.** Fig. 3 show the results of Example 1. Fig. 3 shows that while the target is to increase the system reliability (see Fig. 3(a)), more tasks are included in the maintenance projects than if the goal is to keep serviceability with a minimal budget (see Fig. 3(b)) or with minimal project duration (see Fig. 3(c)). Comparing Fig. 3(b-c) also shows that different sets of tasks can perform the minimal increase of system reliability. Moreover, different kinds of tasks may be included in the maintenance project if the target is the minimal duration or the minimal budget.

### 3.5. Complexity of the problem

Kosztyán[11] has shown that the number of possible project structures is  $2^d$ , where  $d$  is the number of flexible dependencies. If uncertain task completion exists, the number of possible project scenarios is  $2^u$ , where  $0 \leq u \leq n$ . A project scenario has  $0 \leq 2^d \leq n(n-1)/2$  project structures. Therefore, the number of possible project structures is  $\mathcal{N} = \sum_{i=1}^u 2^d$ . If every task can be completed via  $m$  possible modes, the total number of possible project structures is  $\mathcal{M} = \mathcal{N}n^m = n^m \sum_{i=1}^u 2^d$ . Therefore, brute force algorithms cannot be applied, even in small problems. For example, if only *one* project scenario exists, all dependencies are flexible and there are only 3 completion modes ( $n := 8, d := n(n-1)/2 = 28, m := 3$ ); then,  $\mathcal{M} = 2^{28*82} = 4, 294, 967, 296$ .

In PHASE ONE and PHASE TWO, the decision tree is a binary heap. Every node in this tree has sharp bounds for the time and cost demands and for the increase in system reliability. In this case, the expected computation time is  $O(u + d)$  (see Ref. [11]). Since the output of PHASE TWO

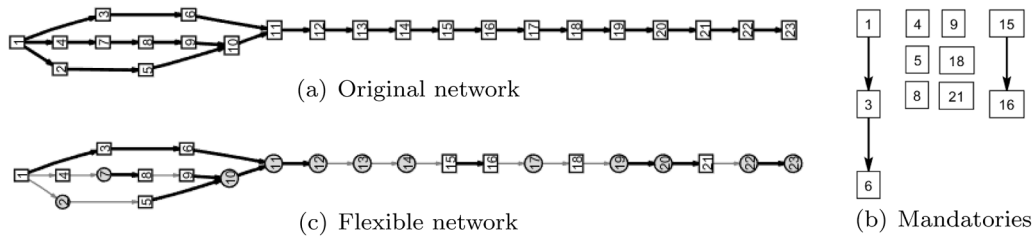


Fig. 4. Production/maintenance (C2014-02) project from the empirical project database [43].

is an MRCPSP, in the final phase, we can use heuristic or exact methods to evaluate the specified project structure. Unfortunately, in this case, the decision tree is not a binary heap but an  $m$ -heap, where  $m$  is the number of modes. Therefore, no fast tree traversal algorithm exists for finding the best project schedule within polynomial time. Since few alternatives usually exist in practice, in the final phase, we implement ACO in accordance with Ref. [37] to determine the project schedule.

#### 4. Simulations

The aim of this section is threefold. First, the proposed algorithm and a recent genetic algorithmic-based model (see Ref. [7]) are compared (see the comparison results in Section 4.3), where tradeoffs among time, cost, growth of component reliability and resources can be assumed and the target function is to minimize project duration (see problem 1-3/2b, the problem of serviceability). Since the proposed method can also minimize the budget while satisfying the constraints (see problem 1-3/2a) and maximize the increase in system reliability (see problem 1-3/1) without the assumption of time, cost, component reliability, and resource trade-offs, the results of different targets are compared.

Therefore, the second goal of the simulation is to compare the impacts of the project and structural parameters or the maintainability (for both serviceability and repairability). The third goal is to compare the effectiveness of traditional long-term shutdowns, where (almost) all system components are checked and/or improved, and novel continuous maintenance approaches, which involve fewer maintenance tasks that can be completed in parallel (see the results in Section 5).

##### 4.1. Specified parameters for generating project plans

In Section 3, we show that the maximal cost demand of a maintenance project occurs if every maintenance task is completed with the maximal task costs ( $C_{max}$ ). Similarly, the maximal time demand ( $T_{max}$ ) results if every maintenance task is completed and every possible relation between tasks is realized. Similarly, the maximal increase in system reliability ( $\Delta r_{max}$ ) occurs if every system component is maintained. The maximal resource demand ( $TPR_{max}$ ) can be calculated if every mandatory and supplementary task is realized and all the dependencies are ignored for the parallel run. These values are used to specify the time, cost, resource and reliability constraints.

We use  $T\%, C\%, R\% \in \{50\%, 60\%, \dots, 100\%\}$  and  $\Delta r\% \in \{50\%, 60\%, \dots, 90\%\}$  to specify the time, cost, resource and reliability constraints (e.g.,  $C_t := T\% \cdot T_{max}$ ). These parameters' (rate of demands) are called the *project parameters*. The project parameters represent the traditional project triangle (time, cost, and quality, i.e., the increase in reliability).

The other group of parameters is the structural parameters, namely, the number of blocks ( $n \in \{25, 50, 75\}$ ) and the ratio of flexible tasks ( $F\% \in \{50\%, 60\%, \dots, 90\%\}$ ). A higher ratio of flexible tasks represents continuous preventive maintenance, while a lower ratio models planned system shutdowns.

The ratio of serial blocks was  $S\% \in \{50\%, 60\%, \dots, 100\%\}$ , where a

higher serial ratio represents a (serial) production line and a lower ratio of serial blocks models a more reliable power plant.

While the goal is either to find the maximal increase in reliability of system components (repairability, see problem 1-3/1) or the minimal project cost (serviceability, see problem 1-3/2a) or project duration (serviceability, see problem 1-3/2b) within the project (time/cost/resource/quality = increase in system reliability) constraints, the dependent variable is the (key) performance of optimization,  $K_T\% = 1 - \frac{T - T_{min}}{T_{max} - T_{min}} \in [0, 1]$ ,  $K_C\% = 1 - \frac{C - C_{min}}{C_{max} - C_{min}} \in [0, 1]$ ,  $K_{\Delta r}\% = \frac{\Delta r - \Delta r_{min}}{\Delta r_{max} - \Delta r_{min}} \in [0, 1]$ , where  $\Delta r$  is the increase in system reliability,  $C$  is the project cost, and  $T$  is the project duration of the optimal solution. A value of 1 indicates that we can find the maximal increase in system reliability/minimal project cost/minimal project duration within the project constraints. This value represents the effectiveness of maintenance.

##### 4.2. Applied datasets

Project plans are generated by the free project generator ProGen [42], and one production/maintenance project is from the empirical project database of Batselier et al. (2015) [43] (see Fig. 4(a)). The structure of the maintenance projects is similar to that of the production project [44], and in the case of shutdowns, there are more serial than parallel tasks in a maintenance project. Additionally, continuous maintenance approaches, such as TPM and the novel agile approaches, prefer shorter maintenance plans that contain fewer, parallel corrective maintenance tasks. Therefore, most of the dependencies are treated as flexible (see Fig. 4(c)), and the system components must be maintained only when the reliability is under a specified (criticality) threshold (see Fig. 4(c)).

The ratio of nonmandatory (noncritical) or supplementary tasks is denoted as  $F\%$ . A low  $F\%$  value represents long-term shutdowns, while a larger value represents continuous maintenance and agile/lean approaches.

In terms of the possible values of  $T\%$ ,  $C\%$ ,  $R\%$ ,  $\Delta r\%$ ,  $F\%$ ,  $n$ , and  $S\%$ , there are 291,600 possible settings, and we generate 20 system structures for each. Therefore, 5,832,000 maintenance projects must be solved.

##### 4.3. Comparison of maintenance project scheduling methods

In the first step, before showing the results of the proposed algorithm, the time/cost/resource constraints and increase in component reliabilities are sorted to satisfy the assumptions of trade-off methods, and in this way, we can compare the proposed method to the meta-heuristic solver proposed by Kosztyán [7] (see the comparison in Fig. 5). Unfortunately, because of the large scale of possible project structures, the optimal solution cannot be calculated; nevertheless, if there are no constraints,  $K_T$  is 1. Thus, we can choose the shortest project structure. While satisfying the constraints,  $K_T\% = 1$  can usually be approached but cannot be reached. Therefore, a higher  $K_T\%$  indicates a better value for a feasible project.

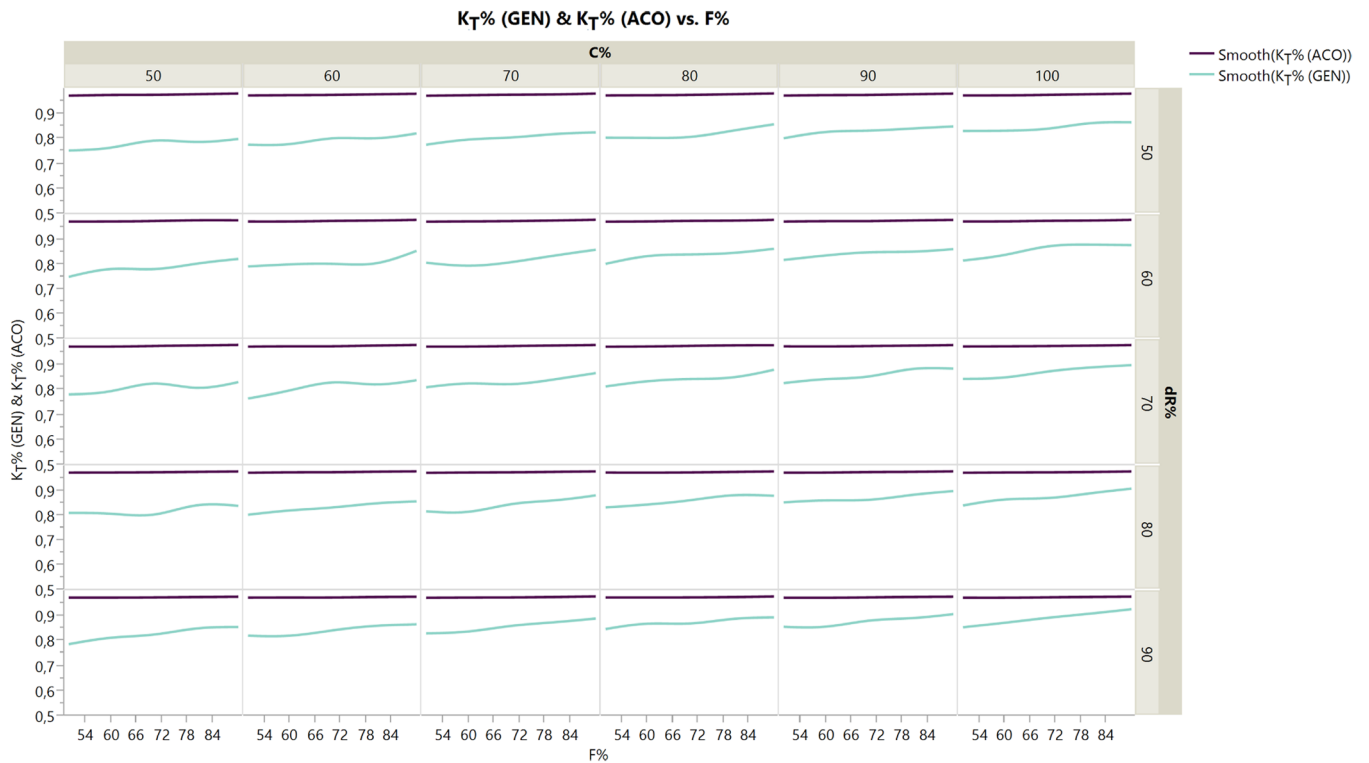


Fig. 5. Comparison of the results of the proposed algorithm (ACO) and the genetic algorithm proposed by Ref. [7]

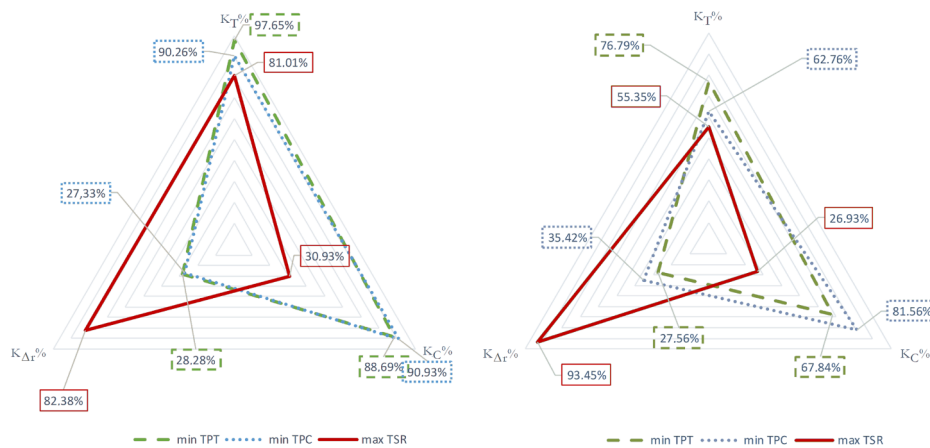
5. Results of the proposed algorithm

The tradeoff assumption is ignored when analyzing the results of the proposed method. The performance of the maintainability methods is compared when different target functions are applied and flexible tasks and flexible dependencies are allowed (see Fig. 6(a)) and when flexibility is not allowed (every corrective/preventive task in the maintenance project is decided) (see Fig. 6(b)). Fig. 6(b) shows the results of a long-term system shutdown, while Fig. 6(a) shows the result of the agile/lean approaches, where to increase system reliability/decrease

the budget/decrease the project duration, flexible (nonmandatory) tasks and dependencies can be excluded from the project.

In addition, to assess the computational time of the proposed algorithm, we investigate how the parameters/constraints influence the maintainability performance ( $K_{\Delta r}\%$ ,  $K_C\%$ ,  $K_T\%$ ).

Table 1 shows the parameter estimates of the simulation results and the impacts of the two parameter types: project constraints ( $T\%$ ,  $C\%$ ,  $R\%$ , and  $\Delta r\%$ ) and structural parameters ( $F\%$ ,  $n$ , and  $S\%$ ). The positive/negative standardized beta coefficients indicate the same/opposite direction for the performance in terms of maintainability. The LogWorth



(a) Flexible tasks/dependencies are allowed (b) Flexibility is not allowed

Fig. 6. Results of performance in terms of repairability  $K_{\Delta r}\%$  and serviceability  $K_C\%$ ,  $K_T\%$ .

**Table 1**  
Parameter estimation (\* significance level is lower than 0.05).

Term	max <i>TSR</i>		min <i>TPC</i>		min <i>TPT</i>	
	Std Beta	LogWorth	Std Beta	LogWorth	Std Beta	LogWorth
<i>T</i> %	0.008920	0.733	0.001046	0.072	0.010830*	1.712
<i>C</i> %	0.004110	0.266	0.008070	0.858	0.001426	0.120
<i>R</i> %	0.003540	0.314	0.007073	0.821	0.001426	0.121
$\Delta$ <i>r</i> %	-0.345566*	532.362	-0.277820*	525.017	-0.069100*	49.209
<i>F</i> %	0.130943*	82.668	0.207190*	302.221	0.248071*	574.562
<i>n</i>	-0.237960*	263.220	-0.169880*	206.439	0.758171*	3432.190
<i>S</i> %	-0.271180*	337.780	-0.608220*	2007.786	-0.104340*	109.803
Adj. <i>R</i> <sup>2</sup>	0.6543		0.5863		0.5545	
Independent	<i>K<sub>A</sub></i> %		<i>K<sub>C</sub></i> %		<i>K<sub>T</sub></i> %	

for each model effect, defined as  $-\log_{10}(p\text{-value})$ , can be treated as an importance value for the regression. The positive significant coefficients of time, cost and resource constraints indicate that if we relax these constraints (increase the constraint), the maintainability performance will also increase, and the negative values of the coefficient of the increase in the system reliability constraint indicate that if the minimal value of the increase in system reliability increases, the performance of the maintainability will decrease. The positive (standardized beta) coefficients of the flexibility parameters indicate that more flexible projects can improve system reliability while improving the performance in terms of serviceability. The negative coefficients of the number of system components (*n*) mean that the performance of maintainability is decreased if the number of components is increased. The negative coefficients of the ratio of serial components show that the performance of the maintainability decreases if the ratio of serial components increases.

## 6. Discussion

Few algorithms can be applied to both system and project structures that include both traditional and agile project management approaches to schedule maintenance projects in an algorithmic manner. Kosztyán [7] proposes a solution for only serviceability and only when the tradeoff assumptions are satisfied; however, we can obtain significantly better solutions if we combine the exact method and ant colony optimization (see Fig. 5).

The proposed algorithm can be used for both maximizing the increase in system reliability and minimizing time/cost demands while satisfying the constraints. Fig. 6 shows that when optimizing the repairability and serviceability, we obtain different types of projects. However, Fig. 6 also shows that better serviceability performance is obtained if flexible projects (flexible tasks and flexible dependencies) are allowed. However, better performance in terms of repairability is achieved when following the traditional maintenance management approaches.

Table 1 shows that the structural parameters are more important than the project constraints. Since *n* did not exceed 75 elements, the analysis of the positive coefficient of block size (*n*) is not essential. Solving the HPMPSP in 75 blocks demands less than a minute on a Pentium Core i5 computer; therefore, the size effect will be analyzed in more detail in a future study.

Since a higher flexible ratio (*F*%) represents continuous maintenance, the positive coefficient of *F*% indicates that *continuous maintenance can be more effective*.

Clearly, more parallel blocks (fewer serial blocks) in a system results in higher system reliability; however, considering the negative coefficient of *S*%, we obtain a novel result. The effectiveness of maintenance

is higher in a more reliable system, where *S*% is lower.

The negative coefficient of the increase in reliability parameter ( $\Delta r$ %) is trivial. If we increase the minimal improvement of system reliability, the distance between the feasible project duration and the ideal minimal project duration will increase.

## 7. Conclusion

In this paper, a novel problem, namely, the *hybrid multimode resource-constrained maintenance project scheduling problem* is defined. This problem integrates system reliability and preventive maintenance tasks. The proposed method can improve both the repairability and the serviceability. It can organize a maintenance project to minimize the maintenance project tasks while maintaining a specified budget and achieving the desired increase in system reliability (serviceability). At the same time, it can maximize the reliability of the system when keeping time/cost/resource constraints (repairability). After applying the proposed matrix-based maintenance management method, the set of maintenance tasks (the result of phase 1), their sequence of completion (the result of phase 2) and the time/cost/growth of system components (the result of phase 3) can be specified.

Since not all the system components will be maintained in a preventive maintenance project scheduling problem, the problem specifies a flexible project that consists of not only mandatory but also supplementary tasks. The simulation test showed that a more reliable system can be effectively maintained and that continuous maintenance is more effective.

The developed matrix-based method and the proposed exact algorithm may be important and essential components of a maintenance project expert system. This method can be applied in a computer-aided maintenance system (CMMS), where reliability/availability values are generated from a diagnostic system, but this method can schedule the maintenance tasks.

The proposed algorithm is a static algorithm; however, the system reliability and the reliability of system components decrease as a function of time. By considering the time dependence and integrating the proposed model with forecast methods, this model can be extended to support predictive maintenance, and a new *predictive maintenance project scheduling problem* can be specified.

## Acknowledgments

This research is supported by a János Bolyai Fellowship, Hungary Academy of Science and by the Research Centre at Faculty of Business and Economics (No. PE-GTK -GSKK A095000000-1) of University of Pannonia (Veszprém, Hungary).

## Appendix A. Nomenclature

Table 2.

Abbr.	Description
$A, \bar{A}, \bar{A}$	set of tasks, set of supplementary tasks, set of mandatory tasks
ACO	ant colony optimization
$c_{r_i}$	critical reliability value of system component $k_i$
$C_b, C_c, C_R, C_{\Delta r}$	time, cost, resource, increase in reliability constraint
$F\%$	rate of flexible tasks and dependencies
GA	genetic algorithm
(H)MRC(M)PSP	(hybrid) multimode resource-constrained (maintenance) project scheduling problem
$K$	finite set of system components
$K\%$	(key) performance of optimization
$k_i$	$i$ -th system component
$M$	matrix representation of project plan
MMp	multimode protocol
$M^4P$	multimode maintenance management problem
NP	nondeterministic polynomial
PPM	preventive & predictive maintenance
$r(k_i)$	reliability value of the system component $k_i$
$\rho$	number of resources
RBD	reliability block diagram
RBM, RCM, TPM	risk-based maintenance, reliability-centered maintenance, total productive maintenance
$S, \bar{X}, \bar{W}$	project scenario, project structure, project schedule
$S\%$	ratio of serial blocks
$t_b, c_i, R_b, \Delta r_i$	time, cost, resource demands and increase in reliability value of tasks $i$
$T\%, C\%, R\%, \Delta r\%$	rate of demands of time, cost, resource, increase in reliability
(TP)C	total project cost
(TP)R	(maximum values of) total project resources
(TP)T	total project time
T(Q)CTP	time-(quality)-cost tradeoff problem
TSR	total system reliability
$E(A)$	set of project scenarios
$<, \sim, \bowtie$	strict dependency, flexible dependency, no dependency between tasks

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at [10.1016/j.orp.2019.100129](https://doi.org/10.1016/j.orp.2019.100129).

## References

- [1] Moubraj J. *Reliability-Centered Maintenance*. second ed. Industrial Press, Inc.; 1997. 2 Revised edition
- [2] Khan FI, Haddara MM. Risk-based maintenance (RBM): a quantitative approach for maintenance/inspection scheduling and planning. *J. Loss Prev. Process Ind.* 2003;16(6):561–73. <https://doi.org/10.1016/j.jlp.2003.08.011>.
- [3] Wireman T. *Total Productive Maintenance*. Industrial Press; 2004.
- [4] Ghobakhloo M, Azar A. Business excellence via advanced manufacturing technology and lean-agile manufacturing. *J. Manuf. Technol. Manag.* 2018;29(1):2–24.
- [5] Packianather MS, Soman S, Davies A, White J. Predictive maintenance in a manufacturing environment through fit manufacturing and discrete event simulation. 2018 World Automation Congress (WAC). 2018. p. 1–6. <https://doi.org/10.23919/WAC.2018.8430400>.
- [6] Samrout M, Châtelet E, Kouta R, Chebbo N. Optimization of maintenance policy using the proportional hazard model. *Reliabil. Eng. Syst. Saf.* 2009;94(1):44–52.
- [7] Kosztyán ZT. Serviceability of large-scale systems. *Simul. Modell. Pract. Theory* 2018;84:222–31. <https://doi.org/10.1016/j.simpat.2018.03.002>.
- [8] Kosztyán ZT, Szalkai I. Hybrid time-quality-cost trade-off problems. *Oper. Res. Perspect.* 2018;5:306–18. <https://doi.org/10.1016/j.orp.2018.09.003>.
- [9] De P, Dunne EJ, Ghosh JB, Wells CE. The discrete time-cost tradeoff problem revisited. *Eur. J. Oper. Res.* 1995;81(2):225–38. [https://doi.org/10.1016/0377-2217\(94\)00187-H](https://doi.org/10.1016/0377-2217(94)00187-H).
- [10] De P, Dunne EJ, Ghosh JB, Wells CE. Complexity of the discrete time-cost tradeoff problem for project networks. *Oper. Res.* 1997;45(2):302–6. <https://doi.org/10.1287/opre.45.2.302>.
- [11] Kosztyán ZT. Exact algorithm for matrix-based project planning problems. *Expert Syst. Appl.* 2015;42(9):4460–73. <https://doi.org/10.1016/j.eswa.2015.01.066>.
- [12] Kelley Jr JE, Walker MR. Critical-path planning and scheduling. Papers Presented at the December 1–3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference. IRE-AIEE-ACM '59 (Eastern). New York, NY, USA: ACM; 1959. p. 160–73. <https://doi.org/10.1145/1460299.1460318>.
- [13] Pritsker AAB. GERT: Graphical Evaluation and Review Technique. *Rand Corporation*; 1966.
- [14] Fulkerson DR. A network flow computation for project cost curves. *Manage Sci.* 1961;7(2):167–78.
- [15] Elmaghraby SE. *Activity Networks: Project Planning and Control by network Models*. New York: Wiley; 1977.
- [16] Ahuja RK, Magnanti TL, Orlin JB. Some recent advances in network flows. *SIAM Rev.* 1993;33(2):175–219. <https://doi.org/10.1137/1033048>.
- [17] Babu A, Suresh N. Project management with time, cost, and quality considerations. *Eur. J. Oper. Res.* 1996;88(2):320–7. [https://doi.org/10.1016/0377-2217\(94\)00202-9](https://doi.org/10.1016/0377-2217(94)00202-9).
- [18] Said SS, Haouari M. A hybrid simulation-optimization approach for the robust discrete time/cost trade-off problem. *Appl. Math. Comput.* 2015;259:628–36. <https://doi.org/10.1016/j.amc.2015.02.092>. <http://www.sciencedirect.com/science/article/pii/S0096300315002933>
- [19] Chen WN, Zhang J, Chung HSH, Huang RZ, Liu O. Optimizing discounted cash flows in project scheduling - an ant colony optimization approach. *IEEE Trans. Syst. Man Cybern. Part C* 2010;40(1):64–77. <https://doi.org/10.1109/TSMCC.2009.2027335>.
- [20] Pellerin R, Perrier N. A review of methods, techniques and tools for project planning and control. *Int. J. Prod. Res.* 2019;57(7):2160–78. <https://doi.org/10.1080/00207543.2018.1524168>.

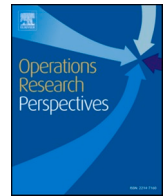
- [21] Tyagi M, Munisamy S, Reddy L. Traditional and hybrid software project tracking technique formulation: state space approach with initial state uncertainty. *CSI Trans. ICT* 2014;2(2):141–51. <https://doi.org/10.1007/s40012-014-0037-5>.
- [22] Gertsbakh I. *Reliability Theory - With Applications to Preventive Maintenance*. Berlin: Springer; 2000.
- [23] Yang L, sheng Ye Z, Lee C-G, fen Yang S, Peng R. A two-phase preventive maintenance policy considering imperfect repair and postponed replacement. *Eur. J. Oper. Res.* 2019;274(3):966–77. <https://doi.org/10.1016/j.ejor.2018.10.049>.
- [24] Idhammar C. *Preventive Maintenance, Essential Care and Condition Monitoring Book*. IDCON inc; 1999.
- [25] Fang C, Marle F. A simulation-based risk network model for decision support in project risk management. *Decis. Support Syst.* 2012;52(3):635–44. <https://doi.org/10.1016/j.dss.2011.10.021>. <http://www.sciencedirect.com/science/article/pii/S0167923611002016>
- [26] Ben-David I, Raz T. An integrated approach for risk response development in project planning. *J. Oper. Res. Soc.* 2001;52(1):14–25. <https://doi.org/10.1057/palgrave.jors.2601029>.
- [27] Wu D, Li J, Xia T, Bao C, Zhao Y, Dai Q. A multiobjective optimization method considering process risk correlation for project risk response planning. *Inf. Sci.* 2018;467:282–95. <https://doi.org/10.1016/j.ins.2018.07.013>. <http://www.sciencedirect.com/science/article/pii/S0020025518305279>
- [28] Nurdiana A, Wibowo MA. Risk management application at performance based contracts based on project life cycle using AHP method (case study at road maintenance project in Indonesia). *Adv. Sci. Lett.* 2018;24(12):9537–9.
- [29] El-Rayes K, Kandil A. Time-cost-quality trade-off analysis for highway construction. *J. Constr. Eng. Manag.* 2005;131(4):477–86. [https://doi.org/10.1061/\(ASCE\)0733-9364\(2005\)131:4\(477\)](https://doi.org/10.1061/(ASCE)0733-9364(2005)131:4(477)).
- [30] Mungle S, Benyoucef L, Son Y-J, Tiwari M. A fuzzy clustering-based genetic algorithm approach for tim-cost-quality trade-off problems: a case study of highway construction project. *Eng. Appl. Artif. Intell.* 2013;26(8):1953–66. <https://doi.org/10.1016/j.engappai.2013.05.006>.
- [31] Tiwari V, Patterson JH, Mabert VA. Scheduling projects with heterogeneous resources to meet time and quality objectives. *Eur. J. Oper. Res.* 2009;193(3):780–90. <https://doi.org/10.1016/j.ejor.2007.11.005>. <http://www.sciencedirect.com/science/article/pii/S0377221707010831>
- [32] Flehinger B. System reliability as a function of system age; effects of intermittent component usage and periodic maintenance. *Oper. Res.* 1960;8(1):30–44. <https://doi.org/10.1287/opre.8.1.30>.
- [33] Gopalakrishnan M, Bokrantz J, Ylipää T, Skoogha A. Planning of maintenance activities - a current state mapping in industry. 7th Industrial Product-Service Systems Conference - PSS, Industry Transformation for Sustainability and Business. 30. 2015. p. 480–5. <https://doi.org/10.1016/j.procir.2015.02.093>.
- [34] Schmidt S, Schmidt BS. Maintainability, reliability and serviceability—industrial examples automotive industry. *Editura Politehnica* 2018 2018:87.
- [35] Pour NS, Modarres M, Aryanejad M, Moghadam RT. The discrete time-cost-quality trade-off problem using a novel hybrid genetic algorithm. *App. Math. Sci.* 2010;4(42):2081–94.
- [36] Afruzi EN, Najafi AA, Roghanian E, Mazinani M. A multi-objective imperialist competitive algorithm for solving discrete time, cost and quality trade-off problems with mode-identity and resource-constrained situations. *Comput. Oper. Res.* 2014;50:80–96. <https://doi.org/10.1016/j.cor.2014.04.003>.
- [37] Li H, Zhang H. Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints. *Autom. Constr.* 2013;35:431–8. <https://doi.org/10.1016/j.autcon.2013.05.030>.
- [38] Dorigo M. *Optimization, Learning, and Natural Algorithms*. Dipartimento di Elettronica, Politecnico di Milano Italy; 1992. Ph.d. Dissertation (in italian).
- [39] Afshar A, Kaveh A, Shoghli O. Multi-objective optimization of time-cost-quality using multi-colony ant algorithm. *Asian J. Civil Eng.* 2007;8(2):113–24.
- [40] Danilovic M, Browning TR. Managing complex product development projects with design structure matrices and domain mapping matrices. *Int. J. Project Manag.* 2007;25(3):300–14. <https://doi.org/10.1016/j.ijproman.2006.11.003>.
- [41] Vartouni AM, Khanli LM. A hybrid genetic algorithm and fuzzy set applied to multi-mode resource-constrained project scheduling problem. *J. Intell. Fuzzy Syst.* 2014;26(3):1103–12.
- [42] Kolisch R, Sprecher A. PSPLIB - a project scheduling problem library: OR software - ORSEP operations research software exchange program. *Eur. J. Oper. Res.* 1997;96(1):205–16. [https://doi.org/10.1016/S0377-2217\(96\)00170-1](https://doi.org/10.1016/S0377-2217(96)00170-1).
- [43] Batselier J, Vanhoucke M. Construction and evaluation framework for a real-life project database. *Int. J. Project Manag.* 2015;33(3):697–710. <https://doi.org/10.1016/j.ijproman.2014.09.004>.
- [44] Ben-Daya M, Ait-Kadi D, Duffuaa SO, Knezevic J, Raouf A. *Handbook of Maintenance Management and Engineering*. vol. 7. Springer; 2009.

**Update**

**Operations Research Perspectives**

Volume 7, Issue , 2020, Page

DOI: <https://doi.org/10.1016/j.orp.2020.100169>



## Erratum regarding missing Declaration of Competing Interest statements in previously published articles

Declaration of Competing Interest statements were not included in the published version of the following articles that appeared in previous issues of Operations Research Perspectives.

The appropriate Declaration/Competing Interest statements, provided by the Authors, are included below:

- 1 "An integrated network design and scheduling problem for network recovery and emergency response" Operations Research Perspectives, 2018; Volume 5 (218–231) 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. <https://doi.org/10.1016/j.orp.2018.08.001>.
- 2 "Integrated approach for computing aggregation weights in cross-efficiency evaluation" Operations Research Perspectives, 2018; Volume 5 (256–264) 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. <https://doi.org/10.1016/j.orp.2018.08.005>.
- 3 "Robust optimization: Sensitivity to uncertainty in scalar and vector cases, with applications" Operations Research Perspectives, 2018; Volume 5 (113–119).
- 4 "Modeling the rational behavior of individuals on an e-commerce system" Operations Research Perspectives, 2018; Volume 5 (22–31).
- 5 "On slowdown variance as a measure of fairness" Operations Research Perspectives, 2018; Volume 5 (33–144) 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. <https://doi.org/10.1016/j.orp.2018.05.001>.
- 6 "Integrating pricing and capacity decisions in car rental: A matheuristic approach" Operations Research Perspectives, 2018; Volume 5 (334–356).
- 7 "Business analytics in manufacturing: Current trends, challenges and pathway to market leadership" Operations Research Perspectives, 2019; Volume 6 (100,127).
- 8 "Construction of currency portfolios by means of an optimized investment strategy" Operations Research Perspectives, 2018; Volume 5 (32–44).
- 9 "Hybrid time-quality-cost trade-off problems" Operations Research Perspectives, 2018; Volume 5 (306–318).
- 10 "Hybrid multimode resource-constrained maintenance project scheduling problem" Operations Research Perspectives, 2019; Volume 6 (100,129).
- 11 "Decision rules for robotic mobile fulfillment systems" Operations Research Perspectives, 2019; Volume 6 (100,128).
- 12 "Impact of TQM on organisational performance: The case of Indian manufacturing and service industry" Operations Research Perspectives, 2018; Volume 5 (199–217).
- 13 "Analysis of a multi-component system with failure dependency, N-policy and vacations" Operations Research Perspectives, 2018; Volume 5 (191–198).

DOIs of original article: <https://doi.org/10.1016/j.orp.2018.08.005>, <https://doi.org/10.1016/j.orp.2019.100127>, <https://doi.org/10.1016/j.orp.2018.07.004>, <https://doi.org/10.1016/j.orp.2018.07.005>, <https://doi.org/10.1016/j.orp.2018.10.002>, <https://doi.org/10.1016/j.orp.2019.100129>, <https://doi.org/10.1016/j.orp.2017.12.001>, <https://doi.org/10.1016/j.orp.2018.01.001>, <https://doi.org/10.1016/j.orp.2018.05.001>, <https://doi.org/10.1016/j.orp.2019.100128>, <https://doi.org/10.1016/j.orp.2018.08.001>, <https://doi.org/10.1016/j.orp.2018.03.001>, <https://doi.org/10.1016/j.orp.2018.09.003>.

<https://doi.org/10.1016/j.orp.2020.100169>

Available online 4 November 2020

2214-7160/© 2020 Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).