Reliability Engineering and System Safety 175 (2018) 234–245

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Optimal selective maintenance decisions for large serial k-out-of-n: G systems under imperfect maintenance

Claver Diallo[a,*], Uday Venkatadri[a], Abdelhakim Khatab[b], Zhuojun Liu[a]

[a] Dalhousie University, 5269 Morris Street Halifax, Nova Scotia B3H-4R2, Canada
[b] Laboratory of Industrial Engineering, Production and Maintenance, Lorraine University, National School of Engineering, Metz, France

ARTICLE INFO

Keywords:
Selective maintenance problem
k-out-of-n systems
Imperfect maintenance
Multidimensional multiple-choice knapsack problem

ABSTRACT

The selective maintenance problem (SMP) arises in many large multicomponent systems which are operated for consecutive missions interspersed with finite breaks during which only a selected set of component repairs or replacements can be carried out due to limited time, budget, or resources. The problem is to decide which components and degree of repairs should be performed in order to guarantee a pre-specified performance level during the subsequent mission. Current SMP formulations in the literature are nonlinear, deal mainly with basic or series-parallel systems and mostly use heuristic methods to obtain solutions.

This paper introduces the first SMP model for serial k-out-of-n systems. Two nonlinear formulations are developed, which can be used to solve the problem for small to moderate size k-out-of-n systems. For large k-out-of-n systems or complex reliability structures, we develop a new two-phase approach which transforms the problem into a multidimensional multiple-choice knapsack problem (MMKP). The new approach is shown to be efficient through multiple sets of numerical experiments.

1. Introduction

The maintenance function has proven to be beneficial for production and service operations by reducing downtime and energy consumption, improving reliability and availability of systems, and extending the equipment lifetimes. Maintenance engineering and optimization has therefore been the subject of many studies. The abundant research literature on the topic covers multiple types of maintenance and attempts with each new paper published to model real practical conditions and constraints. A relatively recent development deals with the selective maintenance problem (SMP) first proposed by Rice et al. [41]. SMP aims at identifying a subset of maintenance actions to be performed on a subset of components from a multicomponent system during a limited window of opportunity for maintenance arising between consecutive missions. SMP covers a very wide range of applications such as military, naval and aerospace equipment, production and manufacturing lines, mining and energy production systems.

The goal of any SMP model is usually to maximize the reliability of the system under consideration during the mission following the repair period (also called intermission break) without exceeding the total budget available. A variant formulation is to minimize the total maintenance cost that guarantees a minimum required reliability level during the subsequent mission. The total duration of the maintenance actions is required to be equal or less than the length of the intermission break.

Summaries of recent SMP developments can be found in [5,17,27,49]. Xu et al. [49] present a literature review on SMP. They review 70 manuscripts on SMP published between 1998 and 2014. They classify the manuscripts based on the problem type (basic, multi-state, multi-mission, fleet-level) and solution methodology (enumeration, heuristic). They then discuss the shortcomings of the literature and highlights future research avenues. One key shortcoming identified by [49] is that all models have considered very basic reliability structures or series-parallel systems. More complex structures such as k-out-of-n are missing or “ignored, intentionally or unintentionally” to quote them. Furthermore, a recent search in leading reliability journals such as Reliability Engineering & System Safety and IEEE Transactions on Reliability, and in Academic Databases such as Compendex, Proquest and Google Scholar found no paper dealing with SMP models for k-out-of-n systems. Thus, to the best of our knowledge, our paper is the first to do so.

The goal of this paper is two-fold: to develop the first SMP formulations to deal with k-out-of-n:G systems and complex reliability structures, and to provide an efficient method to solve the resulting nonlinear optimization models.

* Corresponding author.
E-mail addresses: claver.diallo@dal.ca (C. Diallo), uday.venkatadri@dal.ca (U. Venkatadri), abdelhakim.khatab@univ-lorraine.fr (A. Khatab), zh441868@dal.ca (Z. Liu).

https://doi.org/10.1016/j.ress.2018.03.023
Received 3 November 2017; Received in revised form 9 March 2018; Accepted 20 March 2018
Available online 22 March 2018
0951-8320/© 2018 Elsevier Ltd. All rights reserved.

1.1. Literature review

The basic original SMP model introduced by [41] has been extended in several directions to include components with Weibull lifetime distributions [7], non-identical components [8], imperfect maintenance levels [23,30,38,52], multi-state system reliability [9,12,30,37], structural and economic dependencies [13,14,48,51], global failure propagation and isolation of some components [33], fleet-level SMP [44,45], and a joint SMP and multiple repair-persons assignment problem [17].

Cassady et al. [7] studied the SMP in a series-parallel system, where the components have Weibull distributed lifetimes. Each component is subject of three potential maintenance actions: minimal repair, corrective replacement of a failed component, and preventive replacement of a working/functioning component. Liu and Huang [30] incorporate imperfect maintenance into SMP using the age reduction coefficient approach [35]. An imperfect selective maintenance model was also developed by Zhu et al. [52] and applied to a machining line system. Pandey et al. [38] studied the SMP for binary systems under imperfect maintenance using the hybrid hazard rate approach. In [30] and [38], components are subject to potential maintenance levels ranging from minimal repair to replacement. Zhao and Zeng [50] proposed an SMP model in the multi-mission case and where the break duration is exponentially distributed. Maillart et al. [34] formulated the finite-horizon, and infinite-horizon multi-mission SMP as stochastic dynamic programs. They found that these policies rarely differ and that difference in long-run performance is minimal. Pandey et al. [39] developed a maintenance scheduling model under imperfect maintenance for the finite planning horizon SMP using the hybrid imperfect maintenance model. Recently, Khatab et al. [23] studied the SMP when the quality of the imperfect maintenance actions is stochastic. A nonlinear and stochastic optimization problem was proposed and solved for a series-parallel system. Khatab et al. [27] extended the SMP to deal with the case where mission and break durations are stochastic with known distributions. Liu et al. [31] proposed a sequence planning for selective maintenance of multi-state systems under stochastic maintenance durations. They show that sequencing of the maintenance actions can significantly affect the achieved reliability when the break duration is uncertain. Schneider et al. [44,45] studied the SMP for a fleet of independent and identical systems. The fleet operates a set of sequential missions and returns to a common base where maintenance actions may be performed on some selected components. A nonlinear cost-optimization model and its linearized version are provided and solved. An optimal load distribution model for multi-state systems under SMP was proposed by Chen and Huang [10]. The goal is to increase system reliability by optimally distributing the loads among components subjected to selective maintenance. Dao et al. [15] also studied the SMP for multi-state series systems working in variable loading conditions in the next mission. They proposed a load-dependent degradation model for multi-state components.

Finally, Diallo et al. [17] were the first to propose a model to jointly make selective maintenance decisions and assign maintenance actions to multiple repairpersons.

One common factor among all papers reviewed above is that they all consider multicomponent systems arranged in a series-parallel structure. Since system reliability is a nonlinear function, the formulations proposed above are nonlinear which makes finding optimal solutions for real applications computationally expensive. Furthermore, Rice [42] showed that the SMP is $\mathcal{NP}$-hard. Therefore, several papers have investigated efficient ways for solving the SMP. These solution methods can be grouped in two categories: exact and heuristics methods.

Exact solution methods include full enumeration algorithms, techniques that can reduce the search space through early elimination of uninteresting solutions or depth-first search algorithms, and branch-and-bound (B&B) type procedures. The large majority of papers dealing with SMP use full enumeration method for very small problems [7,8,41]. Full enumeration as a solution method become rapidly cumbersome when the number of the system components increases [7]. To deal with the combinatorial complexity arising from large size systems, four improved enumeration procedures are proposed in Rajagopalan and Cassady [40] to reduce computation times. They show that the best enumeration scheme allows them to improve solution time by up to 99% when the problem size increases by 200% for a series-parallel system with constant failure rate components. An exact method based on the branch-and-bound (B&B) procedure and a Tabu search based algorithm are proposed in Lust et al. [32] to solve the SMP problem for the series-parallel problem in Cassady et al. [7]. They report that their B&B method begins to be computationally intensive when the number of components reaches 20 and they have to resort to the Tabu search heuristic. Cao et al. [5] proposed a Depth-first Branch and Bound (DB&B) method to reduce the number of combinations enumerated. They apply their method to solve the 4-component example with imperfect maintenance dealt with in Pandey et al. [38]. They show a 43% reduction in solution space compared to the enumeration method.

The second category of solution methods includes heuristic and other approximate methods used to find near-optimal solutions [32,38]. Khatab et al. [24] proposed two heuristic methods, adapted from those used to solve the redundancy allocation problem. Genetic algorithms [12,25,30,52] are also used as solution approaches for large size instances of the SMP. Zhao and Zeng [50] used a hybrid intelligent optimization algorithm based on empirical rules to solve their SMP. Liu et al. [31] use an ant colony optimization (ACO) algorithm to solve the constrained combinational optimization problem resulting from their SMP formulation. Sharma et al. [46] used a combination of simulation and genetic algorithm to optimize spare parts forecasting and selective maintenance decisions. Cao et al. [4] used a simulation approach to maximize system availability for an SMP.

1.2. Motivation and contributions of the paper

The main common shortcoming of current SMP models is that their formulations are nonlinear and difficult to solve optimally. Furthermore, the complexity of such formulations limits the application of these models to series-parallel systems and are not able to deal with more general structures such as k-out-of-n systems or more complex reliability structures. The k-out-of-n systems generalizes the series–parallel systems currently used in the SMP setting. The k-out-of-n:G system, also denoted as GA(k, n), consists of n components and operates only when at least k components operate. Two particular reliability structures are derived from the GA(k, n) when k is equal to 1 or n. The GA(1,n) corresponds to a parallel system with n components, while a GA(n, n) system refers to a n-component series system. Examples of GA(k, n) systems include multi-pump systems in hydroelectric plants, servers in a computer networks, multi-display systems in a cockpit, and a multi-engine aircraft that can continue to operate as long as at least two engines are working/functioning [3,29]. These systems have been well-covered in the literature on reliability theory [1,26,28,29].

In this paper, we first propose two new nonlinear formulations of the SMP for serial k-out-of-n systems. We also propose a new two-phase approach for finding optimal solutions using a binary integer programming (BIP) model. A similar two phase approach was proposed by Cassady et al. [6] to optimally solve the reliability redundancy allocation problem (RAP). In their case, they transformed the RAP into a multiple choice knapsack problem (MCKP). In our work, the first phase of the proposed method is a pre-processing phase with a guided generation of all feasible combinations of components and maintenance actions called maintenance patterns. A reliability value is computed for each maintenance pattern as well as its resulting cost and duration and fed into the next phase. The second phase solves a multidimensional multiple-choice knapsack (MMKP) problem to select the optimal mix of patterns.

The proposed two-phase approach allows to efficiently solve the SMP. This efficiency is due to the fact that the evaluation of the nonlinear reliability function is removed from the optimization phase. Our approach also benefits from the computational efficiency of current computers and optimization solvers when dealing with linear programming formulations. Furthermore, given that the system reliability is computed in a separate phase, SMP for large multicomponent systems with complex reliability structures can now be investigated and solved accurately with reasonable computation time.

This paper makes one major and two secondary contributions. It proposes the first SMP models for serial k-out-of-n systems and complex reliability structures, which have been overlooked and ignored in the current SMP literature. Secondly, two tighter and more general nonlinear formulations of the SMP problem which allow to solve the problem for moderately large and complex systems using commercially available nonlinear optimization solvers are proposed. Unfortunately, the nonlinear models are difficult to solve for large systems. Therefore, the third contribution is the development of a new two-phase model to solve the SMP for large and complex systems. The proposed two-phase method removes the evaluation of the nonlinear reliability function from the optimization phase and converts the SMP into a MMKP which can be easily solved by current optimization techniques and linear optimization solvers. Multiple experiments are run to validate the formulations and show that the proposed approach will lead to new SMP extensions by allowing the formulations of more complex problems which will take into account the availability of multiple repair crews/channels, the consideration of performance indicators other than system reliability, etc.

The remainder of the present paper is structured around 6 sections as follows. In Section 2, the notation and main working assumptions are listed. This section also describes the multicomponent system under consideration and the computation of its reliability function. Section 3 presents the imperfect maintenance model and develops the expressions for total maintenance cost and duration incurred by the system components during the break. In Section 4, the classical formulation of the SMP under study is developed and presented as a mixed integer nonlinear program. The proposed two-phase formulation and solution approach is presented and discussed in Section 5 followed by numerical experiments in Section 6. Conclusions are drawn and future research extensions are proposed in Section 7.

2. System description and reliability computation

2.1. Acronyms, notation list and main assumptions

Acronyms:

BIP	Binary integer program
CM	Corrective maintenance
FEA	Full enumeration algorithm
GA(k, n)	k-out-of-n:G system
IM	Imperfect maintenance
ILP	Integer linear program
MINLP	Mixed integer nonlinear program
MMKP	Multidimensional multiple-choice knapsack problem
MR	Minimal repair
PM	Preventive maintenance
SMP	Selective maintenance problem

Notation:

m	Number of subsystems in the multicomponent system
i	Index of subsystems, $i = 1, \dots, m$
n_i	Number of components in subsystem i
j	Index of parts in subsystem i, $j = 1, \dots, n_i$
E_{ij}	The j^{th} component of subsystem i
k_i	Minimum number of working/functioning components required for subsystem i to work
L_{ij}	The highest maintenance level available for component E_{ij}
l	Maintenance level $l \in \{0, \cdots, L_{ij}\}$ available for component E_{ij}
t^c_{ijl} (c^c_{ijl})	Time (cost) of CM when maintenance level l is performed on E_{ij}
t^p_{ijl} (c^p_{ijl})	Time (cost) of PM when maintenance level l is performed on E_{ij}
A_{ij} (B_{ij})	Age of component E_{ij} at the start (the end) of the break
S_{ij} (V_{ij})	Status binary variable of component E_{ij} at the start (the end) of the break
$\mathcal{G}_i$	Set of patterns generated for subsystem i
G_i	Number of patterns generated for subsystem i ($G_i = \lvert\mathcal{G}_i\rvert$)
g	Index of patterns generated for subsystem i, $g = 1, \dots, G_i$
t_{ig}	Duration of pattern g for subsystem i
C_{ig}	Cost of pattern g for subsystem i
C_0	Maximum maintenance budget available
$\mathcal{T}_0$	Limited break duration
R^c_{ij}	Conditional reliability of component E_{ij}
R^s_{ig}	Reliability of subsystem i when pattern g is selected
$\mathcal{R}$	Overall system reliability during the subsequent mission
$\mathcal{R}_0$	Minimum required overall system reliability level
U	Subsequent mission duration

Assumptions:

- The system consists of multiple, repairable binary components (the components and the system are either functioning or failed).
- During the break, system components do not age, i.e. the age of a component is operation-dependent.
- No maintenance activity is allowed during the mission. Maintenance activities are allowed only during the break.
- All required limited resources (budget, repairpersons, tools) are available when needed. Only one repair channel is available meaning that only one component can be worked on at any given time.

2.2. System description

Without loss of generality, the selective maintenance problem addressed in the present work considers a system made of m GA(k, n) subsystems in series. Each subsystem i ($i = 1, \cdots, m$) is composed of n_i components E_{ij}, $j \in \{1, \cdots, n_i\}$. The ith subsystem functions if and only if at least k_i out of the n_i components are functioning. Individual components in each subsystem are independent, their lifetimes are not necessarily identically distributed and they do not have the same age at the start of the break period when maintenance decisions are to be made.

The system is assumed to have just finished a mission and has been turned off during the scheduled break of length $\mathcal{T}_0$ to undergo maintenance activities. The system will be used after the break to carry-out the next mission of duration U. At the end of the current mission (i.e., at the beginning of the current break), each component E_{ij} is described by its current effective age B_{ij}, and its status is given by a binary state variable S_{ij} defined as:

$$S_{ij} = \begin{cases} 1, & \text{if } E_{ij} \text{ is working/functioning at the start of the break} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

At the end of the break, each component E_{ij} is also described by its effective age A_{ij}, and its status is given by a binary state variable V_{ij}:

$$V_{ij} = \begin{cases} 1, & \text{if } E_{ij} \text{ is working/functioning at the end of the break} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The binary variable V_{ij} is necessary because a failed component may be repaired during the break or left as is.

2.3. Probability of successfully completing the next mission

The probability that the system successfully completes the next mission is given by its reliability $\mathcal{R}$. To compute this reliability, we find the expressions of the reliability first for each component R^c_{ij}, then for each subsystem R^s_i and finally for the overall system $\mathcal{R}$.

Let R^c_{ij} denote the conditional probability that component E_{ij} will survive the next mission of length U. This conditional reliability depends on both the status V_{ij} and effective age A_{ij} of the component at the start of the next mission:

$$R^c_{ij} = \frac{R_{ij}(A_{ij}+U)}{R_{ij}(A_{ij})} \cdot V_{ij} \tag{3}$$

where $R_{ij}(t)$ is the unconditional reliability function of component E_{ij}. Without loss of generality, the lifetime of a component E_{ij} is assumed to be governed by a Weibull distribution with shape and scale parameters β_{ij} and η_{ij}. The unconditional component reliability function $R_{ij}(t)$ is $R_{ij}(t) = e^{-\left(\frac{t}{\eta_{ij}}\right)^{\beta_{ij}}}$.

The reliability R^s_i of the ith subsystem during the next mission is obtained from the exact formulation proposed by Arulmozhi [1] to determine the reliability for general k-out-of-n systems:

$$R^s_i = \sum_{j_{k_i}=1}^{n_i} \sum_{j_{k_i-1}=1}^{j_{k_i}-1} \cdots \sum_{j_1=1}^{j_2-1} \left(\prod_{v=j_1}^{j_{k_i}} R^c_{iv}\right)\left(\prod_{\substack{u=1\\ u\neq j_1,\cdots,j_{k_i}}}^{j_{k_i}} (1-R^c_{iu})\right) \tag{4}$$

Finally, we get the reliability $\mathcal{R}$ of the whole system as:

$$\mathcal{R} = \prod_{i=1}^{m} R^s_i = \prod_{i=1}^{m}\left[\sum_{j_{k_i}=1}^{n_i} \sum_{j_{k_i-1}=1}^{j_{k_i}-1} \cdots \sum_{j_1=1}^{j_2-1} \left(\prod_{v=j_1}^{j_{k_i}} R^c_{iv}\right)\left(\prod_{\substack{u=1\\ u\neq j_1,\cdots,j_{k_i}}}^{j_{k_i}} (1-R^c_{iu})\right)\right] \tag{5}$$

Many algorithms and procedures have been developed in the literature to compute the reliability for GA(k, n) systems. The generating function approach proposed by Barlow and Heidtmann [2], although less efficient than the approaches developed by Arulmozhi [1] and Kouchy [28], presents the advantage of being simple and easy to program. Kuo and Zuo [29] present a modified version of their algorithm. The latter approach is used in the patterns generation phase to compute the reliability of the GA(k, n) systems.

3. Maintenance/repair levels, costs and durations

3.1. Imperfect maintenance model

During the break, each failed component E_{ij} can potentially be subjected to a list of $L_{ij}+1$ corrective maintenance (CM) levels l, $l \in \{0, 1, \cdots, L_{ij}\}$. The lowest maintenance level $l = 0$ corresponds to the "Do nothing" case, while the highest level $l = L_{ij}$ corresponds to the perfect replacement or *as good as new* case. Level $l = 1$ corresponds to the minimal repair case which when performed brings the component to an *as bad as old* condition. Intermediate values of $1 < l < L_{ij}$ represent imperfect maintenance actions which bring the component back to a condition between *"as bad as old"* and *"as good as new"*. The age reduction approach proposed by Malik [35] is used to model imperfect maintenance. Hence, when a CM level l is performed on E_{ij}, its age is reduced by a factor α_{ijl}, $(0 < \alpha_{ijl} < 1)$ for a cost c^c_{ijl} and a duration t^c_{ijl}.

Similarly, a working/functioning component with age B_{ij} at the end of the current mission can also be subjected to a number of preventive maintenance (PM) levels l, $l \in \{0, 2, \cdots, L_{ij}\}$. In line with the CM, levels $l = 0$ and $l = L_{ij}$ correspond, respectively, to the "Do nothing" and the perfect replacement or *as good as new* cases. Note that there is no minimal repair equivalent for a PM, therefore the maintenance option $l = 1$ is not available for working/functioning components. Intermediate values of l with $2 \le l < L_{ij}$ represent imperfect maintenance actions which rejuvenate the component by reducing its age by a factor δ_{ijl} $(0 < \delta_{ijl} < 1)$. For example, $\delta_{ijL_{ij}} = 0$ is the perfect replacement case (i.e, the age resets to 0). Each PM action of level l incurs a cost c^p_{ijl} and a duration t^p_{ijl}.

The following binary decision variable ϕ_{ijl} is introduced to model the maintenance level to be selected:

$$\phi_{ijl} = \begin{cases} 1, & \text{if maintenance level } l \text{ is performed on } E_{ij} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Since the component E_{ij} is subjected to maintenance, its corresponding effective age A_{ij} at the end of the break can be expressed as a function of its initial status S_{ij} and the maintenance level l selected. Using Eq. (6), the effective age A_{ij} is given such by:

$$A_{ij} = B_{ij}\left[S_{ij}\sum_{l,\, l\neq 1} \delta_{ijl}\phi_{ijl} + \left(1-S_{ij}\right)\sum_{l} \alpha_{ijl}\phi_{ijl}\right] \tag{7}$$

3.2. Total maintenance cost and time

Depending on the system performance required, a system component E_{ij}, may be selected for maintenance. When it is not selected, the corresponding maintenance cost and duration are ignored. However, if the component E_{ij} is selected for maintenance and an eligible maintenance level $l \in \{1, \cdots, L_{ij}\}$ is selected to be performed on E_{ij}, it incurs a maintenance cost and duration.

The total PM cost is given by:

$$C_{PM} = \sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{l=2}^{L_{ij}} c^p_{ijl} \cdot S_{ij} \cdot \phi_{ijl} \tag{8}$$

where a PM action of level $l \ge 2$ is allowed to be performed on component E_{ij} only if E_{ij} is working/functioning at the start of the break (i.e., $S_{ij} = 1$). Recall that $l = 0$ means that no PM is done, therefore no maintenance cost is incurred. There is no level $l = 1$ for PM.

The total CM cost is given by:

$$C_{CM} = \sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{l=1}^{L_{ij}} c^c_{ijl} \cdot \left(1-S_{ij}\right) \cdot \phi_{ijl} \tag{9}$$

where $\left(1-S_{ij}\right)$ implies that CM actions are available only for failed components.

Similarly, the total PM and CM durations are, respectively, given as follows:

$$T_{PM} = \sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{l=2}^{L_{ij}} t^p_{ijl} \cdot S_{ij} \cdot \phi_{ijl} \tag{10}$$

$$T_{CM} = \sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{l=1}^{L_{ij}} t^c_{ijl} \cdot \left(1-S_{ij}\right) \cdot \phi_{ijl}. \tag{11}$$

Finally, the total maintenance cost C and duration $\mathcal{T}$ are, respectively, given such as:

$$C = C_{PM} + C_{CM} \tag{12}$$

$$\mathcal{T} = T_{PM} + T_{CM}. \tag{13}$$

4. Mixed integer nonlinear programming formulation of the SMP

Assume that a system has just accomplished its current mission and is available for maintenance activities to be carried out on its components. Due to limited resources, only a subset of maintenance actions can actually be performed. Therefore, the goal of the SMP is to jointly select the set of components to be maintained and the maintenance levels to be performed on the selected components. Two new nonlinear models of the generalized SMP may thus be formulated according to the goals of the decision-maker, namely to maximize system reliability during the next mission (MINLP–1), or to minimize the total maintenance cost (MINLP–2).

- **Model MINLP–1**:

$$\text{Max } \mathcal{R} = \prod_{i=1}^{m}\left[\sum_{j_{k_i}=1}^{n_i}\sum_{j_{k_i-1}=0}^{j_{k_i}-1}\cdots\sum_{j_1=0}^{j_2-1}\left(\prod_{v=j_1}^{j_{k_i}} R_{iv}^{c}\cdot\prod_{\substack{u=1\\u\neq j_1,\cdots,j_{k_i}}}^{j_{k_i}}(1-R_{iu}^{c})\right)\right] \tag{14}$$

s.t.:

$$C \le C_0 \tag{15}$$

$$\mathcal{T} \le \mathcal{T}_0 \tag{16}$$

$$\sum_{l}(1-S_{ij})\phi_{ijl} + \sum_{l,l\neq 1} S_{ij}\phi_{ijl} = 1, \quad \forall i,j \tag{17}$$

$$A_{ij} = B_{ij}\left[S_{ij}\sum_{l,l\neq 1}\delta_{ijl}\phi_{ijl} + (1-S_{ij})\sum_{l}\alpha_{ijl}\phi_{ijl}\right], \quad \forall i,j \tag{18}$$

$$V_{ij} = S_{ij} + \sum_{l=1}^{L_{ij}}(1-S_{ij})\phi_{ijl}, \quad \forall i,j \tag{19}$$

$$R_{ij}^{c} = \frac{R_{ij}(A_{ij}+U)}{R_{ij}(A_{ij})}\cdot V_{ij}, \qquad \forall i,j \tag{20}$$

$$\phi_{ijl} \in \{0,1\} \qquad \forall i,j,l \tag{21}$$

In this formulation, the objective function (14) aims to maximize the overall reliability which was developed in Eq. (5). Constraints (15) and (16) ensure respectively that the total maintenance cost and duration do not exceed the budget available and the duration of the break. Constraint (17) states that exactly one maintenance level has to be selected either for PM or CM. Constraint (18) is used to update the effective age of each component at the end of the break period as developed in Eq. (7). Constraint (19) imposes that a component status after the break is the same as its status before unless it has been repaired (because it was failed before the break, i.e. $S_{ij} = 0$). Constraint (20) is used to compute the conditional reliability of component E_{ij}. Constraint (21) defines the binary variable ϕ_{ijl} used in the formulation.

- **Model MINLP–2**:

If a minimum reliability target $\mathcal{R}_0$ is pre-specified for the next mission, then the nonlinear model (MINLP–2) can be formulated as follows:

$$\text{Min } C = \sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{l=1}^{L_{ij}}\left[c_{ijl}^{p}\cdot S_{ij} + c_{ijl}^{c}\cdot(1-S_{ij})\right]\cdot\phi_{ijl} \tag{22}$$

s.t.:

$$\mathcal{R} \ge \mathcal{R}_0 \tag{23}$$

$$\mathcal{T} \le \mathcal{T}_0 \tag{24}$$

$$\sum_{l}(1-S_{ij})\phi_{ijl} + \sum_{l,l\neq 1} S_{ij}\phi_{ijl} = 1, \quad \forall i,j \tag{25}$$

$$A_{ij} = B_{ij}\left[S_{ij}\sum_{l,l\neq 1}\delta_{ijl}\phi_{ijl} + (1-S_{ij})\sum_{l}\alpha_{ijl}\phi_{ijl}\right], \quad \forall i,j \tag{26}$$

$$V_{ij} = S_{ij} + \sum_{l=1}^{L_{ij}}(1-S_{ij})\phi_{ijl}, \quad \forall i,j \tag{27}$$

$$R_{ij}^{c} = \frac{R_{ij}(A_{ij}+U)}{R_{ij}(A_{ij})}\cdot V_{ij}, \qquad \forall i,j \tag{28}$$

$$\phi_{ijl} \in \{0,1\} \qquad \forall i,j,l \tag{29}$$

The MINLP models presented above are computationally intractable due to the complexity of the expression of the reliability $\mathcal{R}$ in the objective function (14) and constraint (23). In practice, currently available solvers for these GA(k, n) system formulations yield local optimum solutions. Hence, there is a need for an approach that allows for more general reliability structures to be modelled and solved optimally.

5. The proposed two-phase approach for a BIP formulation

The first phase is a structured and guided procedure to efficiently generate all feasible combinations of component and maintenance levels. The second phase solves a MMKP to optimally select a series of patterns to maximize the system reliability or minimize the total cost. We give a brief formulation of the MMKP before presenting our two-phase approach. For more details, please refer to [19,20,47].

The MMKP is a binary knapsack problem in which a set of items S partitioned into n disjoint classes $S_1, \cdots, S_n$ is given where each class S_i has n_i items ($i \in \{1, \cdots, n\}$), respectively. Each item j of class S_i has a non-negative performance (or profit) value r_{ij} and requires a certain amount of resources given by a requirement (or weight) vector $W_{ij} = (w_{ij}^{1}, \cdots, w_{ij}^{m})$. The capacities or amounts of available resources are given by a vector $C = (C^1, \cdots, C^m)$. The problem is to fill the multi-constrained knapsack with exactly one item from each class in order to maximize the total performance (or profit) value of the selection while satisfying the capacity constraints. The model uses binary variables $y_{ij} = 1$ if the jth item of class N_i is selected, and $y_{ij} = 0$ otherwise.

$$\text{Max } Z = \sum_{i=1}^{n}\sum_{j=1}^{n_i} r_{ij}\cdot y_{ij}$$

s.t.:

$$\sum_{j=1}^{n_i} y_{ij} = 1 \qquad \forall i \in \{1,\cdots,n\}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n_i} w_{ij}^{k}\, y_{ij} \le C^k \qquad \forall k \in \{1,\cdots,m\}$$

$$y_{ij} \in \{0,1\};\ i \in \{1,\cdots,n\};\ j \in \{1,\cdots,n_i\}$$

5.1. Phase I: pattern generation

A pattern is a combination of components and related maintenance levels to be performed, which results in a discrete reliability R_{ig}^{s} for the i^{th} GA(k_i, n_i) subsystem during the subsequent mission. To estimate the reliability R_{ig}^{s}, the present work uses the approach developed in Kuo and Zuo [29]. The overall principle of Phase I is described in Algorithm (1). This algorithm is developed for a problem with reliability maximization as the objective function (equivalent to MINLP–1). Adapting the algorithm to deal with a cost minimization function (equivalent to MINLP–2) is straightforward.

To illustrate how Phase I works, a GA(1,2) system is considered. At the start of the break, we assume that both components are working/functioning and can undergo 2 levels of maintenance: 0-Do nothing and 1-Replace. The following list $\mathcal{G}$ of 4 patterns is generated $\mathcal{G} = \{1,2,3,4\}$ where pattern $g = 1$ represents the case where both components are replaced, $g = 2$ represents the case where only component 2 is

Algorithm 1 Compute R^s_{ig}, C_{ig}, t_{ig} for all valid patterns for subsystem *i*.

```
Input data: k_i, n_i, c^c_{ijl}, t^c_{ijl}, c^p_{ijl}, t^p_{ijl}, B_{ij}, α_{ijl}, δ_{ijl}, β_{ij}, η_{ij}, m
Initialize: i = 1
while i ≤ m do
    – Generate an integer numbered list 𝒢_i of all valid combination/patterns of components and their PM or CM levels such that at least k_i components will be working/functioning after maintenance. G_i = |𝒢_i|.
    – Calculate the cardinality of the lists: G_i = |𝒢_i|.
    Initialize: g = 1
    while g ≤ G_i do
        – Calculate the related maintenance cost C_{ig} and duration t_{ig} by summing up all the individual costs and durations, respectively.
        if (C_{ig} ≤ C_0) and (t_{ig} ≤ 𝒯_0) then
            – Compute R^c_{ij} the conditional reliability for all components E_{ij} in the current pattern g using Equation (3).
            – Compute R^s_{ig} the reliability of the subsystem i under the current pattern g using the algorithm proposed in [29].
            – Store values of i, g, R^s_{ig}, c_{ig}, t_{ig}.
        else
            – Remove current pattern g from the list 𝒢_i. (all patterns above g get shifted down by one position after the removal of g.)
            – Update g = g − 1 to account for the removed pattern.
            – Update G_i = |𝒢_i|.
        end if
        g = g + 1.
    end while
    i = i + 1.
end while
```

selected to be replaced and component 1 is not selected, $g = 3$ represents the opposite case of pattern $g = 2$, and finally $g = 4$ is the case where no component is selected for maintenance. If at the start of the break both components are failed, then the list of potential patterns is reduced to $\mathcal{G} = \{1, 2, 3\}$ because the Do-nothing option to both components is no longer a valid combination.

At the end of Phase I, a dataset containing the values of the following parameters $i, g, R^s_{ig}, c_{ig}, t_{ig}, G_i$ for each pattern and each subsystem is generated and passed as input data to Phase II for the optimization procedure.

5.2. Phase II: binary integer programming formulation

The aim of Phase II is to select one pattern out of all G_i feasible patterns generated for each subsystem i such that the overall system reliability $\mathcal{R}$ is maximized. This selection is performed such that the total duration and cost of selected patterns do not exceed the break duration and maintenance budget, respectively. The following binary decision variable is used to make the pattern selection:

$$z_{ig} = \begin{cases} 1, & \text{if pattern } g \text{ is selected for subsystem } i \\ 0, & \text{otherwise.} \end{cases}$$

Since the system under consideration is a series arrangement of m GA(k_i, n_i) subsystems ($i = 1, \cdots, m$), the system reliability $\mathcal{R}$ during the next mission can then be written as a function of the decision variable z_{ig} as follows:

$$\mathcal{R} = \prod_{i=1}^{m} \left(\sum_{g=1}^{G_i} R^s_{ig} \, z_{ig} \right). \tag{30}$$

Applying the logarithmic transformation gives:

$$\ln(\mathcal{R}) = \sum_{i=1}^{m} \ln \left(\sum_{g=1}^{G_i} R^s_{ig} \cdot z_{ig} \right).$$

Given that z_{ig} is a binary variable and $\sum_g z_{ig} = 1$, $\ln(\mathcal{R})$ can then equivalently be written as follows:

$$\ln(\mathcal{R}) = \sum_{i=1}^{m} \sum_{g=1}^{G_i} \ln \left(R^s_{ig} \right) \cdot z_{ig} \tag{31}$$

The resulting mathematical formulation for the maximization of the overall system reliability (Model BIP–1) is as follows:

$$\text{Max } \ln(\mathcal{R}) = \sum_{i=1}^{m} \sum_{g=1}^{G_i} \ln(R^s_{ig}) \cdot z_{ig} \tag{32}$$

s.t.:

$$\sum_{g=1}^{G_i} z_{ig} = 1 \qquad \forall i \in \{1, \cdots, m\} \tag{33}$$

$$\sum_{i=1}^{m} \sum_{g=1}^{G_i} C_{ig} \, z_{ig} \le C_0 \tag{34}$$

$$\sum_{i=1}^{m} \sum_{g=1}^{G_i} t_{ig} \cdot z_{ig} \le \mathcal{T}_0 \tag{35}$$

$$z_{ig} \in \{0, 1\}; \; i = 1, \cdots, m; \; g = 1, \cdots, G_i \tag{36}$$

In the above optimization model, constraint (33) ensures that exactly one maintenance pattern is selected per subsystem. Constraints (34) and (35) ensure that the total maintenance cost and duration of all patterns selected do not exceed the budget and break duration, respectively. The last constraint defines the binary decision variable z_{ig} used in the formulation.

Similarly, if a maintenance budget C_0 is pre-specified an alternative formulation (Model BIP–2) is:

$$\text{Min } C = \sum_{i=1}^{m} \sum_{g=1}^{G_i} C_{ig} \cdot z_{ig} \tag{37}$$

s.t.:

$$\sum_{g=1}^{G_i} z_{ig} = 1 \qquad \forall i \in \{1, \cdots, m\} \tag{38}$$

$$\sum_{i=1}^{m} \sum_{g=1}^{G_i} \ln(R^s_{ig}) \cdot z_{ig} \ge \ln(R_0) \tag{39}$$

$$\sum_{i=1}^{m} \sum_{g=1}^{G_i} t_{ig} \cdot z_{ig} \le \mathcal{T}_0 \tag{40}$$

$$z_{ig} \in \{0, 1\}; \; i = 1, \cdots, m; \; g = 1, \cdots, G_i \tag{41}$$

The mathematical model BIP–1 matches the definition of the MMKP given at the beginning of Section 5. BIP–2 is similar to the MMKP because the objective function (37) and constraint (39) would need to be multiplied by -1 to exactly match the definition of the MMKP. Given that the SMP is formulated as an MMKP, its complexity is the same as any other MMKP: it is $\mathcal{NP}$-hard [47]. The computation times increase as more components and subsystems are added. In the numerical experiments section, we optimally solved these models using the Gurobi 7.5 solver [21] called from MPL 5.0 [36] for moderately large systems. It should be noted that if in practice there were a need to solve this problem for a very high number of subsystems, one would be able to use the efficient solution methods recently developed by Voß et al. [47]. Ghasemi and Razzazi [20] have proposed a branch-and-bound (B&B)

Table 1
Parameters for experiment #1 (source: references [7,38]).

E_{ij}	η_{ij}	β_{ij}	S_{ij}	B_{ij}	t^c_{ij1}	t^c_{ij2}	t^p_{ij1}	c^c_{ij1}	c^c_{ij2}	c^p_{ij1}
E_{11}	15	1.5	1	15	3	1	5	6	12	12
E_{12}	15	1.5	1	20	3	1	5	5	12	12
E_{21}	20	3	0	8	2	2	4	5	14	14
E_{22}	20	3	1	15	2	2	4	6	15	15

algorithm to solve very large instances of the MMKP exactly. Their exact method is able to efficiently solve problems with extremely large number of subsystems. Two other exact methods based on the best-first B&B algorithm have been proposed by Khan et al. [22,43]. Other methods can be found in [11,19].

In the following section, numerical experiments are carried out to show that our models yield valid maintenance decisions for complex multicomponent systems more general than any other system considered before. Furthermore, comparisons show that computational requirements are substantially reduced with our proposed approach.

6. Numerical examples

In this section, six sets of numerical experiments are conducted. The first set is a validation experiment based on the example proposed by Cassady et al. [7] and used in Pandey et al. [38]. The second experiment compares the proposed approach to the traditional nonlinear formulation for a series-parallel system. The third experiment compares the proposed approached and the Full Enumeration Algorithm (FEA), which is used as a baseline for all exact methods. The fourth experiment is designed to show how the proposed four models can be used to solve the SMP for a moderately large series-parallel system. The fifth experiment shows how the proposed two-phase approach can be used to solve the SMP for a large serial *k*-out-of-*n:G* system under imperfect maintenance. The sixth experiment shows how the proposed approach can be used to solve the SMP for large complex reliability structures.

All experiments are run on a Intel^{TM}i5 3.4GHz desktop computer with 16GB of RAM running Windows 7^{TM}. The Phase 1 and FEA algorithms were programmed in Python 3.5.

6.1. Set of experiments #1: validation example

For this first set of experiments, the exact same system made of two GA(1,2) subsystems in series considered in [7] and [38] is used. The parameters are also the same and displayed in Table 1. For failed components, three CM levels are considered: $l = 0$ is Do nothing (DN), $l = 1$ is Minimal repair (MR), and $l = 2$ is Replace (R). For working/functioning components, two PM levels are considered: $l = 0$ is Do nothing (DN), $l = 2$ is Replace (R). In the original problem [7], only the break duration is limited at $\mathcal{T}_0 = 19$. Thus, at first we find the optimal solution when only the break duration is limited.

The 10 patterns generated by Phase 1 are listed in Table 2 below. Computation time is 1.56×10^{-6} seconds.

Table 2
Patterns generated for experiment #1.

G_{ig}	$ln(R^s_{ig})$	C_{ig}	t_{ig}	Component-maintenance level
G_{11}	-0.4734	0	0	E_{11}–DN; E_{12}–DN
G_{12}	-0.2123	12	5	E_{11}–DN; E_{12}–R
G_{13}	-0.2296	12	5	E_{11}–R; E_{12}–DN
G_{14}	-0.1099	24	10	E_{11}–R; E_{12}–R
G_{21}	-0.2755	5	2	E_{21}–MR; E_{22}–DN
G_{22}	-0.0226	20	6	E_{21}–MR; E_{22}–R
G_{23}	-0.0422	14	2	E_{21}–R; E_{22}–DN
G_{24}	-0.0039	29	6	E_{21}–R; E_{22}–R
G_{25}	-1.0999	0	0	E_{21}–DN; E_{22}–DN
G_{26}	-0.0640	15	4	E_{21}–DN; E_{22}–R

Phase 2 is then run to select one pattern for each subsystem. Table 3 shows the results obtained for varying values of $\mathcal{T}_0$. Computation time is 0.02 seconds.

For $\mathcal{T}_0 = 16$, we obtain the same results as in [7] and [38]: a maximum optimal reliability of 0.8925 achieved by replacing all 4 components. For $\mathcal{T}_0 = 9$, we obtain the same solution ($\mathcal{R}^* = 0.7753$) obtained by Pandey et al. [38]. The general trend observed in Table 3 is that as the break duration $\mathcal{T}_0$ is shortened, less PM activities can take place and the model thus gives priority to CM actions. This results in decreasing achievable reliability.

In the next experiment, both duration and cost limits are imposed on the maintenance actions to be carried out during the break. The results in Table 4 are obtained for $\mathcal{T}_0 = 9$ and varying values of C_0. For $C_0 = 25$, we obtain the same solution ($\mathcal{R}^* = 0.6140$) obtained by Pandey et al. [38].

The general trend observed in Table 4 is that as the maintenance budget C_0 is reduced, less PM activities can take place and the model thus gives priority to CM actions. This results in decreasing achievable reliability. When both duration and budget limitations are active the achievable maximum reliability is even lower than when only the time limitation is active.

6.2. Set of experiments #2: performance comparison between the nonlinear formulation and the proposed approach

In what follows, we compare the solution quality and computation times obtained by the nonlinear MINLP–1 model and the proposed BIP–1 model. Both models are formulated in MPL 5.0. The MINLP–1 model is solved using Lindo 9.0 while the BIP–1 model is solved with Gurobi 7.5.

Table 5 compares the results obtained for the 2-by-2 system considered in the previous subsection. Table 6 compares the results obtained for a 4-by-2 system obtained by duplicating the 2-by-2 system. Optimal values are in bold. For these two small problems, the BIP–1 model outperforms the MINLP–1 in terms of computation time (CPU_t) and reaches the optimal solution for every instance. In two instances across both tables, the nonlinear method reports a local maximum.

The systems considered in the first two sets of experiments are extremely small in size (4 and 8 components), which is seldom the case in practice. The small size example allows [7] to obtain the optimal solution by enumeration whereas [38] use an undisclosed evolutionary algorithm. For large systems, such methods would be prohibitively time-consuming and not guarantee optimality.

6.3. Set of experiments #3: FEA versus BIP

For this set of experiments, three GA(1,n_i) subsystems are considered with $n_1 = 5$, $n_2 = 8$ and $n_3 = 10$. Components in subsystem i ($i = 1, 2, 3$) are identical and their corresponding lifetimes are Weibull distributed. For subsystem i, shape and scale parameters of Weibull distribution of components' lifetimes are set to $\beta_{1j} = 1.5$ and $\eta_{1j} = 15$ ($j = 1, \ldots, n_1$), $\beta_{2j} = 3$ and $\eta_{2j} = 20$ ($j = 1, \ldots, n_2$), and $\beta_{3j} = 2.1$ and $\eta_{3j} = 10$ ($j = 1, \ldots, n_3$).

For failed components, four CM levels are considered: $l = 0$ is Do-nothing (DN), $l = 1$ is Minimal repair (MR), and $l = 2$ is an imperfect corrective maintenance (IM) that reduces the component age by half, and $l = 3$ is Replace (R). For working/functioning components, three PM levels are considered: $l = 0$ is Do-nothing (DN), $l = 2$ is an imperfect preventive maintenance (IM) that reduces the component age by half, and $l = 3$ is Replace (R). The other parameters are listed in Table 7. These parameters are arbitrarily chosen but satisfy the following relationships: PM actions have duration and cost that are lower than equivalent CM actions, and lower maintenance levels incur less time and cost than higher maintenance levels.

Table 3
Results for experiment with only time limit.

$\mathcal{T}_0$	$\mathcal{R}^*$	$\mathcal{T}^*$	Patterns selected	Component–Maintenance level
16	0.892487	16	$\{G_{14}, G_{24}\}$	E_{11}–R; E_{12}–R; E_{21}–R; E_{22}–R
12	0.858894	12	$\{G_{14}, G_{23}\}$	E_{11}–R; E_{12}–R; E_{21}–R; E_{22}–DN
9	0.775300	7	$\{G_{12}, G_{23}\}$	E_{11}–DN; E_{12}–R; E_{21}–R; E_{22}–DN
5	0.597135	2	$\{G_{11}, G_{23}\}$	E_{11}–DN; E_{12}–DN; E_{21}–R; E_{22}–DN

Table 4
Results for experiment with both cost and time limits: $\mathcal{T}_0 = 9$..

C_0	$\mathcal{R}^*$	$\mathcal{T}^*$	C^*	Patterns selected	Component–Maintenance level
30	0.7753	7	26	$\{G_{12}, G_{23}\}$	E_{11}–DN; E_{12}–R; E_{21}–R; E_{22}–DN
25	0.6140	7	17	$\{G_{12}, G_{21}\}$	E_{11}–DN; E_{12}–R; E_{21}–MR; E_{22}–DN
15	0.5971	2	14	$\{G_{11}, G_{23}\}$	E_{11}–DN; E_{12}–DN; E_{21}–R; E_{22}–DN
10	0.4729	2	5	$\{G_{12}, G_{21}\}$	E_{11}–DN; E_{12}–DN; E_{21}–MR; E_{22}–DN

Table 5
Comparative results for a 2-by-2 system when $\mathcal{T}_0 = 9$..

	MINLP–1		BIP–1	
C_0	$\mathcal{R}$	$CPU_t(s)$	$\mathcal{R}$	$CPU_t(s)$
30	**0.7753**	2.78	**0.7753**	0.02
25	0.6034	2.73	**0.6140**	0.02
15	**0.5971**	2.67	**0.5971**	0.02
10	**0.4729**	0.27	**0.4729**	0.02

Table 6
Comparative results for a 4-by-2 system when $\mathcal{T}_0 = 9$..

	MINLP–1		BIP–1	
C_0	$\mathcal{R}$	$CPU_t(s)$	$\mathcal{R}$	$CPU_t(s)$
100	**0.3566**	1.38	**0.3566**	0.04
50	0.2880	1.14	**0.2904**	0.04
30	**0.2236**	0.83	**0.2236**	0.03

The results in Table 8 show that the BIP is significantly faster than the Full Enumeration Algorithm (FEA) for all three systems considered. The first system is comprised of two parallel subsystems in series GA(1,5)–GA(1,8) with a total of 13 components. When $C_0 = 100$, the FEA takes about 181 seconds to find the optimal solution whereas the BIP takes less than 0.26 s.

The second system is comprised of two parallel subsystems in series GA(1,5)–GA(1,10) with a total of 15 components. When $C_0 = 50$, the FEA takes about 1900 s to find the optimal solution whereas the BIP takes less than 1.7 s. For a system with 23 components GA(1,5)–GA(1,8)–GA(1,10), the FEA reports suboptimal solutions after 2000 s of computation whereas the BIP takes less than 2s to find the optimal solutions. For this third system, computation times are arbitrarily capped at 2000 s because running until complete resolution would not change the conclusion that BIP–1 significantly outperforms FEA.

In the next three experiments, our new approach is used to find the optimal solutions for two moderately large and complex systems in reasonable time.

6.4. Set of experiments #4: optimal solutions for moderately large series-parallel system

The same system made of 3 subsystem considered in the previous experiment is considered here. Two experiments are carried out using

Table 7
Parameters for sets of experiments #3, #4 and #5.

Subsystem	E_{ij}	S_{ij}	B_{ij}	t^c_{ij1}	t^c_{ij2}	t^c_{ij3}	t^p_{ij2}	t^p_{ij3}	c^c_{ij1}	c^c_{ij2}	c^c_{ij3}	c^p_{ij2}	c^p_{ij3}
	E_{11}	0	15	4	6	8	2	4	5	10	14	8	10
	E_{12}	1	12	4	6	8	2	4	5	10	14	8	10
GA(1,5)	E_{13}	0	10	4	6	8	2	4	5	10	14	8	10
	E_{14}	1	18	4	6	8	2	4	5	10	14	8	10
	E_{15}	1	20	4	6	8	2	4	5	10	14	8	10
	E_{21}	0	8	3	4	5	1	2	6	10	20	7	12
	E_{22}	1	15	3	4	5	1	2	6	10	20	7	12
	E_{23}	0	8	3	4	5	1	2	6	10	20	7	12
	E_{24}	1	15	3	4	5	1	2	6	10	20	7	12
GA(1,8)	E_{25}	0	8	3	4	5	1	2	6	10	20	7	12
	E_{26}	1	15	3	4	5	1	2	6	10	20	7	12
	E_{27}	0	8	3	4	5	1	2	6	10	20	7	12
	E_{28}	1	15	3	4	5	1	2	6	10	20	7	12
	E_{31}	0	6	2	2.5	4	2	3	4	8	10	5	7
	E_{32}	1	10	2	2.5	4	2	3	4	8	10	5	7
	E_{33}	0	6	2	2.5	4	2	3	4	8	10	5	7
	E_{34}	1	10	2	2.5	4	2	3	4	8	10	5	7
	E_{35}	0	6	2	2.5	4	2	3	4	8	10	5	7
GA(1,10)	E_{36}	1	10	2	2.5	4	2	3	4	8	10	5	7
	E_{37}	0	6	2	2.5	4	2	3	4	8	10	5	7
	E_{38}	1	10	2	2.5	4	2	3	4	8	10	5	7
	E_{39}	1	10	2	2.5	4	2	3	4	8	10	5	7
	$E_{3,10}$	1	10	2	2.5	4	2	3	4	8	10	5	7

Table 8
Comparison of FEA and BIP–1 for $\mathcal{T}_0 = 20$..

System	# of parts	C_0	C^*	$\mathcal{R}^*$ FEA	$\mathcal{R}^*$ BIP	CPUt (sec) FEA	CPUt (sec) BIP
GA(1,5)–GA(1,8)	13	100	59	0.980	0.980	181.2	0.26
	13	50	49	0.972	0.972	174.8	0.22
	13	25	22	0.912	0.912	173.4	0.21
GA(1,5)–GA(1,10)	15	100	48	0.905	0.905	1,936.5	1.69
	15	50	48	0.905	0.905	1,908.5	1.68
	15	25	24	0.790	0.790	1,868.3	1.56
GA(1,5)–GA(1,8)–GA(1,10)	23	100	60	0.767*	0.861	2,000[a]	1.90
	23	50	48	0.763*	0.852	2,000[a]	1.87
	23	25	25	0.694*	0.695	2,000[a]	1.76

* sub-optimal solution obtained by the FEA after 2000 s.
[a] Calculations interrupted after 2000 s and best solution reported.

Table 9
Comparative results for a moderately large system: $C_0 = 100$..

$\mathcal{T}_0$	Without IM $\mathcal{R}^*$	$\mathcal{T}^*$	C^*	With IM $\mathcal{R}^*$	$\mathcal{T}^*$	C^*
10	0.6709	10	30	**0.7180**	10	27
15	0.8048	15	43	**0.9066**	15	38
20	0.8598	20	56	**0.9593**	20	57
40	0.9716	40	100	**0.9974**	38	100
50	0.9734	43	99	**0.9974**	38	100

Table 10
Comparative results for a moderately large system: $\mathcal{T}_0 = 40$..

C_0	Without IM $\mathcal{R}^*$	$\mathcal{T}^*$	C^*	With IM $\mathcal{R}^*$	$\mathcal{T}^*$	C^*
50	0.8769	22	50	**0.9505**	20	49
60	0.9097	26	60	**0.9737**	24	58
80	0.9481	35	80	**0.9918**	32	79
100	0.9716	40	100	**0.9974**	38	100
150	0.9722	40	105	**0.9988**	40	113

Table 11
Results of BIP–1 for a large and complex system: $\mathcal{T}_0 = 100$ and $C_0 = 180$.

Performance	**Maintenance actions**: Components
$\mathcal{R}^* = 0.8138$ $\mathcal{T}^* = 74$ $C^* = 179$	**DN**: E_{22} E_{28} **MR**: E_{13} E_{21} E_{23} E_{25} E_{27} **IM–PM**: None **IM–CM**: None **R**: all other remaining components.

Table 12
Results for BIP–1 with varying maintenance budget C_0 and $\mathcal{T}_0 = 100$.

C_0	$\mathcal{R}^*$	$\mathcal{T}^*$	C^*	CPU_t (s)
500	0.8440	90	268	2.03
200	0.8415	80	200	1.91
180	0.8138	74	179	1.92
150	0.7125	64	148	1.93
100	0.4316	48	100	1.92

the BIP formulation: with and without the imperfect maintenance (level $l = 2$). The results are displayed in Tables 9 and 10 for $C_0 = 100$ and $\mathcal{T}_0 = 40$, respectively. For the case without imperfect maintenance, Phase I takes about 0.0023s to generate 6552 patterns. Each run of the optimization phase II takes less than 0.12s to find the optimal solution. For the case where the imperfect maintenance is accounted for, Phase I takes 0.079 s to generate 207,792 patterns. Each run of the optimization phase II takes less than 1.95 s to find the optimal solution.

Similarly, when keeping the break duration constant at $\mathcal{T}_0 = 40$ and varying the budget C_0, the optimization phase runs in less than 0.12 s. The results are displayed in Table 10.

The results displayed in Tables 9 and 10 show that when imperfect maintenance is allowed, it is possible to achieve equal or slightly better results (values in bold) than when there is no imperfect maintenance. The inclusion of imperfect maintenance levels gives more flexibility to the model to find a combination of components and maintenance actions that best uses the limited resources.

6.5. *Set of experiments #5: generalized SMP with* k*-out-of-*n *subsystems*

For this set of experiments, a system comprising three k-out-of-n subsystems in series is considered: GA(2,5) – GA(3,8) – GA(4,10). Solving the SMP for such a complex system and moderately large system has not been attempted before. All parameters and maintenance levels are the same used in the previous experiment as listed in Table 7. Only the BIP formulation is used to optimally solve this general SMP as the non-linear formulation would neither be tractable nor guarantee optimal solutions. For this experiment, Phase I takes less than 0.08 s to generate 207,792 patterns. Phase II takes at most 3.93 s to reach optimality.

Table 11 shows the results of the BIP–1 model for $\mathcal{T}_0 = 100$ and $C_0 = 180$. The maximum reliability achieved is 0.8138. All components of subsystems and 1 and 3 are replaced except for the failed component E_{13} which undergoes minimal repair. Failed components E_{21}, E_{23}, E_{25} and E_{27} undergo minimal repair, while functioning components E_{22} and E_{28} are not maintained. The components of subsystem 2 are more reliable than the components in subsystems 1 and 3 ($\beta_2 > \beta_1 > \beta_3$). Therefore, the model suggests better maintenance levels for the components in subsystems 1 and 3 given the limited budget available. When there is sufficient budget, $C_0 = 500$ in Table 12, all components are replaced and the maximum reliability achieved is 0.84396.

Table 12 displays the results obtained with BIP–1 when the values of C_0 are varied from 500 to 100. As the budget is reduced, fewer maintenance actions are performed and the system reliability achieved decreases. Each instance of the optimization runs in less than 2.04 seconds.

Table 13 displays the results obtained with BIP–1 when the values of $\mathcal{T}_0$ are varied from 100 to 25. As the break duration is reduced, fewer

Table 13
Results for BIP–1 with varying break duration $\mathcal{T}_0$ and $C_0 = 250$.

$\mathcal{T}_0$	$\mathcal{R}^*$	$\mathcal{T}^*$	C^*	CPU_t (s)
100	0.8440	88	248	2.04
50	0.6223	50	145	2.04
35	0.3187	35	108	1.92
25	0.1323	25	81	1.91

Table 14
Results for BIP–2 with $\mathcal{R}_0 = 0.70$: case of varying $\mathcal{T}_0$.

$\mathcal{T}_0$	$\mathcal{R}^*$	$\mathcal{T}^*$	C^*	CPU_t (s)
100	0.7084	64	147	1.94
60	0.7156	59	153	1.97
56	0.7018	56	154	2.17
55	Inf.	–	–	2.11

Table 15
Results for BIP–2 with $\mathcal{T}_0 = 200$: case of varying $\mathcal{R}_0$.

$\mathcal{R}_0$	$\mathcal{R}^*$	$\mathcal{T}^*$	C^*	CPU_t (s)
0.85	Inf.	–	–	3.16
0.84	0.8402	78	198	3.18
0.80	0.8034	74	174	3.26
0.75	0.7536	68	157	3.57

maintenance actions are performed and the system reliability achieved decreases. Each instance of the optimization runs in less than 2.04 seconds.

The BIP–2 model is run using the same patterns generated above and for a minimum required reliability $\mathcal{R}_0 = 0.70$. Table 14 displays the results obtained when the values of $\mathcal{T}_0$ are varied from 100 to 56. As the break duration is reduced, less maintenance actions can be performed. Thus, in order to achieve the required system reliability the model suggests performing fewer maintenance activities but with higher quality, which drive the cost up. Each instance of the optimization runs in less than 2.17 s. When the break duration is reduced below 56, the problem becomes infeasible; there are no combination of maintenance levels that would achieve the specified minimum system reliability of 0.70. Table 15 displays the results obtained for $\mathcal{T}_0 = 100$ with the values of $\mathcal{R}_0$ varied from 0.85 to 0.75. When $\mathcal{R}_0 = 0.85$, the problem is infeasible as the duration is not enough to guarantee this minimum reliability. As the required reliability is decreased, less maintenance actions are required to achieve the needed reliability and therefore the incurred cost is also decreased. Each instance of the optimization runs in less than 3.6 s.

6.6. Set of experiments #6: complex reliability structure

This sixth experiment shows that our proposed approach can easily be used to solve the SMP for more complex reliability structures connected in series. In this experiment the two-phase approach is applied to solve the SMP in a system made of three subsystems in series (see Fig. 1). The first subsystem is the well-known bridge configuration, the second subsystem is a parallel system and the third subsystem is a 3-out-10 system.

For subsystems 2 and 3, Algorithm (1) is used as described above to generate all the patterns and calculate the subsystem reliability values for each pattern R^s_{ig} $(i = 2, 3)$. For the bridge subsystem, Step 11 of Algorithm (1) calculates the subsystem reliability R^s_{ig} for each pattern using the following equation:

$$R^s_{ig} = R^c_{13} \cdot \left[\left(1-(1-R^c_{11})(1-R^c_{14})\right)\cdot\left(1-(1-R^c_{12})(1-R^c_{15})\right)\right] + \left(1-R^c_{13}\right)\cdot\left[1-\left(1-R^c_{11}R^c_{12}\right)\cdot(1-R^c_{14}R^c_{15})\right],$$

where the R^c_{ij} are given by Eq. (3).

All parameters and maintenance levels are the same used in the previous experiments as listed in Table 7. Only the BIP formulation is used to optimally solve this general SMP as the nonlinear formulation would neither be tractable nor guarantee optimal solutions. For this experiment, Phase I takes less than 0.065 seconds to generate 207,792 patterns.

Table 16
Results of BIP–1 for a large and complex system: $\mathcal{T}_0 = 100$ and $C_0 = 180$.

Performance	**Maintenance actions**: Components
$\mathcal{R}^* = 0.7454$	**DN**: E_{23} and E_{27}
$\mathcal{T}^* = 73$	**MR**: E_{25}, E_{26} and E_{28}.
$C^* = 180$	**IM–PM**: None
	IM–CM: E_{21}
	R: E_{22}, E_{24} and all components of subsystem 1 and 3.

Table 17
Results of BIP–2 for a large and complex system: $\mathcal{T}_0 = 100$ and $\mathcal{R}_0 = 0.70$.

Performance	**Maintenance actions**: Components
$\mathcal{R}^* = 0.7001$	**DN**: E_{21}, E_{22}, E_{23}, E_{25}, E_{26}, E_{27}, E_{28}, and E_{33}
$\mathcal{T}^* = 58$	**MR**: None
$C^* = 138$	**IM–PM**: None
	IM–CM: E_{13}
	R: all remaining components.

Table 16 shows the results of the BIP–1 model for $\mathcal{T}_0 = 100$ and $C_0 = 180$. The maximum reliability achieved is 0.7454. All components of subsystems 1 and 3 are replaced along with components E_{22} and E_{24}. Components E_{23} and E_{27} are not maintained. The failed components E_{25}, E_{26} and E_{28} undergo MR. Finally, component E_{21} undergoes an IM which reduces its age by half.

Table 17 shows the results of the BIP–2 model for $\mathcal{T}_0 = 100$ and $\mathcal{R}_0 = 0.70$. A reliability of 70.01% is achieved for a minimum cost of 138. All components of subsystems 1 and 3 are replaced except E_{13} and E_{33}. Components E_{21}, E_{22}, E_{23}, E_{25}, E_{26}, E_{27}, E_{28}, and E_{33} are not maintained. The failed component E_{13} undergoes an IM that reduces its age by half.

For both cases in Tables 16 and 17, the optimization runtime is under 2 seconds, which is quite remarkable for a system of such size and complexity.

6.7. Summary and highlights of the experiments

The preceding five experiments show that the proposed formulation of the selective maintenance is efficient and suitable to address complex and large size problems. Experiments 1 and 2 compare our formulations to existing results and showed that the proposed four new formulations are valid and yield the same results. Experiments 3 and 4 showed that our formulations are able to optimally solve the SMP for moderately large series-parallel systems and large serial k-out-of-n systems with non identical components. Finally in experiment 5, the proposed approach is also shown to be very easy to adapt to solve the SMP for complex reliability structures by only modifying one step in Phase 1, while Phase 2 remains unchanged.

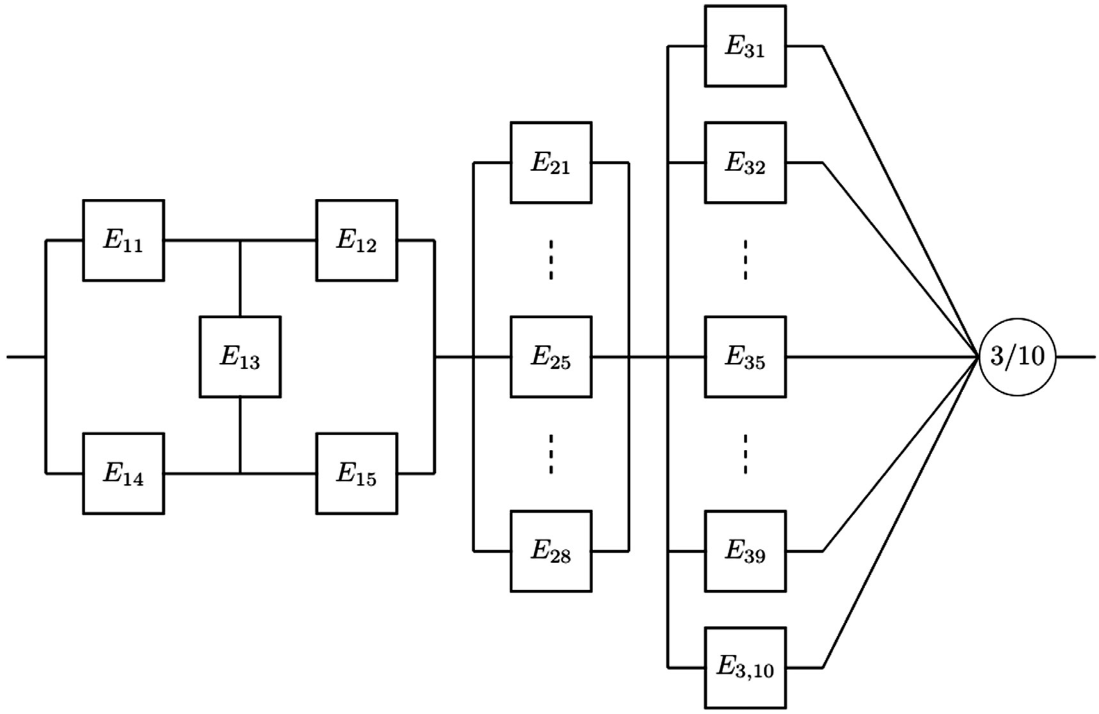

Fig. 1. System structure used in Experiment #6.

7. Conclusion

In this paper, we introduced a two-phase approach combining efficient enumeration of patterns and a multidimensional multiple choice knapsack problem to optimally solve the selective maintenance problem for large serial *k*-out-of-*n* and complex reliability systems since the current nonlinear formulation is known to be inefficient and computationally intensive. Five set of experiments were run to show that the new approach is efficient, reaches the optimal solution in few seconds at most and yields valid maintenance decisions.

Most SMP models assume one repair channel or repairperson. Future extensions that the authors are working on include a generalization to *k*-out-of-*n* subsystems with imperfect maintenance and repairperson eligibility constraints on the subsystems. This can easily be done by generating a list of pattern that each combines a component, a maintenance level and a repairperson.

Current SMP models study only use reliability as the performance indicator. Given that the proposed two-phase approach allows the calculation of the nonlinear reliability functions outside of the optimization phase, it is clear that this method is recommended if other more complex performance indicators such as availability have to be considered within the scope of SMP. Indeed, current SMP models do not use availability as a performance measure in part because the calculations are not analytically tractable for an optimization formulation. One solution to this problem would be to perform the complex calculations required to determine the availability function for each maintenance pattern in Phase I. Phase II would still remain a variant of the MMKP. Other future investigations could include using the proposed approach to formulate and solve fleet condition based maintenance planning problems Feng et al. [18]. Bi-objective selective maintenance problems such as the one proposed by Das Adhikary et al. [16] could also be investigated using the BIP formulation proposed in this paper. Lastly, it would be very interesting to investigate specialized Branch & Bound algorithms, such as [20], specifically designed to solve the MMKP for *k*-out-of-*n* systems under SMP.

Acknowledgment

This work was supported by grants awarded by the Canadian Natural Science and Engineering Research Council to the first and second authors through the Discovery Grants Program. We also thank the Associate Editor and the anonymous reviewers for their comments and suggestions.

References

[1] Arulmozhi G. Exact equation and an algorithm for reliability evaluation of k-out-of-n: G system. Reliab Eng Syst Saf 2002;78(2):87–91.
[2] Barlow RE, Heidtmann KD. Computing k-out-of-n system reliability. IEEE Trans Reliab 1984;R-33(4):322–3.
[3] Bjarnason ET, Taghipour S, Banjevic D. Joint optimal inspection and inventory for a k-out-of-n system. Reliab Eng Syst Saf 2014;131:203–15.
[4] Cao W, Jia X, Hu Q, Song W, Ge H. Selective maintenance for maximising system availability: a simulation approach. Int J Innovative Comput Appl 2017;8(1):12–20.
[5] Cao W, Song W, Hu Q, Du Y. An exact method for solving selective maintenance problems considering imperfect maintenance. In: Intelligent networking and collaborative systems (INCoS), 2016 international conference on. IEEE; 2016. p. 522–6.
[6] Caserta M, Voß S. An exact algorithm for the reliability redundancy allocation problem. Eur J Oper Res 2015;244(1):110–16.
[7] Cassady CR, Murdock WP, Pohl EA. Selective maintenance for support equipement involving multiple maintenance actions. Eur J Oper Res 2001;129:252–8.
[8] Cassady CR, Pohl EA, Murdock WP. Selective maintenance modeling for industrial systems. J Quality Maint Eng 2001;7(2):104–17.
[9] Chen C, Meng MQ-H, Zuo MJ. Selective maintenance optimization for multi-state systems. In: Proceedings of the IEEE Canadian conference on electrical and computer engineering, Edmonton, Canada; 1999.
[10] Chen C, Liu Y, Huang H-Z. Optimal load distribution for multi-state systems under selective maintenance strategy. In: Quality, reliability, risk, maintenance, and safety engineering (ICQR2MSE), 2012 international conference on. IEEE; 2012. p. 436–42.
[11] Chen Y, Hao J-K. A reduce and solve approach for the multiple-choice multidimensional knapsack problem. Eur J Oper Res 2014;239(2):313–22.
[12] Dao CD, Zuo MJ. Selective maintenance for multi-state series systems with s-dependent components. IEEE Trans Reliab 2016;65(2):525–39.
[13] Dao CD, Zuo MJ. Selective maintenance of multi-state systems with structural dependence. Reliab Eng Syst Saf 2017;159:184–95.
[14] Dao CD, Zuo MJ, Pandey M. Selective maintenance for multi-state series–parallel systems under economic dependence. Reliab Eng Syst Saf 2014;121:240–9.
[15] Dao Duc Cuong C, Zuo MJ. Optimal selective maintenance for multi-state systems in variable loading conditions. Reliab Eng Syst Saf 2017;166:171–80.
[16] Das Adhikary D, Bose GK, Jana DK, Bose D, Mitra S. Availability and cost-centered preventive maintenance scheduling of continuous operating series systems using multi-objective genetic algorithm: a case study. Quality Eng 2016;28(3):352–7.

[17] Diallo C, Khatab A, Venkatadri U, Aghezzaf EH. A joint selective maintenance and multiple repair-person assignment problem. 7th international conference on industrial engineering and systems management (IESM 2017); 2017.
[18] Feng Q, Bi X, Zhao X, Chen Y, Sun B. Heuristic hybrid game approach for fleet condition-based maintenance planning. Reliab Eng Syst Saf 2017;157:166–76.
[19] Gao C, Lu G, Yao X, Li J. An iterative pseudo-gap enumeration approach for the multidimensional multiple-choice knapsack problem. Eur J Oper Res 2017;260(1):1–11.
[20] Ghasemi T, Razzazi M. Development of core to solve the multidimensional multiple-choice knapsack problem. Comput Ind Eng 2011;60(2):349–60.
[21] Gurobi Optimization I. Gurobi optimizer reference manual; 2016. http://www.gurobi.com.
[22] Khan MS, Li KF, Manning EG. Quality adaptation in a multisession multimedia system: model, algorithms, and architecture.. University of Victoria; 1999.
[23] Khatab A, Aghezzaf E-H. Selective maintenance optimization when quality of imperfect maintenance actions are stochastic. Reliab Eng Syst Saf 2016;150:182–9.
[24] Khatab A, Ait-Kadi D, Nourelfath M. Heuristic-based methods for solving the selective maintenance problem for series-prallel systems. International conference on industrial engineering and systems management, Beijing, China; 2007.
[25] Khatab A, Dahane M, Ait-Kadi D. Genetic algorithm for selective maintenance optimization of multi-mission oriented systems. Annual European safety and reliability (ESREL) conference, Amsterdam, Netherlands; 2013.
[26] Khatab A, Nahas N, Nourelfath M. Availability of k-out-of-n: G systems with non-identical components subject to repair priorities. Reliab Eng Syst Saf 2009;94:141–51.
[27] Khatab A, Aghezzaf EH, Diallo C, Djelloul I. Selective maintenance optimisation for series-parallel systems alternating missions and scheduled breaks with stochastic durations. Int J Prod Res 2017;55(10):3008–24.
[28] Kouckỳ M. Exact reliability formula and bounds for general k-out-of-n systems. Reliab Eng Syst Saf 2003;82(2):229–31.
[29] Kuo W, Zuo MJ. Optimal reliability modeling: principles and applications. John Wiley & Sons; 2003.
[30] Liu Y, Huang H-Z. Optimal selective maintenance strategy for multi-state systems under imperfect maintenance. IEEE Trans Reliab 2010;59(2):356–67.
[31] Liu Y, Chen Y, Jiang T. On sequence planning for selective maintenance of multi-state systems under stochastic maintenance durations. Eur J Oper Res 2018;268:113–27.
[32] Lust T, Roux O, Riane F. Exact and heuristic methods for the selective maintenance problem. Eur J Oper Res 2009;197:1166–77.
[33] Maaroufi G, Chelbi A, Rezg N. Optimal selective renewal for systems subject to propoagated failures with global effect and failure isolation phenomena. Reliab Eng Syst Saf 2013;114:61–70.
[34] Maillart LM, Cassady CR, Rainwater C, Schneider K. Selective maintenance decision–making over extended planning horizons. IEEE Trans Reliab 2009;58(3):462–9.
[35] Malik M. Reliable preventive maintenance scheduling. AIIE Trans 1979;11(3):221–8.
[36] Maximal Software. Mpl modeling system; 2017. http://www.maximalsoftware.com/mpl/; Retrieved Apr. 17, 2017
[37] Pandey M, Zuo MJ, Moghaddass R. Selective maintenance modeling for multistate system with multistate components under imperfect maintenance. IIE Trans 2013;45:1221–34.
[38] Pandey M, Zuo MJ, Moghaddass R, Tiwari MK. Selective maintenance for binary systems under imperfect repair. Reliab Eng Syst Saf 2013;113:42–51.
[39] Pandey M, Zuo MJ, Moghaddass R. Selective maintenance scheduling over a finite planning horizon. Proc Inst Mech Eng Part O 2016;230(2):162–77.
[40] Rajagopalan R, Cassady CR. An improved selective maintenance solution approach. J Quality Maint Eng 2006;12(2):172–85.
[41] Rice WF, Cassady CR, Nachlas J. Optimal maintenance plans under limited maintenance time. In: Proceedings of industrial engineering conference, Banff, BC, Canada; 1998.
[42] Rice WF. Optimal selective maintenance decisions for series systems. Mississippi State University. Department of Industrial Engineering; 1999. Ph.D. thesis.
[43] Sbihi A. A best first search exact algorithm for the multiple-choice multidimensional knapsack problem. J Comb Optim 2007;13(4):337–51.
[44] Schneider K, Cassady CR. Fleet performance under selective maintenance. In: Reliability and maintainability, 2004 annual symposium-RAMS. IEEE; 2004. p. 571–6.
[45] Schneider K, Cassady CR. Evaluation and comparison of alternative fleet-level selective maintenance models. Reliab Eng Syst Saf 2015;134:178–87.
[46] Sharma P, Kulkarni MS, Yadav V. A simulation based optimization approach for spare parts forecasting and selective maintenance. Reliab Eng Syst Saf 2017;168:274–89.
[47] Voß S, Lalla-Ruiz E. A set partitioning reformulation for the multiple-choice multidimensional knapsack problem. Eng Optim 2016;48(5):831–50.
[48] Xu Q-Z, Guo L-M, Shi H-P, Wang N. Selective maintenance problem for series–parallel system under economic dependence. Defence Technol 2016;12(5):388–400.
[49] Xu Q-Z, Guo L-M, Wang N, Fei R. Recent advances in selective maintenance from 1998 to 2014. J Donghua Univ (English Ed) 2015;32(6):986–94.
[50] Zhao J, Zeng J-C. Maintenance strategy for stochastic selective maintenance of a two-state system. Int J Wireless Mobile Comput 2016;11(4):302–8.
[51] Zhou Y, Lin TR, Sun Y, Bian Y, Ma L. An effective approach to reducing strategy space for maintenance optimisation of multistate series-parallel systems. Reliab Eng Syst Saf 2015;138:40–53.
[52] Zhu H, Liu F, Shao X, Liu Q, Deng Y. A cost-based selective maintenance decision–making method for machining line. Qual Reliab Eng Int 2011;27:191–201.