



**HAL**  
open science

# Optimization of maintenance for complex manufacturing systems using stochastic Remaining Useful Life prognostics

Junkai He, Selma Khebbache, Miguel F. Anjos, Makhlouf Hadji

## ► To cite this version:

Junkai He, Selma Khebbache, Miguel F. Anjos, Makhlouf Hadji. Optimization of maintenance for complex manufacturing systems using stochastic Remaining Useful Life prognostics. *Computers & Industrial Engineering*, 2023, 182, <10.1016/j.cie.2023.109348>. <hal-04115818>

**HAL Id: hal-04115818**

**<https://hal.science/hal-04115818v1>**

Submitted on 9 Jul 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

## Title Page

### Manuscript Title:

Optimization of Maintenance for Complex Manufacturing Systems Using Stochastic Remaining Useful Life Prognostics

### Article Type:

Research Paper

### List of Authors:

Junkai He (corresponding author), Selma Khebbache, Miguel F. Anjos, Makhoulouf Hadji

### Author information:

- *Junkai HE*, Institut Mines-Télécom, Télécom SudParis, 9 rue Charles Fourier, Évry, 91000, France
- *Selma Khebbache*, Technological Research Institute SystemX, 8 Avenue de la Vauve, Palaiseau, 91120, France
- *Miguel F. Anjos*, School of Mathematics, University of Edinburgh, Peter Guthrie Tait Road, Edinburgh, EH9 3FD, Scotland, UK
- *Makhoulouf Hadji*, Technological Research Institute SystemX, 8 Avenue de la Vauve, Palaiseau, 91120, France

# Optimization of Maintenance for Complex Manufacturing Systems Using Stochastic Remaining Useful Life Prognostics

---

## ARTICLE INFO

### Keywords:

Maintenance  
Remaining Useful Life Prognostic  
Complex Manufacturing System  
Mixed Integer Linear Programming  
Chance-Constrained Programming

## ABSTRACT

This paper leverages Remaining Useful Life (RUL) prognostic information for preventive maintenance planning in complex manufacturing factories. Such a factory consists of multiple complex systems that use redundant components as backups to ensure system availability. The purpose is to minimize production fluctuations in the factory due to maintenance or system breakdown. To achieve this, we propose a Mixed-Integer Linear Programming (MILP) model that minimizes the overall production loss. Furthermore, we incorporate random RUL decrease rates according to reliability theory and extend the MILP model using chance-constrained programming. A novel approximation method for dealing with chance constraints is proposed to solve the stochastic model and approximation properties are analyzed. Computational results show that (i) our MILP model can provide more reasonable maintenance decisions compared to the literature. (ii) Sensitivity analysis reports that RUL prognostics and thresholds significantly impact maintenance decisions. (iii) The analysis of the impacts of confidence levels in chance constraints provides decision-makers with the flexibility to adjust maintenance planning under different levels of uncertainty.

---

## 1. Introduction

Manufacturing assets can degrade due to wear and tear in daily operations. Neglecting this reality may lead to sudden system crashes that result in production loss and economic problems for enterprises. Maintenance is an essential element in asset management and can help to slow down the degradation speed. However, as manufacturing systems become increasingly complex, simple maintenance policies may no longer be appropriate. A complex system contains multiple components with interactions among each other (Zhu et al., 2021). Therefore, it is important to see how maintaining interrelated components affects the performance of complex systems. Furthermore, uncertain factors in practice create additional challenges when making maintenance decisions. Industrial managers and academic researchers have thus focused on developing global maintenance management strategies for factories with complex systems, particularly under uncertainty.

The generic factory we consider has a complex structure with several hierarchies, including a series of processes. Each process contains one complex system or parallel complex systems. Each system consists of several component types belonging to different functions, such as compressors and filters. Each component type has a limited number of backups, which can be seen as copies or redundancies (for instance, a system may have four compressors and two filters). An example of a factory diagram can be found in Figure 2 in Section 3. To avoid sudden system failures and ensure factory operation, we propose appropriate maintenance strategies for components, using the prognostic information provided by *Remaining Useful Life* (RUL). The RUL represents the remaining time available for a component to function before it fails. We perform *Preventive Maintenance* (PM) on a component if its RUL reaches a certain condition. This differs from *Corrective Maintenance* (CM) which is carried out when a component has already broken down. The aim of our work, as depicted in Figure 1, is to propose PM planning supported by RUL prognostics to ensure continuous production in the factory. We address both deterministic and uncertain prognostics in maintenance management. The motivation for the latter is that RUL may not decrease as expected due to non-predictable factors in real systems.

In this work, we adopt component redundancy, which enables the use of backup components of any type. It provides higher reliability since backup components can take over operational positions in case of a failure of a component of the same type. By optimizing maintenance on redundant components, we can ensure system availability to the greatest

---

ORCID(s):

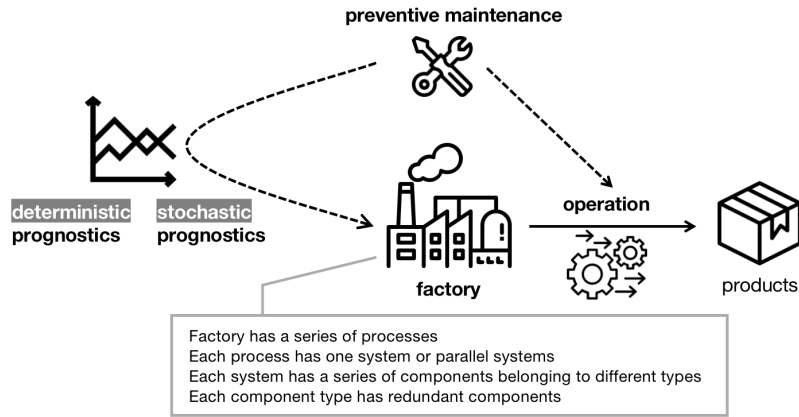


Figure 1: Practical context of our work.

extent possible. The number of operational systems in each process determines the production capacity of that process. Finally, the production volume of the factory depends on the smallest operational capacity in the sequence of processes.

To the best of our knowledge, there is a lack of use of RUL prognostics in the existing literature. Especially, RUL-based mathematical modeling is scarcely carried out. Moreover, system-level availability is rarely addressed through component-level maintenance. As a result, maintenance optimization for a factory with several complex systems using RUL prognostics for redundant components has not been considered before, neither under deterministic nor uncertain environments. To fill these gaps, we propose an optimization approach for global maintenance decision-making based on not only deterministic but also stochastic RUL prognostics and optimal management of redundant components. The goal is to maximize system availability so that the entire factory can operate continuously during the planning horizon. The contributions of this work are summarized as follows:

1. We address a novel deterministic *RUL-based Preventive Maintenance Optimization* (RPMO) problem that minimizes the total production loss. Specific characteristics, such as RUL-based maintenance, component redundancy, system availability, and production, are integrated.
2. We formulate a *Mixed-Integer Linear Programming* (MILP) model to solve the deterministic RPMO problem.
3. We consider a stochastic version of the RPMO problem that accounts for uncertainty in RUL decrease rates, and we formulate a chance-constrained programming model for this stochastic problem based on the MILP model.
4. We design a linear approximation of the chance-constrained model to compute approximate solutions for the stochastic RPMO. We also analyze the mathematical properties of this approximation.
5. We compare the results obtained by our proposed models and by other approaches in the literature. We also perform sensitivity analysis and evaluate the impact of the probability levels in the chance constraints.

For the sake of clarity, we denote deterministic RPMO and stochastic RPMO as DRPMO and SRPMO, respectively. The remainder of this paper is organized as follows. In Section 2, we conduct a comprehensive review of the related literature to highlight our contributions. In Section 3, we describe the DRPMO problem and provide a mathematical formulation using mixed-integer linear programming. In Section 4, we formulate the SRPMO problem using a chance-constrained programming model and propose a linear approximation of this model. In Section 5, we demonstrate the performance and effectiveness of our approaches using computational experiments. Section 6 is dedicated to conclusions, managerial insights, and future research directions.

## 2. Literature Review

Among the vast number of maintenance-related studies, we focus only on the literature about RUL-based maintenance, maintenance in multi-component systems, and maintenance under uncertainty in Section 2.1. Then we summarize maintenance optimization solution methods in both deterministic and stochastic environments in Section 2.2.

## 2.1. Related Maintenance Optimization

Firstly, we review the literature on RUL-based maintenance optimization, including *how to predict RUL*, *how to use RUL prognostics on maintenance*, and *how to leverage RUL prognostics for mathematical formulations*. Secondly, we focus on the availability measure of complex systems via component-level maintenance information. Finally, we proceed to non-deterministic maintenance optimization to position our research.

### 2.1.1. RUL-Related Maintenance Optimization

The RUL is a crucial prognostic parameter that reflects the current status of a component and represents either the time remaining until end-of-life or the time left before the component reaches a warning condition (Si et al., 2011; Diallo et al., 2017). Predicting RUL is a complicated program that involves three pre-processes: data acquisition, health indicator construction, and health stage division. In the literature, RUL prediction approaches can be classified into three mainstream categories. (i) The first is the physics modeled-based approach, which is based on the failure mechanisms or the first principle of damage (Lei et al., 2018). Parameters in physics models are usually correlated to component material properties and stress levels by specific experiments, finite element analysis, and suitable techniques. (ii) The second is the statistical model-based approach, which estimates the RUL by establishing statistical models according to empirical knowledge and presents prediction results as conditional probability density functions. Among them, the Wiener process, Gamma process, and Markov process models are widely used. Recent works include Niu et al. (2023); Wang et al. (2023); Zhang et al. (2022a, 2023a). (iii) The third is the artificial intelligence-based approach, which learns the machinery degradation patterns from available observations (instead of building physics models or statistical models). An increasing number of papers leverage artificial intelligence for RUL prediction recently, such as Chen et al. (2023a); Xiong et al. (2023); Xu et al. (2023). Because our work focuses on leveraging obtained RUL values for maintenance optimization through mathematical programming, RUL prediction is outside the scope of our research. Readers may refer to recent overviews Lei et al. (2018); Hasib et al. (2021); Chen et al. (2023b) for more details.

Instead of RUL prediction, our focus is on effectively utilizing the obtained RUL prognostics in maintenance. In the literature, Do et al. (2015) studied a proactive *Condition-Based Maintenance* (CBM) considering both perfect and imperfect maintenance operations for a deteriorating system based on a Gamma process. The authors proposed a maintenance simulation to decide the next inspection time while considering the system's failure probability. The approach was illustrated using a numerical example. Zhang et al. (2018) presented a CBM policy for a system with Wiener process-based degradation and Poisson process-based random shocks. The degradation level at the next inspection moment was adapted through the estimated RUL to schedule the coming maintenance action. A numerical example was presented and a simulation model was analyzed. Chen et al. (2019) and Liang et al. (2019) took different maintenance actions based on the combination of degradation levels and RUL values. If the degradation level was greater than the warning threshold, then a CM action was required. If the degradation level was smaller than the threshold and the RUL value was lower than the expected one, a PM action was needed. If the degradation level was smaller than the threshold but the RUL value was greater than the expected one, the system operated normally with only the next inspection time to be determined. The main difference between these two papers is the approach for predicting the RUL, which is outside the scope of our research. Lei and Sandborn (2018) proposed a prognostic and health analysis on the maintenance of wind turbines. For multiple wind turbines indicating RUL concurrently, it was assumed that turbines were dependent on each other and system availability relied on the minimum RUL among them. With a similar idea, Cai et al. (2022) and Wang et al. (2022) evaluated the RUL of multi-component systems according to the minimum RUL of all components as well. To the best of our knowledge, maintenance on component redundancy via RUL prognostics in complex systems has not been studied before.

Existing papers also leverage the obtained RUL prognostics in mathematical formulations to take maintenance decisions. Camci (2009) proposed CBM and spare part inventory strategies using given prognostic information to minimize the failure risk. A genetic algorithm was designed and computational results were compared to the ones with a PM strategy. Camci (2015) and Camci et al. (2019) considered maintenance scheduling for geographically distributed assets with forecast reliability and prognostic RUL information. The objective was to minimize the total cost. The former devised a genetic algorithm-based method and the latter formulated a non-linear programming model. Rashidnejad et al. (2018) extended the research of Camci (2015) to a bi-objective problem for minimizing the total cost and maximizing the availability of assets, and applied a non-dominated sorting genetic algorithm II. Bülbül et al. (2019) considered a joint problem with preventive replacement and spare part inventory planning to minimize the total cost. The RUL of a component was formulated as a piece-wise function in a dynamic programming model and three

heuristics were proposed. Wu et al. (2020) presented a maintenance model for complex electro-mechanical equipment via prognostic RUL information to minimize the cost, and proposed an immune particle swarm optimization algorithm. He et al. (2022) considered a similar RUL-based maintenance optimization but for a single complex system and a simpler RUL constraint formulation. In this paper, we extend the work in He et al. (2022) by modeling RUL more accurately and optimizing maintenance for larger complex systems in manufacturing.

The above review of the literature shows that in contrast to the RUL prediction, there is a lack of research using RUL prognostics to take maintenance decisions. Moreover, maintenance optimization based on the obtained RUL prognostics is scarcely carried out, especially in complex production systems with component redundancy.

### 2.1.2. System Availability in Complex Systems

Measuring the availability of complex systems is a significant challenge due to the unplanned downtime of multiple units and components. Most previous articles aim at ensuring the stability of either a component or a system (Cao et al., 2023). As far as we know, only a few works use component-level maintenance decisions to describe system-level availability. Wu and Castro (2020) proposed a linear combination of the degradation processes of components. If this integrated value exceeded a predefined threshold, a PM was performed. The objective was to minimize the number of maintenance actions. Numerical examples were given to illustrate the proposed policy. In Dong et al. (2020), the authors scheduled optimal PM policies considering normal-distributed and independent random shocks. System reliability was described by conditioning on total shock arrival numbers. A Nelder-mead downhill simplex method was introduced to seek the optimal replacement age in minimizing the long-run average maintenance cost. Knofius et al. (2019) studied the configuration of additive manufacturing where materials were added layer by layer during the manufacture of complete parts to guarantee availability. The objective was to minimize the total cost with a system availability constraint. A non-linear programming model and a marginal analysis were proposed. Zhang et al. (2022b) addressed CBM optimization for a two-component system, where component 1 was repairable with only two states (working and failed) and component 2 degraded by a Wiener process with different volatility parameters depending on the working state of component 1. The system was renewed if the degradation of component 2 exceeded a maintenance threshold. Zhang et al. (2023b) further described the state matrix of a complex system by Poisson-distribution-based component degradation. The finite state matrix also depended on an infinitesimal generator and transition probabilities. Azizi and Salari (2023) considered that each unit in a complex system was subject to deterioration by a continuous-time Markov process with two working states and a failure state. Maintenance was initiated when the system had a state with a pre-specified number of failed units. Koopmans and de Jonge (2023) calculated the total production speed of a complex production system by the aggregate production speeds of individual units. The purpose was to keep the system speed at an ideal level to avoid production loss, which thus required more units to operate. Similar to Koopmans and de Jonge (2023), we also assume the need for many systems in a factory to run correctly to ensure the highest possible production, but our considered factory has more complicated structures and we incorporate RUL prognostics as well.

Employing component redundancy is another strong guarantee for complex system availability, which allows in-operation and standby states among alternatively working components (Xenos et al., 2016). Ye et al. (2019) proposed a series architecture consisting of several main operating stages that contained several components. The purpose was to decide the number of redundant components to reduce the total maintenance cost. However, Ye et al. (2019) is related to the design of a production system and is about tactical-level optimization. In our work, the structure of a manufacturing system is provided, and we seek to optimally exploit these redundant components when making maintenance plans. Khorshidi et al. (2016) and Wang et al. (2022) also considered component redundancy in a complex system, in which certain parts must work simultaneously to ensure system availability, called a  $k$ -out-of- $n$  rule. In contrast, we consider more than one complex system, and each of them has redundant components to ensure availability.

We conclude that describing system-level availability through component-level maintenance is rarely addressed. Moreover, to the best of our knowledge, maintenance optimization for a factory with several complex systems using RUL prognostics of redundant components has not been considered before.

### 2.1.3. Non-deterministic Maintenance Optimization

Compared to deterministic maintenance optimization of complex systems, optimization under uncertainty is relatively uncommon. To the best of our knowledge, uncertainty in maintenance optimization literature is usually considered as stochastic maintenance quality, stochastic maintenance duration, and stochastic deterioration, respectively. For *stochastic maintenance quality*, Khatab and Aghezzaf (2016) studied selective maintenance optimization when the quality of maintenance actions was stochastic. This quality was treated as a random variable with identified probability

distributions. Shahraki et al. (2020) studied stochastic imperfect maintenance actions considering the dependency among components. Ghorbani et al. (2022) assumed future operating conditions to be uncertain. The system was subject to several uncertain condition scenarios of exposure, conditional, usage, and stress. Each scenario was modeled with its associated occurrence probability. For *stochastic maintenance duration*, Khatab et al. (2017) studied a selective maintenance problem and the maintenance duration was seen as a stochastic factor. Liu et al. (2022) considered a multi-component maintenance and repairmen assignment problem with stochastic maintenance durations. For *stochastic deterioration*, Alaswad and Xiang (2017) reviewed and summarized commonly used stochastic deterioration models for stochastically deteriorating systems, such as the Wiener process, Gamma process, and Inverse Gaussian process. This motivates us to treat RUL as random values respecting stochastic degradation.

To the best of our knowledge, none of the aforementioned studies on non-deterministic maintenance optimization integrate RUL prognostics, component redundancy, system availability, and production. In particular, stochastic degradation has been modeled for single-component systems using stochastic processes, but it has not been applied to complex systems.

## 2.2. Maintenance Optimization Methods

Various approaches have been proposed for maintenance decision-making such as analytic-based methods (Azizi and Salari, 2023), simulation-based approaches (Cao et al., 2023; Koopmans and de Jonge, 2023), and optimization. In this paper, we focus on maintenance optimization, hence in this section, we review related optimization methods in the literature to properly position our work.

For deterministic maintenance optimization, most recent works propose non-linear formulations or (meta-)heuristics due to problem complexity. Genetic algorithm-based heuristics are proposed in the aforementioned Camci (2009, 2015); Camci et al. (2019); Cai et al. (2022). An immune particle swarm algorithm is devised in Wu et al. (2020); Wang et al. (2022). Moreover, Xiao et al. (2016) assumed that a given set of jobs were processed on a series system. A genetic algorithm was used to determine the production schedule and the PM interval to minimize the failures. Alrabghi and Tiwari (2016) studied non-identical multi-component systems and modeled complex interactions between maintenance strategies and their effects on assets. A method integrating Witness optimizer and simulated annealing was proposed to minimize the total cost. Hajej et al. (2020) studied an integrated maintenance and production strategy based on the degradation level of manufacturing and re-manufacturing units. A two-stage non-linear programming model was proposed to minimize the total cost of production, maintenance, and carbon emission, with the support of an evolutionary algorithm. Chen et al. (2021) extracted the key degradation features from off-line raw data and trained them by a support vector regression model and a long short-term memory network to obtain RUL values. A hybrid grey wolf algorithm was proposed for integrating the two models to enhance prediction robustness. Cui et al. (2021) proposed a data-driven quantitative method to obtain system production loss. A PM decision model was established to minimize the maintenance and production cost, and the optimal maintenance policy was exploited based on an approximate dynamic programming algorithm. Obaidat and Liao (2021) performed scheduled maintenance and unscheduled CM at the time of a true alarm for a discrete manufacturing system. The optimal scheduled maintenance time and sampling parameters are determined by solving mixed-integer non-linear programming to minimize the long-run cost rate. For more details, one can refer to the invited review of Alrabghi and Tiwari (2015) and Shafiee and Sørensen (2019).

For stochastic maintenance optimization, the mentioned Khatab and Aghezzaf (2016) proposed a stochastic non-linear programming model and used identified distributions for model approximation. Khatab et al. (2017) further proposed a chance-constrained programming model that uses the expected value of stochastic durations for approximation. In Ghorbani et al. (2022), Shahraki et al. (2020), and Liu et al. (2022), the authors used the scenarios of uncertain factors for stochastic model approximation. From the existing stochastic solution methods for maintenance optimization, we only see chance-constrained programming in Khatab et al. (2017). This kind of stochastic optimization method allows that some constraints in a model hold with some given probability, and is thus called stochastic probabilistic or chance-constrained programming.

Compared to non-linear formulations and meta-heuristics for deterministic maintenance optimization, linear formulations that provide exact results are limited in the existing literature. Furthermore, for stochastic optimization methods that deal with uncertainty in RUL values, none have been proposed that integrate all the problem characteristics mentioned in Section 2.1.

In conclusion, we briefly summarize the limitations of the state-of-the-art, which corresponds to our research motivations.

- Maintenance optimization leveraging RUL prognostics for complex production systems is rarely studied.
- Describing multiple system availability via component redundancy (backups) has not been considered before.
- Stochastic RUL decrease rates have not been studied in multi-component complex systems and maintenance.
- For deterministic maintenance optimization, linear programming formulations are limited.
- For stochastic maintenance optimization, no related model or algorithm is proposed for the considered problem.

The above research gaps motivate us to consider novel RUL-based preventive maintenance optimization for such a factory with *multiple complex systems*, where *component-level RUL*, *component redundancy*, *system-level availability*, and *production flow balance* are integrated. The purpose is to ensure the continuous operation of the factory by formulating a *MILP model* to get optimal maintenance plans for the addressed DRPMO. Beyond the specifically mentioned characteristics, we model the SRPMO problem using a chance-constrained programming formulation with stochastic RUL decrease rates. Since chance-constrained programming cannot be solved directly, we represent chance constraints by linear approximations in a suitable manner. For clarity, we summarize the most closely related works in Table 1. This table highlights the main differences between our work and the existing literature.

**Table 1**  
Benchmarking our research by recent articles.

Literature	Problem characteristics				Problem solving	
	Component redundancy	RUL usage	System availability	Uncertainty	Deterministic method	Stochastic method
Do et al. (2015)	-	inspection	single	-	SL	-
Zhang et al. (2018)	-	inspection	single	-	SL	-
Chen et al. (2019)	-	combination	single	-	SL	-
Liang et al. (2019)	-	combination	single	-	SL	-
Camci (2009)	-	constraints	single	-	GA	-
Camci (2015)	-	constraints	single	-	GA	-
Bülbül et al. (2019)	-	constraints	single	-	LP	-
Camci et al. (2019)	-	constraints	single	-	NLP	-
Wu et al. (2020)	-	constraints	single	-	NLP, IPSO	-
Xiao et al. (2016)	-	-	single	-	GA	-
Chen et al. (2021)	-	-	single	-	GWA	-
Alrabghi and Tiwari (2016)	-	-	multiple	-	SA	-
Lei and Sandborn (2018)	-	-	multiple	-	SL	-
Knofius et al. (2019)	-	-	multiple	-	NLP	-
Dong et al. (2020)	-	-	multiple	-	HA	-
Wu and Castro (2020)	-	-	multiple	-	SL	-
Cai et al. (2022)	-	-	multiple	-	GA	-
Wang et al. (2022)	-	-	multiple	-	PSO	-
Zhang et al. (2022b)	-	-	multiple	-	HA	-
Azizi and Salari (2023)	-	-	multiple	-	ML	-
Zhang et al. (2023b)	-	-	multiple	-	MDP	-
Koopmans and de Jonge (2023)	-	-	multiple	-	MDP	-
Xenos et al. (2016)	✓	-	multiple	-	LP	-
Ye et al. (2019)	✓	-	multiple	-	MCS	-
Khatab and Aghezaf (2016)	-	-	-	✓	-	SNLP, Dis
Khatab et al. (2017)	-	-	-	✓	-	CCP, Dis
Alaswad and Xiang (2017)	-	-	-	✓	-	SP
Ghorbani et al. (2022)	-	-	-	✓	-	SNLP, Dis
Our Work	✓	constraints	multiple	✓	MILP	CCP, Dis

**Legends:**

For problem characteristics, RUL usage (inspection): determine the next inspection time via RUL information; RUL usage (combination): determine maintenance strategy by the combination of RUL and degradation information; RUL usage (constraints): formulate models with RUL-based constraints; System availability (single): single-level availability; System availability (multiple): use component-level maintenance to identify system-level availability.

For problem solving, SL: simulation; GA: genetic algorithm; LP: linear programming; NLP: non-linear programming; IPSO: immune particle swarm optimization; GWA: grey wolf algorithm; SA: simulated annealing; HA: heuristic approach; ML: maximum likelihood; MDP: Markov decision process; MCS: Monte Carlo simulation; SNLP: stochastic non-linear programming; Dis: probabilistic distribution; CCP: chance-constrained programming; SP: stochastic process.

### 3. Deterministic Problem Description and Formulation

In this section, we focus on the deterministic version of the RPMO. In Section 3.1, we describe in detail the DRPMO problem and its main features. Then in Section 3.2, we formulate a MILP model for solving this problem. In Section 3.3, we briefly comment on the flexibility of our model to adapt to different factory and system architectures.

We make the following assumptions: (i) Systems in the same process are identical in terms of the structure, while systems in different processes may be different. (ii) RUL prognostics are provided as values, including the initial RUL, RUL decrease rate, and RUL threshold. (iii) Preventive maintenance of a component takes place if its RUL is not greater than a prescribed threshold at the beginning of a period. (iv) The RUL threshold is provided by experts and deciding the threshold is not within the scope of this work. (v) The RUL of an unused component decreases relatively slowly compared to the length of its life cycle. For clarity, we summarize the notations used in Table 2.

**Table 2**  
Notations and definitions.

Problem sets	
$\mathcal{T}$	set of periods in the planning horizon (index: $t \in \mathcal{T}$ )
$\mathcal{P}$	set of processes (index: $p \in \mathcal{P}$ )
$S_p$	set of complex systems in process $p$ (index: $s \in S_p$ )
$\mathcal{K}_p$	set of component types in the systems of process $p$ (index: $k \in \mathcal{K}_p$ )
$J_{pk}^{\S}$	set of components belonging to component type $k$ in process $p$ (index: $j \in J_{pk}$ )
Parameters	
$o_{pskj}^{\dagger}$	initial RUL of component $j$
$\alpha_{pk}$	RUL decrease rate if a component is on standby
$\beta_{pk}^t$	RUL decrease rate if a component is in operation, depending on the current period
$r_{pk}$	restored RUL once a component has been maintained
$md_{pk}$	maintenance duration of a component
$\tau_{pk}$	RUL threshold of a component
$c^I$	unit production-loss penalty
$c^{II}$	unit maintenance cost
$b$	budget for maintenance actions
$n$	maximum number of maintenance actions allowed in the factory within a period
$AF^0$	factory maximum or ideal production volume
$\overline{M}$	a sufficiently big value
Binary variables	
$X_{pskj}^t$	= 1 if the RUL of component $j$ reaches the threshold in period $t$ ; = 0 otherwise
$Y_{pskj}^t$	= 1 if component $j$ is working in period $t$ ; = 0 otherwise
$U_{psk}^t$ *	= 1 if component type $k$ has at least one component in operation in period $t$ ; = 0 otherwise
$V_{ps}^t$	= 1 if system $s$ in process $p$ is working in period $t$ ; = 0 otherwise
Continuous variables	
$R_{pskj}^t$	RUL of component $j$ in period $t$
$AP_p^t$	production volume of process $p$ in period $t$
$AF^t$	production volume of the factory in period $t$

$\S$  Subscript  $pk$  denotes component type  $k$  of the systems in process  $p$ .

$\dagger$  Subscript  $pskj$  denotes that component  $j$  belongs to type  $k$  in system  $s$ , and this system belongs to process  $p$ .

\* Subscript  $psk$  denotes component type  $k$  of system  $s$  in process  $p$ .

#### 3.1. Description of the DRPMO Problem

First of all, we present the schematic diagram of a generic factory in Figure 2 with the following interpretations.

- **Factory level.** The factory has a series-parallel architecture. The illustrative example shown in Figure 2 contains a series of four operating processes, i.e.,  $|\mathcal{P}| = 4$ .

- *Process level.* A process either contains one complex system (such as process 4) or consists of parallel complex systems (processes 1, 2, and 3). For instance, process 1 has two parallel systems, that is,  $|S_p| = 2$  when  $p = 1$ .
- *System level.* A complex system incorporates a series of types of components and such a component type fulfills a specific function within the system. In Figure 2, the zoomed-in system in process 2 contains four types of components (functions), represented as circles, triangles, diamonds, and squares, respectively. That is,  $|\mathcal{K}_p| = 4$  when  $p = 2$ . Note that parallel systems in the same process have the same structure.
- *Component type or function level.* Each component type is allowed to have redundant components to enhance system reliability, but only one component is required for operation, namely, the 1-out-of- $n$  rule. Take again the zoomed-in system as an example, the first type represented by circles requires 1-out-of-4 to guarantee the first function in this system. Here  $|\mathcal{J}_{pk}| = 4$  if  $p = 2, k = 1$ .
- *Component level.* Components are the foundation of the entire framework and may need maintenance. Due to component redundancy, a component can thus be in exactly one of the states: working, standby, or maintenance.

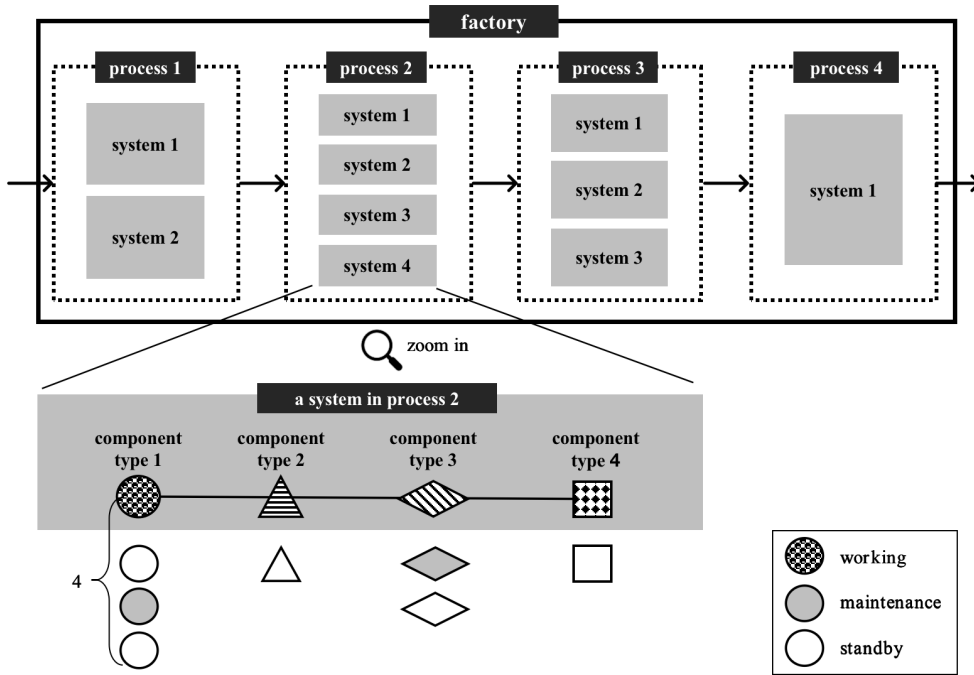


Figure 2: Schematic diagram of a factory.

The core question in our problem is how to plan component-level maintenance to minimize the production loss over the planning horizon. To answer this question, we now explain the main features of our study.

### 3.1.1. RUL-based Maintenance Strategy and RUL Evolution

*RUL-based maintenance strategy.* The basic idea of the proposed RUL-based maintenance is that once the RUL of a component is down to the level of a given threshold, a PM is required for this component. Such a component cannot be used before the maintenance is completed. We provide a total budget for maintenance within the planning horizon and specify a maximum number of maintenance actions per period, as we do not want to have too much maintenance taking place in any period.

*RUL evolution over time.* Since we allow redundant components as backups for each component type and we have the 1-out-of- $n$  rule, we must keep track of RUL values over the optimization horizon. Figure 3 explains how we track RUL values over time. The two components belonging to the same type can be seen as the components noted by triangles in Figure 2. We see from Figure 3 that during periods  $\{1, 2, 3\}$ , component 1 is working with its RUL

decreasing. Meanwhile, component 2 is on standby with its RUL decreasing very slowly (but not constant). In period 4, component 2 is in operation while component 1 is on standby. At the beginning of period 9, the RUL of component 1 reaches the threshold, and therefore a maintenance action is now required for component 1. The RUL of component 1 is restored to a high value once the maintenance is finished, i.e., at the beginning of period 10. As component 2 is available during the maintenance of component 1, this backup ensures the uninterrupted operation of this component type. It is important to note that this figure cannot be obtained before the optimization has been carried out.

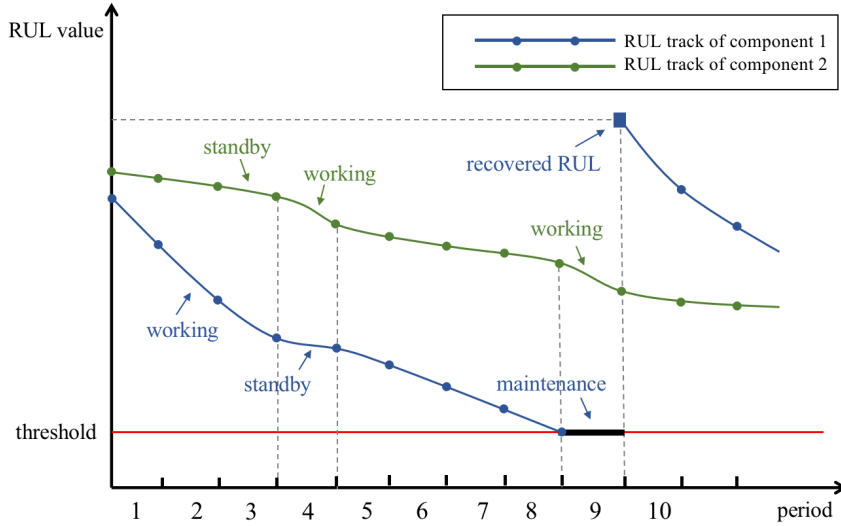


Figure 3: Illustration of RUL evolution.

In summary, we use the following piece-wise function to model the RUL evolution of each component from one period to the next:

$$R_{pskj}^{t+1} = \begin{cases} (1 - \alpha_{pk}) \cdot R_{pskj}^t, & \text{if component } j \text{ is on standby in period } t \\ (1 - \beta_{pk}^t) \cdot R_{pskj}^t, & \text{if component } j \text{ is in operation in period } t \\ r_{pk}, & \text{if component } j \text{ has just been maintained} \end{cases}$$

This function captures the following features, with  $\alpha_{pk} < \beta_{pk}^t < R_{pskj}^t$ . (i) If a component (belonging to type  $k$  in the systems of process  $p$ ) is on standby during period  $t$ , its RUL value decreases by  $\alpha_{pk} \cdot R_{pskj}^t$ . This value depends on the component's current RUL value  $R_{pskj}^t$  and the parameter  $\alpha_{pk} \in (0, 1)$  denotes the wearing coefficient. (ii) If the component is in operation in period  $t$ , its RUL value reduces by  $\beta_{pk}^t \cdot R_{pskj}^t$ , where  $\beta_{pk}^t \in (0, 1)$  is the degradation coefficient of components in period  $t$ . Unlike in the first case, the RUL reduction of a working component depends not only on its current RUL but also on the current period. (iii) If a component is maintained, its RUL is restored to  $r_{pk}$  after a given maintenance duration.

### 3.1.2. Interaction Among Levels

Remember that the objective is to minimize the impact of component-level maintenance on production capacity, namely, **minimize the total penalty due to production loss for all planning periods**. This goal allows the entire factory to maximize its production while keeping maintenance actions in mind. **It requires us to clarify the interactions between levels and identify the potential impact of maintenance on them.**

*Type or function availability.* One component of each type (1-out-of- $n$ ) must operate during a period. For example, in Figure 3, we see that when one component is working, the other component(s) of the same type can be on standby or in maintenance. We note that two situations can put a system out of service: (i) If there is a component type with only one component, then maintaining this component forces the system to stop. (ii) All components belonging to one

type are under maintenance, which is possible due to longer maintenance durations. Obviously, increased component redundancy and shorter maintenance durations reduce the chance of system downtime.

*Complex system availability.* A complex system is usable in a period if and only if all its functions (component types) are operational. For instance, the zoomed-in system in Figure 3 requires four functions to operate simultaneously.

*Process production capacity and production flow.* The production capacity of each process is limited by the number of systems available in it. In Figure 3, the two-system process 1 loses half of its production capacity if one system is out of service. Besides, the flow balance among different processes must be satisfied respecting the sequence. That is, the production volume of a subsequent process should be no less than the output of the previous process.

*Factory production volume.* Factory production depends on the minimum production capacity among the processes. In Figure 2, if we define the ideal yield capacity of the four processes equivalently to be 100, then the ideal production of the factory in each period is 100. As systems may run in parallel in some processes, we thus calculate that the capacities of the individual systems in processes 1, 2, 3, and 4 are 50, 25, 33.3, and 100, respectively. For example, if one system in process 1, one system in process 2, and two systems in process 3 are unavailable during a period, it follows that in this period, the factory production volume is  $\min\{(100 - 1 \times 50), (100 - 1 \times 75), (100 - 2 \times 33.3), 100\} = 33.3$ .

### 3.2. MILP Formulation of the DRPMO Problem

In this section, we formulate a MILP model of the DRPMO problem and denote it as **M1**. The objective function (1) is to minimize the total penalty that is equal to the unit penalty value times the total production loss.

$$\mathbf{M1} : \min C^{penalty} = c^I \cdot \sum_{t \in \mathcal{T}} (AF^0 - AF^t) \quad (1)$$

The optimization of (1) is carried out subject to the following groups of constraints, corresponding exactly to the description in Section 3.1.1 and Section 3.1.2.

**RUL-based maintenance constraints.** Constraints (2) and (3) mean that a maintenance activity is required when the RUL of a component is no greater than the corresponding threshold. Constraints (4) ensure that a component cannot be used during the maintenance duration. Constraints (5) respect the 1-out-of- $n$  rule for each component type or function. Constraint (6) guarantees that the total maintenance expense does not exceed the given budget. Constraints (7) require that the number of maintenance actions in the factory is limited in each period. Note that RUL values are not larger than the maximum value between the initial RUL  $o_{pskj}$  and the recovered RUL  $r_{pk}$ . Therefore, the  $\bar{M}$  in the following constraints can take a value of  $\max\{o_{pskj}, r_{pk}\}$  that is sufficiently large to play this role.

$$R_{pskj}^t \geq \tau_{pk} - \bar{M} \cdot X_{pskj}^t, \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \quad (2)$$

$$R_{pskj}^t \leq \tau_{pk} + \bar{M} \cdot (1 - X_{pskj}^t), \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \quad (3)$$

$$\sum_{t' \in [t, t+md_{pk}-1]} Y_{pskj}^{t'} \leq 1 - X_{pskj}^t, \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in [1, |\mathcal{T}| - md_{pk} + 1] \quad (4)$$

$$\sum_{j \in \mathcal{J}_{pk}} Y_{pskj}^t \leq 1, \forall k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \quad (5)$$

$$c^{II} \sum_{j \in \mathcal{J}_{pk}} \sum_{k \in \mathcal{K}_p} \sum_{s \in \mathcal{S}_p} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} X_{pskj}^t \leq b \quad (6)$$

$$\sum_{j \in \mathcal{J}_{pk}} \sum_{k \in \mathcal{K}_p} \sum_{s \in \mathcal{S}_p} \sum_{p \in \mathcal{P}} X_{pskj}^t \leq n, \forall t \in \mathcal{T} \quad (7)$$

**RUL evolution constraints.** Constraints (8) and (9) state that if a component is on standby during a period, then its RUL reduces by a value of  $\alpha_{pk} \cdot R_{pskj}^t$ . Constraints (10) and (11) require that if a component is in operation during a period, its RUL decreases by a larger value of  $\beta_{pk}^t \cdot R_{pskj}^t$ . Constraints (12) and (13) ensure that when a component has been maintained, its RUL value restores to  $r_{pk}$ . Note that the initial RUL from the manufacturer of each component is provided in constraints (14).

$$R_{pskj}^{t+1} \geq (1 - \alpha_{pk}) \cdot R_{pskj}^t - \bar{M} \cdot (X_{pskj}^t + Y_{pskj}^t), \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \setminus \{|\mathcal{T}|\} \quad (8)$$

$$R_{pskj}^{t+1} \leq (1 - \alpha_{pk}) \cdot R_{pskj}^t + \bar{M} \cdot (X_{pskj}^t + Y_{pskj}^t), \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \setminus \{|\mathcal{T}|\} \quad (9)$$

$$R_{pskj}^{t+1} \geq (1 - \beta_{pk}^t) \cdot R_{pskj}^t - \overline{M} \cdot (1 - Y_{pskj}^t), \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \setminus \{|\mathcal{T}|\} \quad (10)$$

$$R_{pskj}^{t+1} \leq (1 - \beta_{pk}^t) \cdot R_{pskj}^t + \overline{M} \cdot (1 - Y_{pskj}^t), \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \setminus \{|\mathcal{T}|\} \quad (11)$$

$$R_{pskj}^{t+1} \geq r_{pk} - \overline{M} \cdot (1 - X_{pskj}^t), \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \setminus \{|\mathcal{T}|\} \quad (12)$$

$$R_{pskj}^{t+1} \leq r_{pk} + \overline{M} \cdot (1 - X_{pskj}^t), \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \setminus \{|\mathcal{T}|\} \quad (13)$$

$$R_{pskj}^1 = o_{pskj}, \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P} \quad (14)$$

**Type or function availability constraints.** Constraints (15) state that a component type is not functional if all the components it contains are not working. Together with constraints (5), we know that a component type can be used if and only if one component of that type is in operation, i.e.,  $\exists Y_{pskj}^t = 1$ .

$$U_{psk}^t \leq \sum_{j \in \mathcal{J}_{pk}} Y_{pskj}^t, \forall k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \quad (15)$$

**System availability constraints.** Constraints (16) and (17) require that all the component types in it are available to guarantee system availability, namely,  $\forall U_{psk}^t = 1$ . Constraints (18) make sure that if a system is not working, all the components in it are also not working. Note that, in constraints (18), the number of in-operation components within each system is bounded by  $\sum_{k \in \mathcal{K}_p} |\mathcal{J}_{pk}|$ .

$$\frac{1}{|\mathcal{K}_p|} \sum_{k \in \mathcal{K}_p} U_{psk}^t \leq V_{ps}^t + \sum_{k \in \mathcal{K}_p} (1 - U_{psk}^t), \forall s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \quad (16)$$

$$U_{psk}^t \geq V_{ps}^t, \forall k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \quad (17)$$

$$\sum_{j \in \mathcal{J}_{pk}} \sum_{k \in \mathcal{K}_p} Y_{pskj}^t \leq \sum_{k \in \mathcal{K}_p} |\mathcal{J}_{pk}| \cdot V_{ps}^t, \forall s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \quad (18)$$

**Process capacity and production flow constraints.** Constraints (19) provide the upper bound of the production capacity of each process, which depends on the number of available systems in it. Constraints (20) guarantee the production flow such that the production quantity of the subsequent process  $p + 1$  is greater or equal to that of the previous process  $p$ , due to the serial structure in the factory.

$$AP_p^t \leq \frac{AF^0}{|\mathcal{S}_p|} \cdot \sum_{s \in \mathcal{S}_p} V_{ps}^t, \forall p \in \mathcal{P}, t \in \mathcal{T} \quad (19)$$

$$AP_p^t \leq AP_{p+1}^t, \forall p \in \mathcal{P} \setminus \{|\mathcal{P}|\}, t \in \mathcal{T} \quad (20)$$

**Industry production constraints.** Constraints (21) ensure that the production volume of the factory is equal to the minimum production level among the processes. These constraints are linearized as formula  $AF^t \leq AP_p^t$  and this achieves the same outcome due to the form of the objective function. Constraints (22) use the industrial production quantity to bound the number of systems operating in each process. Together with constraints (19), we give lower and upper bounds on the number of operating systems. Take Figure 2 as an example, we suppose that  $AF^t = 33.3$  in period  $t$ . We can calculate, such as process 2, it systems have a capacity of  $AF^0/|\mathcal{S}_2| = 25$ , then we have  $1.33 \leq \sum_{s \in \mathcal{S}_2} V_{2s}^t \leq 1.33 + 1 = 2.33$ , that is,  $\sum_{s \in \mathcal{S}_2} V_{2s}^t = 2$ . Therefore, precisely two systems in process 2 must be working to guarantee this production volume. Similar ideas are used for other processes.

$$AF^t = \min_{p \in \mathcal{P}} [AP_p^t], \forall t \in \mathcal{T} \quad (21)$$

$$\sum_{s \in \mathcal{S}_p} V_{ps}^t \leq AF^t / \frac{AF^0}{|\mathcal{S}_p|} + 1, \forall p \in \mathcal{P}, t \in \mathcal{T} \quad (22)$$

Finally, constraints (23) and (24) specify the nature of the decision variables. Note that  $\mathbb{R}^+$  denotes non-negative real numbers.

$$X_{pskj}^t, Y_{pskj}^t, U_{psk}^t, V_{ps}^t \in \{0, 1\}, \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \quad (23)$$

$$R_{pskj}^t, AP_p^t, AF^t \in \mathbb{R}^+, \forall j \in \mathcal{J}_{pk}, k \in \mathcal{K}_p, s \in \mathcal{S}_p, p \in \mathcal{P}, t \in \mathcal{T} \quad (24)$$

### 3.3. Model Flexibility

In this section, we briefly discuss the flexibility of our MILP model and how it can be adapted to factories and systems with different architectures. Note that it is not the focus of this paper but for future research directions.

As described in Section 3.1, each system has a series structure and requires only one component of each type to be operational in order to ensure the operation of the whole system. This is the so-called 1-out-of- $n$  rule for each component type. Suppose now that the system no longer has a series structure, so that, for example, the system changes from a series structure to a series-parallel one, as depicted in Figure 4. In particular, we now need three components of type 1 running in parallel to ensure the operation of the system, which is a  $k$ -out-of- $n$  rule for that component type.

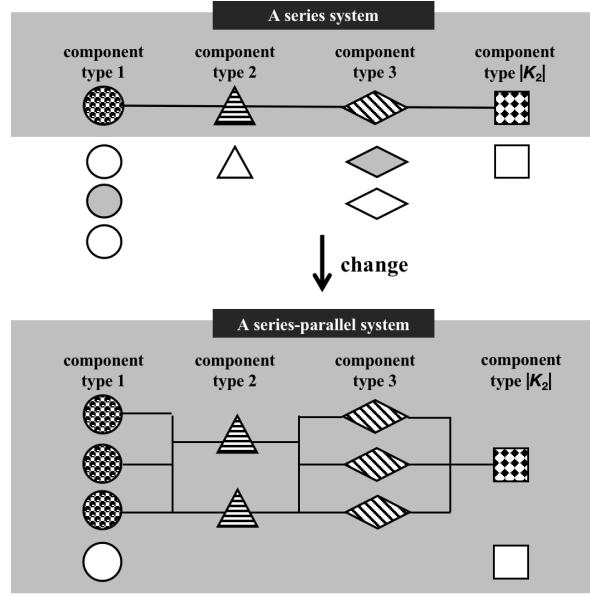


Figure 4: Structure of a system changes from series to series-parallel.

To adapt our model M1 to this changed architecture, we only need to modify constraints (5) and (15) appropriately. First, we need to identify the changed system, that is, to find the corresponding process  $p$  and system  $s$ . Second, we change constraint (5) for  $p$  and  $s$  to ensure that the number of working components of type 1 is no more than three (instead of one). Third, constraint (15) for  $p$  and  $s$  is changed to check whether  $\sum_{j \in \mathcal{J}_{pk}} Y_{pskj}^t = 3$  holds or not, and if not, then this component type cannot guarantee system operations, i.e.,  $U_{psk}^t$  is set to zero.

Naturally, this is only one example to show the flexibility of our model. Many other process changes or system changes can be accommodated by the model for a variety of applications. We carry out computational experiments in Section 5.2.1 to demonstrate this model's flexibility.

## 4. Stochastic Problem Formulation and Approximation

In this section, we introduce uncertainty in the modeling by considering the RUL decrease rates as stochastic so that the RUL evolution is random, as done in reliability theory. In Section 4.1, we formulate the SRPMO problem as a chance-constrained programming model. In Section 4.2, we design an approximation solution approach, and in Section 4.3, we conduct attribute analysis.

#### 4.1. Chance-constrained Programming Formulation

Under uncertainty, the RUL function for on-standby and in-operation conditions introduced in Section 3.1 is replaced by the following function:

$$R_{pskj}^{t+1} = \begin{cases} (1 - \tilde{\alpha}_{pk}) \cdot R_{pskj}^t, & \text{if a component is on standby in period } t \\ (1 - \tilde{\beta}_{pk}^t) \cdot R_{pskj}^t, & \text{if a component is in operation in period } t \end{cases}$$

where  $\tilde{\alpha}_{pk} \in (0, 1)$  denotes the stochastic decrease rate if a component is on standby, and  $\tilde{\beta}_{pk}^t \in (0, 1)$  denotes the stochastic decrease rate if a component is in operation.

Let us first consider the RUL decrease when a component is on standby. We expect the RUL evolution to behave nearly as expected, within a given confidence level even under stochastic decrease rates. For this reason, we set up chance constraints that hold with some given probabilities. For the standby condition, we thus replace constraints (8) and (9) in M1 by chance constraints (25) and (26), with the same ranges:

$$\mathbb{P} \left[ R_{pskj}^{t+1} \geq (1 - \tilde{\alpha}_{pk}) \cdot R_{pskj}^t - \Pi_{\alpha} \right] \geq \Gamma_{\alpha 1} \quad (25)$$

$$\mathbb{P} \left[ R_{pskj}^{t+1} \leq (1 - \tilde{\alpha}_{pk}) \cdot R_{pskj}^t + \Pi_{\alpha} \right] \geq \Gamma_{\alpha 2} \quad (26)$$

where  $\Pi_{\alpha} := \bar{M} \cdot (X_{pskj}^t + Y_{pskj}^t)$  and  $\Gamma_{\alpha 1}, \Gamma_{\alpha 2} \in (0, 1)$  are the probabilities that the respective chance constraints hold.

We have similar chance-constraint formulations for the case when a component is in operation. We replace (10) and (11) in M1 by chance constraints (27) and (28):

$$\mathbb{P} \left[ R_{pskj}^{t+1} \geq (1 - \tilde{\beta}_{pk}^t) \cdot R_{pskj}^t - \Pi_{\beta} \right] \geq \Gamma_{\beta 1} \quad (27)$$

$$\mathbb{P} \left[ R_{pskj}^{t+1} \leq (1 - \tilde{\beta}_{pk}^t) \cdot R_{pskj}^t + \Pi_{\beta} \right] \geq \Gamma_{\beta 2} \quad (28)$$

where  $\Pi_{\beta} := \bar{M} \cdot (1 - Y_{pskj}^t)$  and  $\Gamma_{\beta 1}, \Gamma_{\beta 2} \in (0, 1)$  are the probabilities that the respective chance constraints hold.

With stochastic decrease rates  $\tilde{\alpha}_{pk}$  and  $\tilde{\beta}_{pk}^t$  and constraints (25) - (28), we propose a chance-constrained programming model for the SRPMO, denoted **CCP**.

$$\text{CCP: } \min C^{\text{penalty}} = c^I \cdot \sum_{t \in \mathcal{T}} (AF^0 - AF^t)$$

$$\text{Subject to: } (2) - (7), (12) - (24), (25) - (28)$$

Due to the nature of chance constraints, the model CCP cannot be solved directly. In the next section, we thus design effective methods to compute approximate solutions.

#### 4.2. Approximation of the CCP Model

We assume that  $\tilde{\alpha}_{pk}$  are random variables following normal distributions. The choice of normal distributions is reasonable in this context because we normally expect the stochastic decrease rate to take values near its mean, so values further away from the mean should have lower probabilities. Under this assumption, we prove the following propositions for the proposed chance constraints. These results will be used in the approximation of the model CCP.

Propositions 1 and 2 are related to  $\tilde{\alpha}_{pk}$  and thus are concerned with chance constraints (25) and (26).

**Proposition 1.** *If stochastic decrease rate  $\tilde{\alpha}_{pk}$  follows a normal distribution, i.e.,  $\tilde{\alpha}_{pk} \sim \mathcal{N}(\mu, \sigma^2)$ , then chance constraint (25) can be approximated by the linear formulation*

$$R_{pskj}^{t+1} \geq [(1 - \mu - \sigma \cdot \Phi^{-1}(1 - \Gamma_{\alpha 1})) \cdot R_{pskj}^t - \Pi_{\alpha}], \quad (29)$$

where  $\Phi(\cdot)$  denotes the cumulative distribution function of the standard normal distribution.

PROOF. Chance constraint (25) is equivalent to:

$$\mathbb{P} \left[ \tilde{\alpha}_{pk} \cdot R_{pskj}^t \geq R_{pskj}^t - R_{pskj}^{t+1} - \Pi_{\alpha} \right] \geq \Gamma_{\alpha 1} \quad (30)$$

With the fact that  $R_{pskj}^t \geq \tau_{pk} > 0$ , then constraint (30) can be written as:

$$\mathbb{P} \left[ \tilde{\alpha}_{pk} \geq \frac{R_{pskj}^t - R_{pskj}^{t+1} - \Pi_\alpha}{R_{pskj}^t} \right] \geq \Gamma_{\alpha 1} \quad (31)$$

As  $\tilde{\alpha}_{pk} \sim \mathcal{N}(\mu, \sigma^2)$ , we have that  $\frac{\tilde{\alpha}_{pk} - \mu}{\sigma} \sim \mathcal{N}(0, 1)$ , and thus constraint (31) can be standardized as:

$$\mathbb{P} \left[ \frac{\tilde{\alpha}_{pk} - \mu}{\sigma} \geq \frac{\frac{R_{pskj}^t - R_{pskj}^{t+1} - \Pi_\alpha}{R_{pskj}^t} - \mu}{\sigma} \right] \geq \Gamma_{\alpha 1} \quad (32)$$

namely:

$$\mathbb{P} \left[ \frac{\tilde{\alpha}_{pk} - \mu}{\sigma} \geq \frac{(1 - \mu) \cdot R_{pskj}^t - R_{pskj}^{t+1} - \Pi_\alpha}{\sigma \cdot R_{pskj}^t} \right] \geq \Gamma_{\alpha 1} \quad (33)$$

or equivalently as:

$$1 - \mathbb{P} \left[ \frac{\tilde{\alpha}_{pk} - \mu}{\sigma} \leq \frac{(1 - \mu) \cdot R_{pskj}^t - R_{pskj}^{t+1} - \Pi_\alpha}{\sigma \cdot R_{pskj}^t} \right] \geq \Gamma_{\alpha 1} \quad (34)$$

Now, constraint (34) can be written as follows using the properties of the standard normal distribution:

$$\Phi \left[ \frac{(1 - \mu) \cdot R_{pskj}^t - R_{pskj}^{t+1} - \Pi_\alpha}{\sigma \cdot R_{pskj}^t} \right] \leq 1 - \Gamma_{\alpha 1}. \quad (35)$$

Because the right hand of (35) is revealed as a given value, we thus have the following equivalent constraint:

$$\frac{(1 - \mu) \cdot R_{pskj}^t - R_{pskj}^{t+1} - \Pi_\alpha}{\sigma \cdot R_{pskj}^t} \leq \Phi^{-1}(1 - \Gamma_{\alpha 1}) \quad (36)$$

Using again the fact that  $R_{pskj}^t > 0$ , constraint (36) can be expressed as  $R_{pskj}^{t+1} \geq [(1 - \mu - \sigma \cdot \Phi^{-1}(1 - \Gamma_{\alpha 1})) \cdot R_{pskj}^t - \Pi_\alpha]$ , which is (29) as desired.  $\blacksquare$

**Proposition 2.** *If stochastic decrease rate  $\tilde{\alpha}_{pk}$  follows a normal distribution, i.e.,  $\tilde{\alpha}_{pk} \sim \mathcal{N}(\mu, \sigma^2)$ , then chance constraint (26) can be approximated by the linear formulation*

$$R_{pskj}^{t+1} \leq [(1 - \mu - \sigma \cdot \Phi^{-1}(\Gamma_{\alpha 2})) \cdot R_{pskj}^t + \Pi_\alpha], \quad (37)$$

where  $\Phi(\cdot)$  denotes the cumulative distribution function of the standard normal distribution.

PROOF. Chance constraint (26) is equivalent to:

$$\mathbb{P} \left[ \tilde{\alpha}_{pk} \cdot R_{pskj}^t \leq R_{pskj}^t - R_{pskj}^{t+1} + \Pi_\alpha \right] \geq \Gamma_{\alpha 2} \quad (38)$$

With the fact that  $R_{pskj}^t \geq \tau_{pk} > 0$ , then constraint (38) can be written as:

$$\mathbb{P} \left[ \tilde{\alpha}_{pk} \leq \frac{R_{pskj}^t - R_{pskj}^{t+1} + \Pi_\alpha}{R_{pskj}^t} \right] \geq \Gamma_{\alpha 2} \quad (39)$$

As  $\tilde{\alpha}_{pk} \sim \mathcal{N}(\mu, \sigma^2)$ , we have that  $\frac{\tilde{\alpha}_{pk} - \mu}{\sigma} \sim \mathcal{N}(0, 1)$ , and thus constraint (39) can be standardized as:

$$\mathbb{P} \left[ \frac{\tilde{\alpha}_{pk} - \mu}{\sigma} \leq \frac{\frac{R_{pskj}^t - R_{pskj}^{t+1} + \Pi_\alpha}{R_{pskj}^t} - \mu}{\sigma} \right] \geq \Gamma_{\alpha 2} \quad (40)$$

namely:

$$\mathbb{P} \left[ \frac{\tilde{\alpha}_{pk} - \mu}{\sigma} \leq \frac{(1 - \mu) \cdot R_{pskj}^t - R_{pskj}^{t+1} + \Pi_\alpha}{\sigma \cdot R_{pskj}^t} \right] \geq \Gamma_{\alpha 2} \quad (41)$$

Now, constraint (41) can be written as follows using the properties of the standard normal distribution:

$$\Phi \left[ \frac{(1 - \mu) \cdot R_{pskj}^t - R_{pskj}^{t+1} + \Pi_\alpha}{\sigma \cdot R_{pskj}^t} \right] \geq \Gamma_{\alpha 2} \quad (42)$$

Because the right hand of (42) is revealed as a given value, we have the following equivalent constraint:

$$\frac{(1 - \mu) \cdot R_{pskj}^t - R_{pskj}^{t+1} + \Pi_\alpha}{\sigma \cdot R_{pskj}^t} \geq \Phi^{-1}(\Gamma_{\alpha 2}) \quad (43)$$

Using again the fact that  $R_{pskj}^t > 0$ , constraint (43) can be expressed as  $R_{pskj}^{t+1} \leq [(1 - \mu - \sigma \cdot \Phi^{-1}(\Gamma_{\alpha 2})) \cdot R_{pskj}^t + \Pi_\alpha]$ , i.e., (37). ■

Propositions 3 and 4 are related to  $\tilde{\beta}_{pk}^t$  and are used to approximate chance constraints (27) and (28). We omit the proofs of these two propositions as they follow the same flow as those of Propositions 1 and 2, respectively.

**Proposition 3.** *If stochastic decrease rate  $\tilde{\beta}_{pk}^t$  follows a normal distribution, i.e.,  $\tilde{\beta}_{pk}^t \sim \mathcal{N}(\mu', \sigma'^2)$ , then chance constraint (27) can be approximated by the linear formulation*

$$R_{pskj}^{t+1} \geq [(1 - \mu' - \sigma' \cdot \Phi^{-1}(1 - \Gamma_{\beta 1})) \cdot R_{pskj}^t - \Pi_\beta], \quad (44)$$

where  $\Phi(\cdot)$  denotes the cumulative distribution function of the standard normal distribution.

**Proposition 4.** *If stochastic decrease rate  $\tilde{\beta}_{pk}^t$  follows a normal distribution, i.e.,  $\tilde{\beta}_{pk}^t \sim \mathcal{N}(\mu', \sigma'^2)$ , then chance constraint (28) can be approximated by the linear formulation*

$$R_{pskj}^{t+1} \leq [(1 - \mu' - \sigma' \cdot \Phi^{-1}(\Gamma_{\beta 2})) \cdot R_{pskj}^t + \Pi_\beta], \quad (45)$$

where  $\Phi(\cdot)$  denotes the cumulative distribution function of the standard normal distribution.

Using the approximating linear constraints (29), (37), (44), and (45), we reformulate model CCP as a linear programming model, denoted **M2**.

$$\mathbf{M2:} \quad \min C^{penalty} = c^I \cdot \sum_{t \in \mathcal{T}} (AF^0 - AF^t)$$

**Subject to:** (2) – (7), (12) – (24), (29), (37), (44), (45)

Although the above model M2 can be solved directly using off-the-shelf solvers, we still have to establish how to identify the probabilities used in the chance constraints. In the next section, we focus on the effects of choosing confidence levels in the chance constraints through attribute analysis.

### 4.3. Attribute Analysis of the Approximations

Let us recall the linear constraints (29) and (37) in M2, which are the approximations of chance constraints in CCP. Remember that  $\Pi_\alpha = \overline{M} \cdot (X_{pskj}^t + Y_{pskj}^t)$ . Clearly, if  $X_{pskj}^t + Y_{pskj}^t \geq 1$ , then the two constraints have no effect on the optimization. Alternatively, if  $X_{pskj}^t = Y_{pskj}^t = 0$ , we obtain the following constraint for the RUL evolution:

$$\left[1 - \mu - \sigma \cdot \Phi^{-1}(1 - \Gamma_{\alpha 1})\right] \cdot R_{pskj}^t \leq R_{pskj}^{t+1} \leq \left[1 - \mu - \sigma \cdot \Phi^{-1}(\Gamma_{\alpha 2})\right] \cdot R_{pskj}^t \quad (46)$$

Unlike the deterministic RUL evolution  $R_{pskj}^{t+1} = (1 - \alpha_{pk}) \cdot R_{pskj}^t$  or its equivalent version in expectation  $R_{pskj}^{t+1} = (1 - \mu) \cdot R_{pskj}^t$ , the RUL evolution determined by (46) varies within an interval that depends on the values of the probabilities  $\Gamma_{\alpha 1}$  and  $\Gamma_{\alpha 2}$ . We obtain insights on the possible cases from Proposition 5 and its Corollary 1.

**Proposition 5.** *Denote the coefficients of the left and right hands of constraint (46) as  $L = 1 - \mu - \sigma \cdot \Phi^{-1}(1 - \Gamma_{\alpha 1})$  and  $U = 1 - \mu - \sigma \cdot \Phi^{-1}(\Gamma_{\alpha 2})$ . Under different combinations of  $\Gamma_{\alpha 1}$  and  $\Gamma_{\alpha 2}$ , we obtain the following cases for (46):*

- *If  $\Gamma_{\alpha 1} + \Gamma_{\alpha 2} = 1$ , then  $L = U$  and we have  $L \cdot R_{pskj}^t = R_{pskj}^{t+1} = U \cdot R_{pskj}^t$ . Constraint (46) thus holds with an equality.*
- *If  $\Gamma_{\alpha 1} + \Gamma_{\alpha 2} < 1$ , then  $L < U$  and we have  $L \cdot R_{pskj}^t \leq R_{pskj}^{t+1} \leq U \cdot R_{pskj}^t$ . Constraint (46) is an inequality with different lower and upper bounds.*
- *If  $\Gamma_{\alpha 1} + \Gamma_{\alpha 2} > 1$ , then  $L > U$ , and constraint (46) is infeasible.*

PROOF. We consider each case in turn.

- If  $\Gamma_{\alpha 1} + \Gamma_{\alpha 2} = 1$ , then  $1 - \Gamma_{\alpha 1} = \Gamma_{\alpha 2}$  and it follows that  $\Phi^{-1}(1 - \Gamma_{\alpha 1}) = \Phi^{-1}(\Gamma_{\alpha 2})$ , so that  $L = 1 - \mu - \sigma \cdot \Phi^{-1}(1 - \Gamma_{\alpha 1}) = 1 - \mu - \sigma \cdot \Phi^{-1}(\Gamma_{\alpha 2}) = U$ , and hence the lower and upper bounds are equal.
- If  $\Gamma_{\alpha 1} + \Gamma_{\alpha 2} < 1$ , we have  $1 - \Gamma_{\alpha 1} > \Gamma_{\alpha 2}$  and hence  $\Phi^{-1}(1 - \Gamma_{\alpha 1}) > \Phi^{-1}(\Gamma_{\alpha 2})$ , so that the lower bound is always smaller than the upper bound.
- If  $\Gamma_{\alpha 1} + \Gamma_{\alpha 2} > 1$ , we have  $1 - \Gamma_{\alpha 1} < \Gamma_{\alpha 2}$  and  $\Phi^{-1}(1 - \Gamma_{\alpha 1}) < \Phi^{-1}(\Gamma_{\alpha 2})$ , and thus  $L > U$ , which clearly implies infeasibility. ■

In order to further explore the relationship between (46) and the corresponding version in expectation  $R_{pskj}^{t+1} = (1 - \mu) \cdot R_{pskj}^t$ , we have the following corollary for some special values of  $\Gamma_{\alpha 1}$  and  $\Gamma_{\alpha 2}$ .

**Corollary 1.** *Based on Proposition 5, we have the following observations for specific combinations of  $\Gamma_{\alpha 1}$  and  $\Gamma_{\alpha 2}$  under  $\Gamma_{\alpha 1} + \Gamma_{\alpha 2} \leq 1$ :*

- *If  $\Gamma_{\alpha 1} < 0.5$  and  $\Gamma_{\alpha 2} < 0.5$ , we have  $L \cdot R_{pskj}^t \leq (1 - \mu) \cdot R_{pskj}^t \leq U \cdot R_{pskj}^t$ , meaning that the expected RUL is bounded below by  $L$  and above by  $U$ .*
- *If  $\Gamma_{\alpha 1} \geq 0.5$ , we have  $(1 - \mu) \cdot R_{pskj}^t \leq L \cdot R_{pskj}^t \leq U \cdot R_{pskj}^t$ , meaning that even the lower bound is greater than the expected RUL.*
- *If  $\Gamma_{\alpha 2} \geq 0.5$ , we have  $L \cdot R_{pskj}^t \leq U \cdot R_{pskj}^t \leq (1 - \mu) \cdot R_{pskj}^t$ , meaning that the expected RUL is greater than the upper bound.*
- *If  $\Gamma_{\alpha 1} = \Gamma_{\alpha 2} = 0.5$ , we have  $L \cdot R_{pskj}^t = (1 - \mu) \cdot R_{pskj}^t = U \cdot R_{pskj}^t$ , meaning that the expected RUL is equal to the deterministic value.*

We conclude this section by noting that Proposition 5 and Corollary 1 also hold mutatis mutandis for the probabilities  $\Gamma_{\beta 1}$  and  $\Gamma_{\beta 2}$  in (44) and (45).

## 5. Computational Experiments

Extensive computational experiments are carried out to evaluate the performance of the proposed solution approaches. In Section 5.1, we present the generation of instances, the experimental design, and the performance indicators used. Sections 5.2 and 5.3 report the outcomes of the proposed solution methods for the DRPMO problem and the SRPMO problem, respectively. The experiments were carried out on a personal computer with Core I7 with 4GB RAM under Microsoft Windows 10 Operating System. The MILP model is coded in a Python3.7 environment and solved using the commercial solver IBM CPLEX12.9.

### 5.1. Instance, Experiment and Indicator Setups

Experimental data, based on industrial applications, are randomly generated and classified into different sets. The parameters that determine the size of the instances are set as follows.

- Number of processes  $|\mathcal{P}|$  in the factory is up to 8.
- Number of systems  $|S_p|$  in process  $p$  is generated as an integer within the interval  $[1, 5]$ .  $|S_p| = 1$  means a single system in process  $p$ , while  $|S_p| > 1$  indicates  $|S_p|$  identical parallel systems.
- Number of component types  $|\mathcal{K}_p|$  in the systems of process  $p$  is generated as an integer from the interval  $[1, 5]$ .
- Number of components  $|J_{pk}|$  belonging to type  $k$  in the systems of process  $p$  is randomly generated in the interval  $[1, 8]$ .  $|J_{pk}| = 1$  represents one component for this type without any backup, while  $|J_{pk}| \geq 2$  means component redundancy.
- Number of periods  $|\mathcal{T}|$  in the planning horizon is up to 24. We generally consider each period to be one week in practice so that 4 periods denote one month, 12 periods denote one season, and 24 periods mean half a year.

Other parameter settings are presented as follows.

- Initial RUL is generated as a real number from the interval  $[10, 50]$ . Recovered RUL after maintenance is set by the interval  $[50, 80]$ .
- RUL threshold is produced as an integer within the interval  $[1, 10]$ .
- For the on-standby condition, RUL decrease rate  $\alpha_{pk}$  is generated as a real number in the interval  $[0.001, 0.005]$ .
- For the in-operation condition, RUL decrease rate  $\beta_{pk}^t$  is generated as a real number from the interval  $[0.20, 0.35]$  and it becomes smaller over time.
- Unit penalty and unit maintenance cost are set as 1 and 100, respectively.
- Maintenance budget is set to 2000 and the maximum number of maintenance actions in a period is fixed to 5.
- Stochastic decrease rate  $\tilde{\alpha}_{pk}$  follows  $\mathcal{N}(\alpha_{pk}, 0.005^2)$ .
- Stochastic decrease rate  $\tilde{\beta}_{pk}^t$  follows  $\mathcal{N}(\beta_{pk}^t, 0.05^2)$ .

We carry out comparative experiments to examine the effectiveness of the proposed methods for both the deterministic version (DRPMO) and the stochastic version (SRPMO). For the DRPMO, we first evaluate our formulated MILP by comparing it with existing literature from three perspectives. We then conduct sensitivity analyses to see the effects of the initial parameters, which include all the key factors that may affect experimental results. For the SRPMO, we first present an illustrative example to visually show the impact of uncertainty on the results. Then, we perform controlled experiments to see the impact of confidence levels in chance constraints. For ease of reference, we summarize all the experiments in Table 3 so that the reader can quickly find related experiments and view detailed experimental settings and result analysis in the corresponding sections.

To report the computational results, we use ‘Obj’ to denote the objective values obtained by our approaches and ‘CPUT(s)’ to represent the corresponding computational time (CPU time in seconds). Indicator ‘NoM’ represents the number of maintenance. Indicator ‘Avg.’ reports the average values (such as average Obj, CPUT, NoM) over groupings of instances. We set a computational time limit of 2 hours (7200 seconds in CPU time) for the proposed models because of practical applicability. We use ‘Gap(%)’ to denote the gap between the current best and the optimal value when this time limit is reached.

**Table 3**

Summary of experiments and corresponding information.

Problems	Sections	Tests	Purposes	Reports
DRPMO	Section 5.2.1	Comparison to literature	Impact of component redundancy Measure of system availability Difference of 1-out-of- $n$ and $k$ -out-of- $n$	Figure 5 Figure 5 Figure 5
DRPMO	Section 5.2.2	Sensitivity analysis	Impact of problem sizes Impact of initial RUL values Impact of RUL thresholds Impact of RUL decrease rates	Table 4 Table 5 Table 6 Table 7
SRPMO	Section 5.3.1	An illustrative example	Impact of uncertainty	Figure 6, Table 8
SRPMO	Section 5.3.2	Sensitivity analysis	Impact of confidence levels	Figure 7, Figure 8

## 5.2. Experiments for the DRPMO Problem

In this section, we focus on evaluating our MILP model M1 by comparing it to the existing literature and by carrying out sensitivity analysis.

### 5.2.1. Comparison to Literature

As there is no literature directly comparable to our proposed approach, we can only achieve comparability to the literature by relaxing some constraints or adopting other maintenance optimization strategies from the literature. Considering that our maintenance optimization problem is RUL-based and exploits component redundancy, we propose comparisons from three different perspectives.

- Impact of component redundancy.** Previous research, such as Do et al. (2015), adopted the use of simulations when considering RUL-based maintenance. As they do not consider component redundancy, the availability of the system thus depends exclusively on the condition of a single component. In other words, due to the absence of component redundancy, there is no optimization to be carried out at the component level. In contrast, since we allow redundancy for each component type, our optimization model must determine for each period which components are in operation, which ones need maintenance, and which ones are on standby. Therefore, this comparison not only show the impact of the presence or absence of component redundancy but also demonstrate the benefit of optimization versus the use of simulations.
- Measure of system availability.** Because complex systems contain multiple components, it is important to use component-level information to assess system availability. For RUL-based maintenance, Lei and Sandborn (2018) assumed that system availability depended on the minimum RUL among all components. This is different from our work because we consider different RUL thresholds for different types of components. We compare these two approaches by optimizing twice, the first with different RUL thresholds of components (our approach) and then with a unique RUL threshold for all components (similar to the approach of Lei and Sandborn (2018)).
- Difference between 1-out-of- $n$  and  $k$ -out-of- $n$ .** Khorshidi et al. (2016) focused on the  $k$ -out-of- $n$  rule for component redundancy. Although we focus on the 1-out-of- $n$  rule, we have illustrated in Section 3.3 the flexibility of our model to accommodate a  $k$ -out-of- $n$  rule, if desired. We report results using the 1-out-of- $n$  rule and  $k$ -out-of- $n$  rule (with  $k = 2$ ), respectively. This not only provides a comparison with the approach of Khorshidi et al. (2016) but also illustrates the versatility of our model.

Figure 5 summarizes the total penalty (optimal objective values) obtained by our approach and by others from the literature, over 10 problem instances. In this figure, the  $x$ -axis is indexed by the instance number and the  $y$ -axis by the optimal value. The black curve represents the results obtained using our approach. The blue, green, and red lines present the results obtained by using the ideas in the aforementioned works.

We make the following observations from Figure 5: (i) with respect to the benefit of mathematical optimization (black and blue curves), we see that our optimization approach outperforms the simulation method from the literature in

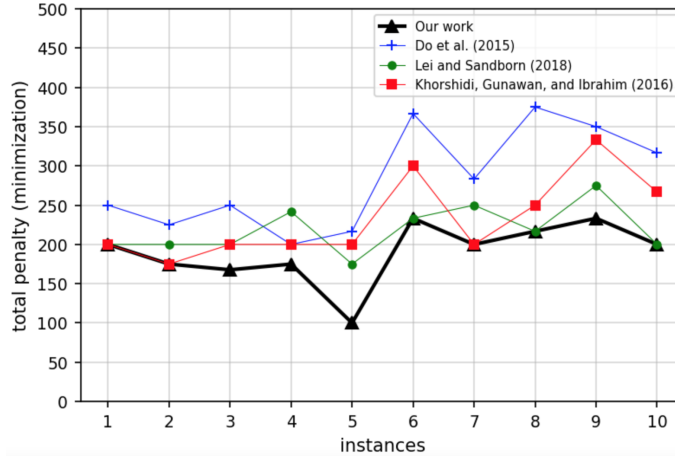


Figure 5: Comparison between our approach and others from the literature.

terms of total penalty minimization for every instance. This is understandable because our optimization takes advantage of component redundancy, providing greater reliability and thus yielding lower total penalty. (ii) With respect to system availability (black and green curves), it is clear that the total penalty of our model is never greater than that of Lei and Sandborn (2018), meaning that our approach is more effective in ensuring system availability. (iii) With respect to the redundancy rules (black and red curves), our approach with the 1-out-of- $n$  rule consistently outperforms the use of the  $k$ -out-of- $n$  rule. This is to be expected because, for each type, requiring more components to operate necessarily leads to increased maintenance. Note that we do not compare their computational times because the results do not show a meaningful difference between the approaches in this regard.

We recall that these comparisons were all carried out using our optimization model with appropriate adjustments to represent each of the four approaches. This further illustrates the flexibility of our optimization model and its potential for application in a variety of industrial contexts.

### 5.2.2. Sensitivity Analysis

In this section, we consider four groups of computational experiments to analyze the effects of different initial parameters on the performance of M1. Computational results are reported in Tables 4-7.

- **Impact of instance sizes (TestA).** We use the number of processes  $|\mathcal{P}|$  and the planning production horizon  $|\mathcal{T}|$  as hyperparameters. The change in the value of  $|\mathcal{P}|$  leads to a change in the internal structure of the factory, which implies the change in other problem-size related parameters such as  $|S_p|$  and  $|\mathcal{K}_p|$ . Therefore, we look at the impact of instance sizes by varying only  $|\mathcal{P}|$  and  $|\mathcal{T}|$ , as we observed clear trends in pre-experiments.
- **Impact of initial RUL values (TestB).** We fix the factory structure and production horizon. For component information, the initial RUL value is the controlled parameter and is generated respectively from three different ranges, namely  $[10, 20]$ ,  $[40, 50]$ , and  $[70, 80]$ , as indicated in the top row of Table 5. A lower initial RUL means a worse starting point because it is closer to the given RUL threshold.
- **Impact of RUL thresholds (TestC).** We fix the factory structure and production horizon. The controlled parameter RUL threshold is taken from the ranges  $[1, 3]$ ,  $[4, 6]$ , and  $[7, 10]$  shown in the top row of Table 6. A larger RUL threshold requires earlier maintenance.
- **Impact of RUL decrease rates (TestD).** We fix the factory structure and production horizon. The controlled parameter decrease rate is generated from three different ranges, namely  $[0.2, 0.25]$ ,  $(0.25, 0.3]$ , and  $(0.3, 0.35]$ , corresponding to the top row of Table 7. A lower degradation rate means that the component's RUL decreases at a slower pace, allowing it to be used for a longer time before maintenance is required.

For TestA in Table 4, we set  $|\mathcal{P}|$  as 4 and 8 and  $|\mathcal{T}|$  as 12 and 24, respectively. We thus have 4 different combinations of  $(|\mathcal{T}|, |\mathcal{P}|)$  to define the instance size, noted from A1 to A4. For example, A1( $|\mathcal{T}| = 12, |\mathcal{P}| = 4$ ) implies an instance

**Table 4**  
Impact of the instance size.

Test A	$ \mathcal{T}  = 12$				$ \mathcal{T}  = 24$			
	no.	Obj	Gap(%)	CPUT(s)	no.	Obj	Gap(%)	CPUT(s)
$ \mathcal{P}  = 4$	A1-1	50	0	13.9	A2-1	0	0	8.1
	A1-2	100	0	26.7	A2-2	200	0	28.7
	A1-3	60	0	20.7	A2-3	625	67.6	7200
	A1-4	0	0	4.8	A2-4	100	0	942.5
	A1-5	0	0	4.4	A2-5	200	0	1020.2
	A1-6	100	0	3.1	A2-6	300	0	1484.8
	A1-7	100	0	39.8	A2-7	100	0	1285.7
	A1-8	0	0	2.9	A2-8	300	47.7	7200
	A1-9	100	0	6.1	A2-9	100	0	306
	A1-10	0	0	1.1	A2-10	1233.3	79.3	7200
	A1-11	100	0	4.6	A2-11	366.7	12.0	7200
	A1-12	100	0	43.1	A2-12	100	0	117.8
	A1-13	100	0	2.5	A2-13	250	20.0	7200
	A1-14	100	0	27.7	A2-14	0	0	93.5
	A1-15	175	0	47.3	A2-15	433.3	53.8	7200
Avg.	-	72.3	0	16.6	-	287.2	18.7	3232.5
$ \mathcal{P}  = 8$	A3-1	33.3	0	11.4	A4-1	1895	89.4	7200
	A3-2	200	0	14.6	A4-2	966.7	80.1	7200
	A3-3	0	0	7.8	A4-3	500	26.6	7200
	A3-4	200	0	34.5	A4-4	300	27.6	7200
	A3-5	0	0	4.6	A4-5	475	57.6	7200
	A3-6	100	0	18.7	A4-6	1578.3	83.1	7200
	A3-7	260	0	308.4	A4-7	550	62.9	7200
	A3-8	183.3	0	4194.9	A4-8	590	83.1	7200
	A3-9	200	0	31.9	A4-9	200	0	1721.1
	A3-10	100	0	32.2	A4-10	375	30.2	7200
	A3-11	150	0	442.4	A4-11	685	73.5	7200
	A3-12	108.3	0	908.2	A4-12	800	79.9	7200
	A3-13	100	0	13.5	A4-13	925	65.9	7200
	A3-14	141.6	0	1188.9	A4-14	1608.3	86.1	7200
	A3-15	0	0	6.1	A4-15	1083.3	85.5	7200
Avg.	-	118.4	0	481.2	-	835.4	62.1	6834.7

of size with 12 periods and 4 processes. For each size, 15 different instances are tested, and therefore in total  $15 \times 4 = 60$  instances are executed. Other parameters are randomly generated as described in Section 5.1. The information about  $|\mathcal{T}|$  and  $|\mathcal{P}|$  is given in the first row and the first column, respectively. In this table, we can see the optimal objective values and computational times under each instance size. For example under A1, objective values change from 0 to 175 with an average value of 72.3 and the corresponding computational time varies from 1.1s to 47.3s with an average of 16.6s. The average optimal values for sizes A1 and A2 grow from 72.3 to 287.2, while their average computational time increases rapidly from 16.6s to 3232.5s. Therefore, for a given number of processes  $|\mathcal{P}|$ , the objective values and computational time increase in general with an increasing number of periods  $|\mathcal{T}|$ . On the other hand, comparing the average Obj and CPUT values of sizes A1 and A3 (A2 and A4), we observe, as expected, that the optimal values and computational time increase with increasing  $|\mathcal{P}|$ . Note that in some instances, the objective values are the current best when the time limit is reached. For example in A2-3, the current objective value is 625 and the gap is 67.6%. This gap generally grows larger as the instance size increases.

A less intuitive observation in Table 4 is that computational time may vary significantly for a given combination of  $|\mathcal{T}|$  and  $|\mathcal{P}|$ . We suspect that other parameters, such as the number of systems in each process  $|S_p|$ , may contribute to these variations. A further observation is that the results for the four sizes A1-A4 do not suggest a linear relationship

**Table 5**  
Impact of the initial RUL.

Test B	B1: $o'_{pskj} \in [10, 20]$			B2: $o'_{pskj} \in [40, 50]$			B3: $o'_{pskj} \in [70, 80]$		
	Instance no.	NoM	Obj	CPUT(s)	NoM	Obj	CPUT(s)	NoM	Obj
B-1	15	100	31.7	5	100	23.0	4	100	14.3
B-2	15	100	29.4	7	100	25.2	3	100	16.5
B-3	20	100	10.2	4	100	17.1	3	100	16.8
B-4	14	100	8.7	4	100	16.4	3	100	13.4
B-5	18	100	10.8	6	100	18.0	4	100	16.9
B-6	20	100	9.9	4	100	10.3	4	100	14.1
B-7	20	100	8.6	4	100	17.1	0	100	12.8
B-8	17	100	19.4	4	100	20.2	3	100	17.1
B-9	20	100	9.4	6	100	19.4	3	100	11.5
B-10	15	100	20.8	6	100	13.6	2	100	11.7
B-11	19	100	11.6	5	100	15.5	3	100	11.7
B-12	17	100	20.6	4	100	20.2	3	100	12.4
B-13	19	100	9.0	5	100	12.0	3	100	18.8
B-14	16	100	33.9	6	100	19.1	4	100	16.7
B-15	17	100	25.3	4	100	8.9	4	100	12.6
B-16	19	100	18.9	4	100	13.5	3	100	12.9
B-17	16	100	20.6	4	100	7.6	4	100	11.8
B-18	14	100	26.6	4	100	19.7	3	100	11.8
B-19	13	100	28.4	4	100	23.9	4	100	14.1
B-20	19	100	17.7	7	100	20.8	3	100	20.2
Avg.	17.2	100	18.6	4.9	100	17.1	3.2	100	14.4

between the number of variables (or constraints) and the corresponding computational time. This means that the instance size is not the only factor impacting the computational time. These observations motivate us to carry out sensitivity analysis (TestB, TestC, and TestD) to try to uncover which factors, apart from instance size, affect the model's computational time.

For TestB in Table 5, we can see that for lower initial RUL (columns under B1), the optimal objective values are always 100. The computational time changes from 8.6s to 33.9s with an average value of 18.6s. Compared to B1, the corresponding objective values of B2 (with the medium initial RUL) have the same average of 100 and the mean computational time has a small decrease from 18.6s to 17.1s. Similar results appear for the instances in B3, and the average computational time reduces to 14.4s. There are significant differences in terms of the indicator NoM of B1, B2, and B3. It is observed that B1 has an average of 17.2, ranging from 13 to 20, while the average values of NoM under B2 and B3 are only 4.9 and 3.2, respectively. We conclude that even if the initial RUL may not significantly affect the optimal objective values, the computational time will decrease with larger initial RUL values. In particular, this parameter has an obvious influence on the number of maintenance actions because the lower the initial RUL, the greater the number of maintenance actions required. The fact that the objective value remains unchanged for different numbers of maintenance is due to component redundancy: more maintenance activities do not necessarily mean more factory production loss.

For TestC in Table 6, we can see that with the increase of the RUL threshold (from C1 to C3), the average objective values increase from 60 to 106.3 and finally reach 203.8 and the average number of maintenance actions increases from 2.6 to 7.1 to 13.4. These two indicators demonstrate that setting a greater RUL threshold may result in more maintenance activities, and this can further affect the production of the whole factory (i.e. lead to greater production loss). However, a greater RUL threshold also means earlier PM, which may enhance component reliability. Exploring the impacts of this trade-off is a question for future research. Lastly, we observe that the computational time obviously increases when the threshold increases.

For TestD in Table 7, we observe that the average numbers of maintenance of D1, D2, and D3 are 3.9, 5.9, and 9.4, respectively. At the same time, the corresponding average objective values increase from 92.5 to 102.5 to 177.5. These two indicators show that with an increasing RUL deterioration rate, a component's RUL reaches the threshold

**Table 6**  
Impact of the RUL threshold.

Test C	C1: $\tau_{pk} \in [1, 3]$			C2: $\tau_{pk} \in [4, 6]$			C3: $\tau_{pk} \in [7, 10]$		
	Instance no.	NoM	Obj	CPUT(s)	NoM	Obj	CPUT(s)	NoM	Obj
C-1	2	0	3.5	4	150	64.2	12	200	24.6
C-2	5	100	19.1	7	100	8.6	12	200	115.5
C-3	0	0	3.6	9	100	7.5	16	200	38.2
C-4	4	100	21.1	5	100	10.9	12	200	17.3
C-5	4	100	17.6	6	100	10.2	20	200	17.8
C-6	5	100	22.5	8	100	8.2	9	200	32.7
C-7	5	100	11.1	8	100	8.6	8	225	263.7
C-8	4	100	13.3	7	100	9.9	11	200	77.4
C-9	0	0	4.1	10	100	7.2	7	225	243.1
C-10	4	100	17.8	10	150	78.2	16	200	28.2
C-11	4	100	20.2	7	100	6.4	11	200	36.1
C-12	4	100	24.5	7	100	6.9	19	200	16.6
C-13	4	100	13.2	7	100	10.8	13	200	16.7
C-14	0	0	3.8	9	100	7.8	12	200	25.5
C-15	0	0	3.0	6	125	22.2	12	200	20.5
C-16	0	0	3.0	6	100	9.8	16	200	14.1
C-17	4	100	11.3	7	100	8.2	17	200	27.1
C-18	0	0	3.4	6	100	7.2	11	200	14.9
C-19	3	100	35.6	5	100	10.2	18	200	27.1
C-20	0	0	4.1	7	100	7.1	16	225	170.9
Avg.	2.6	60	12.8	7.1	106.3	15.5	13.4	203.8	61.4

more quickly, and this may bring about more maintenance actions and lead to greater production loss. Concerning their computational time, the average values increase from 16.1s to 51.2s, meaning that a larger rate may cause increased computational time.

Based on the results of our comparative experiments and sensitivity analysis, we can conclude that: (i) larger problem sizes generally result in greater production loss (total penalty) and longer computational time. (ii) Lower initial RUL values lead to a greater number of required maintenance actions. (iii) A lower RUL threshold may result in more maintenance activities, which can further affect the production of the whole factory. (iv) A higher RUL deterioration rate may result in more maintenance actions and greater production loss. From the analysis in Section 5.2.1, we also observe that (v) having more backup components provides greater reliability and thus yields a lower total penalty.

**Table 7**  
Impact of the RUL decrease rate.

Test D	D1: $\beta_{pk}^t \in [0.20, 0.25]$			D2: $\beta_{pk}^t \in (0.25, 0.30]$			D3: $\beta_{pk}^t \in (0.30, 0.35]$		
	Instance no.	NoM	Obj	CPUT(s)	NoM	Obj	CPUT(s)	NoM	Obj
D-1	4	100	19.4	4	100	8.6	10	125	8.2
D-2	3	75	8.1	8	100	9.1	8	200	69.8
D-3	5	100	13.3	6	100	7.2	12	200	66.6
D-4	5	100	17.9	7	100	10.6	6	200	59.4
D-5	5	100	9.7	4	100	7.4	12	200	107.9
D-6	4	100	11.2	8	100	8.3	9	200	110.1
D-7	4	100	10.3	10	125	43.1	8	200	82.1
D-8	4	75	10.3	5	100	5.6	8	200	57.6
D-9	4	100	24.5	4	100	7.9	7	125	46.4
D-10	3	75	18.1	6	125	51.3	7	200	12.7
D-11	3	75	15.2	5	100	6.8	11	200	44.6
D-12	4	100	14.9	6	100	8.1	11	200	72.1
D-13	4	100	23.2	6	100	7.3	11	125	11.9
D-14	4	100	9.1	7	100	19.4	11	125	46.1
D-15	4	100	18.7	5	100	11.4	13	200	52.2
D-16	3	75	22.5	5	100	10.1	11	125	12.7
D-17	4	100	19.9	5	100	17.3	9	200	64.6
D-18	4	100	17.7	4	100	7.8	10	125	14.2
D-19	3	75	15.1	6	100	7.5	6	200	50.7
D-20	3	100	22.1	7	100	9.1	8	200	33.1
Avg.	3.9	92.5	16.1	5.9	102.5	13.2	9.4	177.5	51.2

### 5.3. Experiments for the SRPMO Problem

We proceed to evaluate the approximated model M2 of chance-constrained programming for the SRPMO problem. In Section 5.3.1, we present an illustrative example to show the effect of stochastic decrease rates on maintenance optimization. In Section 5.3.2, we explore the impact of the choices of probabilities in chance constraints.

#### 5.3.1. An Illustrative Example of the Impact of Uncertainty

We present a small example to illustrate the behaviour of M2, using M1 as the benchmark. We consider an instance with 4 processes (the structure corresponds to Figure 2) and 15 periods, and we set  $\Gamma_{\alpha_1} = \Gamma_{\beta_1} = 0.1$  and  $\Gamma_{\alpha_2} = \Gamma_{\beta_2} = 0.9$  for M2. The computational results obtained for M1 and M2 are presented in Figure 6 and Table 8.

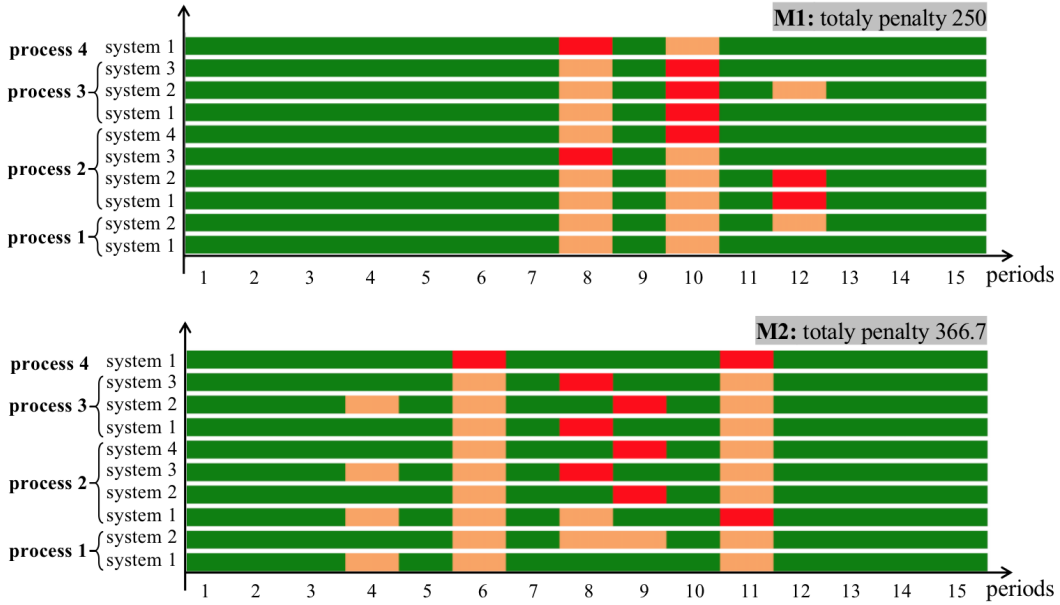


Figure 6: Status of systems over time obtained by M1 and M2.

Table 8

Track of RUL values of the same component by M1 and M2.

$t$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
M1	38	27.74	20.31	14.90	10.97	8.09	5.99	4.44	4.42	3.30	3.28	2.46	20.00	15.08	11.4
M2	38	25.30	16.90	11.32	11.19	7.54	7.46	5.06	4.99	3.40	2.33	20.0	13.76	9.49	6.57

In Figure 6, the top and bottom subfigures represent the system states over time for M1 and M2, respectively. The x-axis is indexed by the time period in the planning horizon and the y-axis by the system. For example, ‘process 1 system 2’ refers to the second system in process 1. A green state means that a system is operating, red means that a system is not operating because of its own maintenance, and orange means that a system is not working due to the statuses of other processes.

We observe from Figure 6 that the total penalty values obtained by M1 and M2 are respectively 250 and 366.7. For M1, we see that in periods 1-7, 9, 11, and 13-15, all the systems are operating so that the production volume of the factory is at its maximum. This is clear from Figure 6 because all the icons are green in these periods. In period 8, system 3 in process 2 and system 1 in process 4 are not operating because of maintenance (red). Therefore, the production capacity of process 4 is 0 in this period, which forces the systems in other processes to become non-operating (orange) in order to respect the production flow balance. Similar situations occur in periods 10 and 12. For M2, only ten periods achieve the maximum production volume: periods 1-3, 5, 7, 10, and 12-15. In the other periods, such as period 8, system 3 in process 2 and systems 1 and 3 in process 3 are not operating due to maintenance, meaning that the production of

the factory is only 33.3 in this period. Regarding the flow balance, we can stop one system in process 1 and one in process 2 to achieve this quantity. Note that in period 4, the shutdown of systems is not due to its own maintenance or the impact of other systems, but the global decision made for the entire planning horizon.

By comparing the two subfigures in Figure 6, it is straightforward to observe that there are more system perturbations for the optimal solution of M2 than for that of M1. This shows that the use of stochastic RUL decrease rates can influence maintenance decisions and production volumes. Specifically for this instance, uncertainty brings more maintenance activities and thus leads to greater production loss.

Table 8 depicts the RUL evolution over time for a component of type 1 belonging to process 2, system 1. We observe that (i) the RUL values obtained by M1 and M2 show significant differences because the RUL decreases faster for M2 than for M1. For example, in periods 1-4, the optimal RUL values for M1 are  $38.0 \rightarrow 27.74 \rightarrow 20.31 \rightarrow 14.9$ , while they are  $38.0 \rightarrow 25.30 \rightarrow 16.90 \rightarrow 11.32$  for M2. (ii) The maintenance decisions taken by M1 and M2 are different. Specifically, this component is on standby under M1 from period 8 to period 9 and 10 to 11, with a slight decrease of RUL, and the maintenance is conducted in period 12. By contrast in M2, the component is on standby from period 4 to period 5, 6 to 7, and 8 to 9, and is maintained in period 11.

Note that we only present the RUL values for one component as an example of the impact of stochastic decrease rates on the RUL evolution. Naturally, other components are also subject to uncertainty.

### 5.3.2. Impact of Confidence Level and Its Specificity

In this section, we test the model M2 with 36 generated instances, using M1 as the benchmark. For each instance, M2 is solved under different combinations of probabilities to illustrate the attribute analysis in Section 4.3. For the sake of simplicity, we set  $\Gamma_{\alpha 1} = \Gamma_{\beta 1} = \Gamma_1$  and  $\Gamma_{\alpha 2} = \Gamma_{\beta 2} = \Gamma_2$ .

We start with the case  $\Gamma_1 + \Gamma_2 = 1$  and test 5 different combinations of  $\Gamma_1$  and  $\Gamma_2$ . The optimal objective values obtained by M1 and M2 are reported in Figure 7, where the  $x$ -axis is indexed by the tested instance and the  $y$ -axis by the optimal objective value. The black curve reports the optimal values of M1, while the colored curves correspond to the values of M2 under different probability combinations. For example, the red curve is for M2 under  $\Gamma_1 = 0.1$  and  $\Gamma_2 = 0.9$ . We observe in Figure 7 that (i) M1 and M2 under  $\Gamma_1 = 0.5$  and  $\Gamma_2 = 0.5$  achieve exactly the same objective values under all the tested instances. This confirms that assigning 0.5 for both probabilities provides an average idea and corresponds to the deterministic case. (ii) M1 outperforms M2 whenever  $\Gamma_2 > 0.5$ . The reason is that the upper bound of  $R_{pskj}^{t+1}$  is tightened while the lower bound is relaxed. As a consequence, lower RUL values are obtained by M2 compared to M1, and M2 thus requires more maintenance activities than M1, leading to a larger total penalty. We also observe that with increasing  $\Gamma_2$ , and hence decreasing  $\Gamma_1$ , the optimal values become larger, as expected. (iii) In contrast to (ii), M2 outperforms M1 whenever  $\Gamma_1 > 0.5$ . The reason is that the upper bound of  $R_{pskj}^{t+1}$  is relaxed and the lower bound is tightened, and therefore greater RUL values are obtained by M2 compared to M1, so that M2 has lower maintenance requirements. All these observations are in agreement with the attribute analysis in Section 4.3.

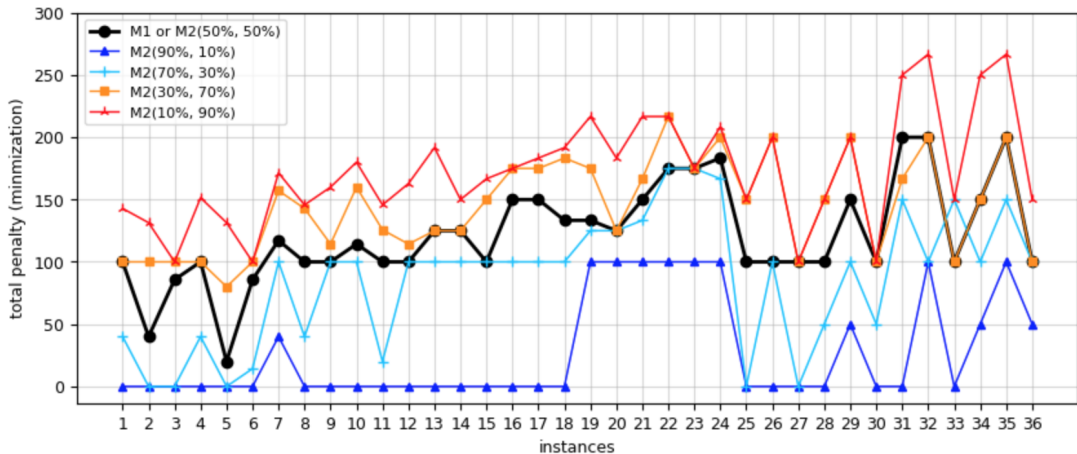


Figure 7: Results of M1 and M2 ( $\Gamma_1 + \Gamma_2 = 1$ ).

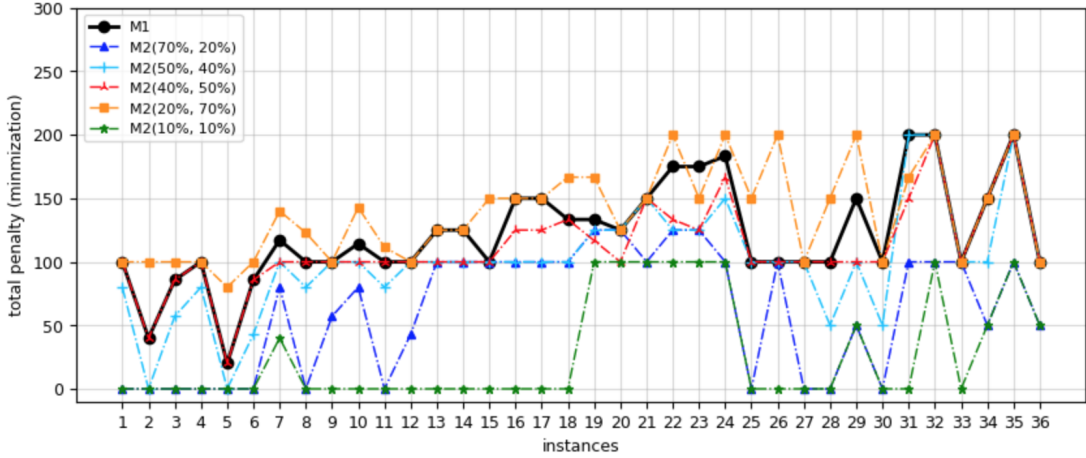


Figure 8: Results of M1 and M2 (under  $\Gamma_1 + \Gamma_2 < 1$ ).

We now proceed to the case  $\Gamma_1 + \Gamma_2 < 1$  with 6 different combinations of  $\Gamma_1$  and  $\Gamma_2$ . The objective values obtained by M1 and M2 are presented in Figure 8. From this figure, we observe that (i) the black curve (corresponding to the deterministic version) no longer delineates the various combinations as it did for the case  $\Gamma_1 + \Gamma_2 = 1$ . The reason is two-fold: first, we are now assigning a confidence level less than 1 for both lower and upper bounds, hence there is a small probability that the model is either more tightened or more relaxed because of the uncertainty in the value of  $R_{pskj}^{t+1}$ . Second, if  $\Gamma_1 + \Gamma_2 < 1$ , the constraints on  $R_{pskj}^{t+1}$  in M2 are inequalities and can be interpreted as a relaxation of M1. This also explains why the relationship between the optimal objective values obtained by M1 and M2 varies from one instance to another. (ii) In general, the optimal values obtained by M2 are smaller with larger  $\Gamma_1$  and smaller  $\Gamma_2$ , which is the same trend as in Figure 7. In particular, when  $\Gamma_1$  is greater than 0.5, M2 outperforms M1 because M2 has more relaxed upper bounds. (iii) Under the combination  $\Gamma_1 = 0.1$  and  $\Gamma_2 = 0.1$ , M2 obtains the smallest optimal values for the tested instances. The reason is that M2 now has the most relaxed upper and lower bounds, providing more choices for the possible values of  $R_{pskj}^{t+1}$  and this leads to fewer maintenance actions. Again, these observations agree with the results in Section 4.3. Finally, we mention that the computational time of M2 shows similar trends to those of M1 as discussed in Section 5.2.

## 6. Conclusions, Managerial Insights, and Future Work

In this paper, we addressed RUL-based preventive maintenance optimization for a generic factory with complex systems, under both deterministic RUL decrease rates and stochastic RUL decrease rates. Our approach integrates distinctive features, such as component redundancy, RUL-based formulation, system availability, and production. For the deterministic case, we proposed a mixed-integer linear programming model to minimize the total penalty due to factory production loss. For the stochastic case, we extended the MILP model into a chance-constrained programming model. We presented an approximation approach and carried out attribute analysis for this stochastic model. We concluded from our experimental results that: (i) Using our proposed MILP model, modern solvers can find optimal solutions for the addressed RUL-based maintenance optimization problems, within a reasonable time. (ii) Compared to the literature, our approaches demonstrate the rationale of applying mathematical optimization and the value of our approach using RUL-based maintenance, component redundancy, and system availability. (iii) Optimal objective values and computational times depend not only on the instance sizes but also on key parameter settings, such as initial RUL values, RUL thresholds, and RUL decrease rates. (iv) Our proposed methods are effective at approximating the stochastic model, and choosing appropriate confidence levels in the chance constraints is important for the performance of this approximation.

Managerial insights drawn from this research include: (i) knowledge about the RUL evolution can be used to optimize preventive maintenance planning. (ii) Accounting for stochastic RUL values can impact the behavior of industrial productivity. (iii) The optimization can trade off maintenance budget and production loss, in making decisions over a planning horizon.

Future work includes but is not limited to (i) introducing other uncertain factors in our maintenance optimization problem. We assumed here that the RUL evolution is affected by uncertainty and depends on the state (working, standby, or maintenance) of a component. However, other uncertain factors such as stochastic maintenance durations can be integrated. (ii) Integrating buffers between subsequent processes in the factory. This would mean that the maintenance of a process does not directly result in stopping other processes, so their stocked production in the buffers is available as soon as the process's maintenance is completed. This concept changes the balance between production and maintenance during the planning horizon. (iii) Considering multi-objective optimization. It would be interesting to study the trade-off between productivity and availability so that an appropriate maintenance budget can be provided for decision-making. (iv) Designing alternative approaches to provide feasible solutions within a short time for very large instances.

## References

- Alaswad, Suzan and Yisha Xiang. 2017. "A review on condition-based maintenance optimization models for stochastically deteriorating system." *Reliability Engineering & System Safety* 157: 54–63.
- Alrabghi, Abdullah and Ashutosh Tiwari. 2015. "State of the art in simulation-based optimisation for maintenance systems." *Computers & Industrial Engineering* 82: 167–182.
- Alrabghi, Abdullah and Ashutosh Tiwari. 2016. "A novel approach for modelling complex maintenance systems using discrete event simulation." *Reliability Engineering & System Safety* 154: 160–170.
- Azizi, Fariba and Nooshin Salari. "A novel condition-based maintenance framework for parallel manufacturing systems based on bivariate birth/birth–death processes." *Reliability Engineering & System Safety* 229: 108798.
- Bülbül, Pınar, Bayındır Z. Pelin, and Bakal I. Serdar. 2019. "Exact and heuristic approaches for joint maintenance and spare parts planning." *Computers & Industrial Engineering* 129: 239–250.
- Cai, Baoping, Yuandong Wang, Yanping Zhang, Yiliu Liu, Weifeng Ge, Rongkang Li, Yonghong Liu, and Guijie Liu. 2022. "Condition-based maintenance method for multi-component system based on RUL prediction: Subsea tree system as a case study." *Computers & Industrial Engineering* 173: 108650.
- Camci, Fatih. 2009. "System maintenance scheduling with prognostics information using genetic algorithm." *IEEE Transactions on Reliability* 58(3): 539–552.
- Camci, Fatih. 2015. "Maintenance scheduling of geographically distributed assets with prognostics information." *European Journal of Operational Research* 245(2): 506–516.
- Camci, Fatih, Kamal Medjaher, Vepa Atamuradov, and Ashyrmuhammet Berdinyazov. 2019. "Integrated maintenance and mission planning using remaining useful life information." *Engineering Optimization* 51(10): 1794–1809.
- Cao, Yingsai, Jianqiang Luo, and Wenjie Dong. 2023. "Optimization of condition-based maintenance for multi-state deterioration systems under random shock." *Applied Mathematical Modelling* 115: 80–99.
- Chen, Zhen, Yaping Li, Tangbin Xia, and Ershun Pan. 2019. "Hidden Markov model with auto-correlated observations for remaining useful life prediction and optimal maintenance policy." *Reliability Engineering & System Safety* 184: 123–136.
- Chen, Chuang, Ningyun Lu, Bin Jiang, and Cunsong Wang. 2021. "A risk-averse remaining useful life estimation for predictive maintenance." *IEEE/CAA Journal of Automatica Sinica* 8(2): 412–422.
- Chen, Jiaxian, Dongpeng Li, Ruyi Huang, Zhuyun Chen, and Weihua Li. 2023a. "Aero-engine remaining useful life prediction method with self-adaptive multimodal data fusion and cluster-ensemble transfer regression." *Reliability Engineering & System Safety* 234: 109151.
- Chen, Jiaxian, Ruyi Huang, Zhuyun Chen, Wentao Mao, and Weihua Li. 2023b. "Transfer learning algorithms for bearing remaining useful life prediction: A comprehensive review from an industrial application perspective." *Mechanical Systems and Signal Processing* 193: 110239.
- Cui, Penghao, Junqiang Wang, and Yang Li. 2021. "Data-driven modelling, analysis and improvement of multistage production systems with predictive maintenance and product quality." *International Journal of Production Research* 1–18.
- Diallo, Claver, Uday Venkatadri, Abdelhakim Khatab, and Sriram Bhakthavatchalam. 2017. "State of the art review of quality, reliability and maintenance issues in closed-loop supply chains with remanufacturing." *International Journal of Production Research* 55(5): 1277–1296.
- Do, Phuc, Alexandre Voisin, Eric Levrat, and Benoit Lung. 2015. "A proactive condition-based maintenance strategy with both perfect and imperfect maintenance actions." *Reliability Engineering & System Safety* 133: 22–32.
- Dong, Wenjie, Sifeng Liu, Yingsai Cao, Saad A. Javed, and Yangyang Du. 2020. "Reliability modeling and optimal random preventive maintenance policy for parallel systems with damage self-healing." *Computers & Industrial Engineering* 142: 106359.
- Ghorbani, Milad, Mustapha Nourelfath, and Michel Gendreau. 2022. "A two-stage stochastic programming model for selective maintenance optimization." *Reliability Engineering & System Safety* 223: 108480.
- Hajej, Zied, Nidhal Rezg, and Ali Gharbi. 2020. "Maintenance on leasing sales strategies for manufacturing/remanufacturing system with increasing failure rate and carbon emission." *International Journal of Production Research* 58(21): 6616–6637.
- Hasib, Shahid A., S. Islam, Ripon K. Chakraborty, Michael J. Ryan, D. K. Saha, Md H. Ahamed. 2021. "A comprehensive review of available battery datasets, RUL prediction approaches, and advanced battery management." *IEEE Access* 9: 86166–86193.
- He, Junkai, Miguel F. Anjos, Makhlouf Hadji, and Selma Khebbache. 2022. "Prognostic-based maintenance optimization in complex systems with resource limitation constraints." *In the Proceedings of International Conference on Operations Research and Enterprise Systems*. Online, Portugal.
- Khatab, Abdelhakim and ElHoussaine Aghezzaf. 2016. "Selective maintenance optimization when quality of imperfect maintenance actions are stochastic." *Reliability engineering & system safety* 150: 182–189.

- Khatab, Abdelhakim, ElHoussaine Aghezaf, Imene Djelloul, and Zaki Sari. 2017. "Selective maintenance optimization for systems operating missions and scheduled breaks with stochastic durations." *Journal of manufacturing systems* 43: 168–177
- Khorshidi, H. Akbarzade, Gunawan Indra, and Ibrahim M. Yousef. 2016. "A value-driven approach for optimizing reliability-redundancy allocation problem in multi-state weighted k-out-of-n system." *Journal of Manufacturing Systems* 40: 54–62.
- Knofius, Nils, Matthieu C. van der Heijden, and W. Henk M. Zijm. 2019. "Consolidating spare parts for asset maintenance with additive manufacturing." *International Journal of Production Economics* 208: 269–280.
- Koopmans, Marco and Bram de Jonge. 2023. "Condition-based maintenance and production speed optimization under limited maintenance capacity." *Computers & Industrial Engineering* doi.org/10.1016/j.cie.2023.109155.
- Lei, Xin and Peter A. Sandborn. 2018. "Maintenance scheduling based on remaining useful life predictions for wind farms managed using power purchase agreements." *Renewable Energy* 116: 188–198.
- Lei, Yaguo, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. 2018. "Machinery health prognostics: A systematic review from data acquisition to RUL prediction." *Mechanical systems and signal processing* 104: 799–834.
- Liang, Zeming, Jianmin Gao, and Hongquan Jiang. 2019. "A maintenance support framework based on dynamic reliability and remaining useful life." *Measurement* 147: 106835.
- Liu, Lujie, Jun Yang, Xuefeng Kong, and Yiyong Xiao. 2022. "Multi-mission selective maintenance and repair persons assignment problem with stochastic durations." *Reliability Engineering & System Safety* 219: 108209.
- Niu, Huifang, Jianchao Zeng, Hui Shi, Xiaohong Zhang, and Jianyu Liang. 2023. "Degradation modeling and remaining useful life prediction for a multi-component system with stochastic dependence." *Computers & Industrial Engineering* 175: 108889.
- Obaidat, Sinan and Haitao Liao. 2021. "Integrated decision making for attributes sampling and proactive maintenance in a discrete manufacturing system." *International Journal of Production Research* 59(18): 5454–5476.
- Rashidnejad, Mehdi, Sadoullah Ebrahimejad, and Jalal Safari. 2018. "A bi-objective model of preventive maintenance planning in distributed systems considering vehicle routing problem." *Computers & Industrial Engineering* 120: 360–381.
- Shafiee, Mahmood and John D. Sørensen. 2019. "Maintenance optimization and inspection planning of wind energy assets: Models, methods and strategies." *Reliability Engineering & System Safety* 192: 105993.
- Shahraki, A. Forouzandeh, Om P. Yadav, and Chrysafis Vogiatzis. 2020. "Selective maintenance optimization for multi-state systems considering stochastically dependent components and stochastic imperfect maintenance actions." *Reliability Engineering & System Safety* 196: 106738.
- Si, Xiaosheng, Wenbin Wang, Changhua Hu, and Donghua Zhou. 2011. "Remaining useful life estimation - a review on the statistical data driven approaches." *European Journal of Operations Research* 213: 1–14.
- Wang, Yukun, Xiaopeng Li, Junyan Chen, and Yiliu Liu. 2022. "A condition-based maintenance policy for multi-component systems subject to stochastic and economic dependencies." *Reliability Engineering & System Safety* 219: 108174.
- Wang, Liying, Yushuang Song, Wenhua Zhang, and Xiaoliang Ling. 2023. "Condition-based inspection, component reallocation and replacement optimization of two-component interchangeable series system." *Reliability Engineering & System Safety* 230: 108907.
- Wu, Minghui, Xuemin Wang, and Xianzhong Liu. 2020. "On condition maintenance model for complex electromechanical equipments based on remaining useful life and wiener process." *Journal of Physics: Conference Series. IOP Publishing* 1678(1): 012014.
- Wu, Shaomin and Inma T. Castro. 2020. "Maintenance policy for a system with a weighted linear combination of degradation processes." *European Journal of Operational Research* 280(1): 124–133.
- Xenos, Dionysios P., Georgios M. Kopanos, Matteo Ciccotti, and Nina F. Thornhill. 2016. "Operational optimization of networks of compressors considering condition-based maintenance." *Computers & Chemical Engineering* 84: 117–131.
- Xiao, Lei, Sanling Song, Xiaohui Chen, and David W. Coit. 2016. "Joint optimization of production scheduling and machine group preventive maintenance." *Reliability Engineering & System Safety* 146, 68–78.
- Xiong, Jiawei, Jian Zhou, Yizhong Ma, Fengxia Zhang, and Chenglong Lin. 2023. "Adaptive deep learning-based remaining useful life prediction framework for systems with multiple failure patterns." *Reliability Engineering and System Safety* doi.org/10.1016/j.res.2023.109244.
- Xu, Zifei, Musa Bashir, Qinsong Liu, Zifan Miao, Xinyu Wang, Jin Wang, and Nduka Ekere. 2023. "A novel health indicator for intelligent prediction of rolling bearing remaining useful life based on unsupervised learning model." *Computers & Industrial Engineering* doi.org/10.1016/j.cie.2023.108999.
- Ye, Yixin, Ignacio E. Grossmann, Jose M. Pinto, and Sivaraman Ramaswamy. 2019. "Modeling for reliability optimization of system design and maintenance based on Markov chain theory." *Computers & Chemical Engineering* 124: 381–404.
- Zhang, Beikun, Liyun Xu, Yiping Chen, and Aiping Li. 2018. "Remaining useful life based maintenance policy for deteriorating systems subject to continuous degradation and shock." *Procedia CIRP* 72: 1311–1315.
- Zhang, Jiushi, Yuchen Jiang, Xiang Li, Mingyi Huo, Hao Luo, and Shen Yin. 2022a. "An adaptive remaining useful life prediction approach for single battery with unlabeled small sample data and parameter uncertainty." *Reliability Engineering & System Safety* 222: 108357.
- Zhang, Nan, Kaiquan Cai, Jun Zhang, and Tian Wang. 2022b. "A condition-based maintenance policy considering failure dependence and imperfect inspection for a two-component system." *Reliability Engineering & System Safety* 217: 108069.
- Zhang, Nan, Sen Tian, Jiatao Xu, Yingjun Deng, and Kaiquan Cai. 2023a. "Optimal production lot-sizing and condition-based maintenance policy considering imperfect manufacturing process and inspection errors." *Computers & Industrial Engineering* doi.org/10.1016/j.cie.2022.108929.
- Zhang, Nan, Yingjun Deng, Bin Liu, and Jun Zhang. 2023b. "Condition-based maintenance for a multi-component system in a dynamic operating environment." *Reliability Engineering & System Safety* 231: 108988.
- Zhu, Zhicheng, Yisha Xiang, and Bo Zeng. 2021. "Multi component maintenance optimization: a stochastic programming approach." *INFORMS Journal on Computing* 33(3): 898–914.