國立臺灣大學資訊管理學研究所

碩士論文

Department of Information Management
College of Management
National Taiwan University

Master’s Thesis

多產品流線型環境中考慮隨機良率的製造與預防性保養整合規劃問題

Integrated multi-product production and preventive maintenance planning in a flow shop system with random yield rate changes

郭仲嘉
Chung-Chia Kuo

指導教授：孔令傑 博士
Advisor: Ling-Chieh Kung, Ph.D.

中華民國 112 年 6 月

June, 2023

doi:10.6342/NTU202403120

致謝

這篇論文的完成有賴於許多人的幫助。首先要謝謝孔令傑老師的指導。與老師思辨討論、激盪想法的過程，使這個研究更為完整。在實驗室的兩年當助教、做專案，都是讓我的軟、硬實力更上一層的經驗。能夠遇到老師並在老師門下學習，我打從心底覺得幸運與感謝。

謝謝奕璇、稚凱、周荷、正一無時無刻溫暖的陪伴我。謝謝王于宸和我完成近乎瘋狂的日本行。謝謝陳洞杋跟我一起在碩士班努力奮鬥。謝謝學姊、學弟妹與同屆的同學們豐富我的實驗室生活，以及包容我的種種不足。謝謝 SCLL 寫出了 cast a spell on her，陪我渡過無數個清醒的夜晚。最後謝謝我的爸媽、姊姊、阿公阿嬤，總是堅定的作為我的依靠。如果這篇論文有一點價值與貢獻的話，那一定都是這些人的功勞。

doi:10.6342/NTU202403120

摘要

生產規劃問題和預防性保養計劃問題在學術研究中被視為獨立的主題，而常被分開研究。然而在現實情況下，保養活動和生產活動的影響經常互相關聯。保養機臺會使生產效能降低，並可能導致供不應求。忽視保養還可能導致生產設備持續惡化，增加生產過程中的不確定性並降低生產效率。因此，涉及生產規劃和預防性保養計劃的整合決策過程是一個需要關注的議題。

在這項研究中，我們考慮一個決策者同時決定生產計劃和預防性保養計劃的問題。在每個週期決策者決定生產數量以及是否進行維護，以最小化總成本。我們將此問題以動態規劃模型描述，並證明此問題的確定性版本為NP-困難。

我們提出了兩個演算法，根據每個週期的狀態，在週期的開頭決定計劃。我們透過兩個啟發式規則，即設置良率閾值和比較保養決策的成本，來決定保養計畫，之後通過求解線性規劃模型來獲得生產計劃。我們通過數值實驗評估算法的性能和在不同情境下的有效性。

關鍵字：生產規劃、預防性保養排程、生產隨機性、流線型環境、啟發式演算法

doi:10.6342/NTU202403120

Abstract

Production planning and preventive maintenance planning are typically treated as separate subjects in academic studies. However, in real-world scenarios, maintenance activities can significantly impact production capacity, potentially resulting in a shortage of supply to meet demand. Neglecting maintenance can also lead to a continuous deterioration of production equipment, causing uncertainty and inefficiency in the production process. Consequently, the integrated decision-making process involving both production planning and preventive maintenance planning becomes a critical and practical challenge that requires attention.

In this study, we consider a problem where the decision maker determines both production plan and preventive maintenance plan. The production quantity and whether to conduct maintenance are decided every period, aiming to minimize the total cost. We formulate the problem with dynamic programming model, then show the deterministic version of our problem to be NP-hard.

We propose two algorithms, where planning is conducted at the beginning of each period given observed states. The maintenance plan is decided by two heuristic ideas, setting a yield rate threshold and comparing costs of decisions. After that, the production plan may be obtained through solving linear programming model. We evaluate the algorithms performance and their effectiveness under different scenarios through numerical experiments.

Keywords: production planning, preventive maintenance scheduling, production uncertainty, flow shop, heuristic algorithm

doi:10.6342/NTU202403120

Contents

doi:10.6342/NTU202403120

doi:10.6342/NTU202403120

List of Figures

doi:10.6342/NTU202403120

doi:10.6342/NTU202403120

List of Tables

doi:10.6342/NTU202403120

doi:10.6342/NTU202403120

Chapter 1

Introduction

1.1 Background and motivation

Production planning and preventive maintenance planning are well-studied subjects during the past decades. The extensive research in academia and wide application in manufacturing industry reflect their significance. Production planning aims to minimize total cost by deciding production quantity at every period. In contrast, preventive maintenance planning tries to maximize the system reliability or product quality, by choosing the appropriate timing to conduct maintenance.

The two subjects are usually studied separately. In production planning problems, the machines are implicitly assumed to be reliable, and the system consistently achieves the planned production quantity. The assumptions, however, are often impractical in many real-world scenarios. In practice, production environments are much more complex, requiring the manufacturers to make decisions from various perspectives.

doi:10.6342/NTU202403120

1.2 Research objectives

We intend to study an integrated production planning and preventive maintenance planning problem under a flow shop system. Similar to production planning problems, most studies regarding flow shop environment assume machine conditions remain unchanged as time passes. In this study we consider an environment where machine conditions deteriorate over time and are restored through maintenances.

In our case, the system consists of multiple stages. At each period, the planner decides the input quantity and whether to conduct a maintenance at each stage. A random amount of output, influenced by the condition of the stage, is produced at the beginning of next period, then it serves as the input of the next stage. The output of each stage is regarded as WIP (work in progress) except for that of the final stage, which is regarded end product. There are demands for end products at each period. If the demands fail to be realized, a shortage cost will be incurred. Although in some studies deterioration of machines is related to the probability of breakdown, this study focuses solely on the declining yield rate of stages over time without considering machine breakdown. Performing maintenance improves the yield rate of stages, but comes with costs and reduces capacity, thereby limiting the input quantity at each stage.

The objective of this study is to decide the optimal input quantity and preventive maintenance plan. The decision aims to balance the costs, capacity, and stage condition, to meet the demand by efficient production. Since the optimal production and maintenance plan depend on the state at the beginning of every period, we formulate the problem as a dynamic programming model and design algorithms to find the solution.

doi:10.6342/NTU202403120

1.3 Research plan

The remainder of this study is organized as follows. Chapter 2 briefly reviews the literature. Chapter 3 describes the abstracted problem and presents the dynamic programming model. Chapter 4 analyzes the problem complexity and describes the proposed algorithm. Chapter 5 conducts numerical experiments to show the effectiveness of the proposed algorithm, and Chapter 6 concludes the study.

doi:10.6342/NTU202403120

Chapter 2

Literature Review

2.1 Studies of production planning and flow shop production environment

Most studies regarding production planning problems are conducted from several aspects, including material requirement planning, lot sizing, hierarchical planning, and integrated planning (Gelders and Van Wassenhove, 1981). According to the number of stages and relation between stages, the production environment can be roughly classified into single stage, flow shop, and job shop. In this study, the production environment is a flow shop system.

Florian et al. (1980) considered deterministic production planning problems over a finite planning horizon. They investigated complexity for problems with various types of cost functions, set-up costs, and capacity limits, then proposed several algorithms to solve a partition of problems. Escudero et al. (1993) considered uncertainty in demand,

doi:10.6342/NTU202403120

and characterized the uncertainty using individual scenarios. Solutions for each scenario are obtained, and are combined to form an implementable policy. They proposed an LP to determine the product inventory and product volume, and an MIP model to support the decision for material supply.

Johnson (1954) is a pioneer in the study of flow shop problems. He investigated the flow shop system consisting of two machines and gave an algorithm that yields the optimal scheduling plan with minimized makespan. Since then, extensive research regarding production problem under flow shop environment is conducted.

Allaoui and Artiba (2006) investigated a two-stage hybrid flow shop problem considering machine availability, where hybrid flow shop problem implies stages may include one or more machines, and that at least one stage includes multiple machines. They indicated that in industries, machines may be unavailable due to breakdowns or preventive maintenance and showed that the problem is NP-hard, since it is a generalization of another well-known NP-hard problem. A Branch and Bound model is proposed, and performance between LIST algorithm, LPT algorithm, and H-heuristic, is compared.

Ebrahimi et al. (2014) studied the scheduling problem under a hybrid flow shop system, considering uncertainty of setup time and due dates. The problem is shown to be NP-hard, and two GA-based algorithms are proposed. Chu et al. (2022) studied an integration of production planning and scheduling under a hybrid flow-shop system. They established a two-level decision making model for a forging system, as well as a particle swarm-genetic hybrid algorithm to obtain the feasible integrated solution.

All of the studies above assume the yield rates of machines are stable, i.e., no loss occurs during production. The assumption is impractical in reality, therefore in our study

doi:10.6342/NTU202403120

we consider the loss during production. Decision of conducting a preventive maintenance must be made to sustain a certain level of yield rate.

2.2 Studies of integrated production planning and preventive maintenance planning

Recently, extensive research about multi-decision optimization problems are conducted. Integrated production planning and preventive maintenance planning problems received much attention due to its practicality in industries.

Aghezzaf et al. (2007) studied a problem assuming that the production system is subject to random failure, and that maintenance caused by a failure reduces the production capacity. They formulated and solved the problem as a multi-item capacitated lot-sizing problem. Aghezzaf and Najid (2008) then extended the model by assuming the system consists of parallel machines and deteriorates over time. It is assumed that when a production line fails, a minimal repair is carried out to restore it to an "as-bad-as-old" status. A preventive maintenance restores the production line to a "as-good-as-new" status, but reduces the production capacity. They modeled the problem as a linear mixed-integer program when noncyclical preventive maintenance policies are allowed, then proposed a Lagrangian-based heuristic procedure for the solution.

Yalaoui et al. (2014) studied a problem applying similar settings with that of Aghezzaf and Najid (2008), including multiple items, parallel machines, and the system deteriorates over time. The difference between theirs and Aghezzaf and Najid (2008) is that the

doi:10.6342/NTU202403120

deterioration is reflected by the decrease of production capacity in their model. They showed that their approach deals with wider range of problems than that of Aghezzaf and Najid (2008). They developed a relaxation technique to reduce the computation time, as well as a heuristic algorithm for complex problems.

Alimian et al. (2019) studied the integrated problem where the demand fluctuates, assuming the components of the system face both independent and common cause failures. They used robust optimization to model the problem and yield the maintenance plan including decision of conducting perfect or imperfect maintenance. Khatab et al. (2019) investigated a problem integrating production quantity planning and conditional-based maintenance, in a system that deteriorates stochastically. The system is considered to be in "fail mode" whenever its degradation level exceeds a predetermined threshold. The system in "fail mode" produces non-conforming items, which will be replaced via overtime production or spot market purchases. An optimization model is developed to minimize total cost, by determining the optimal inspection cycle and the degradation threshold level.

The studies above applied the single-stage production environment. We intend to generalize the problem by applying the setting of flow shop production environment. To the best of our knowledge, the works done by Aghezzaf and Najid (2008) and Yalaoui et al. (2014) is the closest to our work by applying the aforementioned settings. The major difference of our work and theirs is that we conduct planning under a flow shop system, and that the system deterioration reflects on the yield rate in our formulation. Moreover, to make the model fit reality better, we assume that the actual output given the observed yield rate is random, as well as the deterioration rate and the result after

doi:10.6342/NTU202403120

maintenance. These settings come directly from a real world example, and are more suitable for some cases in manufacturing industry.

doi:10.6342/NTU202403120

Chapter 3

Problem Description and Formulation

3.1 Problem description

Consider a multi-stage flow shop system under a finite number of periods indexed by $t = 1, ..., T$. Multiple products are produced, indexed by $i = 1, ..., I$. The stages processing the products are indexed by $j = 0, ..., J$, where J indicates the stage producing end product. We abuse the notation $j = 0$ as it does not represent an actual production stage, but the process of purchasing raw material.

At each period, the input quantity of product i to stage j, denoted by x_{ijt}, is decided. x_{ijt} has an index j ranging from 0 to J. $x_{i,0,t}$ represents the purchasing quantity of raw material. $x_{i,J,t}$ denotes the input quantity to stage J, namely, the production quantity of end products. Finally, x_{ijt} where $j = 1, ..., J-1$, denotes the input quantity to stage j.

doi:10.6342/NTU202403120

$z_{jt} \in \{0, 1\}$, denoting whether to conduct maintenance at stage j, is also decided at each period t. z_{jt} equals 1 means a maintenance is conducted, or 0 otherwise. Both x_{ijt} and z_{ijt} have index t ranging from 1 to $T - 1$, since no decision has to be made at period T.

In reality, there might be multiple machines operating in a single stage. In this study, we view all machines in the same stage as a whole, or that a stage simply consists of only one machine.

We introduce y_{ijt} and w_{jt} as the state variables in our dynamic programming formulation. Let y_{ijt} denote the beginning inventory level of product i at period t. Index j of y_{ijt} ranges from 0 to J, implying meaning similar with that of x_{ijt}. $y_{i,0,t}$ and $y_{i,J,t}$ are the inventory level of raw material and end product, respectively. $y_{i,1,t}, ..., y_{i,J-1,t}$ are the inventory level of WIP(work in progress) at corresponding stages. Let w_{jt} denote the beginning yield rate of stage j at period t. The beginning inventory level and yield rate at period 1 are parameters, denoted by $Y_{i,j,1}$ and $W_{j,1}$. The constraints in the formulation restrict the value of $y_{i,j,1}$ and $w_{j,1}$ in the later section.

For $j = 1, ..., J$, x_{ijt} units of product-i input are processed by corresponding stages at each period t. A random quantity of output, influenced by w_{jt}, is produced at the beginning of period $t+1$, regarded as $y_{i,j,t+1}$. The random quantity of output is denoted as $H(w = w_{jt}, x = x_{ijt})$. It follows a binomial distribution, where the number of experiments equals x, and the success probability equals w. On the other hand, $x_{i,0,t}$ units of raw material are purchased at each period t. They are realized at the beginning of period $t+1$ regarded as $y_{i,0,t+1}$. Without being processed by a stage, the quantity of raw material purchased equals the quantity of raw material inventory increased at the beginning of next period.

doi:10.6342/NTU202403120

The input quantity to a stage is limited by beginning inventory level of the previous stage. i.e., $x_{ijt} \leq y_{i,j-1,t}$ for all $i = 1, ..., I, j = 1, ..., J, t = 1, ..., T-1$. A similar constraint applies to $x_{i,0,t}$, that the purchase of raw material must not exceed the external supply limit M_{it}, i.e., $x_{i,0,t} \leq M_{it}$ for all $i = 1, ..., I, t = 1, ..., T-1$. Furthermore, each stage has its own per period capacity, denoted by A_j^H, meaning the maximum input it may process at each period. The input quantity to stage j must not exceed the capacity, i.e., $\sum_{i=1}^{I} x_{ijt} \leq A_j^H$ for all $i = 1, ..., I, j = 1, ..., J, t = 1, ..., T-1$.

If no maintenance is conducted at a stage, w_{jt} falls by a random rate F_j, which follows a p.d.f. $\tilde{F}$, at the beginning of next period. A maintenance takes extra cost and decreases the per period capacity of stage j from A_j^H to A_j^L. In return, a maintenance restores w_{jt} to a random level K_j, which follows a p.d.f. $\tilde{K}$, at the beginning of the next period.

The objective is to decide optimal input quantity x_{ijt}, and maintenance decision z_{jt} at each period t, in order to minimize the total cost. P_{ijt} and R_{ij} denote production cost and inventory cost, respectively. Every unit of unfulfilled demand incurs a shortage cost Q_i. A maintenance incurs a maintenance cost S_j. Disposal cost U_{ij} is charged if any inventory is left at the end of period T. The indices used are stated in Table 3.1. The decision variables and the state variables are listed in Table 3.2. The meaning of parameters are listed in Table 3.3. Variables s_{ij} and r_{ij} are introduced later in this section.

doi:10.6342/NTU202403120

Notation	Description
i	Index of products, $i = 1, ..., I$.
j	Index of stages, $j = 0, ..., J$, where 0 is the stage of raw material, and J is the stage of end product.
t	Index of time periods, $t = 1, ..., T$.

Table 3.1: List of indices

doi:10.6342/NTU202403120

Notation	Description
x_{ijt}	Input quantity of product i to stage j at period t. $i = 1, ..., I$, $j = 0, ..., J, t = 1, ..., T-1$. $x_{i,0,t}$ is the purchase quantity of product i's raw material at period t, and $x_{i,J,t}$ is end product i's production quantity at period t.
z_{jt}	Whether a maintenance is conducted at stage j, period t, $z_{jt} \in \{0, 1\}$. Equals 1 if a maintenance is conducted, or 0 otherwise. $j = 1, ..., J$, $t = 1, ..., T-1$.
s_{it}	The quantity of product-i demand fulfilled at period t, $i = 1, ..., I$, $t = 1, ..., T$.
r_{jt}	Binary variable used to relax non-linear constraints. $r_{jt} \in \{0, 1\}, j = 1, ..., J$, $t = 1, ..., T-1$.
y_{ijt}	Beginning stage-j inventory level of product i at period t, $i = 1, ..., I$, $j = 0, ..., J, t = 1, ..., T$. $y_{i,0,t}$ denotes the beginning inventory level of product i's raw material in period t, and $y_{i,J,t}$ denotes the beginning inventory level of end product i at period t.
w_{jt}	The beginning yield rate of stage j at period t, $j = 1, ..., J, t = 1, ..., T$.

Table 3.2: List of variables

doi:10.6342/NTU202403120

Notation	Description
P_{ijt}	Unit production cost of product-i raw material, WIP, or end product at stage j, period t.
Q_i	Unit shortage cost of the demand for end product i.
R_{ij}	Unit inventory cost of product-i raw material, WIP, or end product in stage j.
S_j	One-period maintenance cost of stage j.
U_{ij}	Disposal cost if stage-j inventory of product i is left at the end of period T.
F_j	The random rate yield rate of stage j decreases if no maintenance is conducted.
$\tilde{F}$	The probability density function followed by F_j.
K_j	The random yield rate of stage j after a maintenance is conducted.
$\tilde{K}$	The probability density function followed by K_j.
B^L	The lowest yield rate stages would reach.
D_{it}	The demand for end product i at period t.
A_j^H	The per period capacity of stage j if no maintenance is conducted.
A_j^L	The per period capacity of stage j if a maintenance is conducted.
M_{it}	The limit of material supply of product i at period t.
$H(w,x)$	The output quantity given input quantity x and yield rate w. Follows binomial distribution where the number of experiments equals x and the success probability equals w.
$Y_{i,j,1}$	The beginning stage-j inventory level of product i at period 1.
$W_{j,1}$	The beginning yield rate of stage j at period 1.

Table 3.3: List of parameters

doi:10.6342/NTU202403120

Denote $(w_{1,t}, w_{2,t}, ..., w_{J,t})$ as vector w_t, and $(y_{1,0,t}, y_{1,1,t}, ..., y_{1,J,t}, y_{2,0,t}, ..., y_{2,J,t}, ..., y_{I,0,t}, ..., y_{I,J,t})$ as vector y_t. We use superscripted plus sign to transform a mathematical expression to itself when it is negative, and to 0 when it is positive, i.e., $E^+ = \max\{0, E\}$. Given the states and the decisions at period t, the states at period $t+1$ may be derived as

$$w_{j,t+1} = \max\{w_{jt} - F_j, B^L\}(1 - z_{jt}) + K_j z_{jt} \quad \forall j = 1, ..., J, t = 1, ..., T-1, \tag{3.1}$$

$$y_{i,0,t+1} = y_{i,0,t} + x_{i,0,t} - x_{i,1,t} \quad \forall i = 1, ..., I, t = 1, ..., T-1, \tag{3.2}$$

$$y_{i,j,t+1} = y_{ijt} + H(w_{jt}, x_{ijt}) - x_{i,j+1,t} \quad \forall i = 1, ..., I, j = 1, ..., J-1, t = 1, ..., T-1, \tag{3.3}$$

$$y_{i,J,t+1} = \big(y_{i,J,t} + H(w_{J,t}, x_{i,J,t}) - D_{it}\big)^+ \quad \forall i = 1, ..., I, t = 1, ..., T-1. \tag{3.4}$$

Denote $(z_{1,t}, z_{2,t}, ..., z_{J,t})$ as vector z_t and $(x_{1,0,t}, x_{1,1,t}, ..., x_{1,J,t}, x_{2,0,t}, ..., x_{2,J,t}, ..., x_{I,0,t}, ..., x_{I,J,t})$ as vector x_t. We denote $V_t(w_t, y_t)$ as the minimum cost at period t, given all state variables per period and formulate the problem as a dynamic programming model

$$\begin{aligned} V_t(w_t, y_t) = \min_{x_t, z_t} \quad & \mathbb{E}\Big(\sum_{i=1}^{I}\sum_{j=0}^{J}(P_{ij}x_{ijt} + R_{ij}y_{i,j,t+1}) + \sum_{j=1}^{J} S_j z_{jt} \\ & + \sum_{i=1}^{I} Q_i\big(D_{it} - y_{i,J,t} - H(w_{J,t}, x_{i,J,t})\big)^+ + V_{t+1}(w_{t+1}, y_{t+1})\Big), \\ \text{s.t.} \quad & (3.1) - (3.4) \\ & y_{i,j,1} = Y_{i,j,1} \quad \forall i = 1, ..., I, j = 0, ..., J, && (3.5) \\ & w_{j,1} = W_{j,1} \quad \forall j = 1, ..., J, && (3.6) \\ & x_{ijt} \le y_{i,j-1,t} \quad \forall i = 1, ..., I, j = 1, ..., J, && (3.7) \\ & x_{i,0,t} \le M_{it} \quad \forall i = 1, ..., I, && (3.8) \end{aligned}$$

doi:10.6342/NTU202403120

$$\sum_{i=1}^{I} x_{ijt} \le A_j^H(1 - z_{jt}) + A_j^L z_{jt} \quad \forall j = 1, ..., J, \tag{3.9}$$

$$x_{ijt} \ge 0 \quad \forall i = 1, ..., I, j = 1, ..., J, \tag{3.10}$$

$$z_{jt} \in \{0, 1\} \quad \forall j = 1, ..., J. \tag{3.11}$$

Denote our problem as $\mathcal{P}_S$. We analyze the problem by addressing the complexity of a deterministic version of $\mathcal{P}_S$ in the next section.

3.2 Complexity analysis

We formulate a deterministic version of $\mathcal{P}_S$, denoted as $\mathcal{P}_D$, by making stochastic parameters deterministic. Specifically, F_j and K_j are constants instead of random variables, while $H(w, x)$ is a linear function wx, instead of a random variable following binomial distribution. $\mathcal{P}_D$ can then be formulated as an MINLP from the dynamic programming model. Denote $(z_{1,1}, z_{2,1}, ..., z_{J,1}, z_{1,2}, ..., z_{J,T-1})$ as vector z and $(x_{1,0,1}, x_{1,1,1}, ..., x_{1,J,1}, x_{2,0,1}, ..., x_{I,J,1}, x_{1,0,2}..., x_{I,J,T-1})$ as vector x. The MINLP is formulated as

$$\begin{aligned}
\min_{x,z} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J-1}\sum_{t=1}^{T-1}\Big(P_{ij}x_{ijt} + R_{ij}(y_{ijt} + w_{jt}x_{ijt} - x_{i,j+1,t})\Big) + \sum_{j=1}^{J}\sum_{t=1}^{T-1} S_j z_{jt} \\
& + \sum_{i=1}^{I}\sum_{t=1}^{T-1}\Big(P_{i,J}x_{i,J,t} + R_{i,0}(y_{i,0,t} + x_{i,0,t} - x_{i,1,t}) + R_{i,J}(y_{i,J,t} + w_{J,t}x_{i,J,t} - s_{it}) \\
& + Q_i(D_{it} - s_{it})\Big) + \sum_{i=1}^{I}\Big(\sum_{j=0}^{J-1} U_{i,j}(y_{i,j,T}) + U_{i,J}(y_{i,J,T} - s_{i,T})\Big),
\end{aligned}$$

$$\text{s.t.} \quad x_{ijt} \le y_{i,j-1,t} \quad \forall i = 1, ..., I, j = 1, ..., J, t = 1, ..., T - 1, \tag{3.12}$$

$$x_{i,0,t} \le M_{it} \quad \forall i = 1, ..., I, t = 1, ..., T - 1, \tag{3.13}$$

$$\sum_{i=1}^{I} x_{ijt} \le A_j^H(1 - z_{jt}) + A_j^L z_{jt} \quad \forall j = 1, ..., J, t = 1, ..., T - 1, \tag{3.14}$$

doi:10.6342/NTU202403120

$$w_{j,1} = W_{j,1} \quad \forall j = 1, ..., J, \tag{3.15}$$

$$w_{j,t+1} \leq \hat{M}(1 - z_{jt}) + K_j z_{jt} \quad \forall j = 1, ..., J, t = 1, ..., T-1, \tag{3.16}$$

$$w_{j,t+1} \leq \hat{M} z_{jt} + \hat{M} r_{jt} + w_{jt} - F_j \quad \forall j = 1, ..., J, t = 1, ..., T-1, \tag{3.17}$$

$$w_{j,t+1} \leq \hat{M} z_{jt} + \hat{M}(1 - r_{jt}) + B^L \quad \forall j = 1, ..., J, t = 1, ..., T-1, \tag{3.18}$$

$$y_{i,j,1} = Y_{i,j,1} \quad \forall i = 1, ..., I, j = 0, ..., J, \tag{3.19}$$

$$y_{i,j,t+1} = y_{ijt} + w_{jt} x_{ijt} - x_{i,j+1,t} \quad \forall i = 1, ..., I, j = 1, ..., J-1, t = 1, ..., T-1, \tag{3.20}$$

$$y_{i,0,t+1} = y_{i,0,t} + x_{i,0,t} - x_{i,1,t} \quad \forall i = 1, ..., I, t = 1, ..., T-1, \tag{3.21}$$

$$y_{i,J,t+1} = y_{i,J,t} + w_{J,t} x_{i,J,t} - s_{it} \quad \forall i = 1, ..., I, t = 1, ..., T-1, \tag{3.22}$$

$$s_{i,1} \leq y_{i,J,1} \quad \forall i = 1, ..., I, \tag{3.23}$$

$$s_{it} \leq y_{i,J,t} + w_{J,t} x_{i,J,t} \quad \forall i = 1, ..., I, t = 2, ..., T, \tag{3.24}$$

$$s_{it} \leq D_{it} \quad \forall i = 1, ..., I, t = 1, ..., T, \tag{3.25}$$

$$x_{ijt} \geq 0 \quad \forall i = 1, ..., I, j = 1, ..., J, t = 1, ..., T-1, \tag{3.26}$$

$$s_{it} \geq 0 \quad \forall i = 1, ..., I, t = 1, ..., T, \tag{3.27}$$

$$z_{jt} \in \{0, 1\} \quad \forall j = 1, ..., J, t = 1, ..., T-1, \tag{3.28}$$

$$r_{jt} \in \{0, 1\} \quad \forall j = 1, ..., J, t = 1, ..., T-1. \tag{3.29}$$

Constraint (3.1) is linearized by introducing binary variable r_{jt}. Constraints (3.17), (3.18) and (3.29) keep w_{jt} in the correct range by r_{jt} and a large number $\hat{M}$. The objective function of $\mathcal{P}_S$ and constraint (3.4) are linearized by introducing variable s_{it}. By adding constraints (3.23), (3.24), (3.25), and (3.27), s_{it} is guaranteed to be the quantity of fulfilled demand of product i at period t.

doi:10.6342/NTU202403120

Following similar process of Florian et al. (1980), we show $\mathcal{P}_D$ to be NP-hard by reducing a well-known NP-hard problem SUBSET-SUM to $\mathcal{P}_D$. SUBSET-SUM is defined as follows.

SUBSET-SUM: Given a set $\{a_1, ..., a_N\}$ and a number G, decide if there is a subset P such that $\sum_{i \in P} a_i = G$.

Let $\bar{M}$ be a large number. Given a SUBSET-SUM instance, we may construct a $\mathcal{P}_D$ instance $\mathcal{I}_{\mathcal{P}_D}$ with $I = 1, J = 1, T = 3N$, and the remaining parameter settings shown in table 3.4.

doi:10.6342/NTU202403120

The MINLP model of $\mathcal{I}_{\mathcal{P}_D}$ may be derived as

$$\min_{x,z} \quad \sum_{t=1}^{3N-1}\Big(\sum_{j=0}^{1} P_{1,j,t}x_{1,j,t} + \bar{M}(D_{1,t} - s_{1,t}) + z_{i,t}\Big) + \bar{M}(y_{1,1,3N} - s_{1,3N}),$$

$$\text{s.t.} \quad x_{1,0,t} \leq M_{1,t} \quad \forall t = 1, ..., 3N-1,$$

$$x_{1,1,t} \leq y_{1,0,t} \quad \forall t = 1, ..., 3N-1,$$

$$x_{1,1,t} \leq A_1^H(1 - z_{1,t}) + A^L z_{1,t} \quad \forall t = 1, ..., 3N-1,$$

$$w_{1,1} = W_{1,1},$$

$$w_{1,t+1} = z_{1,t} \quad \forall t = 1, ..., 3N-1,$$

$$y_{1,0,1} = Y_{1,0,1},$$

$$y_{1,1,1} = G(N-1),$$

$$y_{1,1,t+1} = y_{1,1,t} + w_{1,t}x_{1,1,t} - s_{1,t+1} \quad \forall t = 1, ..., 3N-1,$$

$$s_{1,t} \leq y_{1,1,t} + x_{1,1,t} \quad \forall t = 2, ..., 3N,$$

$$s_{1,t} \leq D_{1,t} \quad \forall t = 1, ..., 3N,$$

$$x_{1,j,t} \geq 0 \quad \forall j = 0, 1, t = 1, ..., 3N-1,$$

$$s_{1,t} \geq 0 \quad \forall t = 1, ..., 3N,$$

$$z_{1,t} \in \{0, 1\} \quad \forall t = 1, ..., 3N-1.$$

In this instance, the stage fully recovers after every maintenance, then immediately break down at the next period given that both F_1 and K_1 have a value of one. To obtain a cost equals G, all purchased materials must be consumed at the next period, given that the inventory cost of raw material equals a large number. Shortage cost and disposal cost of end product are also a large number, making total cost to be G only if demands at every

doi:10.6342/NTU202403120

Parameter	Value
$P_{1,0,t}$	$0 \quad \forall t = 1, ..., 3N$
$P_{1,1,t}$	$\begin{cases} \frac{a_{(t+1)/3}-1}{a_{(t+1)/3}}, & t \equiv 2 \pmod 3 \\ \bar{M}, & \text{otherwise} \end{cases} \quad \forall t = 1, ..., 3N$
Q_1	$\bar{M}$
$R_{1,0}$	$\bar{M}$
$R_{1,1}$	0
S_1	1
$U_{1,0}$	0
$U_{1,1}$	$\bar{M}$
F_1	1
K_1	1
$D_{1,t}$	$\begin{cases} G, & t \equiv 0 \pmod 3 \\ 0, & \text{otherwise} \end{cases} \quad \forall t = 1, ..., 3N$
A_1^H	$\bar{M}$
A_1^L	0
$M_{1,t}$	$\begin{cases} a_{(t+2)/3}, & t \equiv 1 \pmod 3 \\ 0, & \text{otherwise} \end{cases} \quad \forall t = 1, ..., 3N$
B^L	0
$Y_{1,0,1}$	0
$Y_{1,1,1}$	$(N-1)G$
$W_{1,1}$	0

Table 3.4: Settings of $\mathcal{I}_{\mathcal{P}_D}$

doi:10.6342/NTU202403120

period fulfilled and no inventory of end product is left at the end of period $3N$.

There are G units of demand for end product every three periods. Whenever a planner wish to produce a product, a period must be taken to maintain the stage and to purchase raw materials. After that, the production is conducted at the next period. Finally, the end product is realized at the beginning of the third period. Therefore each set of three periods may be viewed as a group, resulting in a total of N groups. In each period group n, a_n units of raw material may be input into stage 1 after maintenance. Moreover, the production cost subtracts maintenance cost equals a_n if exactly a_n units of products are produced.

There are NG units of demand in total and $(N-1)G$ units of initial inventory of end product, hence G units of products to be produced to avoid shortage and disposal cost. If there exists a plan with total cost of G, there must be a set of period groups whose capacities sum up to G. The supply of raw material correspond to $\{a_1, ..., a_N\}$ in SUBSET-SUM problem. Therefore, the SUBSET-SUM problem returns True if there exists a feasible production plan for $\mathcal{I}_{\mathcal{P}_D}$ with cost G, thereby completing the proof. The only exception is that any number in $a_1, ..., a_N$ lower than 1 may result in a negative production cost, as indicated by the function $\frac{a_n - 1}{a_n}$. In such cases, a transformation that ensures all values of $a_1, ..., a_N$ are greater than 1 can help maintain the validity of the proof.

In the dynamic programming model of $\mathcal{P}_S$, the number of variables increases exponentially by the instance size, making it impractical to solve the model. Therefore we propose heuristic algorithms for $\mathcal{P}_S$ in the next chapter.

doi:10.6342/NTU202403120

Chapter 4

The Algorithms

4.1 The overall procedure

We propose algorithms that plan maintenance by heuristic rules first. Production plan is then decided by solving linear programming model according to the maintenance plan. Finally the algorithms apply the plans for one period, yields states of the next period, then repeat the process until the last period. The overall process is illustrated in Figure 4.1.

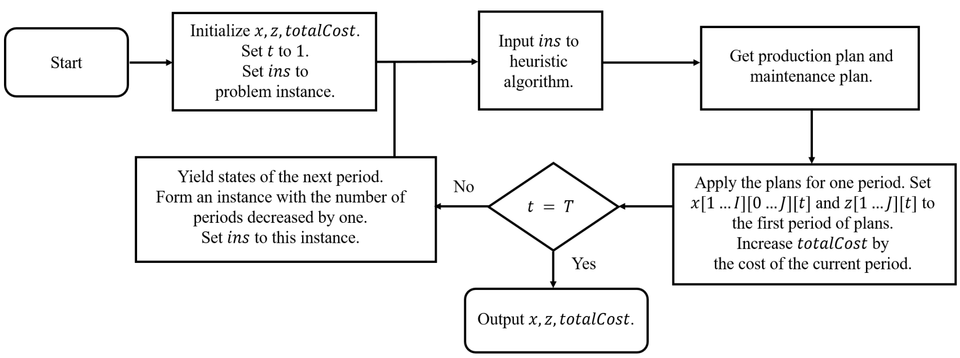

Figure 4.1: An illustration of the overall process

doi:10.6342/NTU202403120

The maintenance plan is determined using heuristic algorithms described in the following sections. After obtaining the maintenance plan, the expected yield rate of all periods may also be acquired. Denote the maintenance plan and expected yield rate as Z and W respectively, a linear programming model may be formed from the MINLP in Chapter 3 since now maintenance plan and yield rate are parameters instead of variables. The linear programming problem may be formulated as

$$\begin{aligned}
\min_{x} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J-1}\sum_{t=1}^{T}\Big(P_{ij}x_{ijt} + R_{ij}(y_{ijt} + W_{jt}x_{i,j,t} - x_{i,j+1,t})\Big) + \sum_{j=1}^{J}\sum_{t=1}^{T} S_j Z_{jt} \\
& + \sum_{i=1}^{I}\sum_{t=1}^{T}\Big(P_{i,0}x_{i,0,t} + P_{i,J}x_{i,J,t} + R_{i,0}(y_{i,0,t} + x_{i,0,t} - x_{i,1,t}) \\
& + R_{i,J}(y_{i,J,t} + W_{J,t}x_{i,J,t} - s_{it}) + Q_i(D_{it} - s_{it})\Big) \\
& + \sum_{i=1}^{I}\Big(\sum_{j=0}^{J-1} U_{ij}y_{i,j,T} + U_{i,J}(y_{i,J,T} - s_{i,T})\Big),
\end{aligned}$$

$$\text{s.t.} \quad x_{ijt} \le y_{i,j-1,t} \quad \forall i = 1, ..., I, j = 1, ..., J, t = 1, ..., T, \tag{4.1}$$

$$x_{i,0,t} \le M_{it} \quad \forall i = 1, ..., I, t = 1, ..., T, \tag{4.2}$$

$$\sum_{i=1}^{I} x_{ijt} \le A^H(1 - Z_{jt}) + A^L Z_{jt} \quad \forall j = 1, ..., J, t = 1, ..., T, \tag{4.3}$$

$$y_{i,j,1} = Y_{i,j,1} \quad \forall i = 1, ..., I, j = 1, ..., J-1, \tag{4.4}$$

$$y_{i,j,t+1} = y_{ijt} + W_{jt}x_{ijt} - x_{i,j+1,t} \quad \forall i = 1, ..., I, j = 1, ..., J-1, t = 1, ..., T-1, \tag{4.5}$$

$$y_{i,0,t+1} = y_{i,0,t} + x_{i,0,t} - x_{i,1,t} \quad \forall i = 1, ..., I, t = 1, ..., T-1, \tag{4.6}$$

$$y_{i,J,t+1} = y_{i,J,t} + W_{J,t}x_{i,J,t} - s_{it} \quad \forall i = 1, ..., I, t = 1, ..., T-1, \tag{4.7}$$

$$s_{i,1} \le y_{i,J,1} \quad \forall i = 1, ..., I, \tag{4.8}$$

$$s_{it} \le y_{i,J,t} + W_{J,t}x_{i,J,t} \quad \forall i = 1, ..., I, t = 2, ..., T, \tag{4.9}$$

doi:10.6342/NTU202403120

$$s_{it} \leq D_{it} \quad \forall i = 1, ..., I, t = 1, ..., T, \tag{4.10}$$

$$x_{ijt} \geq 0 \quad \forall i = 1, ..., I, j = 1, ..., J, t = 1, ..., T, \tag{4.11}$$

$$s_{it} \geq 0 \quad \forall i = 1, ..., I, t = 1, ..., T. \tag{4.12}$$

Solving the LP model yields a production plan. The algorithms apply the maintenance and production plan for one period, then the states of the next period may be obtained, forming an instance with T decreased by 1. After that, the procedure above may be repeated until the last period. The steps in one iteration is shown in Figure 4.2 . The complete procedure is summarized in Algorithm 4.1. It takes the instance to be solved $insToSolve$, then returns the production plan x, the maintenance plan z, and the total cost $totalCost$.

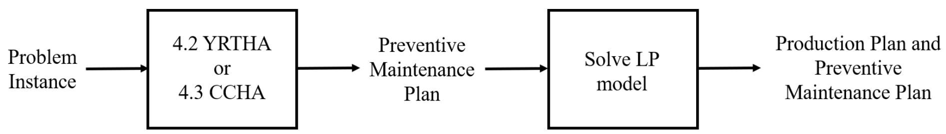

Figure 4.2: The steps conducted in one iteration

4.2 Yield rate threshold heuristic algorithm (YRTHA)

A possible heuristic algorithm for determining the maintenance plan follows a simple idea. The yield rate deteriorates by period, making the production inefficient. The algorithm sets a threshold of yield rate. If the current yield rate of a stage is lower than the threshold, a maintenance is conducted at the current period. For the maintenance plan of remaining periods, the algorithm sets a fixed frequency to conduct maintenances according to the falling speed of yield rate, e.g., the mean of $\tilde{F}$. The steps are shown

doi:10.6342/NTU202403120

Algorithm 4.1 The overall process

function SOLVE($insToSolve$)

    $totalCost \leftarrow 0$, $ins \leftarrow insToSolve$

    $x[1...insToSolve.I][0...insToSolve.J][1...insToSolve.T-1] \leftarrow 0$

    $z[1...insToSolve.J][1...insToSolve.T-1] \leftarrow 0$

    $numIterations \leftarrow ins.T - 1$

    for t in $1, ..., numIterations$ **do**

        $ins.Z \leftarrow$ NULL

        $zPlan \leftarrow$ HEURISTIC(ins)

        $ins.Z \leftarrow zPlan$

        $z[0...insToSolve.J][t] \leftarrow zPlan[0...insToSolve.J][1]$

        $w[1...ins.J][1...ins.T-1] \leftarrow 0$

        $w[1...ins.J][1] \leftarrow ins.W[1...ins.J][1]$

        for j in $1, ..., ins.J$ **do**

            for t in $2, ..., ins.T-1$ **do**

                if $ins.Z[j][t-1] > 0$ **then**

                    $w[j][t] \leftarrow \text{mean}(ins.\tilde{K})$

                else

                    $w[j][t] \leftarrow \max\{ins.B^L, w[j][t-1] \text{ - } \text{mean}(ins.\tilde{F})\}$

                end if

            end for

        end for

        $ins.W \leftarrow w$

        $xPlan \leftarrow$ solveLP(ins)

doi:10.6342/NTU202403120

$x[1...insToSolve.I][0...insToSolve.J][\text{t}] \leftarrow xPlan[1...ins.I][0...ins.J][1]$

$wNext[1...\ ins.J] \leftarrow 0$

for j in $1, ..., ins.J$ **do**

  if $zPlan[j] = 1$ **then**

    $wNext[j] \leftarrow \text{RandomSample}(ins.\tilde{K})$

  else

    $wNext[j] \leftarrow \max\{w[j][1] - \text{RandomSample}(ins.\tilde{F}),\ ins.B^L\}$

  end if

end for

$yNext \leftarrow \text{CalculateInventoryLeft}()$

$costPerPeriod \leftarrow \text{CalculateCostPerPeriod}()$

$totalCost \leftarrow totalCost + costPerPeriod$

$ins.T \leftarrow ins.T - 1$

$ins.W[1...ins.J][1] \leftarrow wNext$

$ins.Y[1...ins.I][1...ins.J][1] \leftarrow yNext$

Remove $ins.D[1...ins.I][1]$

Remove $ins.M[1...ins.I][1]$

end for

return x, z, $totalCost$

end function

doi:10.6342/NTU202403120

in 4.3, and the procedure in detail is summarized in Algorithm 4.2. It takes a problem instance ins and a manually set threshold w, then returns a maintenance plan z.

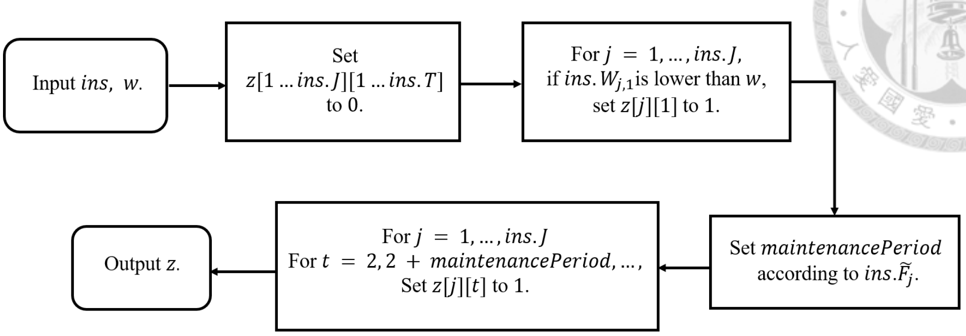

Figure 4.3: The steps of YRTHA

YRTHA is applied in Algorithm 4.1 by substituting HEURISTIC in line 8 to YRTHA and provide an additional threshold parameter w. The algorithm has polynomial time complexity by solving LP and repeating procedure for $T-1$ times. While it is an efficient algorithm, deciding an appropriate threshold is a complicated task. It may be viewed as a searching problem to decide an optimal threshold from B^L to 1. In the overall process of our algorithm, we manually assign several threshold levels to perform grid search, with the best threshold returned. We measure the performance of this algorithm in Chapter 5 and propose another heuristic algorithm in the next section.

4.3 Cost comparison heuristic algorithm (CCHA)

4.3.1 Algorithm description

In this section we propose another possible way to decide whether the stage at current period should be maintained. The idea is to measure the cost of conducting and not conducting a maintenance by observing the state of current period and a few periods in

doi:10.6342/NTU202403120

Algorithm 4.2 Yield Rate Threshold Heuristic Algorithm

function YRTHA(ins, w)
  $z[1...ins.J][1...ins.T-1] \leftarrow 0$
  for j in 1...ins.J **do**
    if ins.$W[j][1] < w$ **then**
      z[j][1] $\leftarrow 1$
    else
      z[j][1] $\leftarrow 0$
    end if
  end for
  for j in 1...$ins.J$ **do**
    $maintenancePeriod \leftarrow$ ceil((mean($ins.\tilde{K}$) $-B^L$) / (2×mean($ins.\tilde{F}$)))
    $t \leftarrow 1$
    while $t < ins.T$ **do**
      $z[j][t] \leftarrow 1$
      $t \leftarrow t + maintenancePeriod$
    end while
  end for
  return z
end function

doi:10.6342/NTU202403120

near future. Assuming the stages operate at full capacity, the production at stage j goes through $J-j$ stages and takes $J-j+1$ periods until the end products are realized. In other words, the production of stage j at period p is directly related to the beginning inventory of end product at period $p+J-j+1$. If period $p+J-j+1$ faces a large quantity of demand, the maintenance decision of stage j at period p may be critical. If a maintenance is conducted, and the capacity this period is fallen by C, there may be C units of shortage, hence the corresponding shortage cost.

On the other hand, while a maintenance may sacrifice the demand at a period, it benefits periods after that by keeping the stage in a better condition. In fact, the cost of not conducting maintenances becomes significant in these later periods. Not conducting maintenance may allow stages to fulfill demand temporarily, but sacrifices the chance of bringing yield rate of the stage to a higher level. Extra shortage cost is possible if the stage produce at a lower yield rate level in the later periods.

Therefore, the idea of algorithm is to compare the cost of conducting and not conducting maintenance at the current period, then apply the decision with lower cost. After the maintenance plan of current period is made, the procedure of planning the remaining periods and solving LP to yield production plan is same as the yield rate threshold algorithm.

The algorithm applies a greedy strategy to consider several periods in near future from now. Specifically, we consider the cost of totally 3 periods. It is assumed that stages produce at their maximum capacity. For the first period, the initial inventory is also considered since inventory also limits the production. There may be bottleneck stages, i.e., the stages with rather low capacity level, in a problem instance. This algorithm

doi:10.6342/NTU202403120

identifies the bottleneck stage with lowest capacity. For stages after the bottleneck stage, the capacity of the bottleneck stage also limits their production.

The cost is composed of shortage, inventory, and maintenance cost of 3 periods. Let $\bar{Q}$ denote the average shortage cost of all products, and $\bar{R}$ denote the average inventory cost of all products and stages. μ_K and μ_F denote the mean of $\tilde{K}$ and $\tilde{F}$, respectively. Let A_B^H denote per period capacity of the bottleneck stage. We use superscripted M and N to represent the decision of whether or not to conduct a maintenance. Subscripted BB and AB are used to represent stages before and after bottleneck stage. The cost of whether or not conducting maintenance at each stage is listed in Table 4.1.

To decide the maintenance plan, stages that satisfies $p + J - j > T$ are planned to not conduct maintenances since any production is too late to meet the demand, therefore no maintenance is required. For the other stages, the algorithm calculates the cost of conducting and not conducting a maintenance by the cost functions in Table 4.1. The option with less cost is adopted at the current period. Then a similar approach described in Algorithm 4.2 is adopted to decide the maintenance plan at the remaining periods. The steps are shown in 4.4, and the procedure in detail is described in Algorithm 4.3. It takes a problem instance ins, then returns a maintenance plan z.

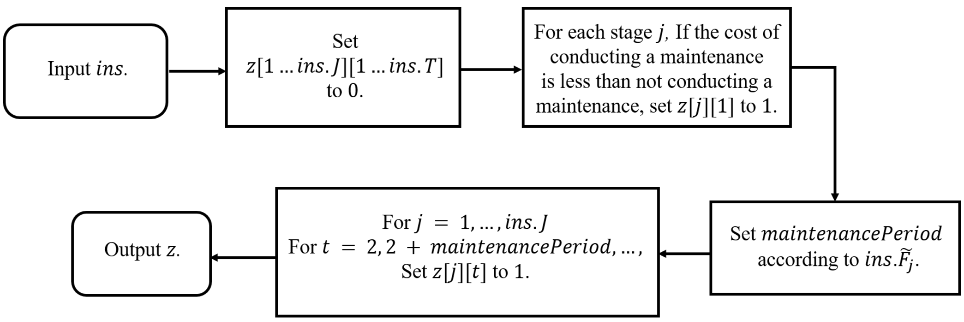

Figure 4.4: The steps of CCHA

doi:10.6342/NTU202403120

Cost	Value
C_{BB}^{M}	$\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+1} - w_{jp}\min(A^{L}, \sum_{i=0}^{I} y_{i,j-1,t}))^{+}$ $+\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+2} - \mu_{K}A^{H})^{+}$ $+\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+3} - (\mu_{K} - \mu_{F})A^{H})^{+}$ $-\bar{R}(\min(A^{L}, \sum_{i=0}^{I} y_{i,j-1,t}) + A^{H} + A^{H})+$ S_j
C_{BB}^{N}	$\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+1} - w_{jp}\min(A^{H}, \sum_{i=0}^{I} y_{i,j-1,t}))^{+}$ $+\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+2} - (w_{jp} - \mu_{F})A^{H})^{+}$ $+\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+3} - (w_{jp} - 2\mu_{F})A^{H})^{+}$ $-\bar{R}(\min(A^{H}, \sum_{i=0}^{I}, y_{i,j-1,t}) + A^{H} + A^{H})$
C_{AB}^{M}	$\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+1} - w_{jp}\min(A^{L}, \sum_{i=0}^{I} y_{i,j-1,t}, A^{HB}))^{+}$ $+\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+2} - \mu_{K}\min(A^{H}, A^{HB}))^{+}$ $+\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+3} - (\mu_{K} - \mu_{F})\min(A^{H}, A^{HB}))^{+}$ $-\bar{R}(\min(A^{L}, \sum_{i=0}^{I} y_{i,j-1,t}, A^{HB}) + 2\min(A^{H}, A^{HB}))$ $+S_j$
C_{AB}^{N}	$\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+1} - w_{jp}\min(A^{H}, \sum_{i=0}^{I} y_{i,j-1,t}, A^{HB}))^{+}$ $+\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+2} - (w_{jp} - \mu_{F})\min(A^{H}, A^{HB}))^{+}$ $+\bar{Q}(\sum_{i=0}^{I} D_{i,p+J-j+3} - (w_{jp} - 2\mu_{F})\min(A^{H}, A^{HB}))^{+}$ $-\bar{R}(\min(\sum_{i=0}^{I} y_{i,j-1,t}, A^{H}, A^{HB}) + 2\min(A^{H}, A^{HB}))$

Table 4.1: Cost of whether or not conducting maintenance at each stage

doi:10.6342/NTU202403120

Algorithm 4.3 Cost Comparison Heuristic Algorithm

function CCHA(ins)
    $z[1...ins.J][1...ins.T-1] \leftarrow 0$
    $costMaintenance \leftarrow 0, costNoMaintenance \leftarrow 0$
    $bottleNeckStage \leftarrow$ findBottleNeck(ins)
    for j in 1 ... $bottleNeckStage$ **do**
        if $J - j + 3 > T$ **then**
            break
        end if
        if $j \leq bottleNeckStage$ **then**
            $costMaintenance \leftarrow$ costMaintenanceBeforeBottleNeck(j)
            $costNoMaintenance \leftarrow$ costNoMaintenanceBeforeBottleNeck(j)
        else
            $costMaintenance \leftarrow$ costMaintenanceAfterBottleNeck(j)
            costNoMaintenance $\leftarrow$ costNoMaintenanceAfterBottleNeck(j)
        end if
        if $costMaintenance < costNoMaintenance$ **then**
            $z[j][1] \leftarrow 1$
        else
            $z[j][1] \leftarrow 0$
        end if
    end for
    for j in $1...ins.J$ **do**
        $maintenancePeriod \leftarrow$ ceil((mean($ins.\tilde{K}$) $-B^L$) / (2mean($ins.\tilde{F}$)))

doi:10.6342/NTU202403120

```
        t ← 1
        while t < ins.T do
            z[j][t] ← 1
            t ← t + maintenancePeriod
        end while
    end for
    return z
end function
```

CCHA is applied in Algorithm 4.1 by substituting HEURISTIC in line 8 to CCHA. The time complexity of this algorithm is same as YRTHA. The advantage of this algorithm is that it is simple to be implemented, requiring no extra parameter settings. Moreover, the algorithm considers the relationship between stages and demand. Chapter 5 compares the performance of this algorithm with the others.

4.3.2 A numerical example

In this section, we provide a numerical example to illustrate how the algorithm works. Given a problem instance with 3 products, 5 stages, and 8 periods. Stage 3 is set to be a bottleneck stage. The demand is set as

$$D = \begin{bmatrix} 100 & 100 & 100 & 150 & 250 & 100 & 150 & 200 \\ 100 & 100 & 100 & 50 & 100 & 50 & 200 & 50 \\ 100 & 100 & 100 & 50 & 200 & 100 & 50 & 50 \end{bmatrix},$$

where the row represents products and columns represents periods. There is sufficient initial inventory, and the initial yield rate equals 0.6 for all j. The remaining of the

doi:10.6342/NTU202403120

Parameter	Value
Q_i	$100 \quad \forall i = 1, ..., I$
R_{ij}	$2 \quad \forall i = 1, ..., I, j = 0, ..., J$
S_j	$4000 \quad \forall j = 1, ..., J$
$\tilde{F}$	Normal distribution with mean equals 0.1
$\tilde{K}$	Normal distribution with mean equals 1
A_j^H	$100 \quad \forall j = 1, 2, 4, 5$
A_j^L	$50 \quad \forall j = 1, 2, 4, 5$
A_3^H	50
A_3^L	25

Table 4.2: List of parameters

parameters are set as shown in Table 4.2.

$\bar{Q}$ and $\bar{R}$ are 100 and 2 respectively, which is calculated from average values of Q and R. Similarly, μ_F and μ_K are 0.1 and 1 respectively, calculated from the mean of $\tilde{F}$ and $\tilde{K}$. For stage 1 at period 1, the production is directly related to the beginning inventory of end product at period $1 + 5 - 1 + 1 = 6$, whose demand equals 250 in total, as well as period 7 and 8. Since it is a stage before the bottleneck stage, C_{BB}^M and C_{BB}^N are applied to calculate the costs.

The cost of conducting a maintenance is calculated as $C_{BB}^M = 100 \times (250 - 0.6 \times 50) + 100 \times (400 - 1 \times 100) + 100 \times (300 - 0.9 \times 100) - 2 \times (50 + 100 + 100) + 4000 = 76500$.

The cost of not conducting a maintenance is calculated as $C_{BB}^N = 100 \times (250 - 0.6 \times 100) + 100 \times (400 - 0.5 \times 100) + 100 \times (300 - 0.4 \times 100) - 2 \times (100 + 100 + 100) = 79400$.

doi:10.6342/NTU202403120

Since the cost of conductung a maintenance is cheaper, it is planned to conduct a maintenance at stage 1, period 1.

Take stage 4 at period 2 as another example. the production is directly related to the beginning inventory of end product at period $2+5-4+1=4$, as well as period 5 and 6. Since it is a stage after the bottleneck stage, C^M_{AB} and C^N_{AB} are applied to calculate the costs.

The cost of conducting a maintenance is calculated as $C^M_{AB} = 100 \times (250-0.6\times 25) + 100 \times (550 - 1 \times 50) + 100 \times (250 - 0.9 \times 50) - 2 \times (50 + 2 \times 50) + 4000 = 97750$.

The cost of not conducting a maintenance is calculated as $C^N_{AB} = 100 \times (250 - 0.6 \times 50) + 100 \times (550 - 0.5 \times 50) + 100 \times (250 - 0.4 \times 50) - 2 \times (50 + 2 \times 50) = 97200$.

Since the cost of not conductung a maintenance is cheaper, it is planned to not conduct a maintenance at stage 4, period 2.

doi:10.6342/NTU202403120

Chapter 5

Numerical Study

5.1 Experiment setting

In this chapter, we conduct numerical experiments to examine the effectiveness of proposed algorithms. In our experiments, $W_{j,1}$ is set to 0.96, and $Y_{i,j,1}$ is set to 0. Since there are no initial inventory, the demands of first 5 periods may not be fulfilled. Considering the problem complexity and real-life application, we set T to 21, which would be three weeks if a period is considered as a day. We define four scenarios with different number of products and stages as listed in Table 5.1.

J is set to 4 in multiple stage scenarios, making it possible to fulfill the demand starting from period 5 given no initital inventory. Considering the difference of efficiency to produce under single and multiple stage, we lower A_j^H under single stage scenarios to reduce the bias. Specifically, A_j^H under single stage is set to 0.65 times A_j^H under multiple stages according to the benchmark performance. M_{ij} applies the same setting.

doi:10.6342/NTU202403120

Scenario	I	J
SPSS	1	1
SPMS	1	4
MPSS	3	1
MPMS	3	4

Table 5.1: Settings of scenarios

We adopt four factors that would affect the performance of proposed algorithms. First, we set ratio of demand per period and stage capacity to simulate different level of tightness to supply the demand. The ratio is set to 2:1 and 3:1, with A_j^H under multiple stages equals 30000, and average demand per period set to 60000 and 90000. Next, we consider two types of demand distribution, even and uneven. For even distribution, the demand at every period follows normal distribution with small variance. In contrast, uneven distribution has demand only at periods 3, 7, 10, 14, 17, and 21. Following is the ratio of inventory and shortage cost, setting to 1:50 and 1:100. The difference between two costs should be significant enough, otherwise the optimal plan would always be conducting no maintenance and producing zero product. Both costs are identical for all products and stages. Finally, we randomly assign one stage to be the bottleneck stage under SPMS and MPMS scenarios. The capacity of bottleneck stage is set to A_j^H, $0.75A_j^H$, and $0.5A_j^H$.

The remaining of the parameters are set as follows. $P_{ij} = 2$ for all $i = 1, ..., I, j = 0, ..., J$, $S_j = 20$ for all $j = 1, ..., J$, and $U_{ij} = 0$ for all $i = 1, ..., I, j = 0, ..., J$. A_j^L is set to half of A_j^H. B^L equals 0.8. $\tilde{F}$ is a normal distribution with mean equals 0.01 and

doi:10.6342/NTU202403120

standard deviation equals 0.0001. $\tilde{K}$ is a normal distribution with mean equals 1 and standard deviation equals 0.001. The random sample from $\tilde{K}$ would be automatically adjusted to 1 if it exceeds 1.

The above four scenarios and four factors generate $4 \times 2 \times 2 \times 2 \times 3 = 96$ scenarios in total. We generate 30 instances for each scenario and report the average performance. The experiments were performed on a laptop equipped with two 2.30Gz Intel(R) Core i5-2600U CPUs and 12GB RAM. The heuristic algorithms are implemented using Python 3.9.13. Gurobi Optimizer version 9.5.1 build v9.5.1rc2 (win64) is invoked to solve mathematical models. We apply $\{0.1, 0.2, ..., 0.9\}$ in YRTHA to perform grid search and report the best performance.

5.2 Benchmarks

5.2.1 The first benchmark: A lower bound

To evaluate the performance of each algorithm, we solve a special scenario of $\mathcal{P}_S$ as benchmark. In this scenario, F_j equals 0 and $W_{j,1}$ equals 1. In other words, the yield rate is always 100%, allowing all production to be fully realized without loss, therefore no maintenance is required. In this case, the only decision left is x_{ijt}, and this problem may be formulated as a linear program. The program is formulated as

$$\begin{aligned}\min_{x} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J-1}\sum_{t=1}^{T-1}\Big(P_{ij}x_{ijt} + R_{ij}(y_{ijt} + x_{i,j,t} - x_{i,j+1,t})\Big) \\ & + \sum_{i=1}^{I}\sum_{t=1}^{T-1}\Big(P_{i,0}x_{i,0,t} + P_{i,J}x_{i,J,t} + R_{i,0}(y_{i,0,t} + x_{i,0,t} - x_{i,1,t})\end{aligned}$$

doi:10.6342/NTU202403120

$$+ R_{i,J}(y_{i,J,t} + x_{i,J,t} - s_{it}) + Q_i(D_{it} - s_{it})\Big)$$

$$+ \sum_{i=1}^{I}\Big(\sum_{j=0}^{J-1} U_{i,j} y_{i,j,T} + U_{i,J}(y_{i,J,T} - s_{i,T})\Big),$$

$$\text{s.t.} \quad x_{ijt} \le y_{i,j-1,t} \quad \forall i = 1, ..., I, j = 1, ..., J, t = 1, ..., T-1, \tag{5.1}$$

$$x_{i,0,t} \le M_{it} \quad \forall i = 1, ..., I, t = 1, ..., T-1, \tag{5.2}$$

$$\sum_{i=1}^{I} x_{ijt} \le A_j^H \quad \forall j = 1, ..., J, t = 1, ..., T-1, \tag{5.3}$$

$$y_{i,j,1} = Y_{i,j,1} \quad \forall i = 1, ..., I, j = 0, ..., J, \tag{5.4}$$

$$y_{i,j,t+1} = y_{ijt} + x_{ijt} - x_{i,j+1,t} \quad \forall i = 1, ..., I, j = 1, ..., J-1, t = 1, ..., T-1, \tag{5.5}$$

$$y_{i,0,t+1} = y_{i,0,t} + x_{i,0,t} - x_{i,1,t} \quad \forall i = 1, ..., I, t = 1, ..., T-1, \tag{5.6}$$

$$y_{i,J,t+1} = y_{i,J,t} + x_{i,J,t} - s_{it} \quad \forall i = 1, ..., I, t = 1, ..., T-1, \tag{5.7}$$

$$s_{i,1} \le y_{i,J,1} \quad \forall i = 1, ..., I, \tag{5.8}$$

$$s_{it} \le y_{i,J,t} + x_{i,J,t-1} \quad \forall i = 1, ..., I, t = 2, ..., T, \tag{5.9}$$

$$s_{it} \le D_{it} \quad \forall i = 1, ..., I, t = 1, ..., T, \tag{5.10}$$

$$x_{ijt} \ge 0 \quad \forall i = 1, ..., I, j = 1, ..., J, \tag{5.11}$$

$$s_{it} \ge 0 \quad \forall i = 1, ..., I, t = 1, ..., T. \tag{5.12}$$

The solution of this problem may be immediately obtained by invoking an LP solver. Since this is an idealized scenario where the productions are maximally efficient, the optimal value is a loose lower bound.

doi:10.6342/NTU202403120

5.2.2 The second benchmark: MINLP heuristic algorithm (MHA)

In order to avoid possible bias caused by the over-idealized settings in the benchmark algorithm, we introduce another algorithm that yields more reasonable optimal values. The algorithm is denoted as *MHA*, and the steps are described as follows. First, invoke solver to solve $\mathcal{P}_D$ and apply the solution plan for one period. The plan yields states of the next period, then an instance with period less by one may be formed. It then can be solved by MINLP solver again, then the procedure repeats until the last period. The complete steps are summarized in Algorithm 5.1. It takes a instance to be solved $insToSolve$, then returns the production plan x, the maintenance plan z, and the total cost $totalCost$.

MHA yields solution with small gap to the lower bound. However, it is time consuming to solve MINLP as the instance size grows. In our experiments, we set 90 seconds as time limit for solving MINLP. All models in the testing instances yield feasible solution.

5.3 Experimental results

Denote an optimal value as z^*, and the value to be measured as z. The optimality gap is calculated as $\frac{z-z^*}{z^*}$. Denote gap between the algorithms and benchmark 1 as GAP_{OPT}. Denote gap between the algorithms and benchmark 2 as GAP_{MHA}. We show the performance of YRTHA and CCHA, and how the selected factors influence performance of the algorithms.

Table 5.2 shows the numerical result of the single-product-single-stage scenario. It is shown that the benchmark algorithm indeed yields a looser bound, comparing to MHA.

doi:10.6342/NTU202403120

Algorithm 5.1 MINLP Heuristic Algorithm

```
function MHA(insToSolve)
    totalCost ← 0, ins ← NULL
    x[1...insToSolve.I][0...insToSolve.J][1...insToSolve.T − 1] ← 0
    z[1...insToSolve.J][1...insToSolve.T − 1] ← 0
    ins ← insToSolve
    numIterations ← ins.T − 1
    w ← ins.W[1...ins.J][1]
    for t in 1, ..., numIterations do
        if t = 1 then
        end if
        xPlan, zPlan ← minlp(ins)
        x[1...insToSolve.I][0...insToSolve.J][t] ← xPlan[1...ins.I][0...ins.J][1]
        z[0...insToSolve.J][t] ← zPlan[0...ins.J][1]
        wNext[1...ins.J] ← 0
        for j in 1, ..., ins.J do
            if zPlan[j][0] = 1 then
                wNext[j] ← RandomSample(ins.K̃)
            else
                wNext[j] ← max{w[j]− RandomSample(ins.F̃), ins.B^L}
            end if
        end for
        yNext ← CalculateInventoryLeft()
        costPerPeriod ← CalculateCostPerPeriod()
```


doi:10.6342/NTU202403120

$totalCost \leftarrow totalCost + costPerPeriod$

$ins.T \leftarrow ins.T - 1$

$ins.W[1...ins.J][1] \leftarrow wNext$

$ins.Y[1...ins.I][1...ins.J][1] \leftarrow yNext$

Remove $ins.D[1...ins.I][1]$

Remove $ins.M[1...ins.I][1]$

end for

return x, z, $totalCost$

end function

In general, YRTHA yields better results than CCHA.

Both YRTHA and CCHA have smaller optimality gap when facing higher demand given fixed capacity. The advantage of benchmark algorithm is limited by the capacity crunch, hence the closer gaps. Similar results are shown under different levels of bottleneck stage capacity. The lower capacity of bottleneck stage yields smaller gap between proposed algorithms and the benchmark. The gaps are shown larger under uneven demand distribution, since the benchmark algorithm plans with perfect yield rate and meets the peak demands better. On the other hand, both algorithms have close optimality gaps under even demand distribution, while YRTHA reaches almost the same performance of MHA, showing that steady demand is well-handled by the strategies of proposed algorithms. Finally, the performance of proposed algorithms do not show big difference when inventory-shortage cost ratio are 1:50 and 1:100. It is first observed an improvement when the ratio changes from 1:9 to 1:50, since the shortage cost is too low that no maintenance and production would be the optimal plan. The improvement, however, does not appear

doi:10.6342/NTU202403120

significant as shortage cost increases after the ratio reaches 1:50. This may suggest a possible way to improve the algorithm, since the control of yield rate should be stricter as shortage cost increases.

Factor	Level	GAP_{OPT}		GAP_{MHA}	
		YRTHA	CCHA	YRTHA	CCHA
Demand-capacity ratio	2:1	18.21%	23.79%	8.06%	13.14%
	3:1	15.96%	19.34%	9.50%	12.69%
Demand distribution	even	9.77%	15.43%	3.59%	8.88%
	uneven	24.39%	27.69%	13.97%	16.96%
Inventory-shortage cost ratio	1:50	16.98%	21.28%	9.26%	13.25%
	1:100	17.18%	21.84%	8.30%	12.58%
Bottleneck stage capacity	$0.5A_j^H$	15.28%	18.00%	9.45%	12.04%
	$0.75A_j^H$	16.99%	21.38%	9.41%	13.52%
	A_j^H	18.08%	25.30%	7.48%	13.19%

Table 5.2: Numerical result of SPSS

Similar results are obtained under environment multiple products, as shown in Table 5.3. Since there exists no big difference of costs and demand between products in our experiment settings, the proposed algorithms yield similar performance as the number of products increases. However, the gaps decrease under even demand distribution, and increase under uneven demand distribution. This further indicates the efficiency of our algorithm planning under even demand distribution.

Table 5.4 and 5.5 show the results of multi-stage environments, with single and mul-

doi:10.6342/NTU202403120

Factor	Level	GAP_{OPT}		GAP_{MHA}	
		YRTHA	CCHA	YRTHA	CCHA
Demand-capacity ratio	2:1	18.26%	23.89%	8.36%	13.45%
	3:1	16.00%	19.40%	9.95%	13.16%
Demand distribution	even	9.87%	15.59%	1.7%	6.92%
	uneven	24.39%	27.70%	16.61%	19.68%
Inventory-shortage cost ratio	1:50	17.03%	21.37%	9.26%	13.25%
	1:100	17.23%	21.93%	9.05%	13.35%
Bottleneck stage capacity	$0.5A_j^H$	15.32%	18.06%	10.42%	13.03%
	$0.75A_j^H$	17.04%	21.46%	9.42%	13.53%
	A_j^H	19.04%	25.42%	7.63%	13.34%

Table 5.3: Numerical result of MPSS

doi:10.6342/NTU202403120

tiple products. While the result remains similar when number of products increases, $\text{GAP}_{textMHA}$ increases under all scenarios when stages increase, indicating the advantage of benchmark 1 increases given the increased problem complexity. CCHA suffers slightly higher gap increase compared to YRTHA. While YRTHA yields better results, they are returned after a search enumerating several possibilities, which requires extra knowledge and settings. CCHA, on the other hand, has the advantage of requiring no extra settings. The performance may be improved by designing other cost functions considering more properties of problem instances. On the other hand, decrease of $\text{GAP}_{textOPT}$ is observed as stages increase. Since we set only 90 seconds to solve MINLPs, solutions with lower cost may not be obtained in time. The proposed algorithms may obtain solution with small gaps compared to MHA in reasonable time.

Factor	Level	GAP_{OPT}		GAP_{MHA}	
		YRTHA	CCHA	YRTHA	CCHA
Demand-capacity ratio	2:1	24.67%	31.75%	7.44%	13.46%
	3:1	19.80%	24.07%	10.12%	14.03%
Demand distribution	even	15.30%	21.68%	2.93%	8.51%
	uneven	29.17%	34.14%	14.62%	18.98%
Inventory-shortage cost ratio	1:50	21.84%	27.15%	9.05%	13.73%
	1:100	22.63%	28.67%	8.51%	13.76%
Bottleneck stage capacity	$0.5A_j^H$	17.67%	21.67%	10.20%	13.94%
	$0.75A_j^H$	21.18%	26.79%	8.46%	13.45%
	A_j^H	27.85%	35.27%	7.67%	13.85%

Table 5.4: Numerical result of SPMS

doi:10.6342/NTU202403120

Factor	Level	GAP_{OPT}		GAP_{MHA}	
		YRTHA	CCHA	YRTHA	CCHA
Demand-capacity ratio	2:1	24.69%	31.75%	6.30%	12.24%
	3:1	19.79%	24.08%	8.44%	12.30%
Demand distribution	even	15.31%	21.68%	1.27%	6.74%
	uneven	29.16%	34.16%	13.47%	17.80%
Inventory-shortage cost ratio	1:50	21.83%	27.10%	6.64%	11.17%
	1:100	22.64%	28.74%	8.10%	13.36%
Bottleneck stage capacity	$0.5A_j^H$	17.61%	21.66%	9.52%	13.28%
	$0.75A_j^H$	21.23%	26.88%	8.20%	13.22%
	A_j^H	27.87%	35.21%	4.38%	10.31%

Table 5.5: Numerical result of MPMS

doi:10.6342/NTU202403120

Chapter 6

Conclusion

In this study, we consider an integrated production planning and preventive maintenance planning problem under a flow shop system. We define the problem by formulating a dynamic programming model. We formulate an MINLP model for the determinsitc version of our problem, then prove it to be NP-hard. Since directly solving the dynamic programming model is difficult, we propose heuristic algorithms with steps of solving MINLP. Due to the fact that solving MINLP is time consuming, we propose two other algorithms. The idea of setting yield rate threshold and comparing costs are used to determine preventive maintenance plan, then production plan may be obtained by solving LP models. Through numerical studies, we show that the proposed algorithm yields near-optimal solutions under different scenarios.

There are some possible directions to extend this study. The inventory cost may be generalized to convex function or other types of function. The convexity and optimization of the problem may require further studies. Next, the proposed algorithms may be improved by setting threshold in a more reasonable way, or developing better cost functions.

doi:10.6342/NTU202403120

Furthermore, the proposed cost function barely considers cost of different products, or the decisions in the near future. Further studies on the relationship between parameters may be required.

doi:10.6342/NTU202403120

Bibliography

Aghezzaf, E.H., M.A. Jamali, D. Ait-Kadi. 2007. An integrated production and preventive maintenance planning model. *European Journal of Operational Research* **181**(2) 679–685.

Aghezzaf, E.H., N.M. Najid. 2008. Integrated production planning and preventive maintenance in deteriorating production systems. *Information Sciences* **178**(17) 3382–3392.

Alimian, M., M Saidi-Mehrabad, A Jabbarzadeh. 2019. A robust integrated production and preventive maintenance planning model for multi-state systems with uncertain demand and common cause failures. *Journal of Manufacturing Systems* **50** 263–277.

Allaoui, H., A. Artiba. 2006. Scheduling two-stage hybrid flow shop with availability constraints. *Computers & Operations Research* **33**(5) 1399–1419.

Chu, H., K. Dong, R. Li, Q. Cheng, C. Zhang, K. Huang, C. Yang, Y. Zheng. 2022. Integrated modeling and optimization of production planning and scheduling in hybrid flow shop for order production mode. *Computers & Industrial Engineering* **174** 108741.

Ebrahimi, M., S.M.T.F. Ghomi, B. Karimi. 2014. Hybrid flow shop scheduling with

doi:10.6342/NTU202403120

sequence dependent family setup time and uncertain due dates. *Applied Mathematical Modelling* **38**(9-10) 2490–2504.

Escudero, L.F., P.V. Kamesam, A.J. King, R.J.B. Wets. 1993. Production planning via scenario modelling. *Annals of Operations Research* **43**(6) 309–335.

Florian, M., J.K. Lenstra, A.H.G. Rinnooy Kan. 1980. Deterministic production planning: Algorithms and complexity. *Management Science* **26**(7) 669–679.

Gelders, L.F., L.N. Van Wassenhove. 1981. Production planning: a review. *European Journal of Operational Research* **7**(2) 101–110.

Johnson, S. M. 1954. Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly* **1**(1) 61–68.

Khatab, A., C. Diallo, E.H. Aghezzaf, U. Venkatadri. 2019. Integrated production quality and condition-based maintenance optimisation for a stochastically deteriorating manufacturing system. *International Journal of Production Research* **57**(8) 2480–2497.

Yalaoui, A., K. Chaabi, F. Yalaoui. 2014. Integrated production planning and preventive maintenance in deteriorating production systems. *Information Sciences* **278** 841–861.

doi:10.6342/NTU202403120