國立臺灣大學管理學院資訊管理學系
碩士論文
Department of Information Management
College of Management
National Taiwan University
Master's Thesis

流水線環境中考慮內生性良率的生產與預防性保養之整合規劃問題

An integrated production-maintenance problem in a flow shop with endogenous yield rates

陳廷旭
Ting-Hsu Chen

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

中華民國 114 年 2 月
February, 2025

doi:10.6342/NTU202500607

謝辭

能夠完成這本碩士論文，最要感謝指導教授孔令傑老師。在撰寫論文的這一年中，無論老師有多忙碌，總是都會在每週撥出時間給予我一對一的指導，不僅讓我學習如何進行嚴謹的研究，也培養我獨立思考的能力；指導論文之餘，老師平時親切的關心，以及待人處事的智慧，更是讓我在學術之外獲益良多。也要感謝口試委員林妙聰老師與藍俊宏老師，兩位老師提出的寶貴建議，使我瞭解論文的不足之處並加以修正，讓這本論文更加完整且更具品質。

除此以外，感謝臺大資管系致力於營造良好的學習環境，不僅開設多面向的課程，讓我深入瞭解資訊管理的價值，更時常邀請國內外知名學者演講，讓我能有機會站在研究前緣、拓展學術視野，如果沒有系上完善的課程規劃與師長們的悉心栽培，便不會有今天的成果。

在研究所的兩年時光中，最常互動的就是實驗室的夥伴金梁、柄瑞、文新、盈穎、予瑄、琳瑄和元婷，和大家一起學習，互相支持鼓勵，無疑是一件幸福的事情。學長姊們的經驗傳承，以及學弟妹們的交流與陪伴，都讓我的碩士求學時光更加充實。

最後，謝謝我親愛的父母一路以來的支持，讓我可以無憂無慮地完成學業，也讓我擁有自信去迎接挑戰。期許自己在未來長遠的路上，能夠謹記師長的教誨與家人的支持，穩健踏實地朝夢想前進。

陳廷旭 謹識
于臺大資訊管理學研究所
民國一百一十四年二月

doi:10.6342/NTU202500607

摘要

本研究探討流水線環境中單產品的生產與預防性保養之整合規劃問題。當工廠機台未進行保養時，機況會隨時間下降，導致生產成本上升；另一方面，儘管保養有助於改善機況，但是過於頻繁的保養將增加機台停機時間，進而提升存貨成本或缺貨成本。因此，如何決定適當的保養時機與對應的生產計畫，以最小化總成本，便成為工廠能否在成本與效益間取得平衡的關鍵議題。

本研究透過一個非線性整數規劃模型來精確描述此問題。由於在問題規模龐大時，求解該數學模型需要耗費大量時間，因此本研究提出數種演算法，以在多項式時間內求得最佳解或近似解。

當生產環境僅包含單一階段時，本研究首先證明保養週期與存貨週期呈現巢狀結構，並利用此特性，進一步根據保養時機將規劃時程分割為若干個子問題，最後透過線性規劃求解子問題，以及將整體問題轉化為最短路徑問題，來求得最佳解。當生產環境包含兩個階段時，本研究證明兩個階段的保養時機遵循同步或相鄰的模式，並延伸單階段演算法，設計一套具雙層最短路徑結構的精確演算法來求解。

當生產環境超過兩個階段時，本研究設計一套啟發式演算法，先將問題依階段分解，再透過單階段演算法逐步求解，以獲得近似解，同時設計數值實驗來驗證其品質與效能。最後，本研究展示如何將此啟發式演算法擴展至良率遞減率為隨機的情境，並同樣透過數值實驗說明其表現。

關鍵詞：生產規劃、預防性保養、非線性整數規劃、精確演算法、啟發式演算法

doi:10.6342/NTU202500607

Abstract

In this study, we investigate a single-product flow shop joint production-maintenance planning problem in a deteriorating system. The yield rates of machines decline over time but can be increased through preventive maintenance. However, machines must be shut down during maintenance periods, which may lead to higher inventory costs or demand shortage costs. Determining the appropriate timing for maintenance and the associated production plan to minimize the total cost is therefore a crucial issue for the factory to remain cost-effective. When the system consists of only one stage, we show that the maintenance and inventory cycles are nested. This property allows us to decompose the planning horizon into subproblems by maintenance and reformulate our problem as a shortest-path problem, where the edge costs can be solved by linear programming. When the system consists of only two stages, we show that the maintenance timing across the two stages follows either a synchronized or a neighboring pattern. An exact algorithm which exhibits a two-layer shortest-path structure is then developed to find an optimal solution. For problems with more than two stages, we develop a heuristic algorithm that decomposes the problem by stage and utilizes the single-stage algorithm to generate a near-optimal solution. Finally, we illustrate how our heuristic algorithm can be extended to problems with stochastic yield declining rates by using a rolling schedule approach. Through numerical experiments, we demonstrate the effectiveness of our heuristic algorithm under both deterministic and stochastic settings.

Keywords: *production planning; preventive maintenance; nonlinear integer program; exact algorithms; heuristic algorithms.*

doi:10.6342/NTU202500607

Contents

謝辭 **i**

摘要 **ii**

Abstract **iii**

List of Figures **vii**

List of Tables **viii**

1 Introduction **1**

1.1 Background and motivation 1

1.2 Research objectives 3

1.3 Research plan 5

2 Literature Review **6**

2.1 Lot-sizing problems 6

2.2 Preventive maintenance with random failure 8

doi:10.6342/NTU202500607

2.3 Preventive maintenance with declining yield rates 9

2.4 Preventive maintenance with multiple stages 10

3 Problem Description and Formulation 12

4 Algorithms 18

4.1 An exact algorithm for the single-stage problem 20

4.2 An exact algorithm for the two-stage problem 27

4.2.1 Properties of the multi-stage problem 27

4.2.2 Solving the two-stage problem 29

4.2.3 Solving the subproblem ($Q_{i\alpha j\beta}$) and complexity analysis 43

4.3 A heuristic algorithm for the multi-stage problem 52

4.4 Numerical study 54

4.4.1 Experiment settings 55

4.4.2 Benchmark 55

4.4.3 Solution performance 57

5 Stochastic Yield Declining Rates 59

5.1 Problem description and algorithms 59

5.2 Numerical study 60

6 Conclusion 63

doi:10.6342/NTU202500607

Bibliography **65**

doi:10.6342/NTU202500607

List of Figures

3.1 Visualization of the decision variables 13

4.1 Schedules (x, z) and (x', z') in the proof of Theorem 2 22

4.2 Decomposing the planning horizon for the single-stage problem 23

4.3 An example of the shortest-path reformulation for solving the single-stage problem 26

4.4 Schedules (x, z) and (x', z') in the proof of Theorem 5 when $l < k$ 32

4.5 An example of maintenance schedule for the two-stage problem 32

4.6 Decomposing the planning horizon for the two-stage problem 37

4.7 Visualization of all valid subproblems 38

4.8 An example of the shortest path reformulation for solving the two-stage problem 44

4.9 Decomposing T_{27} for the subproblem $(Q_{2,S,7,N_+})$ 45

4.10 Flow chart for phase 1 of the heuristic algorithm 53

doi:10.6342/NTU202500607

List of Tables

3.1 List of notations 17

4.1 Conditions on i and j for a valid subproblem 39

4.2 Computation of $c(i, \alpha, j, \beta)$ 50

4.3 Experiment setting of scenarios 56

4.4 Optimality gaps of the algorithms 58

5.1 Experiment setting of scenarios 61

5.2 Optimality gaps of the algorithms 62

doi:10.6342/NTU202500607

Chapter 1

Introduction

1.1 Background and motivation

Production planning and preventive maintenance are two major concerns of a manufacturing company. Production planning problems, also known as lot-sizing problems, focus on developing a production plan that meets customer demands while balancing production, inventory, and demand-related costs (e.g., shortage and backlogging costs). Most studies on production planning assume that production systems operate flawlessly and that manufacturing equipment does not deteriorate. Nevertheless, in reality, machines degrade over time, and this degradation may result in lower yield rates and a higher probability of machine breakdowns. As a common strategy for tackling these issues, manufacturers implement preventive maintenance to raise yield rates and reduce the risk of machine breakdowns. Based on machinery conditions and system status, the planning of preventive maintenance seeks to determine the best maintenance schedule that maximizes system reliability.

doi:10.6342/NTU202500607

Interestingly, while extensive research has been conducted on production planning and preventive maintenance, they are typically treated as separate problems in the literature. However, these decisions are inherently interdependent: though maintenance improves machine health and enhances productivity, it also requires temporary machine shutdowns, which may lead to excessive storage of work-in-process (WIP) or even demand shortages. The best timing for maintenance therefore depends not only on machine status but also on WIP levels and demand quantities, and the optimization of production planning and maintenance scheduling should be carried out together to further minimize costs or maximize profits.

Another key issue when developing production and maintenance plans is to understand the manufacturing process. In many cases, it is required to go through several processing stages in sequence to produce an end product; that is, the production process follows a flow shop structure. This poses additional challenges for manufacturers as they have to coordinate maintenance timing and manage the inventory of work-in-processes (WIPs) across multiple stages. For example, in a two-stage system, when the second stage is under maintenance, the first stage may reduce production to avoid high inventory cost as these WIPs cannot be processed by the second stage immediately. Therefore, to develop an effective production plan for the first stage, it is necessary to consider not only its maintenance schedule but also that of the second stage. These considerations further complicate the problem and require the decision maker to determine the production plan and maintenance schedule from multiple perspectives.

To the best of our knowledge, while some literature is devoted to integrating production and maintenance decisions in a single-stage setting, there is almost no research that

doi:10.6342/NTU202500607

considers a multi-stage setting (see Chapter 2 for details). We therefore attempt to extend the research scope in this area and believe that the integrated production-maintenance problem in a deteriorating multi-stage system is worthwhile to study.

1.2 Research objectives

In this study, we investigate a single-product joint production-maintenance planning problem in a deteriorating flow shop system. We are given a planning horizon that consists of several periods. To produce an end product, it is required to go through several sequential processing stages, and we assume that it is viable to finish all stages consecutively within a single planning period (Zangwill, 1969; Love, 1972; Hwang et al., 2013; Zhao and Zhang, 2020). The demand quantity in each period is known and deterministic and is given before planning begins.

At the beginning of a period, the decision maker determines (1) the input quantity and (2) whether to perform maintenance at each stage. Note that since we want to find the overall production plan for each stage, we view all the machines in the same stage as a whole, or simply say that there is only one machine in a stage. Maintenance takes exactly one period to complete, and production is not allowed during this maintenance period.For each stage, the realized output quantity is assumed to be the input level multiplied by the yield rate at that stage, and the defective items are discarded. Moreover, we view the outputs of each stage (except for the last stage) as work-in-progresses (WIPs), and they constitute the input of the next stage.

At the end of a period, machine yield rates decrease (until they reach a lower bound)

doi:10.6342/NTU202500607

according to an exogenous yield declining rate if maintenance is not performed in that period. On the other hand, if maintenance is conducted, the yield rates are restored to as-new ones. Note that while some studies model system deterioration as an increasing probability of random breakdowns (Aghezzaf et al., 2007; Nourelfath et al., 2010; Aghezzaf et al., 2016), here we follow another stream of literature and models the deterioration as a decline in yield rates (Sloan and Shanthikumar, 2000; Xiang et al., 2014; Zhang et al., 2023). The company is allowed to strategically reject the demand requests but pay a shortage cost to compensate customers. Our goal is to develop a joint production-maintenance schedule that minimizes the sum of production, inventory, and demand shortage costs over the entire planning horizon.

To address our problem, we first formulate it as a nonlinear integer program. However, the high computational burden of solving this model drives us to look for more efficient algorithms. When the system consists of one or two stages, we derive theoretical properties of an optimal solution and develop exact algorithms to solve the problem. When the system consists of more than two stages, we utilize the single-stage algorithm and develop a heuristic algorithm to find a near-optimal solution efficiently. We also demonstrate how our heuristic algorithm can be extended to the case when the yield declining rates are stochastic. Numerical experiments are conducted to show the average performance and the efficiency of our proposed algorithms.

doi:10.6342/NTU202500607

1.3 Research plan

Our research plan is organized as follows. In Chapter 2, we review related studies. In Chapter 3, we precisely describe our problem and formulate it as a nonlinear integer program. To approach our problem, we develop both exact and heuristic algorithms in Chapter 4. A numerical study that demonstrates the effectiveness of the heuristic algorithm is also provided in the same chapter. In Chapter 5, we extend our problem to the scenario when the yield declining rates are random and explain how our heuristic algorithm can be modified accordingly. Finally, Chapter 6 concludes our study.

doi:10.6342/NTU202500607

Chapter 2

Literature Review

In this chapter, we review related studies on production planning and preventive maintenance problems. In Section 2.1, we review classic works on lot-sizing problems. From Sections 2.2 to 2.4, we review research that integrates maintenance decisions into production planning problems. Section 2.2 presents studies that model system deterioration as an increasing probability of random failure. On the other hand, Section 2.3 investigates studies that characterize deterioration using multiple states to describe machine conditions. Finally, Section 2.4 focuses on research that considers a multi-stage deteriorating manufacturing system.

2.1 Lot-sizing problems

Lot-sizing problems have been studied extensively in the literature (Brahimi et al., 2006; Buschkuehl et al., 2010; Brahimi et al., 2017). In the seminal work done by Wagner and Whitin (1958), a single-stage production system with dynamic (period-dependent

doi:10.6342/NTU202500607

and non-constant) demand is investigated. They first show that production occurs only when the inventory is empty. Building on this property, they demonstrate how the problem can be divided into subproblems and develop an exact algorithm based on dynamic programming to find an optimal solution.

To better handle lot-sizing problems, Zangwill (1968) proposes a novel network formulation that facilitates the development of theoretical properties and efficient algorithms. Using this technique, Zangwill (1969) extends the work of Wagner and Whitin (1958) in two directions. First, he demonstrates how the problem and algorithm can be modified to incorporate demand backlogging. Moreover, he generalizes the framework to accommodate systems with multiple processing stages (also referred to as multi-echelon systems) and develops an exact algorithm to solve the problem.

Unlike the above works which assume unlimited production capacity, capacitated lot-sizing problems have also received considerable attention in the literature. Florian and Klein (1971) study a single-stage production system with a constant capacity across all planning periods. Building on the theoretical results and solution approaches in the uncapacitated problem, they utilize dynamic programming to solve the capacitated one. For multi-stage capacitated lot-sizing, Hwang et al. (2013) first introduce the novel concept of a "basis path" to characterize an optimal solution. Following this idea, they develop a sophisticated algorithm that finds an optimal solution in polynomial time.

While the above works develop elegant solution approaches to tackle single- and multi-stage lot-sizing problems, all of them assume a perfect production system where machine yield rates remain constant over the planning horizon. This distinguishes our study from theirs.

doi:10.6342/NTU202500607

2.2 Preventive maintenance with random failure

For studies that consider production planning in a deteriorating system, Aghezzaf et al. (2007) propose an integrated model that incorporates both production and maintenance decisions. Both unplanned corrective maintenance and planned preventive maintenance can be conducted to restore machine conditions and reduce the probability of machine breakdowns. Through a numerical study, they demonstrate how much cost is saved when production and maintenance decisions are made simultaneously rather than considered separately.

Nourelfath et al. (2010) adopt a similar setting to Aghezzaf et al. (2007) to model the joint production and maintenance problem but consider multiple machines (referred to as components) within the same stage. After formulating the problem, they propose a sophisticated approach to evaluate the expected maintenance cost, the expected maintenance duration, and the average production capacity in each period. In addition, they develop a genetic algorithm that searches for an effective maintenance schedule to solve large-scale instances.

In contrast to the above studies which assume that maintenance restores machines to an as-good-as-new state, some studies consider imperfect preventive maintenance. For example, Aghezzaf et al. (2016) consider a scenario where maintenance restores machines to a state between as-bad-as-old and as-good-as-new. To address the problem, they demonstrate how it can be formulated by a mixed-integer program and propose an iterative heuristic algorithm to find near-optimal solutions.

doi:10.6342/NTU202500607

Although the above studies are devoted to integrating production planning and preventive maintenance, their modeling choices, which involve an increasing probability of machine breakdown as the time from the last maintenance increases, are different from our work. In our setting, we assume that machine yield rates decrease over time and do not consider random breakdowns. Therefore, their modeling techniques and solution approaches cannot be applied to our problem.

2.3 Preventive maintenance with declining yield rates

For research that incorporates declining machine yield rates, Sloan and Shanthikumar (2000) and Xiang et al. (2014) both consider joint production and maintenance planning problems in deteriorating systems, with deterministic demand in the former and stochastic demand in the latter. A Markov decision process is used to describe the state (yield rate) of machines, which directly impacts system productivity. The optimal policies for both problems are presented in their respective studies.

Extending these works, Zhang et al. (2023) study a problem where production output is not only determined by yield rates but is also a random variable. Their problem is even more complicated as the transition probabilities between different machine states are no longer homogeneous but depend on the workload of the previous period. A stochastic dynamic programming model is proposed to formulate the problem, and the structure of the optimal policy is characterized in their work.

Aldurgam (2020) investigates a problem that considers maintenance, production, and inspection at the same time. Besides preventive maintenance, the decision maker deter-

doi:10.6342/NTU202500607

mine whether to inspect some products and repair the defective ones when necessary. Balancing the costs of inspection and product repair against the compensation costs incurred from selling defective items thus become the trade-off of the problem. Theoretical properties are derived to gain insights to the problem.

While the modeling choices for describing system deterioration in the above works are similar to ours, they focus on single-stage systems rather than multi-stage ones. The differentiates our work from all the above.

2.4 Preventive maintenance with multiple stages

For studies that integrate production planning and preventive maintenance in a multi-stage system, Shao et al. (2022) investigate a problem that involves two deteriorating stages in a just-in-time system. For each stage, the decision maker determines the production quantity per period and the interval between two consecutive maintenance to minimize the cost. Though their study considers multiple stages, they model system deterioration as an increasing risk of machine breakdown, which differs from this study.

Guo and Gu (2020) examines a two-stage, multi-product assembly system with delayed differentiation configuration. In this system, the first stage performs the common operations required by all products, while the second stage consists of multiple machines, each responsible for producing a specific product type. In each period, a product type is selected for manufacturing and the corresponding machine at the second stage is activated, and the decision maker determines whether to perform preventive maintenance on the other machines in the second stage. Nevertheless, given that the two stages must

doi:10.6342/NTU202500607

operate in tandem in the same period, their problem is essentially equivalent to that of a single-stage system with multiple machines. On the contrary, we consider a true multi-stage system in this study.

To the best of our knowledge, there is only a limited number of studies that consider a multi-stage system with both maintenance and production decisions. We therefore believe our problem is worth of investigation.

doi:10.6342/NTU202500607

Chapter 3

Problem Description and Formulation

In this chapter, we precisely describe our production-maintenance problem and formulate it as a nonlinear integer program.

Let $S = \{1, ..., m\}$ be the set of stages and $m = |S|$ be the number of stages. Moreover, we assume that the planning horizon consists of n periods and let $T = \{1, 2, ..., n\}$ be the set of periods. In the sequel, we use $s \in S$ as the index of a stage and $t \in T$ as the index of a period.

Regarding our decision variables, we define x_{st} as the input quantity to stage s in period t, and y_{st} as the amount of inventory of the work-in-process that has been processed through s stages at the beginning of period t. We include $y_{s,n+1}$ to represent the inventory level at stage s at the end of the planning horizon. For simplicity, we sometimes abbreviate the "work-in-process that has been processed through s stages" as the "WIP at stage s",

doi:10.6342/NTU202500607

and we assume that the ending inventory of a period equals the beginning inventory of the next period. For maintenance decisions, we define $z_{st} \in \{0,1\}$ as 1 if stage s is under maintenance in period t or 0 otherwise, and $w_{st} \in [0,1]$ as the yield rate of stage s in period t. Figure 3.1 provides an example with 2 stages and 4 periods that visualizes the above decision variables.

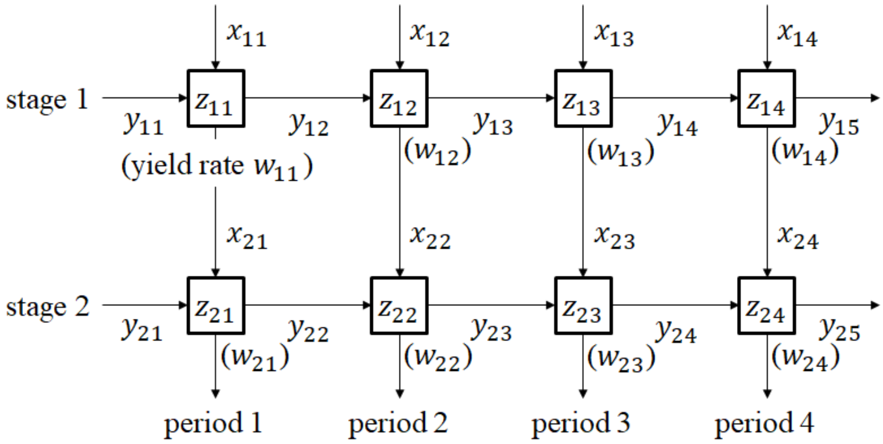

Figure 3.1: Visualization of the decision variables

In period t, there is a known and deterministic demand D_t, and we let the decision variable r_t be the demand fulfilled in this period. The company pays a shortage cost F to compensate for each unit of unfulfilled demand, and demands are lost if not met on time. We use P_s and R_s to denote the production cost per input to stage s and the inventory cost per WIP at stage s, respectively. With the above notations, the total cost that the company seeks to minimize is then

$$\min \quad \sum_{s\in S}\sum_{t\in T}\left(P_s x_{st} + R_s y_{st}\right) + F\sum_{t\in T}\left(D_t - r_t\right), \tag{3.1}$$

which is the sum of the production cost, inventory cost, and demand shortage cost. Moreover, we assume that $F > \sum_{s\in S} P_s$ to exclude the trivial case where all demand

doi:10.6342/NTU202500607

requests are rejected in an optimal solution.

It is evident that there are some relationships between our decision variables. To be more specific, the inventory level should depend on the input quantity and the yield rate, and no production is allowed during a maintenance period. Let M be a large enough number, we then formulate the inventory balancing and production-maintenance constraints

$$y_{s,t+1} = y_{st} + w_{st}x_{st} - x_{s+1,t} \quad \forall s \in S \setminus \{m\}, t \in T \tag{3.2}$$

$$y_{m,t+1} = y_{m,t} + w_{m,t}x_{m,t} - r_t \quad \forall t \in T \tag{3.3}$$

$$x_{st} \leq M(1 - z_{st}) \quad \forall s \in S, t \in T. \tag{3.4}$$

Constraints (3.2) and (3.3) balance the inventory at each stage in each period. Since both w_{st} and x_{st} are continuous variables, their multiplication makes our formulation nonlinear. Constraint (3.4) stops production when maintenance takes place.

To precisely describe our deteriorating production system, we assume that the yield rate at stage s drops by B_{st} at the end of period t when there is no maintenance, and it continues to decrease from period to period until it reaches a lower bound L_s. On the contrary, the yield rate is restored to 100% upon the completion of maintenance. To make sure the yield rate stays above the lower bound, we further introduce a binary variable $v_{st} \in \{0, 1\}$, which is 1 if $w_{st} - B_s < L_s$ or 0 otherwise. The constraints that formulate the change in yield rates according to the maintenance decision are then

$$w_{st} \leq 1 \quad \forall s \in S, t \in T \tag{3.5}$$

doi:10.6342/NTU202500607

$$w_{s,t+1} \leq Mz_{st} + Mv_{st} + (w_{st} - B_{st}) \quad \forall s \in S, t \in T \setminus \{n\} \tag{3.6}$$

$$w_{s,t+1} \leq Mz_{st} + M(1 - v_{st}) + L_s \quad \forall s \in S, t \in T \setminus \{n\}. \tag{3.7}$$

Constraints (3.5) to (3.7) collectively ensure that maintenance can restore the yield rate to 100%. In addition, when no maintenance is performed ($z_{st} = 0$), the binary variable v_{st} activates exactly one of constraints (3.6) and (3.7), and it sets the yield rate $w_{s,t+1}$ to the maximum of $w_{st} - B_{st}$ (drops by B_{st}) and L_s (reaches the lower bound).

Finally, let $\overline{W}_s \geq L_s$ be the initial yield rate of stage s. The formulation is then completed by adding the following constraints

$$w_{s,1} = \overline{W}_s \quad \forall s \in S \tag{3.8}$$

$$y_{s,1} = y_{s,n+1} = 0 \quad \forall s \in S \tag{3.9}$$

$$r_t \leq D_t \quad \forall t \in T \tag{3.10}$$

$$x_{st}, y_{st}, w_{st}, r_t \geq 0 \quad \forall s \in S, t \in T \tag{3.11}$$

$$z_{st}, v_{st} \in \{0, 1\} \quad \forall s \in S, t \in T. \tag{3.12}$$

Constraint (3.8) initializes the yield rate. Constraint (3.9) sets the inventory at both the beginning and the end of the planning horizon to zero (Zangwill, 1969; Florian and Klein, 1971; Hwang et al., 2013). Constraint (3.10) sets the upper bound of the fulfilled demand. Constraints (3.11) and (3.12) are nonnegativity and binary constraints.

When $m > 1$, we refer to our problem as the *multi-stage problem*. For the special cases when $m = 1$ or $m = 2$, we refer to the problems as the *single-stage problem* and the *two-stage problem*, respectively. Moreover, we define $x = [x_{st}]^{m \times n}$, $z = [z_{st}]^{m \times n}$, $y =$

doi:10.6342/NTU202500607

$[y_{st}]^{m\times(n+1)}$, and $w = [w_{st}]^{m\times n}$ as the *production plan*, *maintenance schedule*, inventory level, and yield rate for the entire production system, respectively. Since y and w can be computed once x and z are given, we use solely (x, z) to represent a joint production-maintenance schedule. The notation $f(x, z)$ is adopted to denote the associated objective value of schedule (x, z).

All the notations mentioned above are summarized in Table 3.1.

doi:10.6342/NTU202500607

Sets	
S	Set of stages; $S = \{1, ..., m\}$.
T	Set of periods; $T = \{1, ..., n\}$.
Parameters	
m	Number of stages.
n	Number of periods.
D_t	Demand in period t.
P_s	Production cost per input to stage s.
R_s	Inventory cost per WIP at stage s.
F	Shortage cost per end product.
B_{st}	Yield declining rate of stage s in period t.
L_s	Lower bound of yield rate of stage s.
$\overline{W}_s$	Initial yield rate of stage s.
Decision variables	
x_{st}	Input quantity to stage s in period t.
y_{st}	Amount of inventory at stage s at the beginning of period t.
r_t	Demand fulfilled in period t.
w_{st}	Yield rate of stage s in period t.
z_{st}	1 if stage s is under maintenance in period t or 0 otherwise.
v_{st}	1 if $w_{st} - B_{st} < L_s$ or 0 otherwise.

Table 3.1: List of notations

doi:10.6342/NTU202500607

Chapter 4

Algorithms

In this chapter, we develop algorithms to approach our production-maintenance problem. In Sections 4.1 and 4.2, we show that the problem is polynomial-time solvable when the system consists of one or two stages, respectively, and present our exact algorithms for these cases. In Section 4.3, we propose a heuristic algorithm to address the problem with an arbitrary number of stages. The effectiveness of the heuristic algorithm is evaluated through a numerical study in Section 4.4.

To keep the article concise and avoid overly technical details when proving the theoretical properties, we assume that the production cost, the inventory cost, the yield declining rate, the lower bound of the yield rate, and the demand are all strictly positive throughout this chapter. All our theorems and algorithms can be easily extended to the case where these values are nonnegative.

Assumption 1. *$P_{st}, R_{st}, B_{st}, L_s, D_t > 0$ for all $s \in S$ and $t \in T$.*

Before discussing how to solve our production-maintenance problem, we first focus

doi:10.6342/NTU202500607

on a category of schedule that better captures the relationship between our two major decisions, production and maintenance. As machines are restored after maintenance, it is natural that maintenance should be followed by production; otherwise, the maintenance would either be wasted or better rescheduled. To formalize this idea, we define an *effective schedule* in Definition 1, which captures the relationship between maintenance and production described above.

Definition 1. *A feasible schedule (x, z) is said to be* effective *if $z_{st} = 1$ implies $x_{s,t+1} > 0$ for all $s \in S$ and $t \in T \setminus \{n\}$, and $z_{s,n} = 0$ for all $s \in S$.*

ollowing Definition 1, a natural question is that whether we can limit our attention to those effective schedules when searching for an optimal schedule. Since in an optimal schedule that is not effective, there exists "redundant" maintenance that is not immediately followed by production in the next period. Such maintenance can be removed without affecting the optimality of the schedule. This observation is formally confirmed by Theorem 1.

Theorem 1. *There exists an optimal schedule that is effective.*

Proof. Suppose that in an optimal schedule (x, z), there exists an index $k \in T \setminus \{n\}$ such that $z_{sk} = 1$ and $x_{s,k+1} = 0$ for some $s \in S$. We want to show that setting $z_{sk} = 0$ preserves optimality.

If $x_{st} = 0$ for all $t = k+1, ..., n$, our claim obviously holds. Now assume that there exists another index $l > k+1$ such that $x_{sl} > 0$ and $x_{st} = 0$ for all $t = k+1, ..., l-1$. Because (x, z) is optimal and $x_{s,l-1} = 0$, we must have $z_{s,l-1} = 1$, which immediately implies that the maintenance in period k is redundant. Since the objective value remains

doi:10.6342/NTU202500607

the same after setting $z_{sk} = 0$, we can repeat this process to eliminate all redundant maintenance and eventually obtain an optimal schedule that is effective. □

With effective schedules, we can now better handle the structure of an optimal schedule from at least two perspectives. First, it is clear by definition that consecutive maintenance is not allowed as production should be activated after maintenance. This eliminates many schedules to consider when searching for an optimal one. Second, maintenance and production are more closely connected since maintenance always implies positive production in the next period. This facilitates the development of our algorithms when deciding when to perform maintenance and when to initiate production. Finally, since there is always an optimal schedule that is effective, in the remainder of this article we will focus on effective schedules only.

4.1 An exact algorithm for the single-stage problem

In this section, we present an exact algorithm that solves our single-stage problem. Since the problem here involves only one stage, we omit the stage index s from all notations for simplicity.

To find an optimal solution, it is important to first understand the relationship between the decision variables. Recall that we have characterized the relationship between production and maintenance through an effective schedule. We now examine how inventory, another key decision variable in our problem, relates to maintenance.

Since maintenance raises the machine's yield rate to 100%, it suggests that one should not carry inventory across a maintenance period: producing immediately after the ma-

doi:10.6342/NTU202500607

chine is restored saves both production costs (as the yield rate is 100%) and inventory costs (as no inventory is held over the period). This intuition is formally validated by Theorem 2, which turns out to play a pivotal role in the development of our exact algorithm.

Theorem 2. *There exists an effective optimal schedule* (x, z) *satisfying that* $z_t = 1$ *implies* $y_{t+1} = 0$ *for all* $t \in T \setminus \{n\}$.

Proof. Assume for contradiction that in an effective optimal schedule (x, z), there exists an index $k \in T \setminus \{n\}$ such that $z_k = 1$ and $y_{k+1} > 0$. Let $i < k$ denote the only period index satisfying $x_i > 0$ and $x_t = 0$ for all $t = i + 1, ..., k$, and let $w_i \in [0, 1]$ be the yield rate in period i. Clearly, we have $y_{i+1} \geq y_{i+2} \geq \cdots \geq y_k \geq y_{k+1} > 0$, and we let $\epsilon = \min\{w_i x_i, y_{k+1}\} > 0$. Now consider another schedule (x', z') defined as

$$x'_t = \begin{cases} x_t - \frac{\epsilon}{w_i} & \text{if } t = i \\ x_t + \epsilon & \text{if } t = k + 1 \\ x_t & \text{otherwise} \end{cases}$$

and $z'_t = z_t$ for all $t \in T$. We then have the corresponding inventory level

$$y'_t = \begin{cases} y_t - \epsilon & \text{if } i + 1 \leq t \leq k + 1 \\ y_t & \text{otherwise} \end{cases}$$

according to the inventory balancing constraint. Note that (x', z') is feasible since (1) the value of ϵ ensures that both x'_t and y'_t are nonnegative for all $t \in T$ and (2) (x, z) is

doi:10.6342/NTU202500607

effective and $z_k = 1$ together implies $z_{k+1} = 0$, which further ensures that it is valid to set x'_{k+1} to a positive value. Nevertheless, since

$$f(x', z') - f(x, z) = \epsilon P\left(1 - \frac{1}{w_i}\right) - \epsilon R(k - i + 1) < 0,$$

it follows that (x', z') is a better solution, which contradicts the optimality of (x, z). (Figure 4.1 illustrates the proof by visualizing schedules (x, z) and (x', z') and highlighting the differences of inventory levels.) □

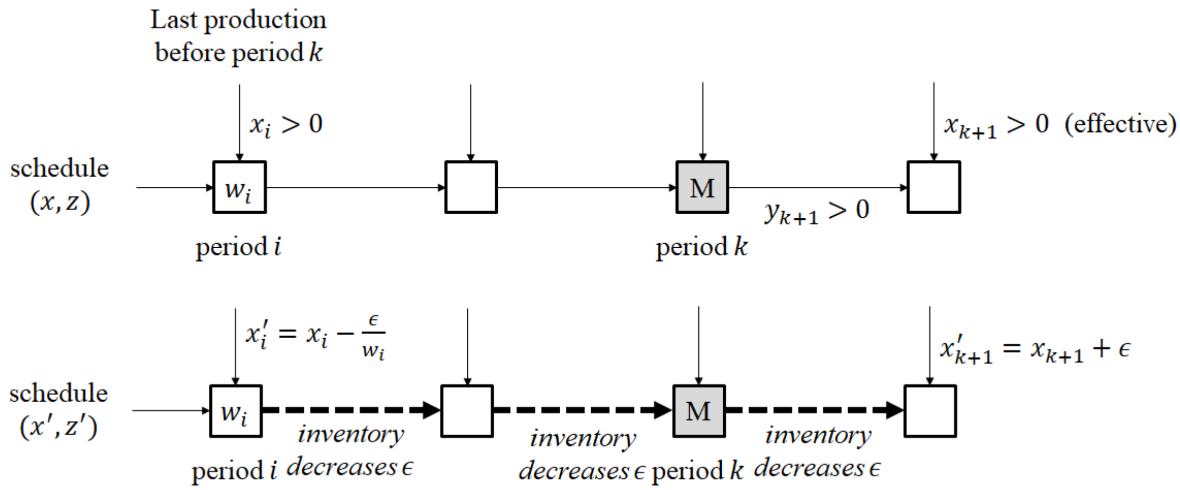

Figure 4.1: Schedules (x, z) and (x', z') in the proof of Theorem 2

While it is clear from Theorem 2 that maintenance always leads to zero inventory, the theorem does not guarantee that the reverse is always true. This means the inventory level may return to zero multiple times between two consecutive maintenance, which in turn suggests that the maintenance and inventory cycles are nested, with the maintenance cycle being longer than the inventory cycle.

Our algorithm utilizes this nested-cycle property and decomposes the entire planning horizon into several *subproblems* by maintenance. Each subproblem consists of several

doi:10.6342/NTU202500607

consecutive periods, with maintenance scheduled only in the last period except for the last subproblem (as no maintenance is conducted at the end of the planning horizon). The goal of a subproblem is to determine the optimal production plan (the input quantity for each period considered by the subproblem) that minimizes the total cost within that subproblem. Moreover, since each subproblem ends with maintenance, there should no inventory carried over between consecutive subproblems. This implies the subproblems are independent of one another and allows us to solve each subproblem separately and combine their solutions to construct an optimal schedule for the whole problem.

Figure 4.2 illustrates the decomposition idea with a 7-period example. Suppose that the optimal solution schedules maintenance in periods 2 and 5. This divides the planning horizon into three subproblems: the first one contains periods 1 and 2, the second one contains periods 3, 4, and 5, and the third one contains periods 6 and 7. Moreover, since the inventory level returns to zero after maintenance, we have $y_3 = y_6 = 0$ by Theorem 2. Together with the boundary condition $y_1 = y_8 = 0$, it follows that the beginning and ending inventory of each subproblem are both zero, which ensures the independence of subproblems.

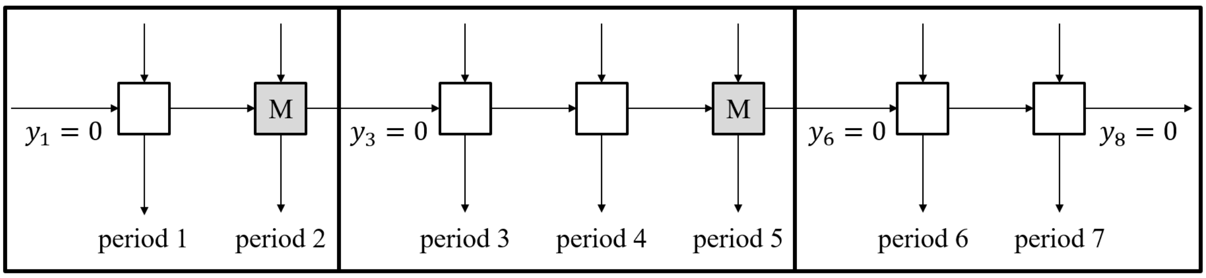

Figure 4.2: Decomposing the planning horizon for the single-stage problem

While the decomposition idea is intuitive, we have to address two key questions to develop it into an algorithm: (1) how to compute the minimum cost for each subproblem,

doi:10.6342/NTU202500607

and (2) how to optimally decompose the whole problem so that the total cost over the entire planning horizon is minimized.

To answer the first question, we begin by precisely formulating a subproblem. We use (Q_{ij}) to denote the subproblem that covers the planning horizon from period $i+1$ to j $(i < j)$, and let $T_{ij} = \{i+1, ..., j\}$ be the corresponding set of periods. Since the maintenance decisions are fixed within a subproblem (scheduled in the last period of the subproblem), the yield rate for each period can be computed based on the yield declining rates and is no longer an endogenous variable. In particular, for a given (Q_{ij}), we define

$$W_t^{(i)} = \begin{cases} \max\left\{\overline{W} - \sum_{t'=i+1}^{t-1} B_{t'}, L\right\} & \text{if } i = 0 \\ \max\left\{100\% - \sum_{t'=i+1}^{t-1} B_{t'}, L\right\} & \text{if } i \neq 0 \end{cases}$$

as the *realized* yield rate in period $t \in T_{ij}$. Note that $W_t^{(i)}$ depends on (Q_{ij}) only through i and is independent of j.

With the realized yield rate, an optimal solution to (Q_{ij}) and the corresponding minimum cost $c(i,j)$ can then be obtained by solving the following linear program

$$c(i,j) = \min \quad \sum_{t \in T_{ij}} \left(Px_t + Ry_t\right) + F \sum_{t \in T_{ij}} \left(D_t - r_t\right) \tag{4.1}$$

$$\text{s.t.} \quad x_j \leq M_j \tag{4.2}$$

$$y_{i+1} = y_{j+1} = 0 \tag{4.3}$$

$$y_{t+1} = y_t + W_t^{(i)} x_t - r_t \quad \forall t \in T_{ij} \tag{4.4}$$

$$x_t, y_t \geq 0 \quad \forall t \in T_{ij} \tag{4.5}$$

doi:10.6342/NTU202500607

$$0 \leq r_t \leq D_t \quad \forall t \in T_{ij}, \tag{4.6}$$

where $M_j = 0$ for all $j \neq n$ and $M_n = \infty$. Constraint (4.2) ensures that when $j \neq n$, no production occurs in the last period of the subproblem as maintenance is performed in that period. Constraint (4.3) ensures that both the beginning and ending inventory of the subproblem are zero. Constraint (4.4) balances the inventory based on the realized yield rate $W_t^{(i)}$.

With the minimum cost of each subproblem, we now reformulate our problem as an equivalent shortest-path problem to find the optimal way to decompose the entire planning horizon. To construct the graph, we first create $n+1$ nodes labeled from 0 to n, where the tth node corresponds to period $t \in T$ and the 0th node serves as an artificial starting node. A directed edge from node i to j, denoted as (i, j), exists if and only if $i < j$. This construction produces a directed graph that is acyclic and contains $\binom{n+1}{2}$ edges. Moreover, when a path from node 0 to n passes through one or more intermediate nodes $t \notin \{0, n\}$, it indicates that maintenance is scheduled in each period t. Therefore, every path from node 0 to n corresponds to a maintenance schedule, and conversely, every maintenance schedule is represented by a path.

To complete the graph, we assign each edge an appropriate cost so that the shortest path from node 0 to n corresponds to the optimal maintenance schedule. Since we only consider effective schedules, for edges $(j-1, j)$ where $2 \leq j \leq n-1$, i.e., maintenance is scheduled in consecutive periods, we set their costs to infinity to prevent the shortest path from visiting them. For all other edges (i, j), we set their costs as $c(i, j)$. With this setup, the cost of the shortest path is then the smallest total cost over the entire planning

doi:10.6342/NTU202500607

horizon. One can then use any shortest-path algorithm to find the shortest path and the corresponding optimal maintenance schedule.

We provide a 4-period example in Figure 4.3 to better illustrate how our problem is reformulated as a shortest-path problem. Since there are four periods, we construct a graph with $4 + 1 = 5$ nodes and $\binom{5}{2} = 10$ edges. The cost of edge (i, j) is given by $c(i, j)$ except for edges (1, 2) and (2, 3), whose costs are set to infinity. Suppose that the shortest path from node 0 to 4 is $0 \to 1 \to 3 \to 4$. This indicates that the minimum cost of the corresponding optimal solution is $c(0, 1) + c(1, 3) + c(3, 4)$. Furthermore, since the shortest path visits intermediate nodes 1 and 3, it suggests that periods 1 and 3 are under maintenance in the optimal solution. Finally, because the entire planning horizon is divided into three subproblems (Q_{01}), (Q_{13}), and (Q_{34}), we can obtain the optimal production plans by solving the corresponding linear programs.

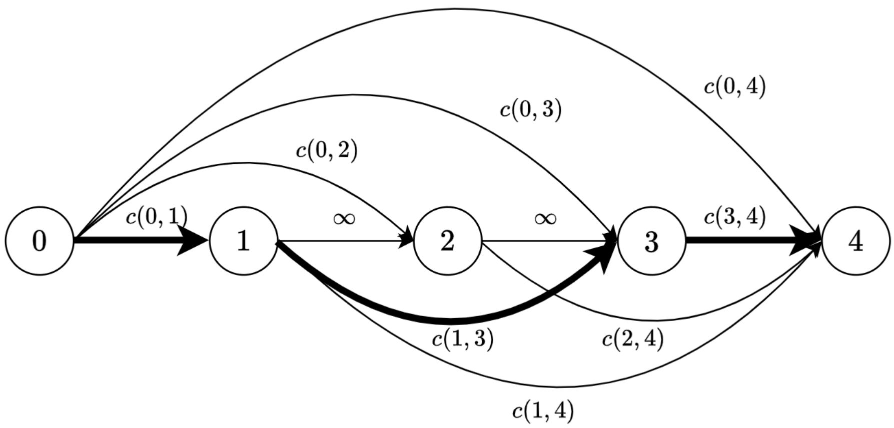

Figure 4.3: An example of the shortest-path reformulation for solving the single-stage problem

Regarding time complexity, note that there are two major steps in our algorithm: computing the edge costs and solving the shortest path. Suppose that the linear programming solver takes a polynomial time of $\mathcal{O}(L_n)$ to solve a linear program with $\mathcal{O}(n)$

doi:10.6342/NTU202500607

variables and $\mathcal{O}(n)$ constraints. We then need $\mathcal{O}(n^2 L_n)$ time to compute all edge costs as there are $\mathcal{O}(n^2)$ edges in the graph. Together with the fact that solving the shortest path on an acyclic graph with $n+1$ nodes requires $\mathcal{O}(n^2)$ time, the overall time complexity of our algorithm is $\mathcal{O}(n^2 L_n)$.

4.2 An exact algorithm for the two-stage problem

In this section, we extend our findings from the single-stage problem and develop an exact algorithm for the two-stage problem. To develop our algorithm, we derive theoretical properties that are general to the multi-stage problem in Subsection 4.2.1. Subsection 4.2.2 then focuses on the two-stage problem and presents the framework of our algorithm. Finally, Subsection 4.2.3 provides the remaining details of our algorithm.

4.2.1 Properties of the multi-stage problem

Since our production-maintenance problem is an extension of the traditional lot-sizing problem (which does not involve maintenance decisions), we follow the idea in the literature and define an *extreme schedule* in Definition 2. Theorem 3 then verifies that a critical property of an optimal solution to the lot-sizing problem also holds for our problem: production occurs only if the inventory is empty.

Definition 2. *A feasible schedule (x, z) is said to be* extreme *if $x_{st} y_{st} = 0$ for all $s \in S$ and $t \in T$.*

Theorem 3. *There exists an optimal schedule that is extreme.*

doi:10.6342/NTU202500607

Proof. For our production-maintenance problem, note that once the maintenance schedule is given, the yield rate in each period at each stage becomes known and fixed, and the problem is simplified to a production planning problem. As indicated by Zangwill (1968), for any maintenance schedule z, there exists an optimal production plan $x(z)$ to the resulting production planning problem such that $(x(z), z)$ is extreme. Let z^* denote the optimal maintenance schedule for our problem. It then follows that $(x(z^*), z^*)$ is an extreme optimal schedule. □

Beyond understanding the connection between production and inventory, it is also crucial to study whether the relationship between inventory and maintenance in the single-stage problem can be extended to the multi-stage problem. Recall that in the single-stage problem, the inventory level always returns to zero upon the completion of maintenance. Fortunately, Theorem 4 confirms that this property is also true for the multi-stage problem.

Theorem 4. *There exists an extreme optimal schedule (x, z) that is effective. Moreover, in such a schedule, if $z_{st} = 1$, then $y_{s,t+1} = 0$ for all $s \in S$ and $t \in T \setminus \{n\}$.*

Proof. For any extreme optimal schedule that is not effective, we can repeat the proof of Theorem 1 to remove all redundant maintenance and obtain an effective optimal schedule. Moreover, such schedule remains extreme since we do not modify x_{st} and y_{st} in the proof. This shows the existence of an extreme effective optimal schedule (x, z). Finally, since (x, z) is effective, we know that $z_{st} = 1$ implies $x_{s,t+1} > 0$, which further leads to $y_{s,t+1} = 0$ according to Theorem 3 as (x, z) is extreme. □

While Theorem 4 helps us understand when the inventory level returns to zero, in the

doi:10.6342/NTU202500607

multi-stage problem, there is no guarantee that all stages will be under maintenance in the same period. This means the inventory levels of different stages may not return to zero simultaneously and thus prevents us from directly extending the decomposition idea in the single-stage algorithm.

To address this issue, our plan is to investigate the possible combinations of maintenance timing across all stages. Nevertheless, when there are more than two stages, the analysis becomes challenging due to the extremely large number of possible combinations. Therefore, we begin with a more tractable problem and focus on the two-stage case in the remainder of this section.

4.2.2 Solving the two-stage problem

Interestingly, for the two-stage problem, when the second stage is under maintenance and stops production, it is reasonable for the first stage to stop production as well. This is because if the first stage continues to produce around the periods when the second stage is under maintenance, the WIP at stage 1 would accumulate (as it cannot be processed by stage 2) and incur additional inventory costs. This suggests that the first stage should stop its production and also schedule its maintenance around the maintenance period of the second stage. Theorem 5 formally characterizes the possible *maintenance patterns* across the two stages.

Theorem 5. *For the two-stage problem, there exists an effective extreme optimal schedule* (x, z) *such that if* $z_{2,t} = 1$*, then either* $z_{1,t} = 1$ *or* $z_{1,t+1} = 1$ *for all* $t \in T \setminus \{n\}$*.*

Proof. Assume for contradiction that in an effective extreme optimal schedule (x, z),

doi:10.6342/NTU202500607

there exists an index $k \in T \setminus \{n\}$ such that $z_{2,k} = 1$ and $z_{1,k} = z_{1,k+1} = 0$. Since (x, z) is effective, we have $x_{2,k+1} > 0$, which further implies that $w_{1,k+1}x_{1,k+1} + y_{1,k+1} \geq x_{2,k+1} > 0$ according to the inventory balancing constraint. Moreover, since (x, z) is extreme, it follows that exactly one of $x_{1,k+1}$ and $y_{1,k+1}$ is strictly positive. Below we show that both cases lead to a contradiction.

For the former case ($x_{1,k+1} > 0$ and $y_{1,k+1} = 0$), we have

$$w_{1,k}x_{1,k} \leq x_{2,k} + y_{1,k+1} = 0,$$

and therefore $x_{1,k} = 0$. It is then obvious that setting $z_{1,k} = 1$ (which restores the yield rate in period $k + 1$ at stage 1 to 100%) is a better solution, which contradicts to the optimality of (x, z).

For the latter case ($y_{1,k+1} > 0$ and $x_{1,k+1} = 0$), let $l \leq k$ be the only index satisfying $x_{1,l} > 0$ and $x_{1,t} = 0$ for all $t = l + 1, ..., k$. Moreover, let

$$\epsilon = \begin{cases} x_{2,k+1} & \text{if } l < k \\ w_{1,k}x_{1,k} & \text{if } l = k \end{cases}$$

be a positive number. Now consider another schedule (x', z') defined as

$$x'_{st} = \begin{cases} x_{st} - \frac{\epsilon}{w_{1,l}} & \text{if } (s,t) = (1,l) \\ x_{st} + \epsilon & \text{if } (s,t) = (1,k+1) \\ x_{st} & \text{otherwise} \end{cases}$$

doi:10.6342/NTU202500607

and

$$z'_{st} = \begin{cases} 1 & \text{if } (s,t) = (1,k) \\ z_{st} & \text{otherwise} \end{cases}.$$

We then have the corresponding inventory level

$$y'_{st} = \begin{cases} y_{st} - \epsilon & \text{if } s = 1 \text{ and } l+1 \leq t \leq k+1 \\ y_{st} & \text{otherwise} \end{cases},$$

and the yield rate of stage 1 in period $k+1$ is restored to $w'_{1,k+1} = 100\%$. Note that (x', z') is feasible since (1) the value of ϵ ensures that both x'_t and y'_t are nonnegative for all $t \in T$ and (2) setting $z'_{1,k} = 1$ is feasible since $x'_{1,k} = 0$. Nevertheless, since

$$f(x', z') - f(x, z) \leq \epsilon \left[P_1 \left(1 - \frac{1}{w_{1,l}} \right) - R_1(k - l + 1) \right] < 0,$$

it follows that (x', z') is a better solution, which contradicts the optimality of (x, z). (Figure 4.4 illustrates the proof for the case when $l < k$ by visualizing schedules (x, z) and (x', z') and highlighting the differences of inventory levels.) $\square$

With Theorem 5, we categorize the patterns of maintenance across the two stages as follows. For the case when $z_{2,t} = z_{1,t} = 1$, we say that the maintenance pattern in period t is *synchronized*. For the case when $z_{2,t-1} = z_{1,t} = 1$, we say that the maintenance pattern in period t is *neighboring*. It is important to note that Theorem 5 does not eliminate the case when stage 1 is under maintenance, but stage 2 is not scheduled for maintenance in any neighboring period.

doi:10.6342/NTU202500607

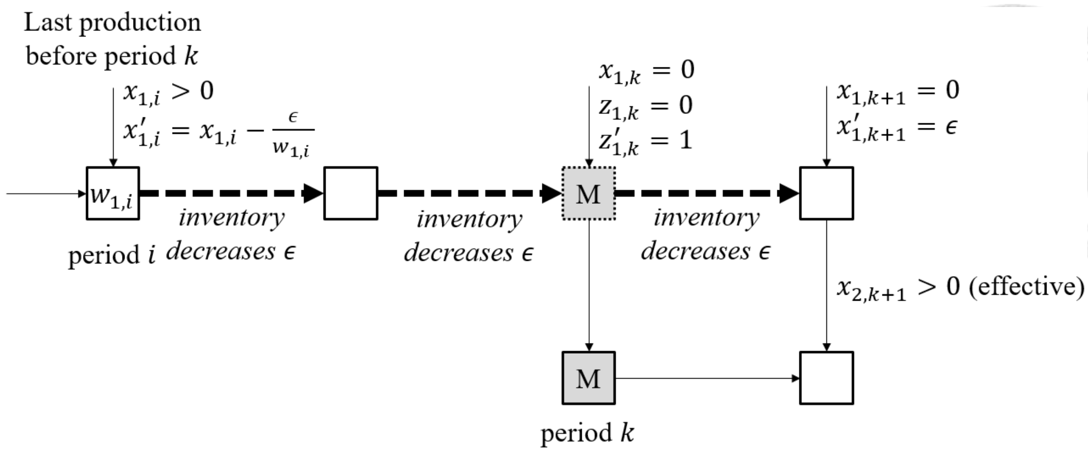

Figure 4.4: Schedules (x, z) and (x', z') in the proof of Theorem 5 when $l < k$

To illustrate the types of maintenance patterns, we consider an example maintenance schedule with 10 periods in Figure 4.5. We can see that the maintenance pattern in period 2 is synchronized, and the patterns in periods 7 and 9 are neighboring. Note that in an optimal schedule, it is possible to have multiple maintenance patterns consecutively concatenated like periods 6 to 9 in the example. Finally, stage 1 is under maintenance in period 4 alone, with no maintenance at stage 2 near this period.

Periods	1	2	3	4	5	6	7	8	9	10
Stage 1		M		M			M		M	
Stage 2		M				M		M		

Figure 4.5: An example of maintenance schedule for the two-stage problem

With the maintenance patterns defined above, our next step is to understand how the pattern is connected to the inventory level. When the maintenance pattern in period t is synchronized, according to Theorem 4, the beginning inventory levels of both stages in period $t + 1$ are both zero. On the other hand, when the maintenance pattern in period t is neighboring, while the beginning inventory level of stage 1 in period $t + 1$ is zero, whether that of stage 2 in period $t + 1$ is also zero requires further analysis. Theorem 6 indicates that it has at most two possible values.

Theorem 6. *For the two-stage problem, there exists an effective extreme optimal schedule* (x, z) *such that if* $(z_{1,t}, z_{2,t}) = (1, 0)$ *for some* $t \in T \setminus \{n\}$*, then*

1. *If* $t = n-1$ *and* $t = n-2$*, we have* $y_{2,t+1} = 0$*;*

2. *If* $t \leq n-3$*, we have either* $y_{2,t+1} = 0$ *or* $y_{2,t+1} = D_{t+1}$*. Moreover, when* $z_{2,t+1} = 0$*, we have* $y_{2,t+1} = 0$*.*

Proof. First, we analyze the case when $t = n-1$ and consider an effective extreme optimal schedule (x, z) such that $(z_{1,n-1}, z_{2,n-1}) = (1, 0)$. According to the inventory balancing and the boundary condition $y_{1,n+1} = 0$, we have

$$x_{2,n} = x_{1,n} + y_{1,n} - y_{1,n+1} = x_{1,n} + y_{1,n} \geq x_{1,n},$$

which is positive since (x, z) is effective. It then follows that $y_{2,n} = 0$ as (x, z) is extreme.

Next, we analyze the case when $t = n-2$ and consider an effective extreme optimal schedule (x, z) such that $(z_{1,n-2}, z_{2,n-2}) = (1, 0)$. Assume for contradiction that $y_{2,n-1} > 0$. Since (x, z) is effective and extreme, we have $x_{1,n-1} > 0$, $x_{2,n-1} = 0$, and $z_{1,n} = 0$. Consider another schedule (x', z'), which postpones the maintenance at stage 1 from period $n-2$ to $n-1$, defined as

$$x'_{st} = \begin{cases} 0 & \text{if } (s,t) = (1, n-1) \\ x_{st} + x_{1,n-1} & \text{if } (s,t) = (1, n) \\ x_{st} & \text{otherwise} \end{cases}$$

doi:10.6342/NTU202500607

and

$$z'_{st} = \begin{cases} 0 & \text{if } (s,t) = (1, n-2) \\ 1 & \text{if } (s,t) = (1, n-1) \\ z_{st} & \text{otherwise} \end{cases} \cdot$$

We then have the corresponding inventory level

$$y'_{st} = \begin{cases} y_{st} - x_{1,n-1} & \text{if } (s,t) = (1,n) \\ y_{st} & \text{otherwise} \end{cases} .$$

Note that (x', z') is feasible as $x'_{1,k} = 0$ allows $z'_{1,k} = 1$. Nevertheless, since

$$f(x', z') - f(x, z) = -R_1 x_{1,n-1} < 0,$$

it follows that (x', z') is a better solution than (x, z).

Next, we analyze the case when $t \leq n-3$, where $y_{2,t+1}$ can take on two possible values. Assume for contradiction that in an effective extreme optimal schedule (x, z), there exists an index $k \in T \setminus \{n\}$ such that $(z_{1,k}, z_{2,k}) = (1, 0)$ and $y_{2,k+1} \notin \{0, D_{k+1}\}$. Below we split our discussion into two cases based on the value of $y_{2,t+1}$.

When $y_{2,k+1} > D_{k+1}$, since $y_{2,k+2} \geq y_{2,k+1} - D_{k+1} > 0$ and (x, z) is extreme, we have $x_{2,k+1} = x_{2,k+2} = 0$. We first show that it is impossible to have $z_{1,k+2} = 1$. In this case, the inventory balancing constraint at stage 1 in period $k+2$ suggests that

$$y_{1,k+2} \leq x_{2,k+2} + y_{1,k+3} = 0,$$

doi:10.6342/NTU202500607

where the last equality holds as $z_{1,k+2} = 1$ implies $y_{1,k+3} = 0$. However, this contradicts the inventory balancing constraint at stage 1 in period $k+1$ since

$$y_{1,k+2} = w_{1,k+1}x_{1,k+1} - x_{2,k+1} - y_{1,k+1} = w_{1,k+1}x_{1,k+1} > 0,$$

where the last equality holds as $z_{1,k} = 1$ implies $y_{1,k+1} = 0$, and the inequality holds as $z_{1,k} = 1$ implies $x_{1,k+1} > 0$. Therefore, we must have $z_{1,k+2} = 0$. Nevertheless, in this case, we can follow a similar proof above by postponing the maintenance at stage 1 from period k to $k+1$ and generating a solution better than (x, z). This shows that the assumption $y_{2,k+1} > D_{k+1}$ leads to a contradiction.

When $0 < y_{2,k+1} < D_{k+1}$, we have $x_{2,k+1} = 0$ since (x, z) is extreme, and therefore $y_{2,k+1} + w_{2,k+1}x_{2,k+1} < D_{k+1}$. This implies the demand in period $k+1$ is partially satisfied. Nevertheless, it is no more costly to increase (respectively, decrease) the production until all the demand in period $k+1$ is either fully satisfied (respectively, fully rejected). This leads to another optimal solution such that $y_{2,k+1} \in \{0, D_{k+1}\}$.

Finally, we prove that $z_{2,k+1} = 0$ leads to $y_{2,k+1} = 0$, or equivalently, $y_{2,k+1} = D_{k+1}$ leads to $z_{2,k+1} = 1$. Since $z_{1,k} = 1$ implies $y_{1,k+1} = 0$ and $y_{2,k+1} > 0$ implies $x_{2,k+1} = 0$, we have

$$w_{1,k+1}x_{1,k+1} = x_{2,k+1} + y_{1,k+2} - y_{1,k+1} = y_{1,k+2} > 0,$$

which further suggests that $x_{1,k+2} = 0$ as (x, z) is extreme. Moreover, according the inventory balancing constraint at stage 1 in period $k+2$, we have

$$x_{2,k+2} + y_{1,k+3} \geq y_{1,k+2} > 0.$$

doi:10.6342/NTU202500607

If $x_{2,k+2} = 0$, we can follow a similar proof above by postponing the maintenance at stage 1 from period k to $k + 1$ and generating a solution better than (x, z). This suggests that $x_{2,k+2} > 0$. Since $x_{2,k+1} = 0$, it immediately follows that it is always true to have $z_{2,k+1} = 1$ in an optimal solution. □

Building on the theoretical properties of the two-stage problem, we now develop an exact algorithm to find an optimal solution. The algorithm extends the decomposition approach in the single-stage algorithm but now divides the planning horizon into subproblems based on maintenance patterns. To be more specific, we split the planning horizon at period t if and only if the maintenance pattern in that period is either synchronized ($z_{1,t} = z_{2,t} = 1$) or neighboring ($z_{1,t} = z_{2,t-1} = 1$). This strategy ensures that each subproblem includes exactly one stage 2 maintenance (except for the last subproblem), and therefore stage 2 maintenance is fixed within each subproblem.

However, recall that Theorem 5 does not rule out the possibility of conducting maintenance at stage 1 without neighboring maintenance at stage 2. Therefore, it is possible that stage 1 is under maintenance in multiple periods within a subproblem. This gives a subproblem two objectives: (1) to determine the optimal maintenance schedule for stage 1, and (2) to determine the optimal production plans for both stages so that the total cost is minimized over the subproblem.

We use the maintenance schedule depicted in Figure 4.6 as an example to explain how the planning horizon is decomposed. Since the maintenance patterns in periods 2, 7, and 9 are either synchronized or neighboring, we use these periods to divide the planning horizon into four subproblems: the first one contains periods 1 and 2, the second one

doi:10.6342/NTU202500607

contains periods 3 to 7, the third one contains periods 8 and 9, and the final subproblem consists of period 10 alone. Note that we do not split the planning horizon at period 4 since the maintenance pattern in that period is neither synchronized nor neighboring, even if stage 1 is under maintenance in that period. Instead, the stage 1 maintenance in period 4 should be identified when solving the second subproblem.

Periods	1	2	3	4	5	6	7	8	9	10
Stage 1		M		M			M		M	
Stage 2		M				M		M		

Figure 4.6: Decomposing the planning horizon for the two-stage problem

Under this decomposition method, each subproblem in the two-stage problem is characterized by the maintenance and inventory status in its last period. While we know that the maintenance pattern in the last period of a subproblem, say period t, is either synchronized or neighboring, in the neighboring case ($z_{1,t} = z_{2,t-1} = 1$ and $z_{2,t} = 0$), Theorem 6 further indicates that the inventory level $y_{2,t+1}$ will either be 0 or equal to $D_{t+1} > 0$. Based on this, we classify the *ending state* of a subproblem into three types: *synchronized*, *neighboring with zero inventory*, and *neighboring with positive inventory*, which are denoted as S, N_0, and N_+, respectively. For convenience, we also refer to the *beginning state* of a subproblem as the ending state of its preceding subproblem.

It is not surprising that the minimum cost of a subproblem depends on both its beginning state and ending state. We use ($Q_{i\alpha j\beta}$) to denote the subproblem that covers the planning horizon from period $i+1$ to j ($i < j$) with beginning state α and ending state β, where $\alpha, \beta \in \{S, N_0, N_+, \emptyset\}$. The notation $\alpha = \emptyset$ (respectively, $\beta = \emptyset$) is adopted to denote the boundary cases when $i = 0$ (respectively, $j = n$), where the subproblem

doi:10.6342/NTU202500607

includes the first (respectively, last) period of the entire planning horizon.

It is important to note that not all pairs of (i, α, j, β) form a valid subproblem. This is because certain combinations would result in schedules that are not effective. For example, $(Q_{1,S,3,N_0})$ is not a valid subproblem since stage 2 is under maintenance in both periods 1 and 2, which obviously makes the schedule not effective. To ensure that a subproblem corresponds to an effective schedule, we impose conditions on i and j for each possible combination α and β to define a *valid* subproblem in Table 4.1. Figure 4.7 further visualizes the structures of subproblems for all 14 valid combinations of α and β.

$(\alpha, \beta) = (S, S)$

Periods	…	i	i+1	…	…	j	…
Stage 1		M				M	
Stage 2		M				M	

$(\alpha, \beta) = (S, N_0)$ or (S, N_+)

Periods	…	i	i+1	…	…	j	…
Stage 1		M				M	
Stage 2		M			M		

$(\alpha, \beta) = (N_0, S)$

Periods	…	i	i+1	…	…	j	…
Stage 1		M				M	
Stage 2	M					M	

$(\alpha, \beta) = (N_0, N_0)$ or (N_0, N_+)

Periods	…	i	i+1	…	…	j	…
Stage 1		M				M	
Stage 2	M				M		

$(\alpha, \beta) = (N_+, N_0)$ or (N_+, N_+)

Periods	…	i		j	…
Stage 1		M		M	
Stage 2	M		M		

$(\alpha, \beta) = (\emptyset, S)$

Periods	i=1	…	…	j	…
Stage 1				M	
Stage 2				M	

$(\alpha, \beta) = (\emptyset, N_0)$ or $(\emptyset, N_+)$

Periods	i=1	…	…	j	…
Stage 1				M	
Stage 2			M		

$(\alpha, \beta) = (\emptyset, \emptyset)$

Periods	i=1	…	…	…	j=n
Stage 1					
Stage 2					

$(\alpha, \beta) = (S, \emptyset)$

Periods	…	i	…	…	j=n
Stage 1		M			
Stage 2		M			

$(\alpha, \beta) = (N_0, \emptyset)$

Periods	…	i	…	…	j=n
Stage 1		M			
Stage 2	M				

Figure 4.7: Visualization of all valid subproblems

All the conditions in the table can be checked easily except for the most special case when $\alpha = N_+$. In this case, we must have $\beta \in \{N_0, N_+\}$ and $j = i+2$. To see the reason,

doi:10.6342/NTU202500607

α	β	Conditions on i	Conditions on j	Conditions on i and j
$\emptyset$	$\emptyset$	$i = 0$	$j = n$	–
$\emptyset$	S	$i = 0$	$1 \leq j \leq n-1$	–
$\emptyset$	N_0	$i = 0$	$2 \leq j \leq n-1$	–
$\emptyset$	N_+	$i = 0$	$2 \leq j \leq n-3$	–
S	$\emptyset$	$1 \leq i \leq n-1$	$j = n$	–
S	S	$1 \leq i \leq n-3$	$3 \leq j \leq n-1$	$j \geq i+2$
S	N_0	$1 \leq i \leq n-4$	$4 \leq j \leq n-1$	$j \geq i+3$
S	N_+	$1 \leq i \leq n-6$	$4 \leq j \leq n-3$	$j \geq i+3$
N_0	$\emptyset$	$2 \leq i \leq n-1$	$j = n$	–
N_0	S	$2 \leq i \leq n-3$	$4 \leq j \leq n-1$	$j \geq i+2$
N_0	N_0	$2 \leq i \leq n-3$	$4 \leq j \leq n-1$	$j \geq i+2$
N_0	N_+	$2 \leq i \leq n-5$	$4 \leq j \leq n-3$	$j \geq i+2$
N_+	N_0	$2 \leq i \leq n-3$	$4 \leq j \leq n-1$	$j = i+2$
N_+	N_+	$2 \leq i \leq n-5$	$4 \leq j \leq n-3$	$j = i+2$

Table 4.1: Conditions on i and j for a valid subproblem

note that $\alpha = N_+$ implies $z_{1,i} = z_{2,i-1} = 1$ and $y_{2,i+1} = D_{i+1} > 0$, which further implies $z_{2,i+1} = 1$ by Theorem 6. This rules out the possibility of $(\alpha, \beta) = (N_+, \emptyset)$. Furthermore, according to Theorem 5, we have either $z_{1,i+1} = 1$ or $z_{1,i+2} = 1$ since $z_{2,i+1} = 1$. The former case ($z_{1,i+1} = 1$) cannot hold as it would lead to consecutive maintenance at stage 1, and therefore it is impossible to have $(\alpha, \beta) = (N_+, S)$. In the latter case ($z_{1,i+2} = 1$), it follows that the maintenance pattern in period $i+2$ is neighboring, which marks the

doi:10.6342/NTU202500607

end of a subproblem, and we thus derive the condition $j = i + 2$ when $\alpha = N_+$.

For a valid subproblem $(Q_{i\alpha j\beta})$, we now precisely formulated it as a mathematical program. Let $T_{ij} = \{i+1, ..., j\}$ be the set of periods covered by $(Q_{i\alpha j\beta})$. Based on the values of i and α (i.e., the beginning period and the beginning state of the subproblem), we define

$$W_{2,t}^{(i,\alpha)} = \begin{cases} \max\left\{\overline{W}_2 - \sum_{t'=i+1}^{t-1} B_{2,t'}, L_2\right\} & \text{if } i = 0 \\ \max\left\{100\% - \sum_{t'=i+1}^{t-1} B_{2,t'}, L_2\right\} & \text{if } i \neq 0 \text{ and } \alpha = S \\ \max\left\{100\% - \sum_{t'=i}^{t-1} B_{2,t'}, L_2\right\} & \text{if } i \neq 0 \text{ and } \alpha \in \{N_0, N_+\} \end{cases}$$

as the *realized* yield rate of stage 2 in period t. Moreover, we let W' be the initial yield rate at stage 1 at the beginning of the subproblem, where $W' = \overline{W}_1$ if $i = 0$ and $W' = 100\%$ if $i > 0$. An optimal solution to $(Q_{i\alpha j\beta})$ and the minimized cost $c(i, \alpha, j, \beta)$ can then be obtained by solving the following nonlinear integer program

$$c(i,\alpha,j,\beta) = \min \quad \sum_{s\in S}\sum_{t\in T_{ij}} \left(P_s x_{st} + R_s y_{st}\right) + F \sum_{t\in T_{ij}} \left(D_t - r_t\right) \tag{4.7}$$

$$\text{s.t.} \quad x_{1,j} \leq M_\beta \tag{4.8}$$

$$A_\beta^S x_{2,j} + A_\beta^N x_{2,j-1} \leq M_\beta \tag{4.9}$$

$$y_{1,i+1} = y_{1,j+1} = 0 \tag{4.10}$$

$$y_{2,i+1} = D'_\alpha \tag{4.11}$$

$$y_{2,j+1} = D''_\beta \tag{4.12}$$

$$w_{1,i+1} = W' \tag{4.13}$$

doi:10.6342/NTU202500607

$$y_{1,t+1} = y_{1,t} + w_{1,t}x_{1,t} - x_{2,t} \quad \forall t \in T_{ij} \tag{4.14}$$

$$y_{2,t+1} = y_{2,t} + W_{2,t}^{(i,\alpha)}x_{2,t} - r_t \quad \forall t \in T_{ij} \tag{4.15}$$

$$x_{1,t} \leq M(1 - z_{1,t}) \quad \forall t \in T_{ij} \tag{4.16}$$

$$w_{1,t} \leq 1 \quad \forall t \in T_{ij} \tag{4.17}$$

$$w_{1,t+1} \leq Mz_{1,t} + Mv_{1,t} + (w_{1,t} - B_{1,t}) \quad \forall t \in T_{ij} \setminus \{j\} \tag{4.18}$$

$$w_{1,t+1} \leq Mz_{1,t} + M(1 - v_{1,t}) + L_1 \quad \forall t \in T_{ij} \setminus \{j\} \tag{4.19}$$

$$x_{st}, y_{st}, w_{st}, r_t \geq 0 \quad \forall s \in S, t \in T_{ij} \tag{4.20}$$

$$z_{1,t}, v_{1,t} \in \{0, 1\} \quad \forall t \in T_{ij}, \tag{4.21}$$

where $M_\beta = 0$ if $\beta \neq \emptyset$ and $M_\emptyset = \infty$; $A_\beta^S = 1$ if $\beta = S$ or 0 otherwise; $A_\beta^N = 1$ if $\beta \in \{N_0, N_+\}$ or 0 otherwise; $D'_\alpha = D_{i+1}$ if $\alpha = N_+$ and 0 otherwise; and finally, $D''_\beta = D_{j+1}$ if $\beta = N_+$ and 0 otherwise.

Constraint (4.8) ensures that when $\beta \neq \emptyset$, no production occurs at stage 1 in period j since maintenance is always performed in that period no matter β is S, N_0, or N_+. Similarly, constraint (4.9) ensures that when $\beta \neq \emptyset$, stage 2 production is not allowed in period j when $\beta = S$ (where the constraint reduces to $x_{2,j} \leq 0$) and in period $j-1$ when $\beta \in \{N_0, N_+\}$ (where the constraint reduces to $x_{2,j-1} \leq 0$). Constraint (4.10) ensures at stage 1, there is no inventory both at the beginning and the end of the subproblem. Constraint (4.11) (respectively, (4.12)) ensures that when $\alpha = N_+$ (respectively, $\beta = N_+$), we have D_{i+1} (respectively, D_{j+1}) end products on hand at the beginning (respectively, end) of the subproblem. Constraint (4.13) sets the initial yield rate at stage 1 at the beginning of the subproblem. The meanings of the rest of the constraints are very similar to that of

doi:10.6342/NTU202500607

the whole problem provided in Chapter 3. The only difference is that the maintenance schedule and the yield rate at stage 2 are now realized and no longer endogenous.

Though the subproblem $(Q_{i\alpha j\beta})$ can be precisely formulated as a nonlinear integer program, we still need to show that it is polynomial-time solvable. Nevertheless, for now we temporarily assume that all $c(i, \alpha, j, \beta)$ values are given and demonstrate how these values can be utilized to design our exact algorithm. We will present an algorithm to solve the subproblem in the next subsection. In the final part of this subsection, we follow the idea of shortest-path reformulation in the single-stage algorithm and adapt it for the two-stage problem to find the optimal way to decompose the entire planning horizon.

To construct the graph, we first create $3n - 1$ nodes as follows. For each period $1 \leq t < n$, we create three copies of nodes that represent the three possible ending states of a subproblem, and we use (t, ω), where $\omega \in \{S, N_0, N_+\}$, to label each node. The source node $(0, \emptyset)$ and the destination node $(n, \emptyset)$ are also added to the graph. A directed edge from node (i, α) to (j, β), denoted as (i, α, j, β), exists if and only if $i < j$. This construction produces a directed graph that is acyclic and contain $\mathcal{O}(n^2)$ edges.[1] Moreover, if the subproblem formed by (i, α) and (j, β) is valid (cf. Table 4.1), we set the edge cost of (i, α, j, β) to $c(i, \alpha, j, \beta)$; otherwise, we set the edge cost to infinity to prevent the shortest path from visiting this edge.

Under this construction, when a path from node $(0, \emptyset)$ to $(n, \emptyset)$ passes through one or more intermediate node (t, ω), it indicates that the maintenance and inventory status

[1]To count the precise number of edges, we first define internal nodes as those that are neither the source nor the destination. We then have $9\binom{n-1}{2}$ edges among internal nodes. Together with $3n - 2$ edges from the source node to an internal node, $3n - 2$ edges from an internal node to the destination node, and one direct edge from the source to the destination, there are $9\binom{n-1}{2} + 2(3n - 2) + 1$ edges in total.

doi:10.6342/NTU202500607

in period t should follow the state ω. Since every path from $(0, \emptyset)$ to $(n, \emptyset)$ represents a unique maintenance schedule, and every maintenance schedule can be represented by a unique path, the shortest path from $(0, \emptyset)$ to $(n, \emptyset)$ is then associated to the optimal maintenance schedule that minimizes the total cost.

To better illustrate our algorithm, we revisit the 10-period example provide in Figure 4.6 and provide the corresponding graph and the shortest path in Figure 4.8 (where many edges are omitted for clarity). For each period $1 \leq t \leq 9$, we create three copies of nodes to represent the three possible cases. Suppose that the shortest path from the source to the destination is $(0, \emptyset) \rightarrow (2, S) \rightarrow (7, N_+) \rightarrow (9, N_0) \rightarrow (10, \emptyset)$. This indicates that the minimum cost is $c(0, \emptyset, 2, S) + c(2, S, 7, N_+) + c(7, N_+, 9, N_0) + c(9, N_0, 10, \emptyset)$, and the maintenance patterns in periods 2, 7, and 9 are synchronized, neighboring, and neighboring in the optimal solution, respectively, as shown in Figure 4.6. Moreover, since we visit the node $(7, N_+)$, the ending inventory in period 7 should be exactly D_8 (the demand in period 8). It is important to note that while stage 2 maintenance are revealed as we travel the shortest path, there may be some independent stage 1 maintenance that are only revealed after we solve the subproblems. We will address this issue in the next subsection.

4.2.3 Solving the subproblem ($Q_{i\alpha j\beta}$) and complexity analysis

In the previous subsection, we formulated the subproblem ($Q_{i\alpha j\beta}$) as a nonlinear integer program (cf. (4.7)–(4.21)) but have not yet explained how it can be solved in polynomial time. In this subsection, we complete the description of our two-stage algorithm by presenting an algorithm that solves ($Q_{i\alpha j\beta}$). A complexity analysis of the entire two-

doi:10.6342/NTU202500607

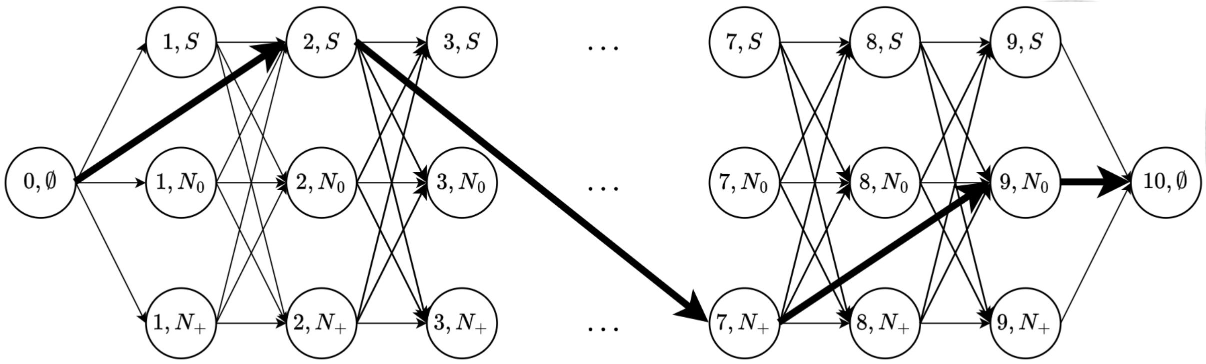

Figure 4.8: An example of the shortest path reformulation for solving the two-stage problem

stage algorithm (which incorporates the algorithm below for solving the subproblems) is provided at the end of this subsection.

Recall that in $(Q_{i\alpha j\beta})$, the maintenance schedule for stage 2 is fixed, and our goal is to determine the maintenance schedule for stage 1 and the production plans for both stages so that the total cost within the subproblem is minimized. For cases when $j \leq i+2$, since the subproblem includes at most two periods, one can solve it easily by enumerating all possible production and maintenance plans. We provide the shortcut formulas (which are derived from the enumeration) for these cases at the end of this subsection for reference. In the following, we focus on the more complex case when $j \geq i+3$.

To solve the subproblem, one can notice that it is actually a variant of the single-stage problem: we still optimize the maintenance schedule for a single stage, but now we make production decisions for two stages. This observation motivates us to adapt the decomposition strategy from the single-stage algorithm to solve the subproblem. To be more specific, we decompose the planning horizon T_{ij} into *inner subproblems* based on stage 1 maintenance.[2] For an inner subproblem, the only stage 1 maintenance is

[2]To clarify the terminology, note that a *subproblem* is created by dividing the entire planning horizon according to maintenance patterns, and each *subproblem* is further divided into *inner subproblems* based

doi:10.6342/NTU202500607

conducted in the last period, and our goal is to determine the optimal production plans for both stages that minimize the total cost within that inner subproblem.

Figure 4.9 illustrates an example of how to decompose the subproblem $(Q_{2,S,7,N_+})$ (which is the second subproblem in the example given at the end of Subsection 4.2.2). According to the ending state of the subproblem, we know that stage 1 is under maintenance in period 7 and stage 2 is under maintenance in period 6. Suppose that stage 1 is also under maintenance in period 4 in the optimal solution. This divides the planning horizon $T_{27} = \{3, 4, 5, 6, 7\}$ into two inner subproblems: the first one covers periods 3 and 4, and the second one covers periods 5, 6, and 7. For each inner subproblem, the yield rate at each stage in each period is realized and no longer endogenous, and our goal is to determine the best production plans for both stages so that the total cost is minimized.

Periods	1	2	3	4	5	6	7	8	9	10
Stage 1		M		M			M		M	
Stage 2		M				M		M		

Figure 4.9: Decomposing T_{27} for the subproblem $(Q_{2,S,7,N_+})$

Although the decomposition approach for solving a subproblem seems applicable, it is necessary to verify whether the inner subproblems are independent of one another. This requires us to check whether $y_{1,t+1}$ and $y_{2,t+1}$ are both zero upon the completion of stage 1 maintenance in period $t \in T_{ij}$ so that no inventory is carried over between consecutive inner subproblems. Unfortunately, while Theorem 4 guarantees that $y_{1,t+1}$ is always zero, Theorem 6 reveals a special case where $y_{2,t+1}$ may not be zero.

The unfortunate case occurs when $\beta \in \{N_0, N_+\}$, where we have $z_{1,j} = z_{2,j-1} = 1$. In

on stage 1 maintenance.

doi:10.6342/NTU202500607

this case, when stage 1 is also under maintenance in period $t = j-2$, i.e., $(z_{1,j-2}, z_{2,j-2}) = (1, 0)$, Theorem 6 is unable to determine whether $y_{2,j-1}$ is 0 or D_{j-1}. To address this issue, we introduce a remedy by considering two *alternative subproblems*. The alternative subproblems is almost the same as the original subproblem but only differ in the demand quantity. Note that this remedy is required only when $\beta \in \{N_0, N_+\}$; when $\beta \in \{S, \emptyset\}$, the subproblem can be divided directly without further adjustment.

To construct this remedy, we notice that in an optimal solution, the demand request in period $j-1$ is either fulfilled or rejected. This generates our two alternative subproblems: one assumes that the demand in period $j-1$ is fulfilled, and the other assumes that it is rejected. We then strategically adjust the demand in periods $j-1$ and $j-2$ to account for these cases in the following way.

In the fulfilled case, given that stage 2 is under maintenance in period $j-1$, it is necessary to prepare D_{j-1} products in period $j-2$ and store them for one period so that we can meet the demand in period $j-1$. Therefore, we treat the demand in period $j-2$ as effectively $D_{j-2} + D_{j-1}$ and that in period $j-1$ as zero, with an additional inventory cost $R_2 D_{j-1}$. Formally, we define $D_t^{(f)}$ as the adjusted demand in period $t \in T_{ij}$ for the fulfilled case, where $D_{j-1}^{(f)} = 0$, $D_{j-2}^{(f)} = D_{j-2} + D_{j-1}$, and $D_t^{(f)} = D_t$ for all $t \in T_{ij} \setminus \{j-1, j-2\}$. The cost of this alternative subproblem is then obtained by first computing the minimum cost of the subproblem with the adjusted demand $D_t^{(f)}$ and then adding the inventory cost $R_2 D_{j-1}$.

In the rejected case, we define $D_t^{(r)}$ as the adjusted demand in period $t \in T_{ij}$. Since the demand in period $j-1$ is not fulfilled, we simply set $D_{j-1}^{(r)} = 0$ and $D_t^{(r)} = D_t$ for all $t \in T_{ij} \setminus \{j-1\}$. The cost of this alternative subproblem is then the minimum cost of

doi:10.6342/NTU202500607

the subproblem with the adjusted demand $D_t^{(r)}$ plus the shortage cost FD_{j-1}.

Note that in both adjusted cases, we have $D_{j-1}^{(f)} = D_{j-1}^{(r)} = 0$, which ensures that $y_{2,j-1} = 0$ if stage 1 is under maintenance in period $j-2$. This allows us to apply the decomposition approach to solve these alternative subproblems (as there is no inventory carried over between consecutive inner subproblems). Finally, we compare the costs of the fulfilled and rejected cases and select the one with the smaller cost.

We use $D = [D_t]_{t \in T_{ij}}$, $D^{(f)} = [D_t^{(f)}]_{t \in T_{ij}}$, and $D^{(r)} = [D_t^{(r)}]_{t \in T_{ij}}$ to denote the vectors of the original demand, the adjusted demand for the fulfilled case, and the adjusted demand for the rejected case over the planning horizon T_{ij}. Since the structure of the original subproblem and both alternative subproblems is almost identical except for the demand values (the original demand D for $\beta \in \{S, \emptyset\}$, and the adjusted demand $D^{(f)}$ and $D^{(r)}$ for $\beta \in \{N_0, N_+\}$), below we provide a general approach to solve a subproblem for any given demand vector $d = [d_t]_{t \in T_{ij}}$, where $d \in \{D, D^{(f)}, D^{(r)}\}$.

To compute the minimum cost of a subproblem, we have to (1) explain how to solve an inner subproblem and (2) determine the optimal way to divide the subproblem. Given the demand vector d, we use the notation $(Q'_{i'j'|d})$ to denote the inner subproblem that covers the planning horizon from period $i'+1$ to j' ($i \leq i' < j' \leq j$). Since the maintenance schedule for stage 1 is fixed within an inner subproblem (scheduled in period j' if $j' \neq n$), we define the *realized* yield rate of stage 1 in period t as

$$W_{1,t}^{(i,\alpha,i')} = W_{1,t} = \begin{cases} \max\left\{\overline{W}_1 - \sum_{t'=i'+1}^{t-1} B_{1,t'}, L_1\right\} & \text{if } i' = 0 \\ \max\left\{100\% - \sum_{t'=i'+1}^{t-1} B_{1,t'}, L_1\right\} & \text{if } i' \neq 0 \end{cases},$$

doi:10.6342/NTU202500607

which is obviously dependent on the beginning period i of the subproblem, the beginning state α of the subproblem, and the beginning period i' of the inner subproblem. Together with stage 2 realized yield rate $W_{2,t}$, we can obtain an optimal solution to $(Q'_{i'j'|d})$ and compute the minimum cost $c'(i', j' \mid d)$ by solving the linear program[3]

$$
\begin{aligned}
c'(i', j' \mid d) = \min \quad & \sum_{s \in S} \sum_{t \in T_{i'j'}} \big(P_s x_{st} + R_s y_{st}\big) + F \sum_{t \in T_{i'j'}} \big(d_t - r_t\big) && (4.22) \\
\text{s.t.} \quad & (4.8)\text{–}(4.12) \\
& x_{1,j'} \leq M_{j'} && (4.23) \\
& y_{1,i'+1} = y_{1,j'+1} = 0 && (4.24) \\
& y_{2,i'+1} = A_{j'} y_{2,j'+1} = 0 && (4.25) \\
& y_{1,t+1} = y_{1,t} + W_{1,t}^{(i,\alpha,i')} x_{1,t} - x_{2,t} \quad \forall t \in T_{i'j'} && (4.26) \\
& y_{2,t+1} = y_{2,t} + W_{2,t}^{(i,\alpha)} x_{2,t} - r_t \quad \forall t \in T_{i'j'} && (4.27) \\
& x_{st}, y_{st} \geq 0 \quad \forall t \in T_{i'j'} && (4.28) \\
& 0 \leq r_t \leq d_t \quad \forall t \in T_{i'j'}, && (4.29)
\end{aligned}
$$

where $M_{j'} = 0$ for all $j' \neq n$ and $M_n = \infty$; and $A_{j'} = 0$ if $j' = j$ or 1 otherwise.

In the above formulation, we include constraints (4.8)–(4.12) to ensure that the maintenance and inventory conditions are aligned with those in $(Q_{i\alpha j\beta})$. Constraint (4.23) ensures that production at stage 1 stops in period j' if $j' \neq n$. Constraints (4.24) and (4.25) ensures zero inventory at the beginning and the end of the inner subproblem, except for the ending inventory at the stage 2 of period j, which is defined by constraint

[3]The linear program may be infeasible in some special cases. In such cases, we set $c'(i', j' \mid d) = \infty$.

doi:10.6342/NTU202500607

(4.12). Finally, constraints (4.26) and (4.27) balances the inventory based on the realized yield rate $W_{1,t}$ and $W_{2,t}$.

Once all the $c'(i', j' \mid d)$ are computed through linear programming, we reformulate the subproblem as a shortest-path problem to find the minimum cost. The process is very similar to what we did in the single-stage problem. The graph consists of $j - i + 1$ nodes labeled from i to j, where node t represents period t and node ith serves as an artificial starting node. A directed edge from node i to j exists if and only if $i < j$. For edges (i', j') where $j' = i' + 1$ and $2 \le j' \le n - 1$, we set the edge cost to infinity; otherwise, the edge cost is $c'(i', j' \mid d)$. A path from node i to j passing through intermediate nodes $t \notin \{i, j\}$ implies that stage 1 maintenance is scheduled in each period t.

Let $g_{ij}(d)$ denote the cost of the shortest path from node i to j when the demand over the planning horizon T_{ij} is given by d. For $\beta \in \{S, \emptyset\}$, we have $c(i, \alpha, j, \beta) = g_{ij}(D)$. For $\beta \in \{N_0, N_+\}$, we compare $g_{ij}(D^{(f)}) + R_2 D_{j-1}$ (where demand in period $j-1$ is fulfilled) and $g_{ij}(D^{(r)}) + F D_{j-1}$ (where demand in period $j - 1$ is rejected) and let $c(i, \alpha, j, \beta)$ be the smaller one. Table 4.2 summarizes the formulas to calculate $c(i, \alpha, j, \beta)$ based on α and β (including those shortcut formulas when $j \le i + 2$).

We conclude the introduction of the two-stage exact algorithm by illustrating how to solve the four subproblems in the example given at the end of Subsection 4.2.2. The first, third, and fourth subproblems each consist of at most two periods, so their minimum costs can be readily obtained from Table 4.2. For the second subproblem, i.e., $(Q_{2,S,7,N_+})$, since its ending stating is N_+, we follow the remedy and create two alternative subproblems by adjusting the demand quantity.

In the case when the demand in period 6 is fulfilled, we set $D_t^{(f)} = D_t$ for all $t \in$

doi:10.6342/NTU202500607

Conditions	α	β	$c(i,\alpha,j,\beta)$
$j=i+1$	$\emptyset$	S	FD_1
	S	$\emptyset$	$(P_1+P_2)D_n$
	N_0	$\emptyset$	$\left(\frac{P_1+P_2}{W_{2,n}}\right)D_n$
$j=i+2$	$\emptyset$	$\emptyset$	$\min\left\{F(D_1+D_2), FD_1+(P_1+P_2)D_2, \left(\frac{P_1}{W_{11}W_{21}}+\frac{P_2}{W_{21}}\right)D_1 + \min\left\{\frac{P_1}{W_{11}W_{21}}+\frac{P_2}{W_{21}}+R_2, \frac{P_1}{W_{11}W_{22}}+\frac{P_2+R_1}{W_{22}}, \frac{P_1}{W_{12}W_{22}}+\frac{P_2}{W_{22}}, F\right\}D_2\right\}$
	$\emptyset$	S	$\min\left\{\frac{P_1}{W_{11}W_{21}}+\frac{P_2}{W_{21}}, F\right\}D_1+\min\left\{\frac{P_1}{W_{11}W_{21}}+\frac{P_2}{W_{21}}+R_2, F\right\}D_2$
	$\emptyset$	N_0	$FD_1+\left(\frac{P_1}{W_{11}}+P_2+R_1\right)D_2$
	$\emptyset$	N_+	$FD_1+\min\left\{\frac{P_1}{W_{11}}+P_2+R_1, F\right\}D_2+\left(\frac{P_1}{W_{11}}+P_2+R_1\right)D_3$
	S	S	$(P_1+P_2)D_{i+1}+\min\{P_1+P_2+R_2, F\}D_{i+2}$
	N_0	S	$\left(\frac{P_1+P_2}{W_{2,i+1}}\right)D_{i+1}+\min\left\{\frac{P_1+P_2}{W_{2,i+1}}+R_2, F\right\}D_{i+2}$
	N_0	N_0	$FD_{i+1}+(P_1+P_2+R_1)D_{i+2}$
	N_0	N_+	$FD_{i+1}+\min\{P_1+P_2+R_1, F\}D_{i+2}+(P_1+P_2+R_1)D_{i+3}$
	N_+	N_0	$R_2D_{i+1}+(P_1+P_2+R_1)D_{i+2}$
	N_+	N_+	$R_2D_{i+1}+\min\{P_1+P_2+R_1, F\}D_{i+2}+(P_1+P_2+R_1)D_{i+3}$
otherwise	S or N_0 or $\emptyset$	S or $\emptyset$	$g_{ij}(D)$
	S or N_0 or $\emptyset$	N_0 or N_+	$\min\left\{g_{ij}(D^{(f)})+R_2D_{j-1}, g_{ij}(D^{(r)})+FD_{j-1}\right\}$

Table 4.2: Computation of $c(i,\alpha,j,\beta)$

$\{3,4,7\}$, $D_5^{(f)}=D_5+D_6$, and $D_6^{(f)}=0$, and compute all $c'(i',j' \mid D^{(f)})$ values. For the other case when the demand in period 6 is rejected, we set $D_t^{(r)}=D_t$ for all $t\in\{3,4,5,7\}$ and $D_6^{(r)}=0$, and compute all $c'(i',j' \mid D^{(r)})$ values. For both cases, we then construct a graph with nodes labeled from 2 to 7 (where edge costs are based on either $c'(i',j' \mid D^{(f)})$ or $c'(i',j' \mid D^{(r)})$), solve the shortest path from node 2 to 7, and obtain the minimum costs $g_{27}(D^{(f)})$ and $g_{27}(D^{(r)})$. Finally, we record $c(2,S,7,N_+)$ as the minimum of $g_{27}(D^{(f)})+R_2D_6$ and $g_{27}(D^{(r)})+FD_6$.

We would like to add a remark on the design of our two-stage exact algorithm. In the previous subsection, we provided an overview of the main framework, which involves

doi:10.6342/NTU202500607

a larger graph with $3n-1$ nodes and edge costs defined as $c(i, \alpha, j, \beta)$. To compute each edge cost, we introduce a secondary and smaller graph with $j-i+1$ nodes and solve the shortest path on this graph to obtain $c(i, \alpha, j, \beta)$. This two-layer shortest-path structure enables us to trace the shortest path on the larger graph to determine stage 2 maintenance decisions, as well as certain stage 1 maintenance decisions that follow from the visited maintenance patterns. When an edge is traversed (on the larger graph), we then turn to its corresponding smaller graph to finalize the remaining stage 1 maintenance.

Regarding the time complexity of our entire two-stage exact algorithm, since the larger graph consists of $\mathcal{O}(n)$ nodes and $\mathcal{O}(n^2)$ edges, finding the shortest path takes $\mathcal{O}(n^2)$ time once all edge costs are known. The bottleneck thus lies in computing each edge cost, which involves solving all the subproblems. For each subproblem, we construct a smaller graph with $\mathcal{O}(n)$ nodes and $\mathcal{O}(n^2)$ edges to find its minimum cost. Let $\mathcal{O}(L_n)$ be the (polynomial) time required for a linear programming solver to solve a linear program with $\mathcal{O}(n)$ variables and $\mathcal{O}(n)$ constraints. We then need $\mathcal{O}(n^2 L_n)$ time to compute all edge costs within the smaller graph. Since finding the shortest path on a smaller graph takes $\mathcal{O}(n^2)$ time, solving a single subproblem requires a total time of $\mathcal{O}(n^2 L_n + n^2) = \mathcal{O}(n^2 L_n)$. Finally, given that there are $\mathcal{O}(n^2)$ subproblems in total, we can conclude that the total time complexity of our algorithm is $\mathcal{O}(n^4 L_n)$.

doi:10.6342/NTU202500607

4.3 A heuristic algorithm for the multi-stage problem

When the system has more than two stages, the combinations of maintenance patterns across all stages can become highly complex. Whether a polynomial-time exact algorithm exists for the multi-stage problem remains an open question. To complement our theoretical results, we propose an efficient heuristic algorithm that generates a near-optimal solution for the multi-stage problem in this section. The average-case performance of our heuristic algorithm will be evaluated in the next section.

Our heuristic algorithm is inspired by the exact algorithm for the single-stage problem. The high-level idea is intuitive: we decompose the multi-stage problem by stage, apply the single-stage exact algorithm to each stage, and combine all resulting schedules to obtain the final solution. To be more precise, let (x^H, z^H) denote the schedule reported by the heuristic algorithm, and let d_t^s denote the "demand" at stage s in period t when applying the single-stage algorithm. Below we introduce our two-phase heuristic algorithm for the multi-stage problem.

In phase 1 of our heuristic algorithm, we start with the last stage (which produces end products) and initialize d_t^m to D_t. We solve the production and maintenance plan for the last stage by applying the single-stage algorithm and obtain the values of $x_{m,t}^H$ and $z_{m,t}^H$ for all $t \in T$. We then move on to the $(m-1)$th stage and construct a single-stage problem for it. Since the input quantity to stage m has just been determined and is known for each period, we set the "demand" at stage $m-1$ in period t to match the input quantity

doi:10.6342/NTU202500607

required by stage m in the same period. More precisely, we define $d_t^{m-1} = x_{m,t}^H$ for the single-stage problem at stage $m-1$ and apply the exact algorithm again to solve it. This process continues for the $(m-2)$th stage, the $(m-3)$th stage, and subsequent stages until reaching the first stage. Therefore, when phase 1 terminates, we have all values of x_{st}^H and z_{st}^H for all $s \in S$ and $t \in T$. The flowchart for phase 1 of the heuristic algorithm is depicted in Figure 4.10.

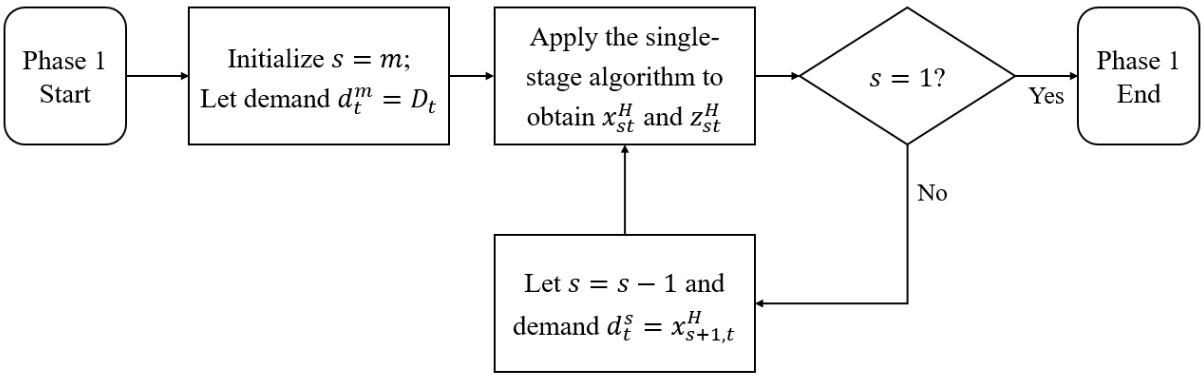

Figure 4.10: Flow chart for phase 1 of the heuristic algorithm

Before we introduce phase 2 of our heuristic algorithm, we would like to add a remark on our problem. Note that in our problem, once the maintenance schedule is given, the yield rate in each period at each stage is realized and becomes exogenous. This implies that we can obtain an optimal production plan under the given maintenance schedule by solving the resulting linear program. Building on this observation, in phase 2, we fix the maintenance schedule z^H obtained in phase 1 and solve the resulting linear program to update the production plan x^H. The updated production plan and the original maintenance schedule are then reported as the heuristic solution. Algorithm 1 presents the pseudocode of our heuristic algorithm.

doi:10.6342/NTU202500607

Algorithm 1 A heuristic algorithm for the multi-stage problem

Phase 1

```
Let $(x^H, z^H)$ denote the heuristic solution.
Let $d_t^s$ be the demand at stage $s$ in period $t$ when solving the single-stage problem.
Set $d_t^m = D_t$ for all $t \in T$.
for $s$ from $m$ to 1 do
    Construct a single-stage problem for stage $s$ with demand $d_t^s$.
    Apply the single-stage algorithm to obtain $x_{st}^H$ and $z_{st}^H$ for all $t \in T$.
    Set $d_t^{s-1} = x_{st}^H$ for all $t \in T$.
end for
```

Phase 2

```
Fix $z^H$ and update $x^H$ by solving the resulting linear program.
Report $(x^H, z^H)$ as the heuristic solution.
```

4.4 Numerical study

In this section, we demonstrate the average-case performance of our heuristic algorithm through numerical experiments. All the experiments in this section are conducted on a laptop with Windows 10, Intel i7-1165G7, 2.80 GHz, and 8 GB RAM. We implement our heuristic algorithm in Python 3.12.

doi:10.6342/NTU202500607

4.4.1 Experiment settings

In our experiments, we consider instances with $m = 3$ stages and $n = 10$ periods. In the basic setting, we set the production cost $P_s = 10$ and the inventory cost $R_s = 3s$ for stage s. Note that the inventory cost increases as the stage approaches the final stage. The unit shortage cost is $F = 50$. The yield declining rate (in %) of stage s is uniformly drawn from $[2, 8]$, and the lower bound of yield rate is set to be $L_s = 50\%$. Finally, for each period, we randomly select an integer between 1 and 20 as the demand quantity.

To further investigate our algorithm under different situations, we design three factors and seven scenarios in our experiments. The first factor is the rate of machine deterioration. The yield declining rate of a stage is uniformly drawn from $[0, 5]$ for the scenario with a low deterioration rate and from $[5, 10]$ for the scenario with a high deterioration rate. The second factor is the volatility of demand. For the scenario with low demand variance, the range of demand is narrowed to 1 to 10; for the scenario with high demand variance, the range of demand is extended to 1 to 40. The last factor is the inventory costs. We set $R_s = s$ and $R_s = 5s$ for scenarios with (relatively) low and high inventory costs, respectively. The information of the seven scenarios is summarized in Table 4.3.

4.4.2 Benchmark

We compare our algorithm with two benchmarks. The first one is our nonlinear integer program. We write Python programs to invoke Gurobi Optimizer 11.0 to solve the mathematical program. The second benchmark is a genetic algorithm (GA). In this algorithm, a solution's chromosome, with size $m \times n$, represents the maintenance plan. Entry (s, t) in

doi:10.6342/NTU202500607

Scenario	Scenario name	B_{st}	D_t	R_s
1	Basic setting	$U[2,8]$	$U[1,20]$	$3s$
2	Low deterioration rate	$U[0,5]$	$U[1,20]$	$3s$
3	High deterioration rate	$U[5,10]$	$U[1,20]$	$3s$
4	Low demand variance	$U[2,8]$	$U[1,10]$	$3s$
5	High demand variance	$U[2,8]$	$U[1,40]$	$3s$
6	Low inventory cost	$U[2,8]$	$U[1,20]$	s
7	High inventory cost	$U[2,8]$	$U[1,20]$	$5s$

Table 4.3: Experiment setting of scenarios

the chromosome is 1 if maintenance is conducted at stage s in period t or 0 otherwise. To generate an initial solution, we randomly set each entry in the chromosome to 0 or 1 with equal probability. We then repeat the above process and generate 30 initial solutions to create the initial solution pool. For each solution, we follow the maintenance plan recorded in its chromosome and solve the resulting linear program to obtain an optimal production plan. The cost of the schedule is then computed and recorded as the fitness value of the solution.

In each GA iteration, we first randomly select 10 solutions from the solution pool. The two solutions with the highest fitness values among them are then chosen as parents to performe crossover. We randomly select a crossover point t_s for each stage s and generate two offspring as follows. For the first offspring, the first t_s entries of stage s are from parent 1, and the remaining $n - t_s$ entries come from parent 2. For the second offspring, the first t_s entries of stage s are from parent 2, and the remaining $n - t_s$ entries come

doi:10.6342/NTU202500607

from parent 1. For each offspring, there is a 5% mutation probability such that a random stage and a random period are selected, and the maintenance decision is reversed. To complete one iteration, we add the two offspring to the solution pool and remove the two worst solutions. After 50 iterations, the best solution in the pool is reported.

4.4.3 Solution performance

For each scenario, we randomly generate 50 instances. For each instance, we first solve the nonlinear integer program to obtain an optimal solution, then solve the same instance using our heuristic algorithm and the genetic algorithm. We use z^*, z^{HEU}, and z^{GA} to denote the cost of an optimal solution, the heuristic solution, and the solution reported by the genetic algorithm, respectively. Table 4.4 summarizes the average and the standard deviation of the optimality gaps under different scenarios. From the last row, we may see that our proposed algorithm reports a solution that has a smaller cost than the solution generated by the genetic algorithm on average. A smaller standard deviation also implies that our algorithms are more robust than the genetic algorithm.

By looking at scenarios 2 and 3, we see that our heuristic algorithm performs better when the yield declining rate is low. When the deterioration rate is high, finding the proper timing to maintain a machine becomes more critical; otherwise, we have to produce products with a poor machine yield rate. This makes the problem more challenging in such cases. Scenarios 4 and 5 show that our algorithm can generate good solutions regardless of the variance of demand. Finally, the last two scenarios suggest that our algorithm achieves better performance when the inventory cost is low. When the inventory cost is high, the trade-off between storing products (conducting maintenance) and pro-

doi:10.6342/NTU202500607

Scenario	Scenario name	Average		S.D.	
		$\frac{z^{\text{HEU}}}{z^*} - 1$	$\frac{z^{\text{GA}}}{z^*} - 1$	$\frac{z^{\text{HEU}}}{z^*} - 1$	$\frac{z^{\text{GA}}}{z^*} - 1$
1	Basic setting	5.03%	6.80%	2.63%	3.12%
2	Low deterioration rate	3.36%	5.54%	2.20%	3.01%
3	High deterioration rate	6.19%	9.05%	3.78%	4.92%
4	Low demand variance	5.16%	6.98%	2.73%	3.32%
5	High demand variance	5.29%	7.58%	2.64%	3.60%
6	Low inventory cost	1.86%	4.56%	1.36%	2.15%
7	High inventory cost	5.59%	6.63%	3.32%	3.59%
	Average	4.64%	6.73%	3.07%	3.69%

Table 4.4: Optimality gaps of the algorithms

ducing with a lower yield rate (delaying maintenance) becomes more difficult. Therefore, our algorithm cannot achieve the same performance quality as in other cases.

Regarding the running time of our heuristic algorithm, in the basic scenario, it takes around 0.111 seconds to solve an instance, while the genetic algorithm takes about 0.664 seconds to report a solution. The running time in other scenarios is similar to that in the basic scenario. This demonstrates that our heuristic algorithm outperforms the genetic algorithm not only in effectiveness but also in efficiency.

doi:10.6342/NTU202500607

Chapter 5

Stochastic Yield Declining Rates

In this chapter, we extend our problem and algorithms by considering stochastic yield declining rates. We assume that the yield declining rates are no longer deterministic but follow a known probability distribution. In Section 5.1, we describe the stochastic version of our problem and explain how to modify our heuristic algorithm to address the randomness. In Section 5.2, we extend the numerical experiments from Section 4.4 to evaluate our heuristic algorithm under a stochastic setting.

5.1 Problem description and algorithms

So far, we have assumed that the yield declining rate in each period at each stage is deterministic and known before planning begins. This assumption relies heavily on an accurate estimation of machine conditions. However, in practice, machine conditions in future periods often exhibit high variance and cannot be precisely predicted in advance. This requires manufacturers to make decisions under random yield declining rates and

doi:10.6342/NTU202500607

motivates us to study the stochastic version of our problem.

We assume that the yield declining rate is stochastic but follows a known probability distribution. Mathematically, let $\widetilde{B}_{st}$ be the random variable that represents the yield declining rate of stage s in period t. Since the distribution is known, the expected yield declining rate $\mathbb{E}[\widetilde{B}_{st}]$ can be readily computed for all $s \in S$ and $t \in T$.

To address the stochastic problem, we modify our heuristic algorithm by adopting a rolling schedule approach as follows. In each period, we first compute the expected yield declining rate and solve the problem as if it were a deterministic problem using our heuristic algorithm in Section 4.3. Once a heuristic schedule is obtained, we execute the production and maintenance plan for the first period. At the end of the first period, the actual yield declining rates are realized, and we update the yield rate accordingly. At the beginning of the second period, we reapply our heuristic algorithm to solve the problem with a planning horizon from the second period to the last period. The updated plan for the second period is then executed, and this process repeats until all periods are visited.

5.2 Numerical study

To evaluate our heuristic algorithm under a stochastic setting, we extend the numerical experiments in Section 4.4. In the basic setting (scenario 1), we follow all the settings, except we let $\widetilde{B}_{st}$ follow a uniform distribution $U[2, 8]$. Regarding the remaining scenarios, we modify scenarios 2 and 3 to test the variance of $\widetilde{B}_{st}$. In scenario 2, we let $\widetilde{B}_{st} \sim U[4, 6]$ to represent a smaller variance; in scenario 3, we let $\widetilde{B}_{st} \sim U[0, 10]$ to represent a larger variance. Scenarios 4, 5, 6, and 7 (which evaluate the impact of demand volatility and

doi:10.6342/NTU202500607

inventory costs) are unchanged. The modified scenarios are presented in Table 5.1.

Scenario	Scenario name	$\widetilde{B}_{st}$	D_t	R_s
1	Basic setting	$U[2,8]$	$U[1,20]$	$3s$
2	Low yield declining rate variance	$U[4,6]$	$U[1,20]$	$3s$
3	High yield declining rate variance	$U[0,10]$	$U[1,20]$	$3s$
4	Low demand variance	$U[2,8]$	$U[1,10]$	$3s$
5	High demand variance	$U[2,8]$	$U[1,40]$	$3s$
6	Low inventory cost	$U[2,8]$	$U[1,20]$	s
7	High inventory cost	$U[2,8]$	$U[1,20]$	$5s$

Table 5.1: Experiment setting of scenarios

Regarding the benchmarks, we modify the genetic algorithm to a rolling schedule version. For our nonlinear integer program, note that once all periods are visited in an experiment, the actual yield declining rates become known information. We imagine a situation in which we were to know the actual yield declining rates in advance and solve the corresponding deterministic problem. This provides the best possible plan under the stochastic setting. However, since this method relies on knowing future information, it is impractical and serves only as a benchmark.

Table 5.2 provides the experiment results. Compared to the experiments conducted in Section 4.4, we observe that the average optimality gaps are slightly larger here. One reason is that the optimal solution here is able to acquire future information in advance, and therefore comparing to such solutions makes our performance not as good as the deterministic ones. The stochastic setting also results in larger standard deviations.

doi:10.6342/NTU202500607

Scenario	Scenario name	Average		S.D.	
		$\frac{z^{\text{HEU}}}{z^*} - 1$	$\frac{z^{\text{GA}}}{z^*} - 1$	$\frac{z^{\text{HEU}}}{z^*} - 1$	$\frac{z^{\text{GA}}}{z^*} - 1$
1	Basic setting	6.55%	9.76%	6.45%	4.97%
2	Low yield declining rate variance	5.54%	9.09%	4.25%	4.31%
3	High yield declining rate variance	6.90%	8.74%	7.95%	4.22%
4	Low demand variance	7.15%	8.52%	7.32%	4.93%
5	High demand variance	6.75%	8.92%	5.72%	4.63%
6	Low inventory cost	1.08%	5.63%	1.10%	3.06%
7	High inventory cost	9.61%	9.64%	7.30%	4.94%
	Average	6.22%	8.61%	6.54%	4.63%

Table 5.2: Optimality gaps of the algorithms

By comparing the results in scenarios 2 and 3, we observe that the performance of our algorithm drops as the variance of the yield declining rates increases. This observation aligns with our intuition since we only use the expectation to estimate the future yield rate. Therefore, the heuristic algorithm struggles when the variance is large. The results and interpretations of the other scenarios are similar to those discussed in Section 4.4.

For the running time, our heuristic algorithm takes around 0.404 seconds to find a 10-period plan for an instance using a rolling schedule approach (so that our algorithm is invoked 10 times). On the contrary, the genetic algorithm takes about 3.010 seconds to generate a schedule. This again confirms that our heuristic algorithm is both effective and efficient under a stochastic setting.

doi:10.6342/NTU202500607

Chapter 6

Conclusion

In this study, we investigate a single-product flow shop joint production-maintenance planning problem in a deteriorating system. Unlike traditional works that treat production planning and preventive maintenance as separate problems, we propose an integrated model that considers both decisions simultaneously. Since solving the mathematical program is time-consuming, we develop algorithms that generate either an optimal or a near-optimal solution efficiently.

When the system consists of only one stage, we prove that the inventory and maintenance cycles are nested and reformulate our problem as a shortest-path problem, where the edge costs can be solved by linear programming. This reformulation allows the problem to be solved in polynomial time using any shortest-path algorithm. When the system consists of two stages, we show that maintenance timing across the two stages follows either a synchronized or a neighboring pattern. This property enables us to extend the single-stage algorithm and design an exact algorithm that exhibits a two-layer shortest-path structure.

doi:10.6342/NTU202500607

For problems with more than two stages, we propose a heuristic algorithm that utilizes the single-stage algorithm to generate a near-optimal solution. We also demonstrate how our heuristic algorithm can be extended to the case when the yield declining rates are stochastic. Through numerical experiments, we verify the effectiveness and efficiency of our heuristic algorithm under both deterministic and stochastic settings.

There are two future directions for our study. The first is the solvability of the multi-stage problem. When the system consists of more than two stages, it is worthwhile to investigate whether there exists an exact algorithm to find an optimal solution. The second direction is to consider a capacitated version of our problem. In this study, we assume that production capacity is unlimited when no maintenance is performed. Investigating whether the problem under capacity constraints is polynomial-time solvable (or show that it is NP-hard) presents another promising research opportunity.

doi:10.6342/NTU202500607

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., A. Khatab, P. L. Tam. 2016. Optimizing production and imperfect preventive maintenance planning's integration in failure-prone manufacturing systems. *Reliability Engineering & System Safety* **145** 190–198.

Aldurgam, M. M. 2020. Dynamic maintenance, production and inspection policies, for a single-stage, multi-state production system. *IEEE Access* **8** 105645–105658.

Brahimi, N., N. Absi, S. Dauzère-Pérès, A. Nordli. 2017. Single-item dynamic lot-sizing problems: an updated survey. *European Journal of Operational Research* **263**(3) 838–863.

Brahimi, N., S. Dauzère-Pérès, N. M. Najid, A. Nordli. 2006. Single item lot sizing problems. *European Journal of Operational Research* **168**(1) 1–16.

Buschkuehl, L., F. Sahling, S. Helber, H. Tempelmeier. 2010. Dynamic capacitated lot-

doi:10.6342/NTU202500607

sizing problems: a classification and review of solution approaches. *OR Spectrum* **32** 231–261.

Florian, M., M. Klein. 1971. Deterministic production planning with concave costs and capacity constraints. *Management science* **18**(1) 12–20.

Guo, W., X. Gu. 2020. Joint decision-making of production and maintenance in mixed model assembly systems with delayed differentiation configurations. *International Journal of Production Research* **58**(13) 4071–4085.

Hwang, H.-C., H.-S. Ahn, P. Kaminsky. 2013. Basis paths and a polynomial algorithm for the multistage production-capacitated lot-sizing problem. *Operations Research* **61**(2) 469–482.

Love, S. F. 1972. A facilities in series inventory model with nested schedules. *Management Science* **18**(5-part-1) 327–338.

Nourelfath, M., M.-C. Fitouhi, M. Machani. 2010. An integrated model for production and preventive maintenance planning in multi-state systems. *IEEE Transactions on Reliability* **59**(3) 496–506.

Shao, X., Z. Chen, B. R. Sarker. 2022. An joint decision of production and maintenance plan (Q, N) for a two-stage deteriorating JIT production system with random breakdowns. *Production Engineering* **16** 89–107.

Sloan, T. W., J. G. Shanthikumar. 2000. Combined production and maintenance scheduling for a multiple-product, single-machine production system. *Production and Operations Management* **9**(4) 379–399.

doi:10.6342/NTU202500607

Wagner, H. M., T. M. Whitin. 1958. Dynamic version of the economic lot size model. *Management science* **5**(1) 89–96.

Xiang, Y., C. R. Cassady, T. Jin, C. W. Zhang. 2014. Joint production and maintenance planning with machine deterioration and random yield. *International Journal of Production Research* **52**(6) 1644–1657.

Zangwill, W. I. 1968. Minimum concave cost flows in certain networks. *Management Science* **14**(7) 429–450.

Zangwill, W. I. 1969. A backlogging model and a multi-echelon model of a dynamic economic lot size production system—a network approach. *Management Science* **15**(9) 506–527.

Zhang, N., K. Cai, Y. Deng, J. Zhang. 2023. Determining the optimal production–maintenance policy of a parallel production system with stochastically interacted yield and deterioration. *Reliability Engineering & System Safety* **237** 109342.

Zhao, M., M. Zhang. 2020. Multiechelon lot sizing: New complexities and inequalities. *Operations Research* **68**(2) 534–551.

doi:10.6342/NTU202500607