Stochastic and multi-criteria optimization for remanufacturing industry

Peng Hu

université PARIS-SACLAY

FUZHOU UNIVERSITY

Stochastic and multi-criteria optimization for remanufacturing industry

Optimisation stochastique et multicritère pour l'industrie remanufacturière

Thèse de doctorat de l'Université Paris-Saclay et Fuzhou University

École doctorale n°580 Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique mathématique
Graduate School: Informatique et sciences du numérique
Référent: Université d'Évry Val d'Essonne

Thèse préparée dans l'unité de recherche **IBISC** (Université Paris-Saclay), sous la direction de **Feng Chu**, Professeure, et le co-encadrement de **Peng Wu**, Professeur

Thèse soutenue à Paris-Saclay, le 24 novembre 2023, par

Peng Hu

Composition du jury

Membres du jury avec voix délibérative

Saïd Mammar Professeur, Université Paris-Saclay	Président
Olga Battaia Professeure, KEDGE Business School	Rapporteur & Examinatrice
Zeqiang Zhang Professor, Southwest Jiaotong University	Rapporteur & Examinateur
Debiao Li Professor, Fuzhou University	Examinateur

THESE DE DOCTORAT

NNT : 2023UPASG070

Praface

Avec le rythme accéléré du développement technologique, divers produits en fin de vie (EOL) sont générés rapidement. Selon les données de la Banque mondiale, environ 2,01 milliards de tonnes de déchets sont générées chaque année dans le monde, et ce chiffre devrait atteindre 3,40 milliards de tonnes d'ici 2050. Une gestion inappropriée des produits EOL peut entraîner la production de déchets pollués substantiels, conduisant à des dommages environnementaux, sanitaires et économiques catastrophiques. Une approche prometteuse pour résoudre ce problème consiste à recycler les produits EOL pour refabriquer des produits de seconde vie afin d'atteindre un objectif d'économie circulaire.

En tant qu'élément crucial du recyclage des produits EOL, le démontage sépare les produits EOL en composants destinés à une destruction ou à une remanufacturation, ce qui a fait l'objet d'une grande attention ces dernières années. Le processus de désassemblage peut être effectué sur une ligne de désassemblage, qui consiste en une série de postes de travail où les produits EOL sont démantelés à travers une séquence de tâches. Pour la ligne de désassemblage, il existe plusieurs concepts et contraintes importants. En termes de concepts, le schéma de désassemblage désigne la combinaison de tâches de désassemblage qui peuvent compléter la décomposition des produits EOL, et le temps de cycle signifie la plus grande quantité de temps de traitement nécessaire pour accomplir les tâches parmi toutes les stations de travail. Quant aux contraintes, premièrement, les relations de préséance des tâches doivent être respectées, ce qui signifie que les tâches de démontage ne peuvent être exécutées qu'après leurs prédécesseurs. Deuxièmement, la somme des temps de traitement de chaque poste de travail ouvert ne peut excéder le temps de cycle permettant d'équilibrer les charges entre les postes de travail.

Le problème d'équilibrage des lignes de désassemblage (DLBP), en tant que l'un des problèmes les plus étudiés concernant le démontage, vise à sélectionner un schéma de désassemblage approprié, à déterminer les postes de travail à ouvrir, et à attribuer les tâches correspondantes du schéma sélectionné aux postes de travail ouverts pour optimiser les indicateurs de performance industrielle, tels que le coût de production, le profit de la ligne de désassemblage ou les émissions de carbone. Dans la littérature existante, 1) la plupart des problèmes stochastiques d'équilibrage de la ligne de désassemblage supposent que les distributions de probabilité des paramètres incertains soient connues ; 2) la majorité des problèmes d'équilibrage de la ligne de désassemblage se concentrent sur un seul produit ; 3) peu de travaux portent sur les problèmes de conception de la chaîne logistique inversée (RSC) liés à l'équilibrage de la ligne de désassemblage. En réalité, plusieurs RSC liées au désassemblage de produits EOL existent dans les industries de la remanufacturation, telles que l'automobile, les téléphones mobiles, etc. Pour combler ces lacunes dans la littérature, trois nouveaux problèmes liés à l'équilibrage de la ligne de désassemblage sont étudiés dans cette thèse.

Tout d'abord, une DLBP à produit unique avec des informations partielles sur les temps de traitement des tâches est étudiée, où seules la moyenne, la borne inférieure et la borne supérieure des temps de traitement des tâches sont connues. L'objectif est de minimiser le coût de désassemblage. Pour le problème étudié, un modèle conjoint à contraintes de probabilités est proposé. Ensuite, une nouvelle formulation sans distribution et une formulation basée sur une approximation de programme de cônes de second ordre sont construites en fonction des propriétés du problème. Les résultats expérimentaux sur 7 instances de référence et sur 81 instances générées aléatoirement montrent l'efficacité de l'approche proposée.

Deuxièmement, une nouve DLBP stochastique multi-produits avec un temps de traitement de tâche incertain est abordée, où seules la moyenne, l'écart type et la limite supérieure des temps de tâche sont disponibles. L'objectif est de minimiser le coût de désassemblage. Pour le problème, un modèle conjoint à contraintes de probabilités est formulé. Ensuite, sur la base de l'analyse du problème, le modèle conjoint à contraintes de probabilités est approximativement transformé en un modèle sans distribution. Ensuite, plusieurs inégalités valides et une méthode

exacte de coupe et de résolution sont conçues pour résoudre efficacement le problème. Les résultats des expériences sur un exemple illustratif et sur 490 instances générées aléatoirement démontrent les bonnes performances du modèle proposé, des inégalités valides et de la méthode de résolution.

Enfin, un nouveau problème de conception de la RSC lié à l'équilibrage de la ligne de désassemblage multi-produits est étudié, où l'approvisionnement en produits EOL, la demande en composants et les temps de traitement des tâches sont supposés incertains. Les objectifs sont de maximiser le profit attendu et de minimiser simultanément les émissions de CO_2. Pour le problème, un modèle bi-objectif de programmation stochastique à deux étapes et non linéaire est formulé, et approximativement transformé en un modèle sans distribution linéaire en fonction des propriétés du problème. Ensuite, une méthode basée sur des contraintes ε-construites est proposée, dans laquelle une décomposition de Benders améliorée est conçue pour résoudre les problèmes transformés à objectif unique. Des expériences numériques comprenant une étude de cas et sur 200 instances générées aléatoirement sont menées pour évaluer les performances des méthodes proposées. De plus, une analyse de sensibilité est réalisée pour tirer des enseignements en matière de gestion.

Acknowledgements

First and foremost, I would like to express my profound gratitude to my supervisor, Professor Feng Chu. Her unwavering support, insightful guidance, and extraordinary patience through this challenging period have been invaluable, particularly considering my transition into a new research direction.

I want to thanks to all the jury members of my thesis defense: Professor Saïd Mammar, Professor Olga Battaia, Professor Zeqiang Zhang, and Professor Debiao Li. Your valuable insights and contributions during the defense process were greatly appreciated.

I am grateful to those who have provided invaluable guidance throughout my research journey, including Peng Wu, Ming Liu, Yunfei Fang, Shijin Wang, and others. My gratitude also extends to my friends, including Liping Gao, Jing Li, Ding Chen, Junkai He, Victoria Bourgeais, and many others, who have shared this academic journey with me.

I want to thank the administrative staff at IBISC, especially Madame Murielle Bourgeois, for their unwavering assistance and support. Additionally, I am grateful to the China Scholarship Council (CSC) for financial support, which allowed me to pursue my doctoral studies in France.

I want to express my deepest gratitude to my family for their unwavering support and understanding. My heartfelt thanks go to my wife, Die Hu, who has been my pillar of strength, supporting me throughout my academic journey and every aspect of my life.

Contents

List of Figures

List of Tables

Acronyms

EOL: End of life

DLBP: Disassembly line balancing problem

ALBP: Assembly line balancing problem

SC: Supply chain

RSC: Reverse supply chain

CLSC: Closed-loop supply chain

CS: Cut-and-solve

LCS: Lifted cut-and-solve

BD: Benders decomposition

IMD: Improved benders decomposition

B&B: Branch and bound

B&C: Branch and cut

BBR: Branch, bound and remember

IP: Integer programming

MIP: Mixed-integer programming

MINP: Mixed-integer nonlinear programming

TSSP: Two-stage stochastic programming

MP: Master problem

SP: Sub-problem

NSGA-II: Non-dominated sorting genetic algorithm-II **GA:** Genetic algorithm

SAA: Sample average approximation

VI: Valid inequality

Parameter Notations

Parameters:

— I : set of supply points indexed by i, and $I = \{1, 2, ..., |I|\}$;

— J : set of collection centers indexed by j, and $J = \{1, 2, ..., |J|\}$;

— K : set of candidate locations for disassembly plants indexed by k, and $K = \{1, 2, ..., |K|\}$;

— L : set of remanufacturing plants indexed by l, and $L = \{1, 2, ..., |L|\}$;

— M : set of machines indexed by m, and $M = \{1, 2, ..., |M|\}$;

— W : set of workstations indexed by w, and $W = \{1, 2, ..., |W|\}$;

— P : set of EOL products indexed by p, and $P = \{1, 2, ..., |P|\}$;

— C : set of components indexed by c, and $C = \{1, 2, ..., |C|\}$;

— R : set of disassembly tasks indexed by r, and $R = \{1, 2, ..., |R|\}$;

— N : set of subassembly nodes indexed by n, and $N = \{1, 2, ..., |N|\}$;

— T : set of time periods indexed by t, and $T = \{0, 1, 2, ..., |T|\}$;

— S : set of scenarios indexed by s, and $S = \{1, 2, ..., |S|\}$;

— ρ_s : the probability of scenario s, where $s \in S$;

— $h_{kk'}$: binary parameter, equal to 1 if $d_{kk'} \geq d_{min}$, 0 otherwise, where $d_{kk'}$ denotes the distance between candidate disassembly plants k and k', d_{min} represents the smallest segregation distance requirement between every pair of opened disassembly plants, and $k, k' \in K$, and $k \neq k'$;

— Ca_j^p : inventory capacity of EOL products of collection center j, where $j \in J$;

— Ca_k^c : inventory capacity of components of disassembly plant k, where $k \in K$;

— Ca_k^p : production capacity of disassembly plant k, where $k \in K$;

— α : a given probability (risk level) that the workloads of all workstations exceeds the cycle time;

— Sv_{ipts} : supply volume of EOL product p at supply point i during period t under scenario s, where $i \in I$, $p \in P$, $t \in T$, $s \in S$;

— De_{lcts} : demand of component c of remanufacturing plant l during period t under scenario s, where $c \in C$, $l \in L$, $t \in T$, $s \in S$;

— pt_r : processing times of task r, where $r \in R_p$;

— C_k^S : fixed cost to set up a disassembly plant at site k, where $k \in K$;

— C_w : fixed cost to open a workstation;

— C_h : fixed cost for handing a hazardous task;

— C_p^p : fixed cost to preprocess (including detect and classify) unit EOL product p in collection center, where $p \in P$;

— C_p^d : fixed cost to disassemble unit EOL product p, where $p \in P$;

— C_{ijp}^{S-C} : transportation cost of one unit EOL product p from supplier i to collection center j, where $i \in I$, $j \in J$, $p \in P$;

— C_{jkp}^{C-D} : transportation cost of one unit EOL product p from collection center j to disassembly plant k, where $j \in J$, $k \in K$, $p \in P$;

— C_{klc}^{D-R} : transportation cost of one unit component c from disassembly plant k to remanufacturing plant l, where $k \in K$, $l \in L$, $c \in C$;

— C_p^{IP} : inventory cost of one unit EOL product p in each time period, where $p \in P$;

— C_c^{IC} : inventory cost of one unit component c in each time period, where $c \in C$;

— σ_c^R : revenue of fulfilled demand of one component $c \in C$ in each time period;

Decision variables:

— x_{wr} : binary variable, equal 1 if task r is assigned to workstation w, 0 otherwise, where $w \in W$, $r \in R$;

— y_w : binary variable, equal 1 if workstation w is opened, 0 otherwise, where $w \in W$;

— h_w : binary variable, equal 1 if workstation w handles hazardous tasks, 0 otherwise, where $w \in W$;

— z_k : binary variable, equal 1 if a disassembly plant is set up at site k, 0 otherwise, where $k \in K$;

— q_{ijpts}^{S-C} : the quantity of EOL product p transported from supply point i to collection center j during period t under scenario s, where $i \in I$, $j \in J$, $p \in P$, $t \in T$, $s \in S$;

— q_{jkpts}^{C-D} : the quantity of EOL product p transported from collection center j to disassembly plant k during period t under scenario s, where $j \in J$, $k \in K$, $p \in P$, $t \in T$, $s \in S$;

— q_{klcts}^{D-R} : the quantity of component c transported from disassembly plant k to remanufacturing plant l during period t under scenario s, where $k \in K$, $l \in L$, $c \in C$, $t \in T$, $s \in S$;

— τ_{jpts}^{IC} : the inventory of EOL product p at collection center j during period t under scenario s, where $j \in J$, $p \in P$, $t \in T$, $s \in S$;

— τ_{kcts}^{ID} : the inventory of component c at disassembly plant k during period t under scenario s, where $k \in K$, $c \in C$, $t \in T$, $s \in S$;

Chapter 1

Introduction

Contents

1.1 Context and Motivations

With the accelerating pace of technology development, various End-of-Life (EOL) products are generated rapidly. According to the data from World Bank (wor, 2018), approximately 2.01 billion tons of solid waste are generated worldwide annually, and this number is estimated to increase to 3.40 billion tons by 2050. For example, China will scrap at least 10 million shared bikes in the next few years, resulting in nearly 300,000 tons of scrap metal (Sun et al., 2023). Inappropriate management of EOL products will result in substantial polluted waste, leading to catastrophic environmental, health and economic damages. One promising approach to address this issue is recycling EOL products by disassembling them into components to remanufacture second-life products to achieve circular objectives. Figure 1.1 depicts the diagram of the flows of EOL and second-life products, where the gray part represents the recycling of EOL products (Diri Kenger et al., 2020). Firstly, EOL products are transferred from end customers to collection centers, where they are inspected, classified and stored. Subsequently, EOL products will be transported to disassembly plants, where they can be decomposed into components, and valuable ones will be shipped to remanufacturing plants and useless ones will be disposed of.

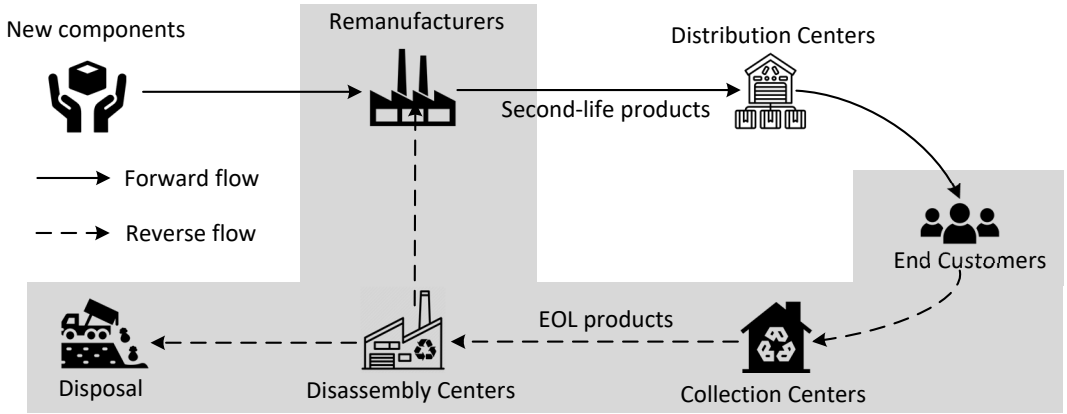


Figure 1.1: An illustrative diagram of forward and reverse flows of products

Disassembly, as one of the crucial parts in EOL products recycling (Paterson et al., 2017), consists of decomposing EOL products into components for further remanufacturing or disposal. The disassembly line often comprises a set of workstations in which EOL products are decomposed by a sequence of tasks. Figure 1.2 shows a disassembly line for an EOL product, where

there are three disassembly tasks and two workstations. For the disassembly line, there are several important concepts and constraints. In terms of concepts, the disassembly scheme denotes the combination of disassembly tasks that can complete the decomposition of EOL products, and cycle time means the largest amount of processing time needed to complete the tasks among all workstations. As for constraints, firstly, task precedence relationships should be respected, which means that disassembly tasks can only be executed after their predecessors. For example, in Figure 1.2, tasks 2 and 3 should be executed after task 1. Secondly, the sum of processing times of each opened workstation can not exceed the cycle time to balance loads between workstations.

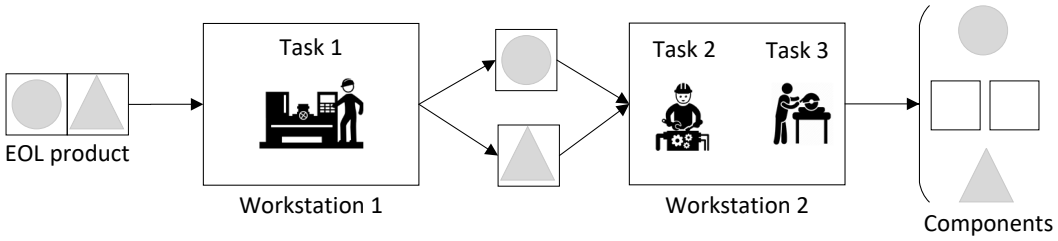


Figure 1.2: An illustrative disassembly line of an EOL product

Disassembly line balancing problem (DLBP), as one of the most studied problems concerning disassembly, aims to select an appropriate disassembly scheme, determine workstations to be opened, and assign corresponding tasks of the selected scheme to opened workstations to optimize industrial performance indicators, such as production cost, disassembly line profit or carbon emission (Güngör and Gupta, 2002). Most of the existing studies focus on deterministic DLBP (Gungor and Gupta, 2001; McGovern and Gupta, 2007b; Riggs et al., 2015; Fang et al., 2020a; Li et al., 2020). However, many uncertain factors should be properly considered during the disassembly process, such as the uncertain task processing times that may be caused by variable operator skills and structure and quality of EOL products (Bentaha et al., 2015a). Özceylan et al. (2019) indicate that only about 30% of the existing studies investigate DLBP with uncertainty. Most DLBP studies with uncertainties assume that the distribution information of the uncertain parameter is known. However, the complete distribution of uncertain parameters may be unattainable, especially when there is not enough historical data (Ng, 2014; Perakis and Roels,

2008). Therefore, we first study a stochastic single-product DLBP with limited information of task processing time.

With the increasing demand for customized products, the scale of EOL products and their variants entering the recycling streams is rapidly expanding. Thus, the traditional single-product disassembly line is inappropriate and uneconomical to disassemble such increasing EOL product variants, which highlights the necessity to design multi-product disassembly lines in practical disassembly issues (Paksoy et al., 2013). While, approximately 96% of related studies concentrate on the single-product DLBP (Özceylan et al., 2019), and multi-product DLBP is rarely studied in the literature, especially in the context of considering uncertainties. To the best of our knowledge, there is no work that investigates the stochastic multi-product DLBP with only knowing partial distribution information. Hence, a new multi-product DLBP considering stochastic task times is investigated in this thesis.

An efficient reverse supply chain (RSC) is important for managing the increasing EOL products, which includes a series of activities, such as collection, recycling, refurbishing, and disassembling, etc (Özceylan and Paksoy, 2014). A RSC is a complex system and its management contains strategic-level decisions relating to the RSC design (e.g., the number and locations of facilities, disassembly equipment procurement), and tactical-level decisions associated with disassembly, transportation and inventory (e.g., disassembly line balancing, transportation amounts, inventory levels) (Yolmeh and Saif, 2021). Most researchers consider RSC design and DLBP separately. In practice, strategic decisions regarding RSC design have a long-term effect on tactical decisions related to disassembly line balancing, while the performance of disassembly lines will in turn influence the profitability of the RSC. Investigating the DLBP related RSC design in an integrated manner may improve the performance of the whole RSC (Özceylan et al., 2014). Moreover, existing studies on DLBP related RSC design focus on single product and deterministic settings. Therefore, addressing a multi-product DLBP related RSC design with considering uncertain features of EOL products is necessary.

1.2 Contents and Contributions

Based on the above observations, this thesis studies the following three novel works:

- **Chapter 3:** Single-product disassembly line balancing problem.

  To support the decision-making for stochastic DLBP under partial information, this study investigates a new stochastic DLBP with partial information of task processing time distribution, i.e., only the mean, upper and lower bounds are known. The problem aims to select a disassembly scheme for the EOL product, and determine the workstations to be opened, and assign the selected tasks to opened workstations. The objective is to minimize the total cost. For the studied problem, a joint chance-constrained model is formulated. Based on problem properties, the chance-constrained model is approximately transformed into a distribution-free model, and then a second-order cone program approximation-based formulation is developed. Experimental results demonstrate the effectiveness and efficiency of the proposed formulation.

- **Chapter 4:** Multi-product disassembly line balancing problem.

  To design an effective disassembly line for multiple EOL products, a novel multi-product DLBP with identical parts and uncertain task processing times is first addressed. The objective is to minimize the total cost. A new joint chance-constrained model is established for the problem, and then approximately converted into a distribution-free model. Two valid inequalities are constructed to reduce the solution space. To solve the large-scale problems, a lifted cut-and-solve method is proposed. Numerical experiments are conducted on benchmark and randomly generated instances to evaluate the performances of the proposed model, valid inequalities and solution method.

- **Chapter 5:** Multi-product disassembly line balancing related reverse supply chain design.

  To efficiently manage multiple EOL products under uncertainty, this chapter studies a new multi-product disassembly line balancing related RSC design, where EOL product supply, component demand, and task processing times are assumed to be uncertain. The problem jointly determines the number and locations of disassembly plants, disassembly machine

procurement, workstations to be opened, machine and tasks assignments, and inventory levels of EOL products and components. The objectives are to maximize the expected profit and minimize CO_2 emissions, simultaneously. A bi-objective two-stage stochastic programming model is formulated for the problem, and then approximately transformed to a linear distribution-free model. An exact ϵ-constrained method is proposed to solve the bi-objective problem. Moreover, an improved Benders decomposition is designed to solve each transformed single-objective problem efficiently. Numerical experiments are conducted to evaluate the performance of proposed methods, and some managerial insights are drawn based on sensitivity analysis.

1.3 Organization of the Thesis

The structure of this thesis is presented in the following Figure 1.3.

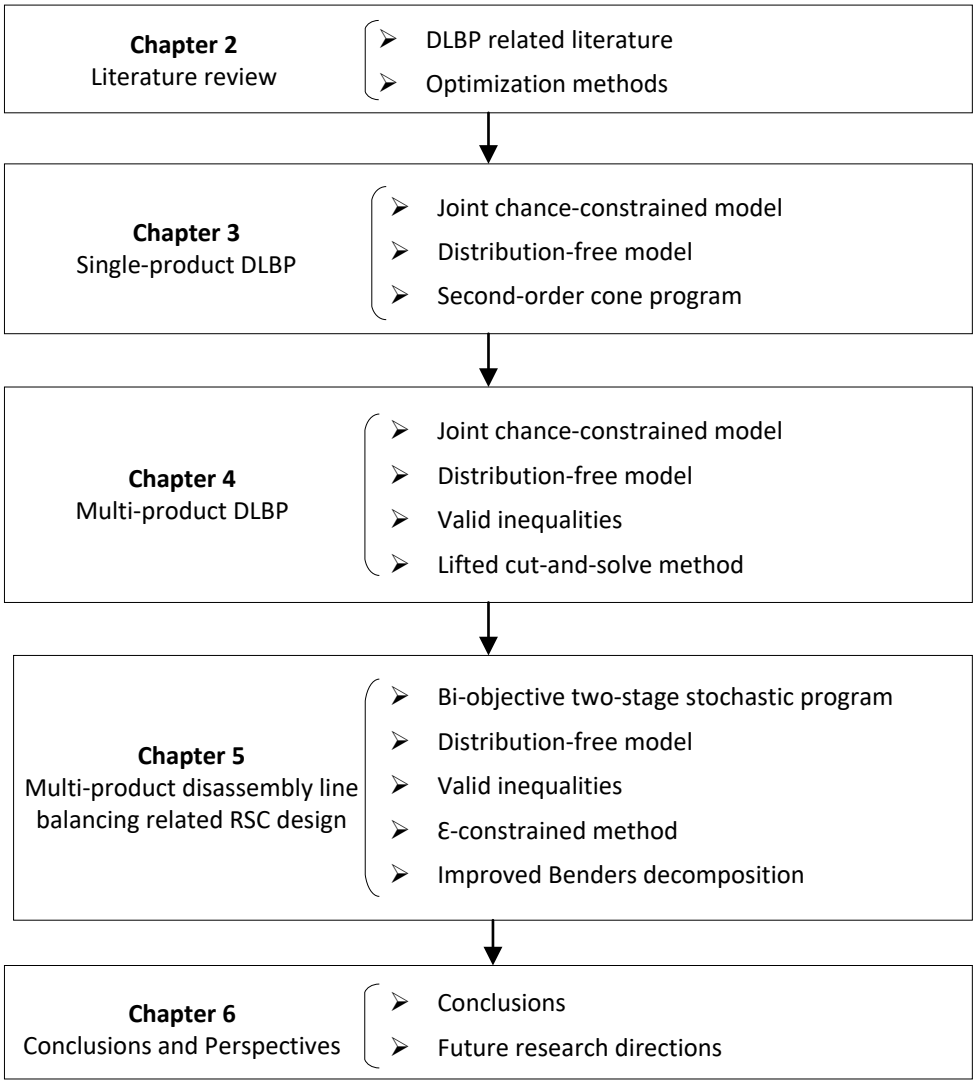


Figure 1.3: The structure of this thesis

Chapter 2

Literature Review

Contents

In this chapter, we first review DLBP related studies in Section 2.1. Then, the solution methods are reviewed in Section 2.2. Finally, Section 2.3 concludes this chapter.

2.1 DLBP Related Literature

DLBP has been widely studied since it was first introduced by Gungor and Gupta (1999). Figure 2.1 illustrates the categories of DLBPs according to problem features and properties, in which the gray parts are studied in this thesis. In the following, we review the related studies from three aspects: single-product DLBP, multi-product DLBP, and integrated DLBP.

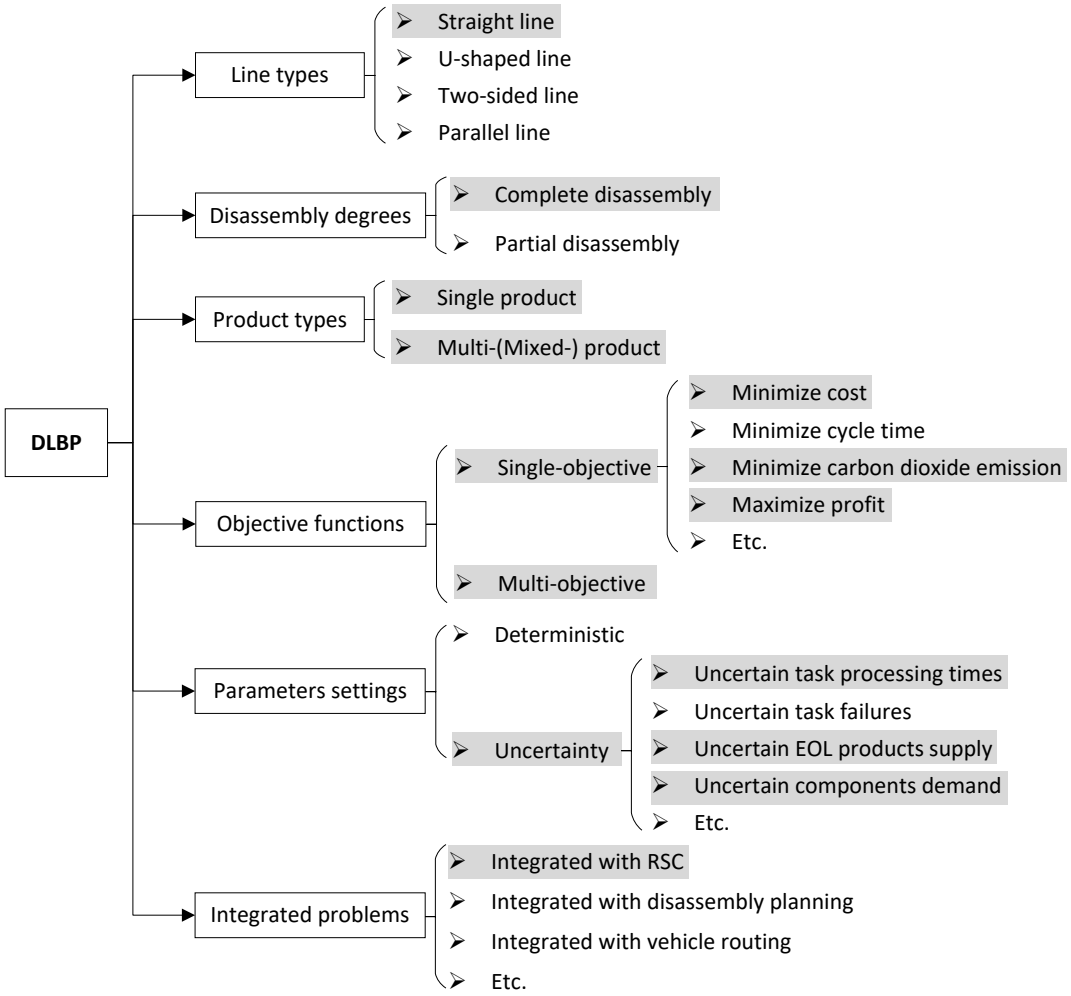


Figure 2.1: The categories of DLBPs

2.1.1 Single-product DLBP

Single-product DLBP refers to balancing the disassembly line for a single type of EOL product. Based on the existence of uncertain factors, the single-product DLBP can be classified into two main categories, i.e., deterministic and uncertain DLBPs.

For the deterministic single-product DLBP, most studies concentrate on straight line and complete disassembly (McGovern and Gupta, 2007b; Koc et al., 2009; Li et al., 2019a, 2020; Y2lmaz et al., 2022). Besides the straight line, many other types of disassembly lines have also been investigated, such as the U-shape disassembly line (Avikal et al., 2013), two-sided disassembly line (Kucukkoc, 2020; Zhang et al., 2022), parallel disassembly line (Hezer and Kara, 2015; Wang et al., 2021b). Recently, some researchers have addressed the partial DLBP, where not all components are dismantled and separated (Ren et al., 2017; Wang et al., 2019; Zhu et al., 2020; Liang et al., 2023).

For the uncertain single-product DLBP, according to Özceylan et al. (2019), most related studies assume that the probability distribution of task processing time is known (Aydemir-Karadag and Turkbey, 2013; Bentaha et al., 2014, 2015a, 2018; Goksoy Kalaycilar et al., 2022). However, the complete distribution of uncertain parameters may be unattainable in real applications, especially when there is not enough historical data (Ng, 2014; Perakis and Roels, 2008). Recently, some researchers have studied the stochastic DLBP with limited information of task processing time. Zheng et al. (2018) is the first to investigate stochastic DLBP with limited distributional information, where the mean, standard deviation and upper bound of task processing times are known. For the problem, a distribution-free model is formulated and a fast algorithm based on problem analysis is proposed. Later, He et al. (2019) extend the above problem by simultaneously optimizing the cost and cycle time, and an improved ε-constraint method is designed to solve the bi-objective problem. He et al. (2020a) study a bi-objective stochastic DLBP with the same limited distributional information of task times. The objectives are to minimize the total cost and contaminant emission, simultaneously. Liu et al. (2019) develop a new robust optimization model with partial information of task processing time, i.e., given mean and covariance matrix. The objective is to minimize the total cost. Then, a two-stage parameter-adjusting heuristic is devised to

solve the problem. Liu et al. (2020b) investigate a two-stage robust stochastic DLBP with the given mean and standard deviation of task processing time. The objective is to minimize the expected system cost in the worst case. A cutting-plane algorithm is proposed for the problem. It can be concluded that the existing single-product DLBP with partial information is studied recently, and all of them assume that the variance, covariance or standard deviation of task processing time is known. However, in some practical cases, there is not enough data to obtain the variance or covariance (Perakis and Roels, 2008). This motivates us to investigate the stochastic DLBP with partial probability distribution of task processing time, i.e., only the mean, upper and lower bounds are known (Chapter 3).

2.1.2 Multi-product DLBP

In this subsection, we review the multi-product DLBP related studies, which are also classified into two categories, deterministic and uncertain.

For deterministic multi-product DLBP, Ilgin et al. (2017) examine a multi-objective mixed-model DLBP in which the same parts of different EOL products are disassembled by so-called common tasks with processing times that may be different between EOL products. Their study assumes that the disassembly schemes of EOL products are predetermined. A mixed-integer linear programming model is proposed for the problem. Then a linear physical programming method is proposed to solve their problem. Fang et al. (2019) investigate a robotic mixed-model DLBP with parallel robots in each workstation to simultaneously minimize four objectives, the cycle time, total energy consumption, peak workstation energy consumption and the number of robots used. Identical parts of multiple EOL products are not considered in their study. A mixed-integer linear program is proposed to formulate the studied problem, and a knowledge-leveraging evolutionary algorithm is designed to solve it. Moreover, they extend their study by considering energy resource constraints (Fang et al., 2020b). A mixed-integer linear programming model is constructed for the problem, and an ε-constraint method based NSGA-II algorithm is proposed to solve it. Mutlu and Güner (2021) study a two-sided mixed-model DLBP with the objectives of minimizing the number of workstations and minimizing the selected task processing

times. For the problem, a mixed-integer linear programming model is formulated, and a memetic algorithm is designed.

For the uncertain multi-product DLBP, Altekin et al. (2008) study a stochastic multi-objective mixed-model U-shaped DLBP where the task processing time follows a normal distribution. The authors assume that the EOL products have no identical parts and the disassembly schemes are predetermined. A non-linear stochastic programming model is formulated for the problem. A collaborative ant colony optimization algorithm is proposed to solve the problem. Paksoy et al. (2013) investigate a multi-objective mixed-model DLBP with fuzzy objectives. For the problem, a mixed-integer linear programming model is proposed and solved using the commercial solver LINGO 11.0. Fang et al. (2020a) recently investigate a stochastic multi-objective mixed-model DLBP with interval task processing times, in which EOL products are disassembled by parallel robots in each workstation. A mixed-integer linear program is proposed for the problem and an evolutionary simulated annealing algorithm is developed to solve the problem. Liu et al. (2022) study a stochastic multi-product DLBP considering workforce assignment. The uncertain task processing time is represented by partial information of the probability distribution, i.e., the mean and covariance matrix. Each workstation in their study can only handle one type of EOL product. A stochastic program with conditional value-at-risk constraints is developed, and an exact cutting-plane method is proposed for the problem. Yin et al. (2022) recently study a multi-product partial DLBP with parallel robots in each workstation, where the uncertain task processing times are described by interval numbers. The disassembly schemes of EOL products are predetermined. A mixed-integer programming model is established for the studied problem. Then, a multi-objective hybrid driving algorithm is proposed to solve the problem.

From the above review on multi-product DLBP, it can be concluded that only a few studies investigate multi-product DLBP under uncertainty, especially with partial information of uncertain parameters. Thus, Chapter 4 addresses a stochastic multi-product DLBP, where uncertain task processing times and identical parts are considered.

2.1.3 Integrated DLBP

In this subsection, we briefly review related integrated DLBP studies, which can be categorized based on the decision level, i.e., operational, tactical, and strategic decision-making.

For operational and tactical decision-making, some researchers study the integrated disassembly line balancing and sequencing problem (Liu et al., 2020a; Edis et al., 2022; Zeng et al., 2022), in which decisions regarding both the disassembly line balancing and task sequencing are jointly made. The DLBP is also integrated with the collect routing problem recently (Habibi et al., 2017, 2019; Diri Kenger et al., 2020; Kenger et al., 2021; Liu et al., 2021), where the decisions on line balancing and routing are made together. More recently, He et al. (2022) investigate an integrated disassembly line balancing and planning problem with considering the machine specificity. Previous studies are pioneer works of the coordination disassembly line balancing with upstream and downstream flows of RSC. Furthermore, integrating the disassembly line balancing and the whole RSC may have a great impact on supply chain performance. Özceylan and Paksoy (2013) is the first to study the integrated RSC design and DLBP, which consists of determining the transportation of EOL products and components and balancing the disassembly line. The objective is to minimize the total cost. For the problem, a mixed-integer nonlinear programming model is constructed, and an illustrative instance is solved by calling GAMS/BARON. Özceylan and Paksoy (2014) investigate a similar integrated RSC design and DLBP to minimize the total cost, where the capacities of plants and customer demands are assumed to be fuzzy. For the problem, a fuzzy model is proposed and solved by calling GAMS/BARON. Özceylan et al. (2014) address a joint closed-loop supply chain (CLSC) design and DLBP to minimize the total cost. The problem is to determine the amounts of collected and dismantled EOL products, transportation, and balance the disassembly line. The problem is formulated as a mixed-integer nonlinear programming model and solved by calling GAMS-COIN/BONMIN.

For strategic decision-making, Budak (2020) investigate a deterministic integrated RSC design and DLBP to minimize the total cost, minimize the total emissions, and maximize the social impact, simultaneously. The problem determines the locations of storage centers, disassembly plants, and recycling centers, transportation of EOL products and components, inventory quan-

tities of EOL products, and balances the disassembly line. For the problem, a mixed-integer non-linear programming model is formulated, and an improved augmented ε-constraint method is proposed. Yolmeh and Saif (2021) study an integrated CLSC design and DLBP for a single-product, in which customer demand and quantity of EOL products are assumed to be uncertain and characterized by scenarios. The problem aims to determine the locations of production, assembly, remanufacturing and disassembly plants, transportation of components and products, and balance the assembly and disassembly lines. The objective is to minimize the total cost. The problem is formulated as a two-stage stochastic programming model, which is equivalently decomposed into a series of line balancing problems and a CLSC optimization problem, and then solved by CPLEX.

To sum up, we can conclude that 1) there is no work investigating multi-product integrated disassembly line balancing related RSC problems; 2) no study jointly considers uncertain supply, demand and task times. Therefore, Chapter 5 investigates a novel integrated multi-product RSC design and DLBP under uncertain supply, demand and task times.

2.2 Solution Methods

In this section, we review the optimization under uncertainty, multi-objective optimization, and single-objective optimization, which provides the foundation for developing effective methods for our problem.

2.2.1 Optimization under uncertainty

There are various optimization methods to deal with uncertainties in the disassembly, such as fuzzy programming (Paksoy et al., 2013; Özceylan and Paksoy, 2014; Zhang et al., 2017), robust programming (Xiao et al., 2021), distributionally robust programming (Liu et al., 2019, 2022, 2021), stochastic programming (Agrawal et al., 2008; Bentaha et al., 2015b; Zheng et al., 2018; Goksoy Kalaycilar et al., 2022), etc. In this thesis, we adopt stochastic programming to solve our problems, because it can efficiently transform uncertain problems into deterministic problems

by characterizing uncertainties using random scenarios or probability distributions. Thus, we review stochastic programming through formulations and solution methods in the following.

(1) Formulations for stochastic programming

Considering such an uncertain optimization problem with two stages of decisions, where the first stage decisions have to be decided before the realization of uncertain events, and the second stage decisions are made subsequently based on the results of first stage decisions (Ruszczyński and Shapiro, 2003). According to Birge and Louveaux (2011), the general formulation of two-stage stochastic programming can be written as:

$$\min \quad z = \boldsymbol{c}^T \boldsymbol{x} + \boldsymbol{E}_\xi \left[\min \boldsymbol{q}(s)^T \boldsymbol{y}(s) \right] \tag{2.1}$$

$$s.t. \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{2.2}$$

$$\boldsymbol{T}(s)\boldsymbol{x} + \boldsymbol{W}\boldsymbol{y}(s) = \boldsymbol{h}(s) \tag{2.3}$$

$$\boldsymbol{x} \geq 0, \boldsymbol{y}(s) \geq 0 \tag{2.4}$$

where vector $\boldsymbol{x}$ denotes the first stage decisions, $\boldsymbol{c}$ and $\boldsymbol{b}$ are known vectors, $\boldsymbol{W}$ is a known matrix, and vector $\boldsymbol{y}(s)$ represents the second stage decisions. For a given realization random events $s \in S$, the associate second stage coefficients $\boldsymbol{q}(s)$, $\boldsymbol{T}(s)$, and $\boldsymbol{h}(s)$ are known. $\boldsymbol{E}_\xi$ denotes the mathematical expectation.

Another widely studied stochastic programming formulation is chance-constrained programming (Charnes and Cooper, 1959). In chance-constrained programming, constraints on random variables should be maintained at prescribed probability levels. Consider the general form of the chance constraint as follows:

$$Pr\left(A(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\right) \geq 1 - \alpha \tag{2.5}$$

where $\alpha \in [0, 1]$, $\boldsymbol{\xi}$ denotes a random vector, and this constraint means that the probability of satisfying $A(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$ is larger than or equal to $1 - \alpha$.

Usually, the chance-constrained formulation is computationally intractable. One promising way is to transform the chance constraint into a solvable one based on the information of random variables (Nemirovski and Shapiro, 2007). Therefore, we will briefly review related solutions methods for stochastic programming in the following subsection.

(2) Solution methods for stochastic programming

In this subsection, we will review the solution methods for stochastic programming via three categories, i.e., scenario-based method, distribution-based method, and partial distribution information-based method.

The scenario-based method approximates the random parameters by a set of scenarios, especially when the distribution of random parameters is not continuous. One of the most used approximation methods is the sample average approximation (SAA), which is based on Monte Carlo sampling techniques (Kleywegt et al., 2002). The SAA generates a set of random samples of realizations of the random parameters and approximates the random parameters by sample average values (Nemirovski and Shapiro, 2007). The main advantages of the scenario-based method are its ease of implementation and no restriction on the distributions of random parameters. The solution quality highly depends on the number of samples, while a large number of samples will sharply increase the computational burden.

The distribution-based method usually assumes that the exact probability distributions of uncertain parameters are known, and then the stochastic problems are transformed into deterministic ones based on the probability density functions of uncertain parameters (Ruszczyński and Shapiro, 2003). Many researchers have adopted the distribution-based method to solve stochastic disassembly related problems (Bentaha et al., 2015b; Liu and Zhang, 2018; Yolmeh and Saif, 2021). However, the probability distributions of random parameters may be challenging to obtain or can not be precisely defined in real applications.

The partial distribution information-based method aims to solve stochastic programming, where only limited information of the probability distribution of the uncertain parameter is known. Ng (2014) and Perakis and Roels (2008) point out that the complete distribution of uncertain pa-

rameters may be unattainable when there is not enough historical data. Therefore, it is necessary to develop efficient approaches for stochastic DLBP with limited information of task processing time. So far, only a few studies have studied stochastic DLBP under partial distribution information (Zheng et al., 2018; He et al., 2019; Liu et al., 2022).

2.2.2 Multi-objective optimization

We first give the general definition for multi-objective optimization problem, then introduce several common solution methods.

(1) Notations for multi-objective optimization

Without loss of generality, consider such a minimization multi-objective optimization problem (MOOP), which can be written as follows:

$$\textbf{MOOP:} \quad \min \ (F_1(\boldsymbol{x}), F_2(\boldsymbol{x}), ..., F_N(\boldsymbol{x})) \tag{2.6}$$

$$s.t. \quad \boldsymbol{x} \in \Psi \tag{2.7}$$

where $\boldsymbol{x}$ is the decision vector, Ψ denotes the solution space, F_n represents the nth objective function, and N is the number of objective functions.

Different from the single-objective optimization problem with one optimal value, MOOP always involves more than one optimal solution, and these objectives often conflict with each other (Collette and Siarry, 2004). The optimal solutions of MOOP are called non-dominated solutions or Pareto optimal solutions, which can be defined as follows:

Definition 2.2.1. *A solution $\boldsymbol{x}^* \in \Psi$ is a non-dominated solution, if there does not exist any other solution $\boldsymbol{x} \in \Psi$ such that $F_n(\boldsymbol{x}) \leq F_n(\boldsymbol{x}^*)$, and $F_n(\boldsymbol{x}) < F_n(\boldsymbol{x}^*)$ for at least one objective function, where $n = \{1, 2, ..., N\}$.*

All the non-dominated solutions form the Pareto front that can reflect the trade-off between different objectives. Figure 2.2 illustrates the Pareto front of a bi-objective optimization problem,

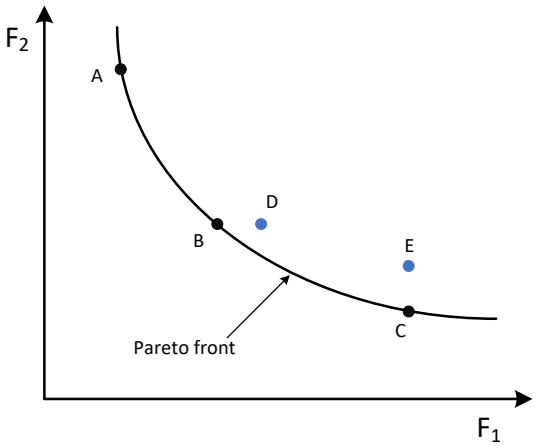


Figure 2.2: An illustrative example of the Pareto front

in which points A, B and C denote the Pareto optimal solutions, and points D and E denote the dominated solutions that are dominated by points A, B and C.

(2) Solution methods for multi-objective optimization

There are various methods for MOOP, such as weighted sum method (Marler and Arora, 2010), multi-objective evolutionary algorithm (Coello and Lamont, 2004), ε-constrained method (Ehrgott and Gandibleux, 2002), etc.

The weighted sum method is a priori method, which assigns a weight to each objective, thus converting a multi-objective problem into a single-objective one for resolution (Zadeh, 1963).

$$\begin{aligned} \min \quad & \sum_{n=1}^{N} \omega_n F_n(\boldsymbol{x}) \\ s.t. \quad & \boldsymbol{x} \in \Psi \end{aligned}$$

where ω_n denotes the weight of objective function $F_n(\boldsymbol{x})$, and usually $\omega_n \geq 0$ and $\sum_{n=1}^{N} \omega_n = 1$.

However, the weights assigned to these objectives depend on the preferences of decision-makers and are determined in advance. Only one Pareto optimal solution can be obtained under one specific combination of weights, which may not reflect the real Pareto front.

For many multi-objective optimization problems, it is difficult or inefficient to obtain the exact Pareto front. Consequently, many researchers develop multi-objective evolutionary algorithms

to obtain an approximation Pareto front (Von Lücken et al., 2014). The multi-objective evolutionary algorithm is an iterative search method that begins with a set of initial solutions (population), and iteratively improves the solution by various evolutionary operations. There are various multi-objective evolutionary algorithms, such as multi-objective genetic algorithm (Deb, 1999; Wang et al., 2021c), non-dominated sorting genetic algorithm (Deb et al., 2000; Bensmaine et al., 2013), strength Pareto evolutionary algorithm (Zitzler et al., 2001; Jiang and Yang, 2017), etc. Although multi-objective evolutionary algorithms may obtain the Pareto solutions fast, they can not guarantee the quality of the obtained solutions.

The ε-constrained method transforms the multi-objective problem into the single-objective problem by setting one objective function as the principal objective and setting other objective functions to the ε-constraints, and then iteratively solving the single-objective problem by varying the value of ε (Ehrgott and Gandibleux, 2002; Laumanns et al., 2006). The following bi-objective optimization problem (BOOP) illustrates the process of the ε-constrained method.

$$\begin{aligned} \textbf{BOOP:} \quad & \min \ (F_1(\boldsymbol{x}), F_2(\boldsymbol{x})) \\ s.t. \quad & \boldsymbol{x} \in \Psi \end{aligned} \tag{2.8}$$

where $F_1(\boldsymbol{x})$ and $F_2(\boldsymbol{x})$ are objective functions, and $\boldsymbol{x}$ is the decision variable whose solution space is Ψ.

Consider that $F_1(\boldsymbol{x})$ is set as the principal objective, and the BOOP can be transformed to the following single-objective optimization problem (SOOP) by introducing a new parameter ε.

$$\begin{aligned} \textbf{SOOP:} \quad & \min F_1(\boldsymbol{x}) \\ s.t. \quad & \boldsymbol{x} \in \Psi \\ & F_2(\boldsymbol{x}) \leq \varepsilon \end{aligned} \tag{2.9}$$

To solve SOOP, we need to predetermine the interval of ε, which can be obtained by solving

the following four single-objective models.

$$
\begin{aligned}
\mathbf{P}^{\mathbf{I}}_{\mathbf{F_1}}: \quad & F_1^I = \min F_1(\boldsymbol{x}) \\
& s.t. \quad \boldsymbol{x} \in \Psi
\end{aligned}
\qquad\qquad
\begin{aligned}
\mathbf{P}^{\mathbf{I}}_{\mathbf{F_2}}: \quad & F_2^I = \min F_2(\boldsymbol{x}) \\
& s.t. \quad \boldsymbol{x} \in \Psi
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{P}^{\mathbf{N}}_{\mathbf{F_1}}: \quad & F_1^N = \min F_1(\boldsymbol{x}) \\
& s.t. \quad \boldsymbol{x} \in \Psi \\
& \quad F_2 = F_2^I
\end{aligned}
\qquad\qquad
\begin{aligned}
\mathbf{P}^{\mathbf{N}}_{\mathbf{F_2}}: \quad & F_2^N = \min F_2(\boldsymbol{x}) \\
& s.t. \quad \boldsymbol{x} \in \Psi \\
& \quad F_1 = F_1^I
\end{aligned}
$$

Thus, the interval of ε is $[{F_2}^I, {F_2}^N]$. Then, the Pareto optimal solutions of BOOP can be obtained by solving a series of SOOPs, where ε is varying in the interval $[{F_2}^I, {F_2}^N]$ with a proper step size.

The ε-constrained method is widely applied to solve multi-objective optimization problems, because it can obtain the exact Pareto front. While it should be careful to choose the principle objective and determine the step size, which may significantly impact the solving efficiency.

2.2.3 Single-objective optimization

In this subsection, we review solution methods for single-objective optimization problems, which can be classified into approximated and exact methods.

(1) Approximated method

Approximated methods are designed to find a feasible solution for the problem, while the solution can not be proven optimal. The commonly approximated methods include constructive heuristic, meta-heuristic, matheuristic, etc.

Heuristic is an algorithm that can quickly find the near-optimal solution for the problem, and usually relies on problem-specific rules, experience, and search strategies (Lin and Kernighan, 1973). For the disassembly line balancing related problems, various heuristics have been applied, such as greedy algorithm (McGovern and Gupta, 2004, 2005), local search (McGovern and Gupta, 2007a; Li et al., 2019b; Tian et al., 2023), etc. Heuristics have the advantage of being easy to

implement and capable of quickly finding feasible solutions. While, the optimality and quality of solutions obtained by heuristics can not be guaranteed.

Meta-heuristic is a more general algorithm, usually inspired by natural phenomena and population based, which can adapt to various problems (Osman and Kelly, 1997; Hussain et al., 2019). Many researchers have developed different kinds of meta-heuristics to solve the disassembly line balancing related problems, such as genetic algorithm (McGovern and Gupta, 2007b; Kalayci et al., 2016; Kucukkoc, 2020), simulated annealing algorithm (Fang et al., 2020a; Wang et al., 2021a), ant colony algorithm (Agrawal et al., 2008; Kalayci and Gupta, 2013a; Çil et al., 2020), particle swarm algorithm (Kalayci and Gupta, 2013c; Çil et al., 2020), artificial bee colony algorithm (Kalayci and Gupta, 2013b; Kalayci et al., 2015; Guo et al., 2023), etc. In particular, meta-heuristic is often designed to find near-optimal or good solutions to complex problems in a reasonable amount of time, such as NP-hard problems and non-linear problems. However, the convergence speed of In particular may be slow, especially for the problems with complex and high-dimensional search spaces.

Matheuristic, short for mathematical heuristics, is a hybrid optimization method that combines mathematical programming techniques and heuristic (Boschetti et al., 2009). There are various matheuristics studied in the literature. For the fix-and-optimize matheuristic, a feasible solution is obtained by a mathematical programming model, and heuristics are used to improve the current feasible solution (Lindahl et al., 2018; Friske et al., 2022). In the decomposition matheuristic, the original problem is decomposed into small sub-problems, and some sub-problems are solved by mathematical programming (Doi et al., 2018; Chitsaz et al., 2019). Matheuristic requires a specific design depending on the studied problem, and there is a trade-off between optimality and computational efficiency.

(2) Exact method

The exact method aims to solve optimization problems to find global optimal solutions. There are several types of exact methods commonly used for optimization problems. Branch and Bound (B&B) is essentially an enumeration method that divides the original problem into sub-problems

(branches), explores these branches using a tree-like search strategy, and prunes these branches that cannot lead to a better solution than the best one found so far (Lawler and Wood, 1966). Branch and Cut (B&C) combines B&B and the cutting-plane method, where cutting-planes (i.e., inequalities) are added to strengthen the relaxation of the mixed-integer programming model, thus improving the efficiency of the search (Tawarmalani and Sahinidis, 2005). Branch and Price (B&P) incorporates B&B and columns generation approach, in which columns (decision variables) are iteratively added at each node of the searching tree (Barnhart et al., 1998). In this thesis, we apply the cut-and-solve and Benders decomposition methods. Consequently, a detailed descriptions of these two methods are introduced in the following.

Cut-and-solve method

The cut-and-solve (CS) method is first introduced by Climer and Zhang (2006) to solve the asymmetric traveling salesman problem, and then has been successfully used to solve many combinatorial optimization problems, such as facility location (Yang et al., 2012; Gadegaard et al., 2018), and lane reservation (Fang et al., 2013; Wu et al., 2017), etc. CS method is a particular branch and bound algorithm, where only two sub-problems are considered at each node, and only the best solution is memorised. Thus, the size of the search tree and the memory required can be reduced.

In the CS method, the original problem is decomposed into two problems, namely Sparse Problem (SP) and Dense Problem (DP). At the $(i-1)$-th iteration, a Piercing Cut (PC_{i-1}) is constructed based on the linear relaxed solution of (DP_{i-1}). At the i-th iteration, PC_{i-1} cuts the solution space of DP_{i-1} into two sub-spaces that correspond to (SP_i) and a new (DP_i). SP_i generally has a small solution space and may be exactly solved to yield a new feasible solution of the original problem, and update the current best upper bound (UB) if possible. DP_i often has a large solution space, and its relaxed problem is generally solved to obtain a lower bound (LB) of the original problem. Especially, DP_0 denotes the original problem. The process continues until the current LB is greater than or equal to the best UB. Then, the solution of the best UB is output as an optimal solution of the original problem.

The convergence rate of the CS method mainly depends on the effectiveness of PC, the quality of the LB and the speed to solve SP. To enhance the performance of the CS method, in this thesis, we improve the traditional CS method through the following strategies: 1) a constructive heuristic is proposed to obtain an initial UB that may be an optimal solution of the original problem. 2) double PCs (PC^1_{i-1}, PC^2_{i-1}) based on partially linear relaxation are generated at each iteration to obtain a better LB. 3) SP_i is further divided into two sub-problems (SP^1_i, SP^2_i) by a second PC, leading to a fast resolution for SP_i.

Benders decomposition

Benders decomposition (BD) is proposed by Benders (1962) and has been widely studied and applied to solve various problems in the past few decades. In the BD procedure, the original problem is decomposed into two coupling problems, i.e., master problem (MP) and sub-problem (SP). MP contains partial constraints and variables of the original problem, which is essentially a relaxation of the original problem. Without loss of generality, considering a maximization problem, an upper bound (UB) can be obtained by optimally solving MP (if MP is feasible). The SP contains the remaining constraints and variables and is solved with the fixed MP solution. If the SP is solved, we can obtain a lower bound (LB) for the original problem, and then an optimality cut can be added to MP. Otherwise, a feasibility cut is generated and added to MP. The process iterates until the optimal solution is found or other termination criteria are met (Rahmaniani et al., 2017).

In the literature, there are various kinds of BD, such as logic-based BD (Hooker and Ottosson, 2003; Heching et al., 2019), generalized BD (Geoffrion, 1972; Gharaei et al., 2020), in which the SP contains integer decision variables or some non-linear constraints. The most studied one is the classical BD, where the integer decision variables and corresponding constraints are retained in MP, and SP only contains continuous decision variables. Thus, SP is a continuous linear sub-problem, and the dual of SP is solved to generate optimal or feasible cuts (Magnanti and Wong, 1981). Our studied problem in Chapter 5 is a stochastic two-stage problem, and the second-stage problem is a scenario-based linear problem. This motivates us to implement BD to solve

our problem.

The performance of BD highly depends on the quality of generated optimality cuts. To accelerate the convergence rate, we enhance the BD via four strategies: 1) add valid inequalities to MP to get a tighten UP; 2) partially decompose our problem, which means we retain partial scenario SPs in MP, which may lead to a high-quality UP; 3) divide scenarios into small groups, and each group generates a cut, this can reduce the size of MP; 4) additionally generate a Pareto optimality cut which improves the quality of cut.

2.3 Conclusions

In this Chapter, we first review DLBP related literature via three aspects, i.e., single-product, multi-product, and integrated problems. Then, related solution methods, including optimization methods under uncertainties, multi-objective optimization methods, and methods for single-objective optimization. We can find that many research issues have not been investigated, such as stochastic DLBPs with partial information of probability distribution, stochastic multi-product DLBP, and multi-product disassembly line balancing related RSC design. Therefore, three corresponding novel problems are studied in Chapters 3-5.

Chapter 3

Single-product Disassembly Line Balancing Problem

Contents

In the literature, the majority of studies focus on deterministic DLBP. Only a few researchers investigate uncertain DLBP. In addition, most of them assume that probability distributions of uncertain parameters are known. However, in real applications, due to various factors, such as the lack of historical data for EOL products, obtaining complete probability distribution information is often challenging. Consequently, in this chapter, we study a DLBP with partial information of task processing time.

The problem studied in this chapter aims to optimally choose a disassembly scheme for EOL product, determine the number of workstations to be opened, and assign disassembly tasks to the opened workstations. The objective is to minimize the disassembly cost. For the problem, a chance-constrained model is proposed. Based on problem property analysis, a new distribution-free formulation is constructed. Subsequently, a second-order cone program approximation-based formulation is developed for it. Experimental results on benchmark instances and new randomly generated instances demonstrate the effectiveness and efficiency of the proposed formulation.

The rest of this chapter is organized as follows. Section 3.1 states the studied problem and proposes a chance-constrained model. Section 3.2 discusses problem properties, and a distribution-free formulation is proposed. Section 3.3 develops a second-order cone program approximation-based formulation. Section 3.4 conducts the numerical experiments and analyses the computational results. Finally, Section 3.5 concludes this chapter.

3.1 Problem Statement and Chance-constrained Model

3.1.1 Problem statement

The studied DLBP investigates a single EOL product with uncertain task processing times, where only limited information of the probability distribution of task processing time is available, i.e., the mean, lower and upper bounds. Consider an EOL product with several alternative disassembly schemes, each containing a set of disassembly tasks with known precedence relationships. There is a set of available workstations. The problem aims to select a disassembly scheme for the EOL

product, determine the number of workstations to be opened and assign selected tasks to the opened workstations to minimize the total cost incurred in the disassembly line.

An example of Hand Light (Tang et al., 2002) is used to intuitively understand the studied DLBP. Figure 3.1 illustrates the disassembly scheme graph of the Hand Light, where three types of lines represent three different disassembly schemes. Let □ and ◯ represent a subassembly node and a disassembly task, respectively. The Hand Light contains 7 parts and N_0 refers to the initial state of the EOL product. The task is represented by R, which is indexed from 1 to 10. Specifically, tasks with a gray background represent handling hazardous parts, which requires an extra cost. The directions of arrows connecting subassemblies and tasks indicate the disassembly order and task precedence relationship. Disassembly scheme 1 represented by the bold line consists of tasks $\{R_2, R_5, R_6, R_8, R_9, R_{10}\}$ and subassemblies $\{N_0, N_2, N_3, N_5, N_6, N_7\}$. Scheme 2 represented by the thin line consists of tasks $\{R_1, R_3, R_6, R_7, R_9, R_{10}\}$ and subassemblies $\{N_0, N_1, N_3, N_4, N_6, N_7\}$. Scheme 3 represented by the dotted line consists of tasks $\{R_2, R_4, R_6, R_7, R_9, R_{10}\}$ and subassemblies $\{N_0, N_2, N_3, N_4, N_6, N_7\}$.

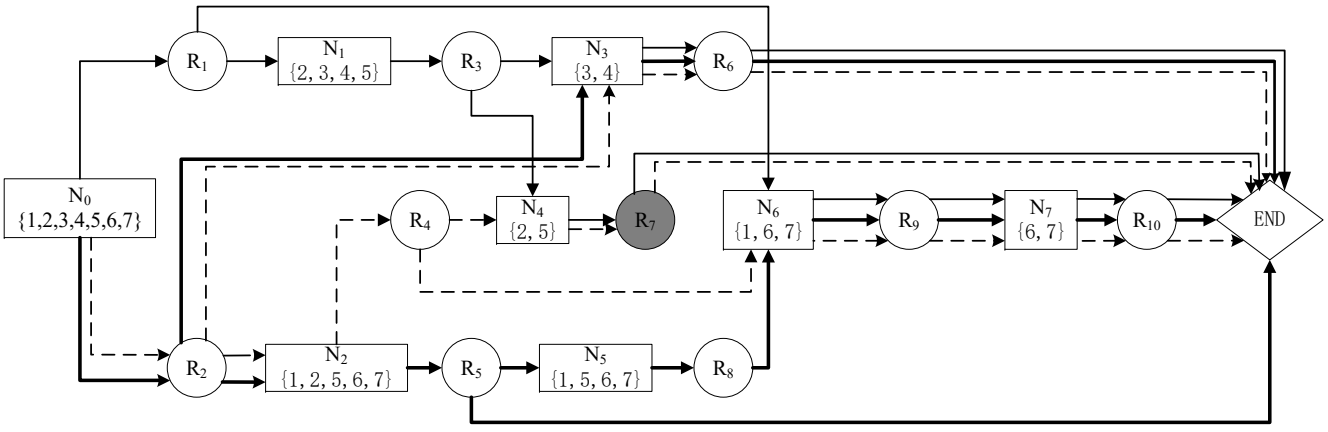


Figure 3.1: The disassembly scheme graph of the Hand Light

In order to well formulate the problem, the following assumptions are given:

- A single type of EOL product is completely disassembled on a straight disassembly line;
- An opened workstation can handle more than one task and each task cannot be assigned to more than one workstation, and workstations are independent;

- Task processing times are independent and can be represented by partial information of probability distribution, i.e. the mean, upper and lower bounds;
- The tasks precedence constraints have to be respected;
- Handling hazardous tasks incurs additional costs.

3.1.2 Chance-constrained model

Before the chance-constrained model, the notations are defined as follows:

Indices:

— r : index of disassembly tasks;

— w, w' : indices of workstations;

— n : index of subassembly nodes;

Parameters:

— R : set of disassembly tasks;

— W : set of workstations;

— N : set of subassembly nodes;

— P_n : set of immediate predecessors of subassembly n, where $n \in N$;

— Q_n : set of immediate successors of subassembly n, where $n \in N$;

— H : set of hazardous tasks;

— pt_r : processing time of task r, where $r \in R$;

— pd_r : the mean processing time of task r, where $r \in R$;

— pd_r^U : the upper bound of processing time of task r, where $r \in R$;

— pd_r^L : the lower bound of processing time of task r, where $r \in R$;

— CT : cycle time of disassembly line;

— α : the probability of task processing time exceeding the cycle time for all workstations;

— c_w : fixed cost to open a workstation;

— c_h : extra cost for handing a hazardous task;

Decision variables:

— x_{wr} : binary variable, equal 1 if task r is assigned to workstation w, 0 otherwise, where $r \in R$, $w \in W$;

— y_w : binary variable, equal 1 if workstation w is open, 0 otherwise, where $w \in W$;

— h_w : binary variable, equal 1 if a hazardous task is assigned to workstation w, 0 otherwise, where $w \in W$;

For the problem, a joint chance-constrained model (P1) is formulated as follows:

$$\textbf{P1:} \quad \min \left(c_w \sum_{w \in W} y_w + c_h \sum_{w \in W} h_w \right) \tag{3.1}$$

$$s.t. \quad \sum_{r \in Q_0} \sum_{w \in w} x_{wr} = 1 \tag{3.2}$$

$$\sum_{w \in w} x_{wr} \leq 1, \ \forall r \in R \tag{3.3}$$

$$\sum_{r \in Q_n} \sum_{w \in W} x_{wr} = \sum_{r \in P_n} \sum_{w \in w} x_{wr}, \ \forall n \in N \backslash \{0\} \tag{3.4}$$

$$\sum_{r \in Q_n} x_{w'r} \leq \sum_{r \in P_n} \sum_{w=1}^{w'} x_{wr}, \ \forall n \in N \backslash \{0\}, \forall w' \in W \tag{3.5}$$

$$x_{wr} \leq y_w, \ \forall r \in R, w \in W \tag{3.6}$$

$$\sum_{w \in W} w x_{wr} \leq \sum_{w \in W} y_w, \ \forall r \in R \tag{3.7}$$

$$x_{wr} \leq h_w, \ \forall w \in W, r \in H \tag{3.8}$$

$$Pr \left(\sum_{r \in R} pt_r x_{wr} \leq CT, \ \forall w \in W \right) \geq 1 - \alpha \tag{3.9}$$

$$x_{wr}, y_w, h_w \in \{0, 1\}, \ \forall r \in R, w \in W \tag{3.10}$$

The objective function (3.1) is to minimize the total cost of the disassembly line, consisting of two parts: (i) the fixed cost of opening workstations; (ii) the additional cost incurred by the tasks handing hazardous parts.

Constraint (3.2) guarantees that only one disassembly scheme is selected. Constraints (3.3) restrict that any tasks can be assigned to at most one workstation. Constraints (3.4) ensure the flux conservation of each subassembly and only one immediate successor task be selected for

each subassembly. Constraints (3.5) indicate the precedence constraint of tasks, i.e., the successor tasks of each subassembly should be assigned to the workstation with a larger or equal index to the preceding one. Constraints (3.6) imply the opened workstations. Constraints (3.7) ensure that all the workstations should be opened from small to large. Constraints (3.8) determine the workstation handing hazardous tasks. Constraint (3.9) represents that the task processing time for all workstations is no more than the cycle time with a possibility of at least $1-\alpha$. Constraints (3.10) give the bounds of decision variables.

Since the general DLBP has been proven to be NP-hard (McGovern and Gupta, 2007b,a), the studied stochastic DLBP is of course NP-hard. Moreover, the joint chance constraints (3.9) make the problem more challenging to solve. In the following, we first transform model P1 into a distribution-free model.

3.2 Problem Analysis and Distribution-free Formulation

In this section, we analyze the problem properties, and then approximately transform the chance-constrained model into a distribution-free model.

3.2.1 Problem analysis

Regarding the joint chance constraint (3.9), based on the assumption that workstations are independent of each other, we have:

$$Pr\left(\sum_{r\in R} pt_r x_{wr} \leq CT, \forall w \in W\right) = \prod_{w\in W} Pr\left(\sum_{r\in R} pt_r x_{wr} \leq CT\right) \tag{3.11}$$

Let β_w denote the individual possibility of workstation w, and the joint chance constraint can further be equivalently transferred into a set of individual chance constraints (Bentaha et al., 2015a; Cheng and Lisser, 2012).

$$Pr\left(\sum_{r\in R} pt_r x_{wr} \leq CT\right) \geq 1-\beta_w,\ \forall w \in W \tag{3.12}$$

where $(1-\beta_w)^{|W|} = 1-\alpha$.

Constraints (3.12) guarantee that the possibility of task processing times of workstation w not exceeding the cycle time is at least 1-β_w. Next, a conservative approximation of (3.12) is constructed by the following proposition.

Proposition 1. *The chance constraint* (3.12) *is equivalent to*

$$\min\left\{\sum_{r\in R} pd_r^U x_{wr}, \sum_{r\in R} pd_r x_{wr} + \sqrt{\frac{-\ln\beta_w \sum_{r\in R}((pd_r^U - pd_r^L)^2 x_{wr}^2)}{2}}\right\} \leq CT,\ \forall w \in W \tag{3.13}$$

where pd_r, pd_r^U, pd_r^L denote the mean, the upper bound, and the lower bound of processing time of task r, respectively.

Proof. By Hoeffding inequality (Hoeffding, 1994), for any independent random variables X_i bounded in $[a_i, b_i]$ with the mean value $E(\bar{X})$ and for any $t > 0$, the following formula holds:

$$P\left(\bar{X} - E(\bar{X}) \geq t\right) \leq \exp^{\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}\right)} \tag{3.14}$$

Accordingly, inequality (3.12) can be transformed into the following form:

$$Pr\left(\frac{\sum_{r\in R} pt_r x_{wr}}{n} - \frac{\sum_{r\in R} pd_r x_{wr}}{n} \geq \frac{CT - \sum_{r\in R} pd_r x_{wr}}{n}\right) \leq \beta_w, \forall w \in W \tag{3.15}$$

where $n = |R|$.

Note that $CT - \sum_{r\in R} pd_r x_{wr} > 0$ which implies by $\alpha < 50\%$ (α is a risk level and usually $\alpha \leq 10\%$) (Bentaha et al., 2015b). Hence, any x_{rw} satisfying

$$\beta_w \geq \exp^{\left(-\frac{2n^2(TI)^2}{\sum_{r\in R}\left(pd_r^U x_{wr} - pd_r^L x_{wr}\right)^2}\right)},\ \forall w \in W \tag{3.16}$$

is a feasible solution for (3.15), where $TI = \frac{CT - \sum_{r\in R} pd_r x_{wr}}{n},\ \forall w \in W$.

Naturally, (3.16) can be rewritten as follows:

$$\sum_{r\in R} pd_r x_{wr} + \sqrt{\frac{-\ln\beta_w \sum\limits_{r\in R}\left(pd_r^U x_{wr} - pd_r^L x_{wr}\right)^2}{2}} \le CT,\ \forall w \in W \tag{3.17}$$

Moreover, when $\sum\limits_{r\in R} pd_r x_{wr} + \sqrt{\frac{-\ln\beta_w \sum\limits_{r\in R}\left(pd_r^U x_{wr} - pd_r^L x_{wr}\right)^2}{2}} > \sum\limits_{r\in R} pd_r^U x_{wr}$, (3.17) can be replaced by the following expression to get a better solution.

$$\sum_{r\in R} pd_r^U x_{wr} \le CT,\ \forall w \in W \tag{3.18}$$

Hence, combining (3.16) and (3.18), the chance constraint (3.15) can be reformulated as follows:

$$\min\left\{\sum_{r\in R} pd_r^U x_{wr}, \sum_{r\in R} pd_r x_{wr} + \sqrt{\frac{-\ln\beta_w \sum\limits_{r\in R}(pd_r^U - pd_r^L)^2 x_{wr}^2}{2}}\right\} \le CT,\ \forall w \in W \tag{3.19}$$

The proof is completed. □

3.2.2 Distribution-free formulation

Based on the above analysis, model P1 can be transformed into the following distribution-free model P2 by replacing constraint (3.9) with constraints (3.19).

$$\textbf{P2:}\quad \min\left(c_w \sum_{w\in W} y_w + c_h \sum_{w\in W} h_w\right)$$
$$s.t.\quad (3.2) - (3.8), (3.10), (3.19)$$

Model P2 is still difficult because of constraints (3.19), and can only be solved by a non-linear solver, which is inefficient. For large-scale instances, it fails to provide a feasible solution within a reasonable amount of time. In the following, the distribution-free model P2 will be approximated

to a more efficient second-order cone programming (SOCP) for its resolution.

3.3 Second-order Cone Programming

The SOCP has been investigated in various studies (Altekin, 2017; Bentaha et al., 2015a,b). The key step of SOCP is to transfer the difficult-solve constraint into a certain type of quadratic constraint, which can be directly solved by commercial solvers.

Inspired by this, the non-linear constraint (3.19) is transferred. Firstly, we introduce a continuous decision variable S_w to denote the deviation of task procession times of workstation w. Then the non-linear part of inequality (3.19), i.e., (3.17) can be rewritten as follows:

$$\sum_{r \in R} pd_r x_{wr} + \sqrt{\frac{-\ln \beta_w}{2}} S_w \leq CT, \ \forall w \in W \tag{3.20}$$

$$\sum_{r \in R} (pd_r^U x_{wr} - pd_r^L x_{wr})^2 \leq S_w^2, \ \forall w \in W \tag{3.21}$$

$$S_w \geq 0, \ \forall w \in W \tag{3.22}$$

Notice that constraints (3.19) are $\min(A, B) \leq C$ structure. According to Altekin (2017), a binary decision variable U_w is introduced, where U_w takes a value of 1 if $A \leq B$, and 0 otherwise.

$$U_w = \begin{cases} 1 & \text{if } \sum_{r \in R} pt_r^U x_{wr} \leq \sum_{r \in R} pd_r x_{wr} + \sqrt{\frac{-\ln \beta_w}{2}} S_w, \\ 0 & \text{otherwise.} \end{cases} \tag{3.23}$$

Therefore, constraints (3.19) can be expressed as two independent inequalities, where M is a large positive number.

$$\sum_{r \in R} pt_r^U x_{wr} \leq CT + M(1 - U_w), \ \forall w \in W \tag{3.24}$$

$$\sum_{r \in R} pd_r x_{wr} + \sqrt{\frac{-\ln \beta_w}{2}} S_w \leq CT + MU_w, \ \forall w \in W \tag{3.25}$$

Then, the SOCP formulation (P3) can be formulated as:

$$\textbf{P3:} \quad \min \left(c_w \sum_{w \in W} y_w + c_h \sum_{w \in W} h_w \right)$$

$$s.t. \quad (3.2) - (3.8), (3.10), (3.20) - (3.22), (3.24) - (3.25)$$

Formulation P3 is a SOCP model with quadratic constraints, which can be directly solved by the commercial solver, such as CPLEX.

3.4 Numerical Experiments

In this section, 7 benchmark instances from the literature are firstly studied to verify the effectiveness of the proposed new formulation, then more randomly generated instances are further tested. Commercial solver CPLEX 12.8 is called in Visual Studio 2019 to solve the proposed formulation. All numerical experiments are implemented on a personal computer with Core I5 with 12GB RAM under Microsoft Windows 8.1 operating system.

3.4.1 Experiments on benchmark instances

The developed new model is firstly tested on 7 benchmark instances from previous literature Bentaha et al. (2012, 2013a); Koc et al. (2009); Lambert (1999); Ma et al. (2011); Tang et al. (2002), containing different kinds of EOL products. Denote each instance by the combination of the first letter of the author and year. Table 3.1 reports all the input data of instances, which are in line with Zheng et al. (2018). 25% of tasks are assumed hazardous for each case. The confidence level parameter α is set to 0.05 and the upper and lower bounds of task time are respectively $1.2pd_r$ and $0.8pd_r$, where pd_r is the mean task time. Columns $|R|$, $|N|$, $|W|$, CT represent the total number of tasks, the number of subassemblies, the number of available workstations and the cycle time, respectively.

Three different formulations, i.e., the known distribution model (Bentaha et al., 2015b) (B model), the distribution-free model (Zheng et al., 2018) (Z model), and the new proposed model

Table 3.1: Benchmark instances

No.	Instance	Product	$\|R\|$	$\|N\|$	$\|W\|$	CT
1	BBD13a	Compass	10	5	3	0.61
2	BBD13b	Piston and connecting rod	25	11	4	120
3	KSE09	Sample product	23	13	6	20
4	L99a	Radio set	30	18	9	50
5	L99b	Ball-point pen	20	13	9	10
6	MJKL11	Automatic pencil	37	22	10	40
7	TZC02	Hand light	10	7	6	90

P3 are tested and compared. Table 3.2 reports the results of studied instances under three models (B model, Z model and P3 model). We use the superscript i, where $i = 1, 2, 3$, to indicate the three models. The columns $|R^i|$, $|W^i|$, $|W_h^i|$, Obj^i represent the number of tasks for the selecting disassembly scheme, number of opened workstations, number of workstations for handling hazardous tasks, objective value, respectively. As shown in Table 3.2, the numbers of selected tasks and opened workstations are the same for the three models. However, the numbers of workstations handling hazardous tasks are different in some cases, which leads to different objective values.

Table 3.2: Experimental results of benchmark instances

Instance	B model				Distribution-free							
					Z model				P3 model			
	$\|R^1\|$	$\|W^1\|$	$\|W_h^1\|$	Obj^1	$\|R^2\|$	$\|W^2\|$	$\|W_h^2\|$	Obj^2	$\|R^3\|$	$\|W^3\|$	$\|W_h^3\|$	Obj^3
BBD13a	3	2	1	4.88	3	2	1	4.88	3	2	1	4.88
BBD13b	4	2	1	960	4	2	1	960	4	2	1	960
KSE09	6	3	2	260	6	3	2	260	6	3	2	260
L99a	9	3	2	650	9	3	1	550	9	3	1	550
L99b	9	3	1	110	9	3	1	110	9	3	1	110
MJKL11	7	3	2	455	7	3	1	385	7	3	1	385
TZC02	6	3	1	990	6	3	1	990	6	3	1	990

To facilitate the description of the solution, Table 3.3 illustrates the detailed solutions of three models, where the number '1' in the table means that the corresponding task is done and assigned to the corresponding workstation in the column. We can observe that for case TZC02, although the numbers of opened workstations are the same, the selected tasks are different. Specifically, B model and P3 model choose tasks $\{R_2, R_4, R_6, R_7, R_9, R_{10}\}$, but Z model selects

tasks $\{R_1, R_3, R_6, R_7, R_9, R_{10}\}$. The main reason for this result is that more than one disassembly scheme is meeting the confidence level. In addition, only one workstation is needed to handle hazardous tasks for Z model and P3 model in cases L99a and MJKL11, while B model needs two. So the total cost of the disassembly scheme under B model is higher because of the higher additional cost of handling hazardous tasks. It can be found that although with less information or data, the performance of the proposed P3 model is the same as Z model, and even better than B model in some cases.

Table 3.3: Experimental results of TZC02 instance

Task\workstation	B model					Z model					P3 model				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
R_1						1									
R_2	1										1				
R_3						1									
R_4	1										1				
R_5															
R_6			1				1						1		
R_7		1					1					1			
R_8															
R_9	1							1			1				
R_{10}		1						1				1			

3.4.2 Experiments on randomly generated instances

The random instances are generated using the method of Koc et al. (2009). Three parameters (a, t, N) determine the instance structure, where a denotes the number of artificial nodes (number of subassemblies) at each level, t denotes the number of tasks for each artificial node, and N denotes the number of parts of the EOL product. Then, the total number of artificial nodes A can be calculated by $A = a(N-2)+1$, and the total number of tasks B can be calculated by $B = a(t(N-3)+2)$. Referring to (Koc et al., 2009), we set $a = \{3, 5, 10\}$, $t = \{2, 3, 5\}$, and $N = \{8, 10, 12\}$. So the largest scale instance has 470 tasks and 101 artificial nodes, while the smallest scale instance has 36 tasks and 19 artificial nodes. Other input parameters keep the same with those in former experiments, i.e., $c_w = 3, c_h = 2$ and 25% of tasks are hazardous. Task

processing times are randomly generated from the interval $[10, 50]$. Besides, the cycle time is set as 80, and the maximal number of workstations is set as 10. To study the impact of the uncertain level, the upper (lower) bound and standard deviation parameters $(pb_i, E[pd_i^2])$ set as $(0.1, 0.01)$, $(0.15, 0.03)$ and $(0.2, 0.05)$, which is the same as Ng (2014) and He et al. (2019). Note that when the upper (lower) bound parameter takes the value of pd_i, the corresponding upper and lower bounds of task processing time are $(1 + pd_i)pd_r$ and $(1 - pd_i)pd_r$, respectively. Computational results are reported in Tables 3.4 - 3.6.

Table 3.4: Experimental results with low uncertain level $(0.1,\ 0.01)$

Set	(a, t, N)	A	B	B model		Z model		P3 model	
				Obj^1	(W^1, W_h^1)	Obj^2	(W^2, W_h^2)	Obj^3	(W^3, W_h^3)
1	$(3, 2, 8)$	19	36	640	$(2, 1)$	640	$(2, 1)$	640	$(2, 1)$
2	$(3, 3, 8)$	19	51	640	$(2, 1)$	480	$(2, 0)$	480	$(2, 0)$
3	$(3, 5, 8)$	19	81	720	$(3, 0)$	720	$(3, 0)$	720	$(3, 0)$
4	$(5, 2, 8)$	31	60	720	$(3, 0)$	720	$(3, 0)$	720	$(3, 0)$
5	$(5, 3, 8)$	31	85	720	$(3, 0)$	720	$(3, 0)$	720	$(3, 0)$
6	$(5, 5, 8)$	31	135	480	$(2, 0)$	480	$(2, 0)$	480	$(2, 0)$
7	$(10, 2, 8)$	61	120	640	$(2, 1)$	640	$(2, 1)$	640	$(2, 1)$
8	$(10, 3, 8)$	61	170	640	$(2, 1)$	640	$(2, 1)$	640	$(2, 1)$
9	$(10, 5, 8)$	61	270	480	$(2, 0)$	480	$(2, 0)$	480	$(2, 0)$
10	$(3, 2, 10)$	25	48	880	$(3, 1)$	880	$(3, 1)$	880	$(3, 1)$
11	$(3, 3, 10)$	25	69	720	$(3, 0)$	720	$(3, 0)$	720	$(3, 0)$
12	$(3, 5, 10)$	25	111	720	$(3, 0)$	720	$(3, 0)$	720	$(3, 0)$
13	$(5, 2, 10)$	41	80	960	$(4, 0)$	960	$(4, 0)$	960	$(4, 0)$
14	$(5, 3, 10)$	41	115	720	$(3, 0)$	720	$(3, 0)$	720	$(3, 0)$
15	$(5, 5, 10)$	41	185	640	$(2, 1)$	640	$(2, 1)$	640	$(2, 1)$
16	$(10, 2, 10)$	81	160	880	$(3, 1)$	880	$(3, 1)$	880	$(3, 1)$
17	$(10, 3, 10)$	81	230	720	$(3, 0)$	480	$(2, 0)$	480	$(2, 0)$
18	$(10, 5, 10)$	81	370	480	$(2, 0)$	480	$(2, 0)$	480	$(2, 0)$
19	$(3, 2, 12)$	31	60	960	$(4, 0)$	960	$(4, 0)$	960	$(4, 0)$
20	$(3, 3, 12)$	31	87	880	$(3, 1)$	880	$(3, 1)$	880	$(3, 1)$
21	$(3, 5, 12)$	31	141	880	$(3, 1)$	880	$(3, 1)$	880	$(3, 1)$
22	$(5, 2, 12)$	51	100	1360	$(5, 1)$	1200	$(5, 0)$	1200	$(5, 0)$
23	$(5, 3, 12)$	51	145	960	$(4, 0)$	960	$(4, 0)$	960	$(4, 0)$
24	$(5, 5, 12)$	51	235	720	$(3, 0)$	720	$(3, 0)$	720	$(3, 0)$
25	$(10, 2, 12)$	101	200	1120	$(4, 1)$	1040	$(3, 2)$	1120	$(4, 1)$
26	$(10, 3, 12)$	101	290	880	$(3, 1)$	720	$(3, 0)$	880	$(3, 1)$
27	$(10, 5, 12)$	101	470	720	$(3, 0)$	720	$(3, 0)$	720	$(3, 0)$
Average	-	49	152	773.3	-	743.7	-	752.6	-

Table 3.4 reports the experiment results under low uncertain level $(0.1, 0.01)$. For the three models, all 27 instances obtain optimal solutions. Comparing columns 5, 8 and 14, we can see that

Table 3.5: Experimental results with middle uncertain level $(0.15,\ 0.03)$

Set	(a,t,N)	A	B	B model		Z model		P3 model	
				Obj^1	(W^1,W_h^1)	Obj^2	(W^2,W_h^2)	Obj^3	(W^3,W_h^3)
1	$(3,2,8)$	19	36	720	$(3,0)$	640	$(2,1)$	640	$(2,1)$
2	$(3,3,8)$	19	51	640	$(2,1)$	640	$(2,1)$	640	$(2,1)$
3	$(3,5,8)$	19	81	720	$(3,0)$	720	$(3,0)$	720	$(3,0)$
4	$(5,2,8)$	31	60	880	$(3,1)$	720	$(3,0)$	720	$(3,0)$
5	$(5,3,8)$	31	85	720	$(3,0)$	720	$(3,0)$	720	$(3,0)$
6	$(5,5,8)$	31	135	480	$(2,0)$	480	$(2,0)$	480	$(2,0)$
7	$(10,2,8)$	61	120	880	$(3,1)$	720	$(3,0)$	720	$(3,0)$
8	$(10,3,8)$	61	170	720	$(3,0)$	640	$(2,1)$	640	$(2,1)$
9	$(10,5,8)$	61	270	480	$(2,0)$	480	$(2,0)$	480	$(2,0)$
10	$(3,2,10)$	25	48	880	$(3,1)$	880	$(3,1)$	880	$(3,1)$
11	$(3,3,10)$	25	69	960	$(4,0)$	720	$(3,0)$	720	$(3,0)$
12	$(3,5,10)$	25	111	720	$(3,0)$	720	$(3,0)$	720	$(3,0)$
13	$(5,2,10)$	41	80	1120	$(4,1)$	960	$(4,0)$	960	$(4,0)$
14	$(5,3,10)$	41	115	880	$(3,1)$	720	$(3,0)$	720	$(3,1)$
15	$(5,5,10)$	41	185	640	$(2,1)$	640	$(2,1)$	640	$(2,1)$
16	$(10,2,10)$	81	160	880	$(3,1)$	880	$(3,1)$	880	$(3,1)$
17	$(10,3,10)$	81	230	720	$(3,0)$	720	$(3,0)$	720	$(3,0)$
18	$(10,5,10)$	81	370	720	$(3,0)$	480	$(2,0)$	480	$(2,0)$
19	$(3,2,12)$	31	60	1120	$(4,1)$	960	$(4,0)$	960	$(4,0)$
20	$(3,3,12)$	31	87	960	$(4,0)$	960	$(4,0)$	960	$(4,0)$
21	$(3,5,12)$	31	141	880	$(3,1)$	880	$(3,1)$	880	$(3,1)$
22	$(5,2,12)$	51	100	1520	$(5,2)$	1360	$(5,1)$	1360	$(5,1)$
23	$(5,3,12)$	51	145	1120	$(4,1)$	960	$(4,0)$	960	$(4,0)$
24	$(5,5,12)$	51	235	880	$(3,1)$	880	$(3,1)$	880	$(3,1)$
25	$(10,2,12)$	101	200	1120	$(4,1)$	1120	$(4,1)$	1120	$(4,1)$
26	$(10,3,12)$	101	290	880	$(3,1)$	880	$(3,1)$	880	$(3,1)$
27	$(10,5,12)$	101	470	880	$(3,1)$	720	$(3,0)$	720	$(3,0)$
Average	-	49	152	856.3	-	785.2	-	785.2	-

Table 3.6: Experimental results with high uncertain level $(0.2,\ 0.05)$

Set	(a,t,N)	A	B	B model		Z model		P3 model	
				Obj^1	(W^1,W_h^1)	Obj^2	(W^2,W_h^2)	Obj^3	(W^3,W_h^3)
1	$(3,2,8)$	19	36	880	$(3,1)$	640	$(2,1)$	640	$(2,1)$
2	$(3,3,8)$	19	51	720	$(3,0)$	640	$(2,1)$	640	$(2,1)$
3	$(3,5,8)$	19	81	720	$(3,0)$	720	$(3,0)$	720	$(3,0)$
4	$(5,2,8)$	31	60	960	$(4,0)$	720	$(3,0)$	720	$(3,0)$
5	$(5,3,8)$	31	85	880	$(3,1)$	720	$(3,0)$	720	$(3,0)$
6	$(5,5,8)$	31	135	480	$(2,0)$	480	$(2,0)$	480	$(2,0)$
7	$(10,2,8)$	61	120	880	$(3,1)$	720	$(3,0)$	720	$(3,0)$
8	$(10,3,8)$	61	170	720	$(3,0)$	720	$(3,0)$	720	$(3,0)$
9	$(10,5,8)$	61	270	640	$(2,1)$	480	$(2,0)$	480	$(2,0)$
10	$(3,2,10)$	25	48	1040	$(3,2)$	880	$(3,1)$	880	$(3,1)$
11	$(3,3,10)$	25	69	960	$(4,0)$	720	$(3,0)$	720	$(3,0)$
12	$(3,5,10)$	25	111	720	$(3,0)$	720	$(3,0)$	720	$(3,0)$
13	$(5,2,10)$	41	80	1200	$(5,0)$	960	$(4,0)$	960	$(4,0)$
14	$(5,3,10)$	41	115	1120	$(4,1)$	880	$(3,1)$	880	$(3,1)$
15	$(5,5,10)$	41	185	720	$(3,0)$	640	$(2,1)$	640	$(2,1)$
16	$(10,2,10)$	81	160	880	$(3,1)$	880	$(3,1)$	880	$(3,1)$
17	$(10,3,10)$	81	230	720	$(3,0)$	720	$(3,0)$	720	$(3,0)$
18	$(10,5,10)$	81	370	720	$(3,0)$	720	$(3,0)$	720	$(3,0)$
19	$(3,2,12)$	31	60	1360	$(5,1)$	1120	$(4,1)$	1120	$(4,1)$
20	$(3,3,12)$	31	87	1120	$(4,1)$	960	$(4,0)$	960	$(4,0)$
21	$(3,5,12)$	31	141	960	$(4,0)$	880	$(3,1)$	880	$(3,1)$
22	$(5,2,12)$	51	100	1680	$(5,3)$	1360	$(5,1)$	1360	$(5,1)$
23	$(5,3,12)$	51	145	1360	$(5,1)$	960	$(4,0)$	960	$(4,0)$
24	$(5,5,12)$	51	235	1120	$(4,1)$	880	$(3,1)$	880	$(3,1)$
25	$(10,2,12)$	101	200	1360	$(5,1)$	1120	$(4,1)$	1120	$(4,1)$
26	$(10,3,12)$	101	290	1120	$(4,1)$	880	$(3,1)$	880	$(3,1)$
27	$(10,5,12)$	101	470	960	$(4,0)$	720	$(3,0)$	720	$(3,0)$
Average	-	49	152	962.96	-	808.89	-	808.89	-

both model A2 and model Z get the same objective value under all the instances except instances 25 and 26. In addition, P3 model obtains a lower cost than B model over instances 2, 17 and 22. That is to say, we can open fewer total workstations or fewer workstations to handle hazardous tasks. From Table 3.5, we can see that P3 model performs as well as Z model under middle uncertain level $(0.15, 0.03)$ since they have the same cost, and they open fewer workstations than B model over instances 1, 4, 7, 8, 11, 13, 14, 18, 19, 22, 23 and 27. Furthermore, Table 3.6 shows the results under high uncertain level $0.2, 0.05$. We can capture that P3 model performs the same as Z model. On the other hand, P3 model gets a better result than B model.

In summary, we can observe that the proposed new model can effectively and efficiently optimize the disassembly scheme under uncertainty. It performs the same as Z model in middle and high uncertain situations, and it performs better than B model in most of instances. As the uncertain level increases, the proposed new model still keeps a relatively good result compared to other models.

3.5 Conclusions

This chapter investigates a new single EOL product DLBP, where partial information of the probability distribution of task processing time is known. For the problem, a chance-constrained model is proposed and transformed into a new distribution-free model based on obtained problem properties. Then, an approximated SOCP formulation is developed to solve the problem effectively. Numerical experiments on benchmark instances and randomly generated instances demonstrate that the proposed new formulation can effectively solve the problem.

However, increasing demand for customized products results in various new products, and the scale of EOL products and their variants is rapidly expanding in the recycling market. Thus, in the next chapter, we investigate a new multi-product DLBP.

Chapter 4

Multi-product Disassembly Line Balancing Problem

Contents

With the increasing scale of EOL products and their variants, the traditional single-product disassembly line may be inappropriate and uneconomical to handle these increasing EOL product variants. Fang et al. (2019) indicate that multi-product disassembly line balancing can satisfy increasing multiple EOL products disassembly demand and reduces line building and maintenance costs.

Therefore, in this chapter, we extend our study to a novel multi-product DLBP, where identical parts of EOL products and uncertainty task times are simultaneously considered. For the problem, a new joint chance-constrained model is formulated and approximately transformed into a distribution-free model based on problem analysis. To efficiently solve the problem, several valid inequalities are provided and an exact lifted cut-and-solve method is designed. Numerical experiments are conducted to evaluate the performances of the proposed model, valid inequalities and solution method.

The remainder of this chapter is organized as follows. Section 4.1 describes the studied problem and formulates the problem by a new joint chance-constrained model. Section 4.2 approximately transforms the joint chance-constrained model into a distribution-free model and develops several valid inequalities. Section 4.3 proposes an exact lifted cut-and-solve method. Section 4.4 conducts numerical experiments. Section 4.5 concludes this chapter.

4.1 Problem Description and Joint Chance-constrained Model

4.1.1 Problem description

Consider that multiple EOL products have to be disassembled. These EOL products may have identical components that can be disassembled by the same machine or workstation. The disassembly of identical parts of these products should be accomplished using identical tasks.

To depict the studied problem, we use an Engine Electronic Control Unit (ECU) example (Cucchiella et al., 2016). Figure 4.1 illustrates the disassembly schemes of two ECUs that are represented by different colors: red and blue for A, and black and green for B. Each of them has 4 identical parts, i.e., front plastic casing (1), left metal piece (2), right metal piece (3), rear metal

casing (4), and different printed circuit boards (5 or 6). The underlined parts $\underline{5}$ and $\underline{6}$ are printed circuit boards and assumed to be hazardous. A disassembly task and a subassembly (disassembly state) are represented by ◯ and □, respectively. R and N represent the sets of disassembly tasks and subassemblies, respectively. Specifically, N_1 and N_2 denote the initial subassemblies of A and B, and N_{FA} and N_{FB} represent the final states of A and B, respectively. Unlike the study in Chapter 3, the task may be executed multiple times because it could belong to multiple products. Let R_r^k denote that task r can be executed k times because it belongs to k products. For example, R_{10}^2, and R_{13}^2 are identical tasks and may be executed twice because of identical parts of A and B. Moreover, tasks 6, 7, 11, 12 with gray color are assumed to be hazardous because they handle hazardous parts $\underline{5}$ and $\underline{6}$.

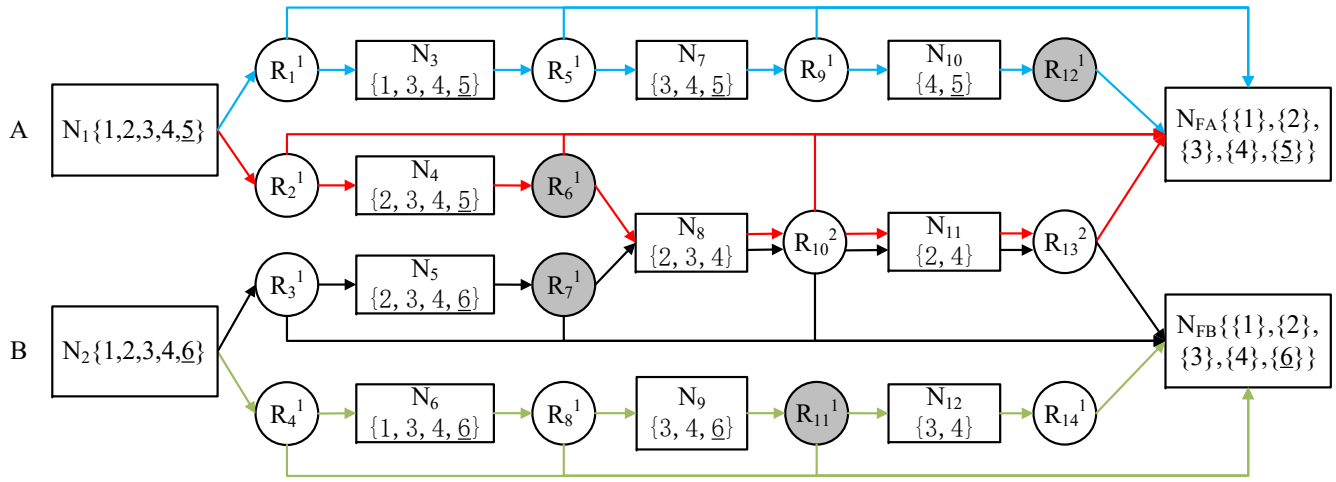


Figure 4.1: Disassembly schemes of products A and B

The studied flexible multi-product DLBP consists of the optimal selection of a disassembly scheme for each EOL product, determination of the workstations to be opened, and assigning tasks to opened workstations respecting task precedence relationships and cycle time within a given risk level. The objective is to minimize the total disassembly cost, including the open workstation cost and the cost of handling hazardous tasks. Besides the assumptions in Chapter 3, we have the following additional assumptions:

- EOL products may have identical parts, and the disassembly of these identical parts can be accomplished by identical tasks;

- One workstation can disassemble several tasks of multiple EOL products, and workstations are independent and homogeneous.
- Task processing times are assumed to be mutually independent and uncertain, and only their limited probability distribution information is available, i.e., the mean, standard deviation and upper bound;

4.1.2 Joint chance-constrained model

Before the mathematical model, the related notations are presented as follows:

Indices:

— r : index of disassembly tasks;

— w, w' : index of workstations;

— n : index of subassembly nodes;

Parameters:

— R : set of disassembly tasks;

— W : set of workstations;

— H : set of hazardous tasks and $H \subset R$;

— N_0: set of initial subassembly nodes;

— N : set of subassembly nodes including initial subassembly nodes, $N_0 \subset N$;

— P_n : set of immediate predecessors of subassembly node n, where $n \in N$;

— Q_n : set of immediate successors of subassembly node n, where $n \in N$;

— pt_r : stochastic processing time of task r, where $r \in R$;

— γ_r : the maximum number of times that task r can be executed, where $r \in R$;

— CT : the cycle time of all workstations;

— α : a given risk level (probability) that the task processing time exceeds the cycle time for all workstations;

— c_w : fixed cost for opening a workstation;

— c_h : fixed cost for handing a hazardous task.

Decision variables:

— x_{wr}: non-negative integer variable, number of times of task r is assigned to workstation w, where $r \in R, w \in W$;

— y_w: binary variable, equal to 1 if workstation w is open, 0 otherwise, where $w \in W$.

For the problem, a joint chance-constrained program P1 is formulated as follows:

$$\textbf{P1:} \quad \min \quad c_w \sum_{w \in W} y_w + c_h \sum_{r \in H} \sum_{w \in W} x_{wr} \tag{4.1}$$

$$s.t. \quad \sum_{r \in S_n} \sum_{w \in W} x_{wr} = 1, \ \forall n \in N_0 \tag{4.2}$$

$$\sum_{w \in W} x_{wr} \leq \gamma_r, \ \forall r \in R \tag{4.3}$$

$$\sum_{r \in S_n} \sum_{w \in W} x_{wr} = \sum_{r \in P_n} \sum_{w \in W} x_{wr}, \ \forall n \in N \backslash N_0 \tag{4.4}$$

$$\sum_{r \in Q_n} \sum_{w=1}^{w'} x_{wr} \leq \sum_{r \in P_n} \sum_{w=1}^{w'} x_{wr}, \ \forall n \in N \backslash N_0, \ \forall w' \in W \tag{4.5}$$

$$\sum_{r \in R} x_{wr} \leq \sum_{r \in R} \gamma_r y_w, \ \forall w \in W \tag{4.6}$$

$$y_{w-1} \geq y_w, \ \forall w \in W \backslash \{1\} \tag{4.7}$$

$$Pr\left(\sum_{r \in R} pt_r x_{wr} \leq CT, \ \forall w \in W \right) \geq 1 - \alpha \tag{4.8}$$

$$y_w \in \{0, 1\}, \forall w \in W \tag{4.9}$$

$$x_{wr} \text{ nonnegative integer}, \ \forall r \in R, \ w \in W \tag{4.10}$$

Objective (4.1) minimizes the total cost, including the workstation opening cost and the extra cost for handling hazardous tasks. Constraints (4.2) mean that only one task is selected at the beginning of disassembly for each EOL product. Constraints (4.3) indicate that task r may be executed at most γ_r times. Constraints (4.4) ensure the flow conservation of each subassembly, i.e., equality of the executed times of preceding tasks and succeeding tasks of each subassembly. Constraints (4.5) respect the precedence relationships of tasks. Constraints (4.6) guarantee that a task may be assigned to a workstation only when the workstation is opened. Constraints (4.7)

number the opened workstations from small to large and avoid empty workstations. Constraint (4.8) ensures the joint probability that the task processing time does not exceed the cycle time for all workstations is greater than or equal to $1-\alpha$. Constraints (4.9) - (4.10) define the domains of decision variables.

The above-mentioned joint chance-constrained model P1 is stochastic and non-linear because of its stochastic task processing time pt_r, and the joint chance constraint (4.8). To efficiently solve the studied problem, P1 is approximately transformed into a distribution-free model P2 in the next section.

4.2 Distribution-free Model and Valid Inequalities

In this section, based on problem analysis, the joint chance-constrained model P1 is transformed into a distribution-free model P2, and valid inequalities are proposed to tighten its solution space.

4.2.1 Distribution-free model

To transform P1 into an approximated distribution-free model, the joint chance constraint (4.8) in P1 needs to be transformed into individual chance constraints based on the assumption that workstations are independent of each other and task processing times are mutually independent. To this end, β_w is introduced to represent the individual risk level for respecting the cycle time of workstation w, $\forall w \in W$, and the joint chance constraint (4.8) can be equivalently transformed to (4.11) and (4.12) as follows:

$$Pr\left(\sum_{r\in R} pt_r x_{wr} \leq CT\right) \geq 1-\beta_w,\ \forall w \in W \tag{4.11}$$

$$\prod_{w\in W}(1-\beta_w) = 1-\alpha \tag{4.12}$$

where $0 \leq \beta_w \leq 1, \forall w \in W$.

Constraints (4.11) mean that the workload of workstation w respects the cycle time with at least a possibility of $1-\beta_w$. constraint (4.12) establishes the relationship between β_w and α. Ac-

cording to the commonly used method for determining the value of individual risk level (Bentaha et al., 2015b; Zheng et al., 2018), which assumes equality of all individual risk levels, parameter β_w is calculated by $\beta_w = 1 - \sqrt[|W|]{1-\alpha}, \forall w \in W$, where $|W|$ is the number of workstations.

With given information of the stochastic task processing time pt_r, the mean $pd_r = E[pt_r]$, the standard deviation σ_r and the upper bound $pd_r(1 + b_r)$, where b_r is the limited deviation ratio to the mean, and similar to Ng (2014) and Zheng et al. (2018), the stochastic task processing time can be expressed as:

$$pt_r = pd_r(1 + Z_r),\ \forall r \in R \tag{4.13}$$

where Z_r is the deviation ratio to the mean of the task processing time and is limited by b_r, i.e., $Z_r \leq b_r$.

Therefore, the individual chance Constraints (4.11) are reformulated as:

$$Pr\left(\sum_{r\in R} pd_r(1 + Z_r)x_{wr} \leq CT\right) \geq 1 - \beta_w,\ \forall w \in W \tag{4.14}$$

To construct the distribution-free model, Constraints (4.14) are approximately transformed to the following constraints:

$$\sum_{r\in R}(pd_r + \nu_r)x_{wr} \leq CT,\ \forall w \in W \tag{4.15}$$

where ν_r is an auxiliary parameter that reflects the uncertainty of task processing time pt_r, which will be explained later.

Then, the distribution-free model P2 is proposed as follows.

$$\textbf{P2:} \qquad \min \quad c_w \sum_{w\in W} y_w + c_h \sum_{r\in H}\sum_{w\in W} x_{wr}$$

$$s.t. \qquad (4.2) - (4.7), (4.9), (4.10), (4.15)$$

As long as the value of ν_r is determined, P2 becomes a mixed-integer linear program that

may be solved using a commercial solver, such as CPLEX, at least for small-sized instances. If the following proposition is true, ν_r can be pre-determined.

Proposition 2. *As long as any ν_r satisfies the following equation*

$$f(\nu_r) := \min_{\lambda_i>0} \left\{ e^{-\lambda_i \nu_r / p d_r} \left(1 + \frac{E[Z_r^2]}{b_r^2} (e^{\lambda_i b_r} - \lambda_i b_r - 1) \right) - \beta_w \right\} = 0 \tag{4.16}$$

then, any solution of P2 must satisfy the individual chance Constraints (4.14)*.*

Before demonstrating Proposition 2, Lemmas 1 and 2 are introduced (refer to Ng (2014)). Firstly, Lemma 1 is introduced below to describe the relationship between Z_r and b_r.

Lemma 1. *For any $\lambda > 0$, the following inequality must hold:*

$$E[e^{\lambda Z_r}] \leq 1 + \frac{E[Z_r^2]}{b_r^2} (e^{\lambda b_r} - \lambda b_r - 1) \tag{4.17}$$

Proof. By the definition of the mean task processing time, we have $p d_r = E[t_r] = E[p d_r (1 + Z_r)]$. Thus, it follows that $E[Z_r] = 0$. Then, $e^{\lambda b_r}$ can be expressed as $1 + \lambda b_r + \sum_{m=2}^{\infty} \frac{(\lambda b_r)^m}{m!}$ (according to the Taylor series expansion). Because $Z_r \leq b_r$ and $b_r > 0$,

$$\begin{aligned} E[e^{\lambda Z_r}] &= 1 + \sum_{m=2}^{\infty} \frac{\lambda^m E[Z_r^m]}{m!} \\ &\leq 1 + E[Z_r^2] \sum_{m=2}^{\infty} \frac{\lambda^m E[b_r^{m-2}]}{m!} \\ &= 1 + \frac{E[Z_r^2]}{b_r^2} \sum_{m=2}^{\infty} \frac{\lambda^m b_r^m}{m!} \\ &= 1 + \frac{E[Z_r^2]}{b_r^2} (e^{\lambda b_r} - \lambda b_r - 1) \end{aligned} \tag{4.18}$$

□

Next, Lemma 2 is proposed to depict the relationship between $E[Z_r^2]$ and σ_r^2.

Lemma 2. *With the given standard deviation of task processing time σ_r, we have $E[Z_r^2] = \frac{\sigma_r^2}{p d_r^2}$.*

Proof. Let $D[t_r]$ denote the variance of processing time of task r, we have:

$$D[t_r] = D[pd_r(1 + Z_r)] = pd_r^2 D[Z_r] \tag{4.19}$$

Note that $E[Z_r] = 0$, hence $D[Z_r] = E[Z_r^2] - (E[Z_r])^2 = E[Z_r^2]$. Accordingly, the following equation

$$D[t_r] = pd_r^2 E[Z_r^2] \tag{4.20}$$

must hold, thus $E[Z_r^2] = \frac{\sigma_r^2}{pd_r^2}$. □

Now, we provide the proof of Proposition 2.

Proof. Given a feasible solution x_{wr} obtained by (4.15), if $\sum_{r \in R} pd_r(1 + Z_r)x_{wr} > CT$, it must follow that $\sum_{r \in R} pd_r Z_r x_{wr} > \sum_{r \in R} \nu_r x_{wr}$. Moreover, if $pd_r Z_r > \nu_r$, it must follows that $\sum_{r \in R} pd_r Z_r x_{wr} > \sum_{r \in R} \nu_r x_{wr}$. Hence, for simplicity, $\sum_{r \in R} pd_r Z_r x_{wr} > \sum_{r \in R} \nu_r x_{wr}$ is conservatively approximated by $pd_r Z_r > \nu_r$, so the following inequality must hold:

$$Pr\left(\sum_{r \in R} pd_r(1 + Z_r)x_{wr} > CT\right) \leq Pr\left(pd_r Z_r > \nu_r\right), \ \forall w \in W \tag{4.21}$$

Obviously, we have $Pr\left(pd_r Z_r > \nu_r\right) = Pr\left(e^{\lambda Z_r} > e^{\lambda \nu_r / pd_r}\right)$. Then, using the Markov inequality: $Pr(X \geq a) \leq \frac{E(X)}{a}$, where $X \geq 0$, and $a > 0$. Let $X = e^{\lambda Z_r}$ and $a = e^{\lambda \nu_r / pd_r}$. With Lemma 1, we have

$$\begin{aligned} Pr\left(e^{\lambda Z_r} > e^{\lambda \nu_r / pd_r}\right) &\leq e^{-\lambda \nu_r / pd_r} E[e^{\lambda Z_r}] \\ &\leq e^{-\lambda \nu_r / pd_r}\left(1 + \frac{E[Z_r^2]}{b_r^2}(e^{\lambda b_r} - \lambda b_r - 1)\right) \end{aligned} \tag{4.22}$$

Since (4.22) is valid for any $\lambda > 0$, the following inequality must holds:

$$Pr\left(\sum_{r \in R} pd_r(1 + Z_r)x_{wr} > CT\right) \leq \min_{\lambda > 0} e^{-\lambda \nu_r / pd_r}\left(1 + \frac{E[Z_r^2]}{b_r^2}(e^{\lambda b_r} - \lambda b_r - 1)\right), \ \forall w \in W \tag{4.23}$$

Recall constraints (4.14) that can be rewritten as:

$$Pr\left(\sum_{r\in R} pd_r(1+Z_r)x_{wr} > CT\right) < \beta_w,\ \forall w \in W \tag{4.24}$$

We can always find a λ such that the right sides of (4.22) and (4.23) to be equal, we can obtain

$$\min_{\lambda>0} e^{-\lambda\nu_r/pd_r}\left(1+\frac{E[Z_r^2]}{b_r^2}(e^{\lambda b_r}-\lambda b_r-1)\right)-\beta_w = 0,\ \forall w \in W \tag{4.25}$$

So any ν_r satisfies (4.25), then any solution x_{wr} of P2 must satisfy the individual chance constraint (4.14). □

The value of ν_r can be obtained by solving equality (4.16). To determine ν_r, Algorithm 1 is presented.

Algorithm 1 Calculation of the value of ν_r

Input: $pd_r, b_r, \sigma_r^2, \nu_r = 0, \lambda_i = 0.001$

1: $f_{value} = \widehat{f}(\nu_r, \lambda_i) = e^{-\lambda_i\nu_r/pd_r}\left(1+\frac{E[Z_r^2]}{b_r^2}(e^{\lambda_i b_r}-\lambda_i b_r-1)\right)-\beta_w$;
2: **while** $f_{value} > 0.001$ **do**
3: **if** $f \leq 1$ **then**
4: $\lambda_i = \lambda_i + 0.001$;
5: $f_{value} = \widehat{f}(\nu_r, \lambda_i)$;
6: **else**
7: $\nu_r = \nu_r + 0.1$;
8: $\lambda_i = 0.001$;
9: $f_{value} = \widehat{f}(\nu_r, \lambda_i)$;
10: **end if**
11: **end while**

Output: The value of ν_r

The basic idea of the algorithm is to determine a combination of $(\nu_r,\ \lambda)$ that satisfies $\widehat{f}(\nu_r, \lambda_i) = 0$. Although P2 is a mixed-integer linear programming model once ν_r is fixed, P2 is still NP-hard because the deterministic single-product DLBP is NP-hard (McGovern and Gupta, 2007b). Therefore, effective valid inequalities are proposed to improve P2 in the next section.

4.2.2 Valid inequalities

This section presents two valid inequalities to tighten the solution space of model P2, and then an improved distribution-free model is proposed.

Valid inequality 1

The first valid inequality attempts to limit the number of workstations to be opened. To propose the valid inequality, we relax the assumption that one workstation can execute several tasks of different EOL products in model P2 to form its relaxed model P2$'$, in which one workstation executes the tasks of only one EOL product. P (indexed by p) denotes the set of EOL products and $|W_p|$ denotes the number of workstations required for product p in a feasible solution of P2$'$. $\sum_{p\in P}|W_p|$ denotes the total number of workstations of the feasible solution of P2$'$. The following proposition is formulated.

Proposition 3. *Let $\sum_{w\in W} y_w$ be the number of opened workstations in a feasible solution of P2, then the following inequality:*

$$VI1: \quad \sum_{w\in W} y_w \leq \sum_{p\in P}|W_p| \tag{4.26}$$

is a valid inequality for P2.

Proof. VI1 indicates that the number of opened workstations of an optimal solution of $P2$ does not exceed $\sum_{p\in P}|W_p|$. Suppose that $\sum_{w\in W} y_w^{''}$ denotes the optimal number of workstations of $P2$ and $\sum_{w\in W} y_w^{''} > \sum_{p\in P}|W_p|$. Let $\sum_{w\in W} y_w^{'}$ denote the optimal number of workstations of $P2^{'}$, and we have $\sum_{p\in P}|W_p| \geq \sum_{w\in W} y_w^{'}$. Thus, $\sum_{w\in W} y_w^{''} > \sum_{w\in W} y_w^{'}$. As $P2^{'}$ is the relaxed model of $P2$, we can always find another optimal number of workstations of $P2$, denoted as $\sum_{w\in W} y_w^{*}$ that is smaller than or equal to $\sum_{w\in W} y_w^{'}$. Apparently, we have $\sum_{w\in W} y_w^{*} \leq \sum_{w\in W} y_w^{'} < \sum_{w\in W} y_w^{''}$. So it is contradictory to the assumption that $\sum_{w\in W} y_w^{''}$ is the optimal number of workstations of $P2$. Thus, (4.26) is a valid inequality of $P2$. □

To determine $\sum_{p\in P}|W_p|$, Algorithm 2 is proposed below, where L_p (indexed by l), R_l^p and AT

are the disassembly scheme set of product p, the task set of disassembly scheme l in L_p and the available time of a workstation, respectively. $|W_p^l|$ denotes the number of open workstations of disassembly scheme l of product p.

Algorithm 2 Upper bound determination for opened workstations $\sum_{p \in P} |W_p|$

Input: Disassembly schemes of EOL products $(L_p, \forall p \in P)$, task processing times $(pt_r, \forall r \in R)$, cycle time CT

1: $p = 1$;
2: **while** $(p \leq |P|)$ **do**
3: $l = 1$, $|W_p| = 0$;
4: **while** $(l \leq |L_p|)$ **do**
5: $station_{num} = 1$, $r = 1$, $AT = CT$;
6: **while** $(r \leq |R_l^p|)$ **do**
7: **if** $pt_r \leq T$ **then**
8: $AT = AT - pt_r$, $r = r + 1$;
9: **else**
10: $station_{num} = station_{num} + 1$, $AT = CT$;
11: **end if**
12: **end while**
13: $|W_p^l| = station_{num}$;
14: **if** $|W_p^l| > |W_p|$ **then**
15: $|W_p| = |W_p^l|$;
16: **end if**
17: $l = l + 1$;
18: **end while**
19: $p = p + 1$;
20: **end while**

Output: The value of $\sum_{p \in P} |W_p|$

Algorithm 2 contains three loops: Lines 6 to 12 calculate the number of workstations of disassembly scheme l of product p in a feasible solution of P2$'$, Lines 4 to 18 determine $|W_p| = \max_{l \in L_p} \{|W_p^l|\}$, and Lines 2 to 20 obtain $\sum_{p \in P} |W_p|$.

Valid inequality 2

The second valid inequality determines the workstations to which disassembly tasks cannot be assigned due to their precedence relationships. The task precedence relationship implies that a task cannot be executed before its predecessors and after its successors. For simplicity, let

PT_r^l and QT_r^l denote the sum of processing times of the predecessors and successors of task r in disassembly scheme l, respectively, where $l \in L_r$ and L_r is the disassembly scheme set containing task r. Consequently, the minimum numbers of workstations for predecessors and successors of task r, n_{pr} and n_{sr} may be determined by the following formulas: $n_{pr} = \min_{l \in L_r} \left\{ \lceil PT_r^l / CT \rceil \right\}$ and $n_{qr} = \min_{l \in L_r} \left\{ \lceil QT_r^l / CT \rceil \right\}$, recall that CT is the cycle time and $\lceil x \rceil$ is the smallest integer greater than or equal to x.

Without loss of generality, the second valid inequality can be analyzed via the case illustrated in Figure 4.2, in which tasks are numbered and represented by ◯. The illustrated EOL product has two disassembly schemes that contain 11 tasks, and the corresponding processing times ($pt_r = pd_r + \nu_r$) are set as 8, 7, 8, 5, 4, 7, 6, 8, 5, 9, and 7 seconds. With the assumption that only one scheme of a product can be selected. For the green scheme ($l = 1$), task 7 has predecessors $\{1, 2, 3\}$ and successor $\{8, 9\}$, and we obtain $PT_7^1 = pt_1 + pt_2 + pt_3 = 23$ seconds and $QT_7^1 = pt_8 + pt_9 = 13$ seconds. Similarly for the blue scheme ($l = 2$), we have $PT_7^2 = pt_4 + pt_5 + pt_6 = 16$ seconds and $QT_7^2 = pt_{10} + pt_{11} = 16$ seconds. We suppose that the cycle time is 10 seconds, and we have $W_{max} = 6$ according to Algorithm 2. Therefore, $n_{p7} = \min(\lceil 23/10 \rceil, \lceil 16/10 \rceil) = 2$ and $n_{q7} = \min(\lceil 13/10 \rceil, \lceil 16/10 \rceil) = 2$. Task 7 cannot be assigned before workstation $n_{p7} = 2$ and after workstation $|W_{max}| - n_{s7} + 1 = 5$. Therefore, for model P2, we have $x_{17} = x_{67} = 0$ and the following proposition.

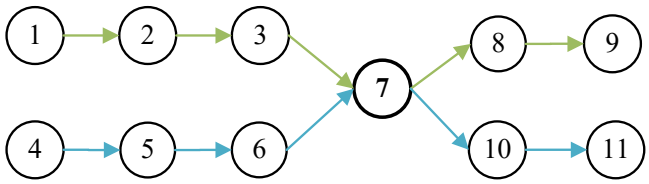


Figure 4.2: An illustration example of valid inequality 2

Proposition 4. *With the pre-determined* $\sum_{p \in P} |W_p|$ *by Algorithm 2, the following inequality:*

$$VI2: \quad x_{wr} = 0, \ \forall r \in R, w < n_{pr}, w > \sum_{p \in P} |W_p| - n_{qr} + 1 \tag{4.27}$$

is a valid inequality for P2.

Proof. VI2 means that task r cannot be assigned to the workstations smaller than n_{pr} or larger than $\sum_{p\in P}|J_p| - n_{qr} + 1$. Suppose that there exists an optimal solution ζ of model $P2$ with $x_{wr} \neq 0$, in which $w < n_{pr}$ or $w > \sum_{p\in P}|W_p| - n_{qr} + 1$. Naturally, we can always find a solution $\zeta^{'}$ that the processors of task r need at least $n_{pr} - 1$ workstations. Thus, it can be observed that solution $\zeta^{'}$ requires fewer workstations than the optimal solution ζ. It is contradictory to our former assumption. Hence, (4.27) is valid for model $P2$. □

Improved distribution-free model

The numerical experiments show that the model with VI1 and VI2 performs best. Therefore, an improved model P3 is presented:

$$\textbf{P3:} \quad \min \quad c_w \sum_{w\in W} y_w + c_h \sum_{r\in H}\sum_{w\in W} x_{wr}$$

$$s.t. \quad (4.2) - (4.7), (4.9), (4.10), (4.15), (4.26), (4.27)$$

Although the improved model P3 is tighter than P2, it is still time-consuming for large-scale instances. Therefore, an exact lifted cut-and-solve method is proposed in the next section to solve model P3 efficiently.

4.3 Lifted Cut-and-solve Method

As introduced in Chapter 2, the CS method is a particular branch and bound algorithm and has been successfully used to solve many combinatorial optimization problems. To enhance the performance of the CS method for solving our problem, a new lifted CS method is devised.

The framework and search tree of the proposed lifted CS method are outlined in Figures 4.3 and 4.4, respectively. Particularly, 1) a constructive heuristic (Algorithm 3) is proposed to obtain an initial UB that may be an optimal solution of the original problem (see subsection 4.3.1). 2) double PCs (PC^1_{i-1}, PC^2_{i-1}) based on partially linear relaxation are proposed at each iteration to obtain a better LB (see subsection 4.3.2). 3) SP is further divided into two sub-problems (SP^1_i,

SP_i^2) by a second PC for an efficient resolution (see subsection 4.3.3).

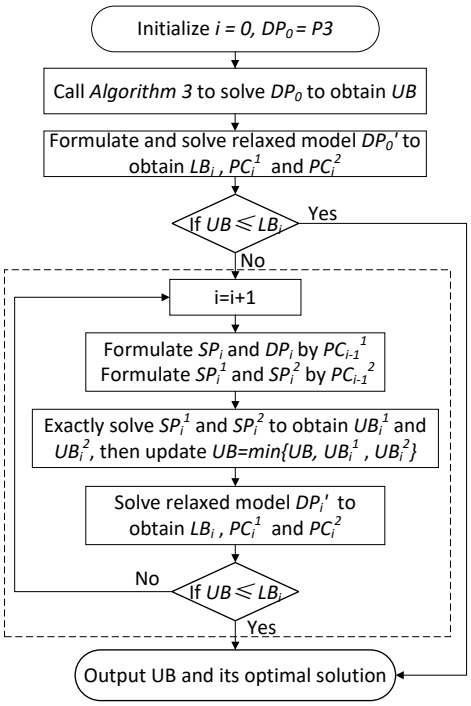


Figure 4.3: The framework of lifted cut-and-solve method

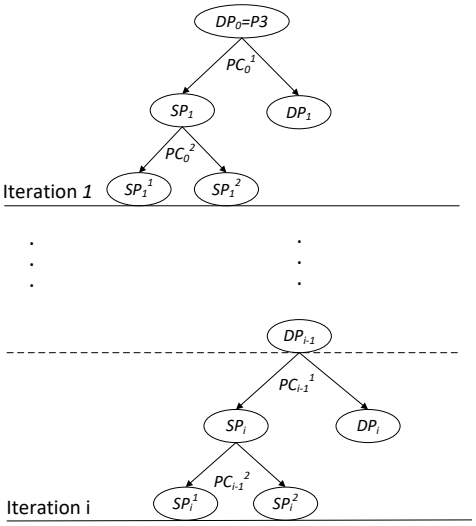


Figure 4.4: The search tree of lifted cut-and-solve method

4.3.1 Heuristic to determine an initial UB

The main purpose of the proposed heuristic is to select a disassembly scheme with the minimum sum of task processing times for each EOL product, open workstations for product p one at a time, $p = 1, ..., |P|$, and assign the related tasks to workstations respecting task precedence relationships and cycle time constraints.

Note that R_p^l denotes the ordered task set in which the precedence relationships of tasks are respected for $l \in L_p$, and L_p is the disassembly scheme set of product p. Recall that $pt_r = pd_r + \nu_r$ denotes the processing time of task $r, \forall r \in R$. The proposed heuristic is summarised as follows.

Algorithm 3 consists of two parts. Lines 1 to 8 select the disassembly scheme with the minimum sum of task processing times for each EOL product, and Lines 9 to 18 decide the workstations to be opened and assign the selected tasks to the opened workstations.

Algorithm 3 The heuristic to obtain an upper bound

Input: $pt_r, R_p^l, \forall l \in L_p, p \in P, r \in R_p^l$

Set $p = 1$;
while $p \leq |P|$ **do**
  for $l = 1 \rightarrow |L_p|$ **do**
    Calculate $T_l = \sum_{r \in R_p^l} pt_r$;
  end for
  $R_p^* = R_p^{l^*}$, where $l^* = arg\min_{l \in L_p}\{R_l\}$;
  $p = p + 1$;
end while
Set $w = 0, T = 0, RP = \bigcup_{p \in P} R_p^*$;
while $(r \leq |RP|)$ **do**
  if $pt_{RP[r]} \leq T$ **then**
    $T = T - pt_{RP[r]}$;
    $x_{wRP[r]} = 1$, $r = r + 1$;
  else
    $w = w + 1$;
    $y_w = 1$, $T = CT$;
  end if
end while
Calculate $UB = C_w \sum_{w \in W} y_w + C_h \sum_{r \in H} \sum_{w \in W} x_{wr}$;

Output: An UB and its corresponding solution

4.3.2 Double piercing cuts

Existing CS methods generally define one PC, and the corresponding SP may still be difficult to solve. The double PCs (PC_{i-1}^1, PC_{i-1}^2) at the $(i-1)$-th iteration are designed based on an optimal solution of linear relaxation of DP_{i-1} in this study, denoted as $(\widetilde{x}_{wr}, \widetilde{y}_w)$. Note that the objective function value is primarily determined by the number of workstations to be opened, i.e., $\sum_{w \in W} y_w$. Because the cost of handling hazardous tasks is the same regardless of which disassembly scheme is selected. A better LB may be obtained with the partial relaxation of DP_{i-1}. Preliminary experiments show that task r has a larger probability of being assigned to workstation w in the optimal solution of $P3$ when $\widetilde{x}_{wr}$ has a large value. Therefore, we propose 1) a partial relaxation strategy for DP_{i-1}, 2) the first PC with the value of $\widetilde{y}_w$, 3) the second PC with the value of $\widetilde{x}_{wr}$.

At the $(i-1)$-th iteration, with an optimal solution of the relaxed DP_{i-1}, the first piercing cut,

PC^1_{i-1}, can be defined as follows.

$$\sum_{y_w \in \Omega^1_{i-1}} y_w = 0 \tag{4.28}$$

where $\Omega^1_{i-1} = \{y_w | \widetilde{y}_w = 0, \forall w \in W\}$.

Let R_{i-1} denote the set of tasks whose values of $\widetilde{x}_{wr}$ are fractions. For any task $r \in R_{i-1}$, the combination of task r assigned to workstation w, i.e., (w, r) corresponding to the largest values of $\widetilde{x}_{wr}$ is presented below:

$$\Psi_{i-1} = \left\{ (w,r) | (w,r) = arg \max_{r \in R_{i-1}, w \in W} \{\widetilde{x}_{wr}\} \right\} \tag{4.29}$$

Consequently, the second piercing cut PC^2_{i-1} can be defined as follows:

$$\sum_{x_{wr} \in \Omega^2_{i-1}} x_{wr} \leq l_{i-1} - 1 \tag{4.30}$$

where $\Omega^2_{i-1} = \{x_{wr} | (w,r) \in \Psi_{i-1}\}$ and l_{i-1} is an integer that takes the value of $\sum_{(w,r) \in \Psi_{i-1}} \lceil \widetilde{x}_{wr} \rceil$.

The first piercing cut (PC^1_{i-1}) divide the current DP_{i-1} into two sub-problems DP_i and SP_i. The second piercing cut (PC^2_{i-1}) further divide SP_i into two sub-problems SP^1_i and SP^2_i. The formulations of sparse and dense problems are presented below.

4.3.3 Sparse and dense problem formulations

According to the above-defined double PCs, the formulations of the dense problem (DP_i) at the $i\text{-}th$ iteration can be presented as follows.

$$\begin{aligned} DP_i: \quad & \min \quad c_w \sum_{w \in W} y_w + c_h \sum_{r \in H} \sum_{w \in W} x_{wr} \\ s.t. \quad & (4.2) - (4.7), (4.9), (4.10), (4.15), (4.26), (4.27) \\ & \sum_{y_w \in \Omega^1_{i-1}} y_w \geq 1 \end{aligned} \tag{4.31}$$

Two sparse problems SP_i^1 and SP_i^2 are defined as follows:

$$
\begin{aligned}
SP_i^1: \quad & \min \quad c_w \sum_{w \in W} y_w + c_h \sum_{r \in H} \sum_{w \in W} x_{wr} \\
s.t. \quad & (4.2) - (4.7), (4.9), (4.10), (4.15), (4.26), (4.27) \\
& \sum_{y_w \in \Omega_{i-1}^1} y_w = 0 \\
& \sum_{x_{wr} \in \Omega_{i-1}^2} x_{wr} \geq l_{i-1}
\end{aligned} \tag{4.32}
$$

$$
\begin{aligned}
SP_i^2: \quad & \min \quad c_w \sum_{w \in W} y_w + c_h \sum_{r \in H} \sum_{w \in W} x_{wr} \\
s.t. \quad & (4.2) - (4.7), (4.9), (4.10), (4.15), (4.26), (4.27) \\
& \sum_{y_w \in \Omega_{i-1}^1} y_w = 0 \\
& \sum_{x_{wr} \in \Omega_{i-1}^2} x_{wr} \leq l_{i-1} - 1
\end{aligned}
$$

To better understand the proposed method, the ECU example mentioned in Section 4.1 is given below. In the example, there are 2 products and 14 tasks with processing times (in seconds) of {30, 35, 45, 35, 35, 20, 18, 35, 31, 38, 50, 20, 25, 25}. The costs of opening a workstation (c_w) and handling a hazardous task (c_h) are 3 and 2 dollars, respectively. There are 6 workstations available, and the cycle time CT is 90 seconds. The detailed process of the lifted cut-and-solve method is depicted below in Figure 4.5.

Next, numerical experiments are conducted to evaluate the proposed lifted CS method's performance, and the results are discussed.

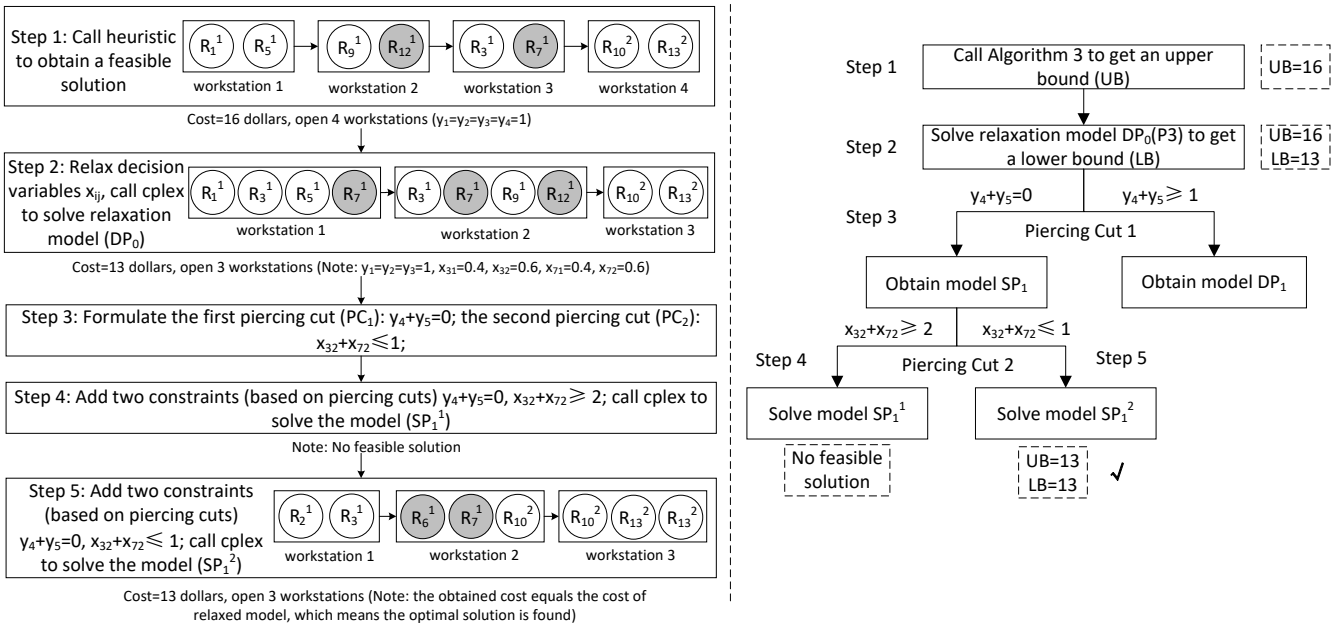


Figure 4.5: The detailed process of the algorithm for solving the instance

4.4 Numerical Experiments

Numerical experiments are performed on an illustrative example, 10 instances based on realistic products and 480 randomly generated instances to evaluate the performance of the proposed integrated approach. The program is coded using the C++ programming language in Microsoft Visual Studio 2019. All numerical experiments are performed on a personal computer with Core I5 and 3.20 GHz CPU with 12GB RAM. All models are solved using the CPLEX solver version 12.9.

4.4.1 An illustrative example

In this part, the ECU example (see Figure 4.1) is investigated to test our models, which contain 14 tasks and 12 subassemblies. The mean task processing times (in seconds) are {30, 35, 45, 35, 35, 20, 18, 35, 31, 38, 50, 20, 25, 25}. The upper bound, standard deviation and risk level are 0.1, 0.01 and 5%, respectively. The costs of opening a workstation and handling a hazardous task are 3 and 2 dollars, respectively. Assume that there are 6 workstations available, and the cycle time CT is 90 seconds.

Figure 4.6 reports the results of the illustrative instance. In the optimal solution, 3 worksta-

tions are opened, and the total cost is 13 dollars. For ECU A, the selected disassembly scheme contains tasks $\{1, 5, 9, 12\}$, and the selected disassembly scheme of ECU B contains tasks $\{3, 7, 10, 13\}$, corresponding to the blue and black lines in Figure 4.2, respectively.

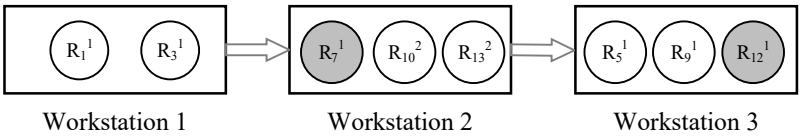


Figure 4.6: The result of ECU example

4.4.2 Instance data and parameter setting

Since there is no standard data set that can be directly used for the multi-product DLBP with identical parts and stochastic task times, we generate 10 instances based on 7 realistic EOL products used in Chapter 3 (see Table 4.1) by emerging different EOL products to generate the multi-product instances, which is similar to Fang et al. (2020b) and Liu et al. (2022). The basic information of these multi-product instances is summarized in Table 4.2, where each instance scale is primarily determined by three parameters, i.e., the number of products $|P|$, the number of tasks $|R|$, and the number of available workstations $|W|$.

Table 4.1: Information of realistic EOL products

Number	EOL product	Reference
1	Hand light	Tang et al. (2002)
2	Compass	Bentaha et al. (2012)
3	Ballpoint pen	Lambert (1999)
4	Sample product	Koc et al. (2009)
5	Piston and connecting rod	Bentaha et al. (2013b)
6	Radio set	Lambert (1999)
7	Automatic pencil	Ma et al. (2011)

Moreover, to thoroughly evaluate the performances of the integrated method, 96 problem sets, each with 5 instances, for a total of 480 instances are randomly generated. For each EOL product, 20% of tasks are set as identical tasks, and 20% are set as hazardous tasks. The mean task processing times (pd_r) (in seconds) are randomly generated from $[10, 50]$ consistent with He

Table 4.2: Information of the multi-product instances based on realistic EOL products

Number	Products	$\|P\|$	$\|R\|$	$\|W\|$
1	1,2	2	20	10
2	1,3	2	30	10
3	1,2,3	3	40	15
4	2,3,4	3	53	15
5	1,2,3,4	4	63	20
6	2,3,4,5	4	78	20
7	1,2,3,4,5	5	88	25
8	2,3,4,5,6	5	108	25
9	1,2,3,4,5,6	6	118	30
10	2,3,4,5,6,7	6	145	30

et al. (2020b). Similar to Ng (2014), the upper bound and standard deviation of the task processing time $(b_r, E[Z_r^2])$ are set as $(0.1, 0.01)$, $(0.2, 0.05)$ and $(0.3, 0.10)$ to represent different uncertainty levels. The unit fixed cost of a workstation c_w and the unit cost for handling hazardous task c_h are set as 3 and 2 dollars, respectively (Bentaha et al., 2015b; He et al., 2020b). The risk level α and the cycle time CT are set as 5% and 90 seconds, respectively. The input parameters are summarized in Table 4.3.

Table 4.3: Input parameters of random instances

Parameters	Values
Number of products $(\|P\|)$	2, 4, ..., 20
Number of tasks $(\|R\|)$	20, 40, ..., 400
number of available workstations $(\|W\|)$	6, 10, ..., 86
Cost to open a workstation (c_w) (dollars)	3
Cost to handle a hazardous task (c_h) (dollars)	2
Cycle time (CT) (seconds)	90
Risk level (α)	5%
Mean task times (pd_r) (seconds)	[10,50]
$(b_r, E[Z_r^2])$	$(0.1, 0.01)$, $(0.2, 0.05)$, $(0.3, 0.10)$

4.4.3 Evaluation of the distribution-free model

The approximated distribution-free model $P2$ (solved via CPLEX) is evaluated by comparison with the corresponding deterministic model (DM) (solved via CPLEX) and a sampling average approximation (SAA) method. The model with the SAA method mainly refers to Luedtke and Ahmed

(2008) and Qiu and Wang (2014). For the approximated model, some new parameters are introduced below:

Ω : the set of randomly generated Monte Carlo samples;

η : index of each sample;

$z_{\eta w}$: binary variable, equal to 1 if the workload of workstation $w \in W$ exceeds the cycle time under sample $\eta \in \Omega$, 0 otherwise ;

M: a big positive number;

The approximated model using SAA method is proposed in the following:

$$\textbf{P4:} \quad \min \quad c_w \sum_{w \in W} y_w + c_h \sum_{r \in H} \sum_{w \in W} x_{wr}$$

$$s.t. \quad (4.2) - (4.7), (4.9), (4.10)$$

$$\sum_{r \in R} pt_r(\eta) x_{wr} - M z_{\eta w} \leq CT, \ \forall w \in W, \ \forall \eta \in \Omega \tag{4.33}$$

$$\sum_{\eta \in \Omega} \sum_{w \in W} z_{\eta w} \leq \lfloor |\Omega| \alpha \rfloor \tag{4.34}$$

$$z_{\eta w} \in \{0, 1\}, \ \forall w \in W, \ \forall \eta \in \Omega \tag{4.35}$$

Constraints (4.33) guarantee that the workloads of all workstations will not exceed the cycle time under all samples by adding a binary variable $z_{\eta w}$. Constraint (4.34) restricts the sum value of $z_{\eta w}$, which is determined by the scale of Monte Carlo samples $|\Omega|$ and the probability α, where $\lfloor x \rfloor$ means the largest integer that does not exceed x. Constraints (4.35) define the domains of decision variables.

The task processing times of the DM are set as the means of the stochastic ones in P2. The task processing times in SAA are assumed to follow normal distributions, in which their means and standard deviations are identical to these values in P2. The number of samples is set to 20 based on preliminary experiments. Numerical experiments are first performed on the realistic EOL product instances.

Tables 4.4-4.6 report the computational results of realistic EOL product instances under 3 uncertainty levels, where Obj_{DM}, T_{DM}, Obj_{SAA}, T_{SAA} and Obj_{P2}, T_{P2} denote the average objective

values and computational times of DM, SAA and P2, respectively. In particular, the computational time unit is seconds denoted by s. We can observe that the average objective value of the DM is the smallest, which takes the value of 29.5, and the objectives of SAA and P2 are almost the same. On the other hand, the SAA requires more computational time than P2 and DM. Moreover, with the uncertainty increase, the objective values and the computational times also increase. Therefore, more workstations are generally needed to deal with the uncertain task times.

Table 4.4: The results of realistic EOL product instances with $(b_r = 0.1, E[Z_r^2] = 0.01)$

		DM		SAA		P2	
Instance	(P, R, W)	Obj_{DM}	$T_{DM}(s)$	Obj_{SAA}	$T_{SAA}(s)$	Obj_{P2}	$T_{P2}(s)$
1	(2,20,10)	14.2	0.10	14.2	0.20	14.8	0.09
2	(2,30,10)	19.0	0.19	20.8	1.05	21.4	0.16
3	(3,40,15)	22.8	0.35	24.0	2.29	24.0	0.36
4	(3,53,15)	22.8	0.43	25.6	17.60	25.6	0.66
5	(4,63,20)	29.6	2.35	30.2	18.78	31.4	2.69
6	(4,78,20)	27.8	1.14	29.6	305.60	29.6	1.21
7	(5,88,25)	34.0	2.83	37.0	2411.27	37.0	2.68
8	(5,108,25)	36.4	4.47	37.4	2782.47	39.4	11.99
9	(6,118,30)	43.8	4.99	45.8	3626.41	47.4	8.40
10	(6,145,30)	44.8	8.72	50.8	4837.71	54.0	17.50
Average	-	29.5	2.56	31.5	1400.35	32.5	4.58

Table 4.5: The results of realistic EOL product instances with $(b_r = 0.2, E[Z_r^2] = 0.05)$

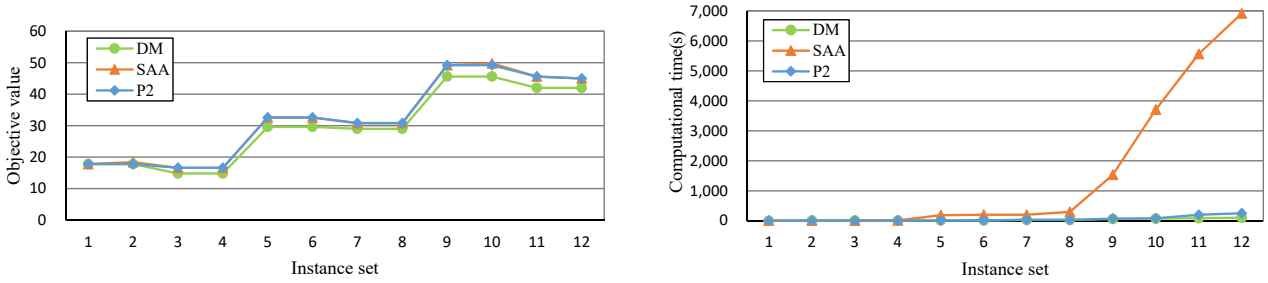
		DM		SAA		P2	
Instance	(P, R, W)	Obj_{DM}	$T_{DM}(s)$	Obj_{SAA}	$T_{SAA}(s)$	Obj_{P2}	$T_{P2}(s)$
1	(2,20,10)	14.2	0.10	16.0	0.30	15.4	0.10
2	(2,30,10)	19.0	0.19	23.2	2.41	22.6	0.22
3	(3,40,15)	22.8	0.35	25.8	12.77	26.4	0.41
4	(3,53,15)	22.8	0.43	28.2	153.42	27.6	0.88
5	(4,63,20)	29.6	2.35	32.6	6824.53	33.2	0.87
6	(4,78,20)	27.8	1.14	30.8	119.36	32.6	11.04
7	(5,88,25)	34.0	2.83	38.5	2333.57	40.0	3.33
8	(5,108,25)	36.4	4.47	43.0	5148.70	41.2	11.56
9	(6,118,30)	43.8	4.99	49.8	6283.63	49.8	32.48
10	(6,145,30)	44.8	8.72	51.7	7218.16	54.0	15.26
Average	-	29.5	2.56	34.0	2209.69	34.3	7.61

To have an observation of the performances on different methods, experiments are further conducted on random instance sets with up to 6 products, 120 tasks and 30 workstations. Com-

Table 4.6: The results of realistic EOL product instances with $(b_r = 0.3, E[Z_r^2] = 0.1)$

		DM		SAA		P2	
Instance	(P, R, W)	Obj_{DM}	$T_{DM}(s)$	Obj_{SAA}	$T_{SAA}(s)$	Obj_{P2}	$T_{P2}(s)$
1	(2,20,10)	14.2	0.10	16.6	0.33	17.2	0.11
2	(2,30,10)	19.0	0.19	24.4	3.19	24.4	0.16
3	(3,40,15)	22.8	0.35	27.0	34.31	28.2	0.63
4	(3,53,15)	22.8	0.43	28.6	115.71	28.6	0.73
5	(4,63,20)	29.6	2.35	36.2	1528.23	36.2	2.91
6	(4,78,20)	27.8	1.14	33.2	233.04	33.2	2.79
7	(5,88,25)	34.0	2.83	42.0	4097.66	42.4	134.70
8	(5,108,25)	36.4	4.47	44.0	5444.01	43.6	68.44
9	(6,118,30)	43.8	4.99	54.0	7874.54	53.3	15.27
10	(6,145,30)	44.8	8.72	56.7	9235.37	55.5	100.11
Average	-	29.5	2.56	36.3	2856.64	36.3	32.58

putational results are illustrated in Figures 4.7-4.9, in which sub-figures (a) and (b) report the objective values and computational times of DM, SAA and P2, respectively. Sub-figures (a) of Figures 4.7-4.9 show that the curves of the objective values of P2 and SAA almost overlap, and they are superior to the values of DM. This result indicates that the solution qualities of P2 and SAA are nearly identical, and DM proposes a better solution because there are no uncertain factors. On the other hand, sub-figures (b) shows that the curves of the computational times of P2 and DM almost overlap, and the computational time of SAA increases sharply.

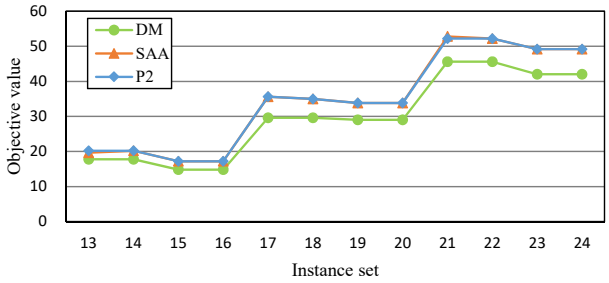


(a) Objective value

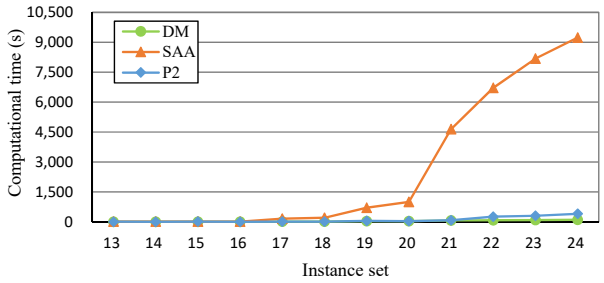
(b) Computational time

Figure 4.7: Comparison results of models with $(b_r = 0.1, E[Z_r^2] = 0.01)$

The detailed results are presented in Tables 4.7-4.9. Based on Tables 4.7-4.9, the average gaps of the objective values between DM and P2 can be calculated by the formula $(Obj_{P2} - Obj_{DM})/Obj_{DM} \times 100\%$. The gaps under the 3 uncertainty levels are 7.72%, 16.44% and 24.83%,

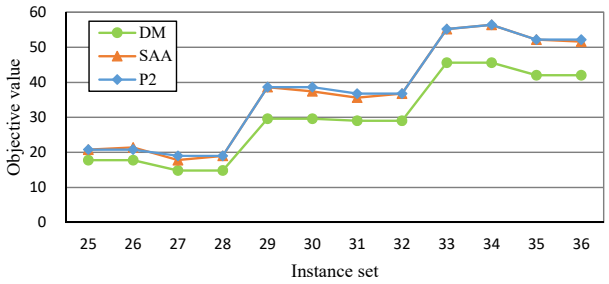


(a) Objective value

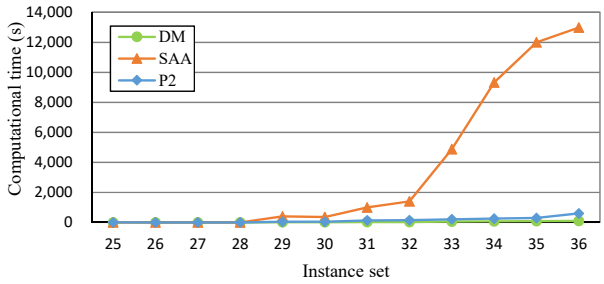


(b) Computational time

Figure 4.8: Comparison results of models with $(b_r = 0.2, E[Z_r^2] = 0.05)$



(a) Objective value



(b) Computational time

Figure 4.9: Comparison results of models with $(b_r = 0.3, E[Z_r^2] = 0.1)$

respectively, which indicate that the gaps increase with levels of uncertainty. Considering the average computational time, SAA are 474, 313 and 258 times of those needed by P2 under 3 different uncertainty levels. These results indicate that DM and P2 perform better than SAA in terms of computational time.

Moreover, we utilize the Wilcoxon signed-rank test, a flexible non-parametric statistical hypothesis test (Garc4a et al., 2009), to compare the performance of the three methods. The principle of the Wilcoxon signed-rank test is to calculate the differences between the two compared methods and analyze these differences to test if they are statistically significantly different. We prefer the Wilcoxon test because it does not assume the normality of the samples (Chica et al., 2010) and it has already been demonstrated to help analyze the behavior of evolutionary algorithms, and it has been well adopted in the disassembly line balancing problems (Fang et al., 2019; Li et al., 2020).

The null-hypothesis H_0 is that the compared two methods have equal performance. The significance level considered in all the tests to be presented is $p = 0.05$. A p-value is smaller than

Table 4.7: The results of random instances with $(b_r = 0.1, E[Z_r^2] = 0.01)$

		DM		SAA		P2	
Instance	(P, I, J)	Obj_{DM}	$T_{DM}(s)$	Obj_{SAA}	$T_{SAA}(s)$	Obj_{P2}	$T_{P2}(s)$
1	(2,20,6)	17.8	0.08	17.8	0.12	17.8	0.13
2	(2,20,10)	17.8	0.10	18.4	0.22	17.8	0.12
3	(2,40,10)	14.8	0.13	16.6	0.33	16.6	0.15
4	(2,40,14)	14.8	0.10	16.6	0.36	16.6	0.13
5	(4,40,14)	29.6	0.41	32.6	6.16	32.6	0.37
6	(4,40,18)	29.6	0.50	32.6	10.92	32.6	0.39
7	(4,80,18)	29.0	0.92	30.8	11.10	30.8	1.13
8	(4,80,22)	29.0	0.92	30.8	26.06	30.8	1.01
9	(6,60,22)	45.6	2.02	49.2	1538.67	49.2	2.56
10	(6,60,26)	45.6	2.57	49.8	3708.64	49.2	2.60
11	(6,120,26)	42.0	3.58	45.6	5568.77	45.6	14.00
12	(6,120,30)	42.0	5.93	45.0	6918.61	45.0	14.95
Average	-	29.8	1.44	32.2	1482.50	32.1	3.13

Table 4.8: The results of random instances with $(b_r = 0.2, E[Z_r^2] = 0.05)$

		DM		SAA		P2	
Instance	(P, I, J)	Obj_{DM}	$T_{DM}(s)$	Obj_{SAA}	$T_{SAA}(s)$	Obj_{P2}	$T_{P2}(s)$
13	(2,20,6)	17.8	0.08	19.6	0.14	20.2	0.12
14	(2,20,10)	17.8	0.10	20.2	0.25	20.2	0.12
15	(2,40,10)	14.8	0.13	17.2	0.41	17.2	0.17
16	(2,40,14)	14.8	0.10	17.2	0.42	17.2	0.13
17	(4,40,14)	29.6	0.41	35.6	28.69	35.6	0.40
18	(4,40,18)	29.6	0.50	35.0	35.44	35.0	0.49
19	(4,80,18)	29.0	0.92	33.8	261.50	33.8	1.34
20	(4,80,22)	29.0	0.92	33.8	348.12	33.8	1.16
21	(6,60,22)	45.6	2.02	52.8	4641.25	52.2	7.85
22	(6,60,26)	45.6	2.57	52.2	6717.90	52.2	12.35
23	(6,120,26)	42.0	3.58	49.2	8182.52	49.2	15.87
24	(6,120,30)	42.0	5.93	49.2	9236.56	49.2	53.99
Average	-	29.8	1.44	34.7	2454.44	34.7	7.83

0.05, denoting a rejection of the null-hypothesis, i.e., the compared two methods perform significantly differently. The results of the Wilcoxon test are illustrated in Table 4.10. We can obtain that DM outperforms P2 and SAA statistically because the p-values are smaller than 0.0001. In comparison, P2 and SAA are not statistically different in terms of the objective values according to the p-value of 0.3047.

Table 4.9: The results of random instances with $(b_r = 0.3, E[Z_r^2] = 0.10)$

		DM		SAA		P2	
Instance	(P, I, J)	Obj_{DM}	$T_{DM}(s)$	Obj_{SAA}	$T_{SAA}(s)$	Obj_{P2}	$T_{P2}(s)$
25	(2,20,6)	17.8	0.08	20.8	0.18	20.8	0.11
26	(2,20,10)	17.8	0.10	21.4	0.24	20.8	0.15
27	(2,40,10)	14.8	0.13	17.8	0.34	19.0	0.16
28	(2,40,14)	14.8	0.10	19.0	0.34	19.0	0.14
29	(4,40,14)	29.6	0.41	38.6	99.29	38.6	1.52
30	(4,40,18)	29.6	0.50	37.4	59.58	38.6	1.23
31	(4,80,18)	29.0	0.92	35.6	409.25	36.8	4.56
32	(4,80,22)	29.0	0.92	36.8	863.36	36.8	9.12
33	(6,60,22)	45.6	2.02	55.2	4875.63	55.2	11.79
34	(6,60,26)	45.6	2.57	56.4	9333.72	56.4	18.40
35	(6,120,26)	42.0	3.58	52.2	12005.20	52.2	20.30
36	(6,120,30)	42.0	5.93	51.6	12984.24	52.2	90.04
Average	-	29.8	1.44	36.9	3385.95	37.2	13.13

Table 4.10: Results of Wilcoxon signed-rank test for three different methods

	Negative difference $(A > B)$		Positive difference $(A < B)$		Equal difference $(A = B)$		p-value
Compared pairs	Number	Ranks	Number	Ranks	Number	Ranks	
$A = DM, B = P2$	0	0	34	595	2	-	< 0.0001
$A = DM, B = SAA$	0	0	35	630	1	-	< 0.0001
$A = P2, B = SAA$	5	31	4	31	27	-	0.3047

DM: deterministic model solved via CPLEX; P2: P2 model solved via CPLEX; SAA: sampling average approximation method

4.4.4 Evaluation of valid inequalities

To evaluate the proposed valid inequalities, numerical experiments are performed on realistic EOL product instances and randomly generated instance sets with up to 10 products, 200 tasks and 46 workstations.

Table 4.11 reports the computational results of realistic EOL product instances, in which columns 2 and 3 represent the instance parameters and objective values. Columns 3-6 represent the computational times of model P2, P2 with valid inequality 1 (P2+VI1), P2 with valid inequality 2 (P2+VI2) and P2 with valid inequalities 1 and 2 (P2+VI1+VI2), which are denoted by T_{P2}, T_{P2+VI1}, T_{P2+VI2}, and $T_{P2+VI1+VI2}$, respectively. From Table 4.11, we can see that the average computational times of these 4 models are 2.56, 1.75, 1.74, and 1.46 seconds, respectively. It indicates that the pro-

posed two inequalities are valid.

Table 4.11: Computational results of valid inequalities on realistic EOL product instances

Instance	(P, R, W)	Obj	$T_{P2}(s)$	$T_{P2+VI1}(s)$	$T_{P2+VI2}(s)$	$T_{P2+VI1+VI2}(s)$
1	(2,20,10)	14.2	0.10	0.09	0.07	0.05
2	(2,30,10)	19.0	0.19	0.10	0.10	0.11
3	(3,40,15)	22.8	0.35	0.24	0.17	0.19
4	(3,53,15)	22.8	0.43	0.31	0.22	0.22
5	(4,63,20)	29.6	2.35	1.58	1.98	1.09
6	(4,78,20)	27.8	1.14	0.75	0.53	0.50
7	(5,88,25)	34.0	2.83	1.55	1.51	1.45
8	(5,108,25)	36.4	4.47	2.40	1.68	1.66
9	(6,118,30)	43.8	4.99	3.83	4.51	3.45
10	(6,145,30)	44.8	8.72	6.66	6.70	5.92
Average	-	29.5	2.56	1.75	1.74	1.46

Moreover, the results of random instances are reported in Figure 4.10 and Table 4.12. Specifically, the curves in Figure 4.10 represent the computational time ratios that are calculated via the formula $R_i = pt_r/T_{P2} \times 100\%$, where $i = P2$, $P2 + VI1$, $P2 + VI2$ and $P2 + VTI1 + VI2$. The results show that each proposed inequality can save computational time, and P2+VI1+VI2 (P3) is the most efficient. Specifically, the average computational time ratios of P2+VI1, P2+V2 and P2+VI1+VI2 (P3) are only 33.93%, 33.80% and 25.17%, respectively. Therefore, the performance of model P3, i.e., P2+VI1+VI2 (P3), is verified.

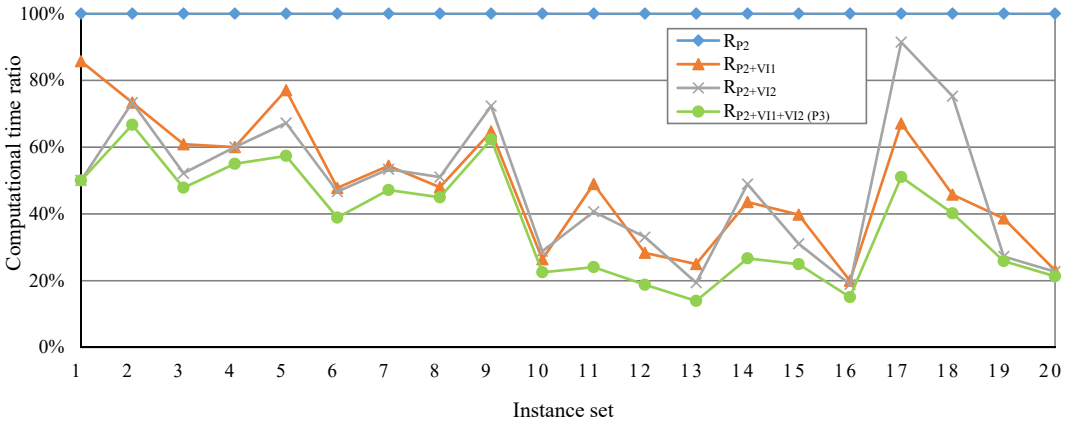


Figure 4.10: Comparison results of valid inequalities

Table 4.12: Computational results of valid inequalities on random instances

Instance	(P, I, J)	Obj	$T_{P2}(s)$	$T_{P2+VI1}(s)$	$T_{P2+VI2}(s)$	$T_{P2+VI1+VI2}(s)$
1	(2,20,6)	17.8	0.14	0.12	0.07	0.07
2	(2,20,10)	17.8	0.15	0.11	0.11	0.10
3	(2,40,10)	14.8	0.23	0.14	0.12	0.11
4	(2,40,14)	14.8	0.20	0.12	0.12	0.11
5	(4,40,14)	29.6	0.61	0.47	0.41	0.35
6	(4,40,18)	29.6	0.90	0.43	0.42	0.35
7	(4,80,18)	29.0	1.93	1.05	1.03	0.91
8	(4,80,22)	29.0	1.98	0.95	1.01	0.89
9	(6,60,22)	45.6	4.19	2.71	3.03	2.61
10	(6,60,26)	45.6	11.29	2.98	3.25	2.54
11	(6,120,26)	42.0	14.81	7.25	6.02	3.56
12	(6,120,30)	42.0	18.21	5.16	6.02	3.42
13	(8,80,30)	62.2	36.12	9.02	7.00	5.03
14	(8,80,34)	62.2	19.50	8.49	9.54	5.20
15	(8,160,34)	56.2	50.19	19.96	15.58	12.51
16	(8,160,38)	56.2	84.51	16.87	15.95	12.73
17	(10,100,38)	75.2	36.45	24.45	33.33	18.60
18	(10,100,42)	75.2	44.33	20.29	33.36	17.81
19	(10,200,42)	68.4	139.75	54.01	38.09	36.14
20	(10,200,46)	68.4	153.52	35.41	34.83	32.69
Average	-	44.1	30.95	10.50	10.46	7.79

4.4.5 Evaluation of the lifted CS method

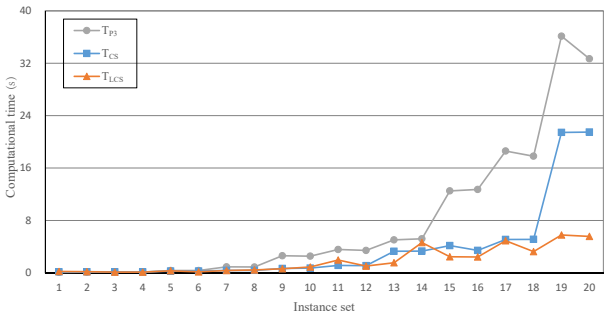
The proposed lifted CS method is first tested on realistic EOL product instances, and then tested on 20 small-scale random instance sets with up to 10 products, 200 tasks and 46 workstations and 20 large-scale random instance sets with up to 20 products, 400 tasks and 86 workstations. The proposed method is compared with P3 solved by CPLEX (version 12.9) and a classic CS method.

Table 4.13 reports the results on instances based on realistic products. Columns 3-5 illustrate the computational times of the CPLEX, the classic CS method and the lifted CS method, which are represented by T_{P3}, T_{CS} and T_{LCS}, respectively. From Table 4.13, we can observe that all instances are solved within a short time, and the average computational times of CPLEX, classic CS method, and lifted CS method are 1.46, 0.71, and 0.55 seconds, respectively. It implies that the lifted CS method can efficiently solve the instances and can reduce 62.3% of the computational time required by CPLEX on average.

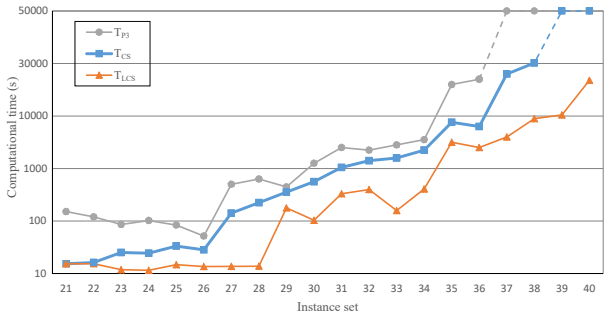
The experimental results on random instances are illustrated in Figure 4.11, and sub-figures

Table 4.13: Computational results of lifted cut-and-solve method on realistic EOL product instances

Instance set	(P, R, W)	Obj	$T_{P3}(s)$	$T_{CS}(s)$	$T_{LCS}(s)$
1	(2, 20, 10)	14.2	0.05	0.11	0.06
2	(2, 30, 10)	19.0	0.11	0.16	0.11
3	(3, 40, 15)	22.8	0.19	0.27	0.10
4	(3, 53, 15)	22.8	0.22	0.30	0.19
5	(4, 63, 20)	29.6	1.09	1.08	0.81
6	(4, 78, 20)	27.8	0.50	0.38	0.35
7	(5, 88, 25)	34.0	1.45	0.89	0.50
8	(5, 108, 25)	36.4	1.66	0.70	0.69
9	(6, 118, 30)	43.8	3.45	1.07	0.87
10	(6, 145, 30)	44.8	5.92	2.19	1.78
Average	-	29.5	1.46	0.71	0.55



(a) Instances with up to 10 products, 200 tasks and 46 available workstations



(b) Instances with up to 20 products, 400 tasks and 86 available workstations

Figure 4.11: Comparison results of methods

(a) and (b) report the results of small-scale and large-scale instances, respectively. Sub-figure (a) shows that the computational times of all methods are less than 40 seconds and the curves of T_{CS} and T_{LCS} are below the T_{P3} curve in most cases, except for instances 1-4. Table 4.14 indicates that the average computational times of CPLEX, classical CS method and lifted CS method are 7.79, 3.64 and 1.84 seconds, respectively. This result signifies that the lifted CS method globally outperforms CPLEX and the classical CS method.

For large-scale instances, sub-figure (b) shows that the computational times of all 3 methods increase with instance set size. However, the computational time of the lifted CS method increases much more slowly than the classical CS method and CPLEX. Notably, CPLEX and the classical CS method cannot propose feasible solutions for instance sets 37-40 and instance sets

Table 4.14: Computational results for random instances with $|P| = 2 - 10$, $|I| = 20 - 200$ and $|J| = 6 - 46$

Instance set	(P, I, J)	Obj	$T_{P3}(s)$	$T_{CS}(s)$	$T_{LCS}(s)$
1	(2, 20, 6)	17.8	0.07	0.15	0.21
2	(2, 20, 10)	17.8	0.10	0.16	0.17
3	(2, 40, 10)	14.8	0.11	0.15	0.11
4	(2, 40, 14)	14.8	0.11	0.13	0.11
5	(4, 40, 14)	29.6	0.35	0.26	0.31
6	(4, 40, 18)	29.6	0.35	0.26	0.15
7	(4, 80, 18)	30	0.91	0.37	0.35
8	(4, 80, 22)	30	0.89	0.37	0.44
9	(6, 60, 22)	45.6	2.61	0.66	0.62
10	(6, 60, 26)	45.6	2.54	0.72	0.89
11	(6, 120, 26)	42.0	3.56	1.13	1.96
12	(6, 120, 30)	42.0	3.42	1.09	1.03
13	(8, 80, 30)	62.2	5.03	3.29	1.54
14	(8, 80, 34)	62.2	5.20	3.32	4.62
15	(8, 160, 34)	56.2	12.51	4.16	2.45
16	(8, 160, 38)	56.2	12.73	3.42	2.42
17	(10, 100, 38)	75.2	18.60	5.09	4.90
18	(10, 100, 42)	75.2	17.81	5.11	3.24
19	(10, 200, 42)	68.4	36.14	21.44	5.78
20	(10, 200, 46)	68.4	32.69	21.49	5.56
Average	-	44.18	7.79	3.64	1.84

39-40 within 50,000 seconds. The computational times of these instances are presented by dotted lines. Table 4.15 shows that the average computational times of CPLEX, the classical CS and the lifted CS methods are 6356.84, 1657.09 and 673.23 seconds, respectively. For instance sets 1-36, the lifted CS method needs only 17.53% and 40.65% of the computational times required by the CPLEX and the classic CS method, respectively.

In summary, the proposed lifted CS method is more efficient than CPLEX and the classical CS method, particularly for large-scale instances.

4.5 Conclusions

This chapter investigates a new multi-product DLBP with identical parts of multiple products and uncertain task processing times. To efficiently solve the problem, an integrated approach is de-

Table 4.15: Computational results for random instances with $|P| = 12 - 20$, $|I| = 120 - 400$ and $|J| = 46 - 86$

Instance set	(P, I, J)	Obj	$T_{P3}(s)$	$T_{CS}(s)$	$T_{LCS}(s)$
21	(12, 120, 46)	88.2	171.59	18.67	18.64
22	(12, 120, 50)	88.2	139.83	21.05	18.73
23	(12, 240, 50)	82.0	93.57	40.08	7.38
24	(12, 240, 54)	82.0	113.40	38.71	6.22
25	(14, 140, 54)	104.2	92.32	52.35	16.88
26	(14, 140, 58)	104.2	71.42	45.21	13.40
27	(14, 280, 58)	95.0	653.86	165.50	13.64
28	(14, 280, 62)	95.0	823.88	357.88	13.98
29	(16, 160, 62)	114.2	563.37	460.63	278.67
30	(16, 160, 66)	114.2	1505.53	722.52	114.81
31	(16, 320, 66)	109.2	3617.95	1299.23	435.62
32	(16, 320, 70)	109.2	3564.20	1875.17	504.30
33	(18, 180, 70)	132.0	3966.16	2315.86	221.47
34	(18, 180, 74)	132.0	5081.14	3153.85	511.58
35	(18, 360, 74)	132.2	18852.11	8524.85	4841.85
36	(18, 360, 78)	132.2	22182.63	7421.85	3754.55
37	(20, 200, 78)	156.2	-	26523.11	5527.60
38	(20, 200, 82)	156.2	-	30087.80	9526.51
39	(20, 400, 82)	156.0	-	-	10854.60
40	(20, 400, 86)	156.0	-	-	21672.45
Average	-	107.1	6356.84	1657.09	673.23

In the table, $'-'$ means that the optimal solution has not been found within 50000 seconds. Hence, the average value is obtained from sets 21 to 36.

vised in which 1) the problem is formulated using a novel joint chance-constrained model, 2) a distribution-free model is proposed based on problem analysis, 3) efficient valid inequalities are developed to reduce the solution space, and 4) a new lifted CS method is provided to solve the problem. Numerical experiments on one case study, realistic EOL product instances and random instances demonstrate the high performance of the integrated approach.

To efficiently manage EOL product disassembly, it is important to focus on disassembly line balancing, which belongs to a tactical decision level. However, to improve the overall performance of the disassembly system, it must consider strategic, tactical and operational decisions together for better resource utilization. Therefore, in the next chapter, we investigate a new disassembly line balancing related RSC design problem for multiple EOL products under uncertainty.

Chapter 5

Multi-product Disassembly Line Balancing Related RSC Design

Contents

The previous two chapters focus on DLBP, which belongs to tactical-level decisions. Scholars and practitioners recognize that strategic-level decisions impact the overall performance of a system in the long term, and these two level decisions interact with each other. To improve the overall performance of disassembly system, it is necessary to consider strategic and tactical decisions together. Motivated by this fact, this chapter investigates a novel disassembly line balancing related RSC design problem to maximize the expected profit and minimize CO_2 emissions, simultaneously. For the problem, a bi-objective nonlinear two-stage stochastic programming model is formulated and approximately transformed to a linear distribution-free model based on problem properties (Section 5.1). Then, an exact ε-constrained based method is proposed to efficiently solve the problem. Significantly, the transformed single objective problems in ε-constrained method are exactly solved by an improved Benders decomposition (Section 5.2). Numerical experiments comprising a case study and 200 randomly generated instances are conducted to evaluate the performance of the proposed methods. Moreover, sensitivity analysis is made to draw managerial insights (Section 5.3).

5.1 Problem Description and Formulation

In this section, the studied problem is described and formulated as a bi-objective nonlinear two-stage stochastic programming model with joint chance constraints. Then, the nonlinear model is approximately transformed into a linear distribution-free model.

5.1.1 Problem description

Consider a RSC to be designed, which comprises a set of supply points (I), collection centers (J), candidate disassembly plants (K), and remanufacturing plants (L), as illustrated in Figure 5.1. Firstly, collected multiple EOL products can be transported from supply points to collection centers, where they are inspected, classified and stored. Then, EOL products will be transported to opened disassembly plants, where they can be decomposed into components, and valuable ones will be shipped to remanufacturing plants. The studied RSC network can be represented by

a directed graph $G = \{N, A\}$, where N and A denote sets of nodes and arcs, respectively. We have $N = I \cup J \cup K \cup L$, and I, J, K, L are mutually disjoint. The values of arcs denote the unit transportation cost for EOL products or components between different nodes.

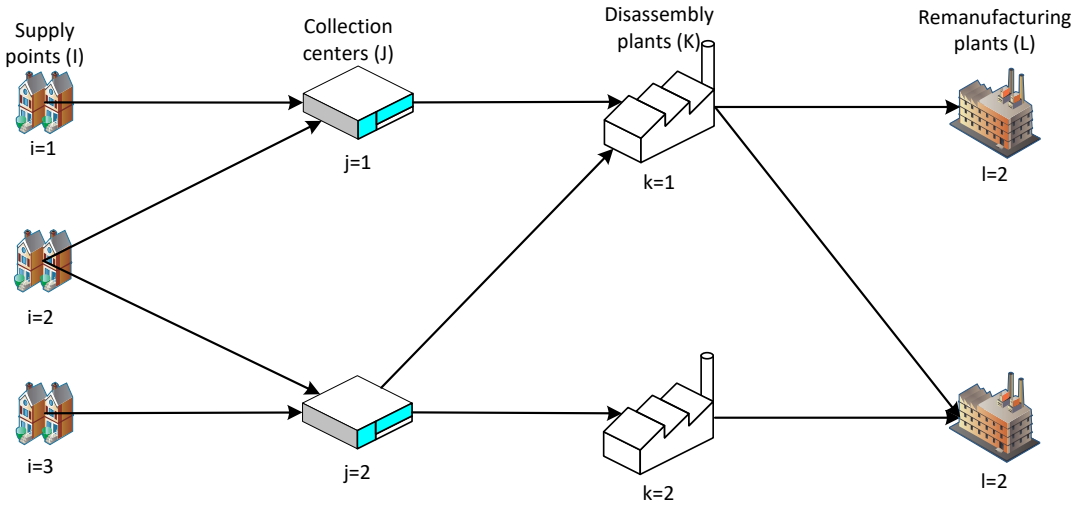


Figure 5.1: An illustrative example of the RSC

In this chapter, we assume that several uncertain EOL products are available at each supply point $i \in I$ in a RSC, and the uncertain amount of EOL product $p \in P$ at period $t \in T$ under scenario $s \in S$ can be represented by O_{ipts}. The percentage of EOL product $p \in P$ at supply point $i \in I$ worth to be disassembled is assumed to be known and denoted by θ_{ip}. A collection center $j \in J$ has a limited capacity V_j to store EOL products. Each candidate disassembly plant $k \in K$ has W_k available workstations, a limited production capacity E_k, and a limited inventory capacity U_k for components. The cycle time CT_k is assumed to be known. At an opened disassembly plant, a number of machines need to be purchased, and disassembly lines must be balanced to accomplish EOL product disassembly. There are M different types of machines with different technology levels, which high technology machines have lower CO_2 emissions but high acquisition costs. Due to variable qualities of EOL products, the processing time pt_r of task $r \in R$ is assumed to be uncertain, and only partial probability distribution information of task time is available, i.e., the mean pd_r, standard deviation E_r^2 and upper bound b_r. Each remanufacturing plant $l \in L$ has an uncertain demand for component $c \in C$ and is denoted by D_{lcts}. For each scenario $s \in S$, the probability is assumed to be known and represented by ρ_s.

For the studied problem, we need to make strategic-level and tactical-level decisions during a planning horizon composed of $|T|$ periods. The strategic decisions involve determining the number of disassembly plants to be opened and their locations, and the procurement of multi-types of equipment (i.e., disassembly machines). The tactical-level decisions consist of assigning tasks to procured machines and assigning procured machines to workstations, deciding the transportation flows of EOL products and components, and determining the inventory levels of EOL products and components at period t, where $t \in T$. In summary, we have the following additional assumptions:

- A minimum segregation distance must be respected for any two selected disassembly plants to mitigate the risk of a geographically-induced disruption to the RSC.
- One disassembly plant can handle multiple EOL products, and the production capacity at a disassembly plant has to be respected.
- Each machine is capable of processing a given set of tasks, and it can handle only one task at a time.
- The percentage of EOL products that are worthwhile to be disassembled is assumed to be known.
- The supply of EOL products and demand for components are uncertain, and are portrayed by a set of scenarios, and the probability of each scenario is assumed to be known.
- The task precedence relationships of EOL products are known and must be respected.
- The cycle time of each disassembly plant is assumed to be known, and the total processing time of all workstations cannot exceed the cycle time under a given risk level.
- The numbers of components obtained from one EOL product are assumed to be known, and the inventory capacities needed for one EOL product and one component are known and have to be respected.

This chapter investigates a multi-period integrated disassembly line balancing related RSC design problem under stochastic EOL products supply, components demand and task times. Strategic-level decisions (the number of disassembly plants to be opened and their locations, and machines procurement) and tactical-level decisions (machines and tasks assignments, transportation and inventory levels of EOL products and components) are simultaneously made. The objectives are to maximize the expected profit and minimize the CO_2 emissions of machines. In the following, a bi-objective nonlinear two-stage stochastic programming model is formulated.

5.1.2 Bi-objective nonlinear two-stage stochastic programming model

Indices:

— i : index of supply points;

— j : index of collection centers;

— k, k' : indices of disassembly plants;

— l : index of remanufacturing plants;

— m : index of machines;

— w : index of workstations;

— p : index of EOL products;

— c : index of components;

— r, r' : indices of disassembly tasks;

— t : index of time periods;

— s : index of scenarios;

Parameters:

— I : set of supply points;

— J : set of collection centers;

— K : set of candidate locations for disassembly plants;

— L : set of remanufacturing plants;

— M : set of machines;

— W_k : set of workstations of disassembly plant k, where $k \in K$;

— P : set of EOL products;

— C : set of components;

— R : set of disassembly tasks;

— R_p : set of disassembly tasks of product p, where $p \in P$;

— B_r : set of immediate successors of task r, where $r \in R$;

— T : set of time periods, and $T = \{0, 1, 2, ..., |T|\}$;

— S : set of scenarios;

— ρ_s : the probability of scenario s, where $s \in S$;

— $h_{kk'}$: binary parameter, equal to 1 if $d_{kk'} \geq d_{min}$, 0 otherwise, where $d_{kk'}$ denotes the arc value between candidate disassembly plants k and k', d_{min} represents the smallest segregation arc value requirement between every pair of opened disassembly plants, and $k, k' \in K$, $k \neq k'$;

— V_j : inventory capacity of EOL products in collection center j, where $j \in J$;

— ι_p^P : inventory capacity needed for one EOL product p, where $p \in P$;

— U_k : inventory capacity of components in disassembly plant k, where $k \in K$;

— ι_c^C : inventory capacity needed for one component c, where $c \in C$;

— E_k : production capacity of disassembly plant k, where $k \in K$;

— ω_p : production capacity needed for dismantling one EOL product p, where $p \in P$;

— O_{ipts} : supply volume of EOL product p at supply point i during period t under scenario s, where $i \in I$, $p \in P$, $t \in T$, $s \in S$;

— D_{lcts} : demand for component c at remanufacturing plant l during period t under scenario s, where $l \in L$, $c \in C$, $t \in T$, $s \in S$;

— θ_{ip} : percentage of EOL product p at supply point i worth to be disassembled, where $i \in I$, $p \in P$;

— δ_{cp} : number of component c obtained from one EOL product p, where $c \in C$, $p \in P$;

— pt_r : task processing time of task r, which is stochastic, where $r \in R$;

— pd_r : mean processing time of task r, where $r \in R$;

— E_r^2 : standard deviation of processing time of task r, where $r \in R$;

— b_r : upper bound parameter of processing time of task r, where $r \in R$;

— CT_k : cycle time of disassembly plant k, where $k \in K$;

— α : a given probability (risk level);

— e_m : CO_2 emission of machine m in the planning horizon T, where $m \in M$;

— a_{mr} : binary parameter, equal to 1 if task r can be accomplished by machine m, 0 otherwise, where $m \in M$, $r \in R$;

— c_k^S : setup cost of a disassembly plant at location k, where $k \in K$;

— c_m^P : procurement cost for machine m, where $m \in M$;

— c_k^W : cost to open a workstation in location k, where $k \in K$;

— c_p^C : cost to preprocess one unit EOL product p in collection centers, where $p \in P$;

— c_p^D : cost to disassemble one unit EOL product p, where $p \in P$;

— c_{ijp}^{S-C} : transportation cost of one unit EOL product p from supplier i to collection center j, where $i \in I$, $j \in J$, $p \in P$;

— c_{jkp}^{C-D} : transportation cost of one unit EOL product p from collection center j to disassembly plant k, where $j \in J$, $k \in K$, $p \in P$;

— c_{klc}^{D-R} : transportation cost of one unit component c from disassembly plant k to remanufacturing plant l, where $k \in K$, $l \in L$, $c \in C$;

— c_p^{IP} : inventory cost of one unit EOL product p in each time period, where $p \in P$;

— c_c^{IC} : inventory cost of one unit component c in each time period, where $c \in C$;

— σ_c^R : revenue of fulfilled demand of one component $c \in C$ in each time period;

In the two-stage stochastic programming (Birge and Louveaux, 2011), the first-stage decisions are made before the random events. The second-stage decisions are made according to the first-stage decisions after the realization of each random event, and the results of the second-stage will also affect the first-stage decision through the recourse cost. Thus, we formulate our problem as a two-stage stochastic programming model. In particular, the first-stage problem determines the locations of disassembly plants, machines procurement, assignments of machines and tasks, and open workstations. The second-stage problem decides the transportation of EOL products and components, and the inventory levels of EOL products and components under each scenario, conforming to the first-stage decisions. The decision variables are defined as follows.

First-stage decision variables:

— x_k : binary variable, equal to 1 if a disassembly plant is set up at location k, 0 otherwise, where $k \in K$;

— y_{kmw} : binary variable, equal to 1 if machine m is procured and assigned to workstation w in disassembly plant k, 0 otherwise, where $k \in K$, $m \in M$, $w \in W_k$;

— z_{kw} : binary variable, equal to 1 if workstation w is opened and used in disassembly plant k, 0 otherwise, where $k \in K$, $w \in W_k$;

— v_{kp} : binary variable, equal to 1 if product p is dismantled in disassembly plant k, 0 otherwise, where $k \in K$, $p \in P$;

— u_{kmwr} : binary variable, equal to 1 if task r is assigned to machine m on workstation w in disassembly plant k, 0 otherwise, where $k \in K$, $m \in M$, $w \in W_k$, $r \in R$;

Second-stage decision variables:

— q_{ijpts}^{S-C} : the quantity of EOL product p transported from supply point i to collection center j during period t under scenario s, where $i \in I$, $j \in J$, $p \in P$, $t \in T$, $s \in S$;

— q_{jkpts}^{C-D} : the quantity of EOL product p transported from collection center j to disassembly plant k during period t under scenario s, where $j \in J$, $k \in K$, $p \in P$, $t \in T$, $s \in S$;

— q_{klcts}^{D-R} : the quantity of component c transported from disassembly plant k to remanufacturing plant l during period t under scenario s, where $k \in K$, $l \in L$, $c \in C$, $t \in T$, $s \in S$;

— τ_{jpts}^{IC} : the inventory of EOL product p at collection center j during period t under scenario s, where $j \in J$, $p \in P$, $t \in T$, $s \in S$;

— τ_{kcts}^{ID} : the inventory of component c at disassembly plant k during period t under scenario s, where $k \in K$, $c \in C$, $t \in T$, $s \in S$;

Let $\xi_s \in R$ denote the objective value of the second-stage problem under scenario s, then model P1 is formulated as follows:

$$\textbf{P1:} \quad \max\ F_1 = \sum_{k \in K} \left(-c_k^S x_k - \sum_{m \in M} \sum_{w \in W_k} c_m^P y_{kmw} - \sum_{w \in W_k} c_k^W z_{kw} \right) + \sum_{s \in S} \rho_s \xi_s \tag{5.1}$$

$$\min\ F_2 = \sum_{k \in K} \sum_{m \in M} \sum_{w \in W_k} e_m y_{kmw} \tag{5.2}$$

The first-stage problem:

s.t. $$x_k + x_{k'} \leq 1 + h_{kk'},\ \forall k, k' \in K | k < k' \tag{5.3}$$

$$v_{kp} \leq x_k,\ \forall k \in K, p \in P \tag{5.4}$$

$$\sum_{m \in M} \sum_{w \in W_k} u_{kmwr} = v_{kp},\ \forall k \in K, p \in P, r \in R_p \tag{5.5}$$

$$u_{kmwr} \leq a_{mr} y_{kmw},\ \forall k \in K, m \in M, w \in W_k, r \in R \tag{5.6}$$

$$\sum_{k \in K} \sum_{w \in W_k} y_{kmw} \leq 1,\ \forall m \in M \tag{5.7}$$

$$z_{kw} \leq x_k,\ \forall k \in K, w \in W_k \tag{5.8}$$

$$u_{kmwr} \leq z_{kw},\ \forall k \in K, m \in M, w \in W_k, r \in R \tag{5.9}$$

$$z_{kw} \leq z_{k(w-1)},\ \forall k \in K, w \in W_k \backslash \{1\} \tag{5.10}$$

$$\sum_{m \in M} \sum_{w \in W_k} w u_{kmwr} \leq \sum_{m \in M} \sum_{w \in W_k} w u_{kmwr'},\ \forall k \in K, r \in R, r' \in B_r \tag{5.11}$$

$$Pr\left\{ \sum_{m \in M} \sum_{r \in R} pt_r u_{kmwr} \leq CT_k,\ \forall w \in W_k \right\} \geq 1 - \alpha,\ \forall k \in K \tag{5.12}$$

$$x_k, y_{kmw}, z_{kw}, v_{kp}, u_{kmwr} \in \{0, 1\},\ \forall k \in K, m \in M, w \in W_k, p \in P, r \in R \tag{5.13}$$

The second-stage problem (for each scenario $s \in S$):

$$\xi_s = \max \sum_{k \in K} \sum_{l \in L} \sum_{c \in C} \sum_{t \in T} \sigma_c^R q_{klcts}^{D-R} - \sum_{i \in I} \sum_{j \in J} \sum_{p \in P} \sum_{t \in T} (c_{ijp}^{S-C} + c_p^C) q_{ijpts}^{S-C} - \sum_{j \in J} \sum_{p \in P} \sum_{t \in T} c_p^{IP} \tau_{jpts}^{IC}$$

$$- \sum_{j \in J} \sum_{k \in K} \sum_{p \in P} \sum_{t \in T} (c_{jkp}^{C-D} + c_p^D) q_{jkpts}^{C-D} - \sum_{k \in K} \sum_{l \in L} \sum_{c \in C} \sum_{t \in T} c_{klc}^{D-R} q_{klcts}^{D-R} - \sum_{k \in K} \sum_{c \in C} \sum_{t \in T} c_c^{IC} \tau_{kcts}^{ID} \tag{5.14}$$

$$\sum_{j \in J} q_{ijpts}^{S-C} \leq \theta_{ip} O_{ipts},\ \forall i \in I, p \in P, t \in T \tag{5.15}$$

$$\sum_{p \in P} \iota_p^P \tau_{jpts}^{IC} \leq V_j,\ \forall j \in J, t \in T \tag{5.16}$$

$$\tau_{jpts}^{IC} = \tau_{jp(t-1)s}^{IC} + \sum_{i \in I} q_{ijpts}^{S-C} - \sum_{k \in K} q_{jkpts}^{C-D},\ \forall j \in J, p \in P, t \in T \tag{5.17}$$

$$\tau_{jp0s}^{IC} = 0,\ \forall j \in J, p \in P \tag{5.18}$$

$$\sum_{p \in P} \sum_{j \in J} \omega_p q_{jkpts}^{C-D} \leq E_k,\ \forall k \in K, t \in T \tag{5.19}$$

$$\sum_{j\in J} \omega_p q^{C-D}_{jkpts} \leq E_k v_{kp},\ \forall k \in K, p \in P, t \in T \tag{5.20}$$

$$\sum_{c\in C} \iota^C_c \tau^{ID}_{kcts} \leq U_k x_k,\ \forall k \in K, t \in T \tag{5.21}$$

$$\tau^{ID}_{kcts} = \tau^{ID}_{kc(t-1)s} + \sum_{j\in J}\sum_{p\in P} \delta_{cp} q^{C-D}_{jkpts} - \sum_{l\in L} q^{D-R}_{klcts},\ \forall k \in K, c \in C, t \in T \tag{5.22}$$

$$\tau^{ID}_{kc0s} = 0,\ \forall k \in K, c \in C \tag{5.23}$$

$$\sum_{k\in K} q^{D-R}_{klcts} \leq D_{lcts},\ \forall l \in L, c \in C, t \in T \tag{5.24}$$

$$q^{S-C}_{ijpts}, q^{C-D}_{jkpts}, q^{D-R}_{klcts}, \tau^{IC}_{jpts}, \tau^{ID}_{kcts} \geq 0,\ \forall i \in I, j \in J, k \in K, l \in L, p \in P, c \in C, t \in T \tag{5.25}$$

Objective F_1 maximizes the expected profit, which is calculated by subtracting the total cost from the total revenue of satisfying component demand. The total cost consists of the first-stage cost, i.e., the costs to set up disassembly plants, procure machines and open workstations, and the expected second-stage cost, including the costs to preprocess and disassemble EOL products, transport EOL products and components, store EOL products and components. Objective F_2 minimizes the total CO_2 emissions of machines in all disassembly plants.

The constraints from (5.3) to (5.13) constitute the first-stage problem. Constraints (5.3) restrict that disassembly plants cannot be set up at the two candidate locations simultaneously if the distance between them is smaller than the smallest segregation requirement. Constraints (5.4) represent that only the open disassembly plants can dismantle EOL products. Constraints (5.5) guarantee that all the tasks of one EOL product should be assigned when the product is dismantled, and each task can only be assigned to one machine in one workstation. Constraints (5.6) ensure that tasks can only be assigned to the procured machines that are eligible. Constraints (5.7) indicate that each machine can be assigned to at most one workstation in one disassembly plant. Constraints (5.8) restrict that workstations can only open within the established disassembly plants. Constraints (5.9) indicate that the tasks can only be assigned to the opened workstations. Constraints (5.10) ensure that workstations are opened in ascending order of their indexes. Constraints (5.11) respect task precedence relationships. Constraints (5.12) ensure that the probability of the processing times of all workstations not exceeding the cycle time is equal

to or larger than $1-\alpha$ at all open disassembly plants. Constraints (5.13) define the ranges of the first-stage decision variables.

The constraints from (5.14) to (5.25) form the second-stage problem. Constraint (5.14) defines the objective function of the second-stage problem, which is the revenue of satisfying component demand minus the costs to preprocess and disassemble EOL products, transport EOL products and components, and store EOL products and components. Constraints (5.15) ensure that the transported volumes of EOL products from different supply points to collection centers do not exceed the supply volumes that are worth disassembling at all supply points. Constraints (5.16) respect the EOL product inventory capacities of collection centers. Constraints (5.17) ensure the EOL products flow conservations of collection centers. Constraints (5.18) regulate the initial inventory of EOL products at collection centers. Constraints (5.19) ensure that the total amount of EOL products dismantled in each disassembly plant is smaller than or equal to its production capacity. Constraints (5.20) indicate that each kind of EOL product should respect the production capacity of disassembly plants. Constraints (5.21) respect the components inventory capacities of disassembly plants. Constraints (5.22) ensure the components flow conservations of disassembly plants. Constraints (5.23) regulate the initial components inventory at disassembly plants. Constraints (5.24) ensure that the total supply of components does not exceed the demand of remanufacturing plants. Constraints (5.25) define the domains of the second-stage decision variables.

Model P1 is nonlinear because of joint chance constraints (5.12). Therefore, it can not be directly solved by calling commercial solvers. Hence, model P1 is firstly linearized and transformed into a distribution-free model in the following.

5.1.3 Approximated linear distribution-free model

Similar to Chapter 4, the joint chance constraints (5.12) can be conservatively transformed into the following ones:

$$\sum_{m\in M}\sum_{r\in R}(pd_r+\nu_r)u_{kmwr}\leq CT_k,\ \forall k\in K,\ w\in W_k \tag{5.26}$$

Thus, model P1 can be approximated into the following linear distribution-free model (P2):

$$\textbf{P2:}\quad \max\ F_1=\sum_{k\in K}\left(-c_k^S x_k-\sum_{m\in M}\sum_{w\in W_k}c_m^P y_{kmw}-\sum_{w\in W_k}c_k^W z_{kw}\right)+\sum_{s\in S}\rho_s\xi_s$$

$$\min\ F_2=\sum_{k\in K}\sum_{m\in M}\sum_{w\in W_k}e_m y_{kmw}$$

$$s.t.\quad (5.3)-(5.11),(5.13)-(5.25),(5.26)$$

To solve the bi-objective model P2, an exact ε-constrained based method is devised in the following.

5.2 The ε-constrained Based Method

In this section, we first describe the framework of the ε-constrained method, and then propose an improved Benders decomposition (IBD) to exactly solve the transformed single objective problems.

5.2.1 The ε-constrained method framework

The basic idea of ε-constrained method is to transform the bi-objective problem into a set of single-objective problems by setting one objective function as the ε-constraint, and then iteratively solve the single-objective problems by varying the value of ε (Ehrgott and Gandibleux, 2002).

In this study, to resolve the single objective problems, the first objective is set as the principal

objective, and the second objective is transformed into the ε-constraint. Accordingly, we can obtain the single-objective mixed-integer linear programming model P3(ε_n) at the n-th iteration:

$$\textbf{P3}(\boldsymbol{\varepsilon_n}): \quad \max\ F_1 = \sum_{k\in K}\left(-c_k^S x_k - \sum_{m\in M}\sum_{w\in W_k} c_m^P y_{kmw} - \sum_{w\in W_k} c_k^W z_{kw}\right) + \sum_{s\in S}\rho_s\xi_s$$

$$s.t. \quad (5.3)-(5.11), (5.13)-(5.25), (5.26)$$

$$\sum_{k\in K}\sum_{m\in M}\sum_{w\in W_k} e_m y_{kmw} \le \varepsilon_n \tag{5.27}$$

where $\varepsilon_n = \varepsilon_{n-1} - \Delta$, ε_{n-1} is the CO_2 emission at $(n-1)$-th iteration, and Δ denotes the step size.

To determine the interval of ε_n, the following single objective problems $\mathrm{P}^{\mathrm{I}}_{\mathrm{F}_1}, \mathrm{P}^{\mathrm{N}}_{\mathrm{F}_1}, \mathrm{P}^{\mathrm{I}}_{\mathrm{F}_2}, \mathrm{P}^{\mathrm{N}}_{\mathrm{F}_2}$ are are formed as follows:

$\mathbf{P^I_{F_1}}$: $F_1^I = \max F_1$

$s.t.$ (5.3) − (5.11), (5.13) − (5.25), (5.26)

$\mathbf{P^N_{F_1}}$: $F_1^N = \max F_1$

$s.t.$ (5.3) − (5.11), (5.13) − (5.25), (5.26)

$F_2 = F_2^I$

$\mathbf{P^I_{F_2}}$: $F_2^I = \min F_2$

$s.t.$ (5.3) − (5.11), (5.13) − (5.25), (5.26)

$\mathbf{P^N_{F_2}}$: $F_2^N = \min F_2$

$s.t.$ (5.3) − (5.11), (5.13) − (5.25), (5.26)

$F_1 = F_1^I$

By solving the four models by calling the IBD approach that is detailed in Subsection 5.2.2, the interval of ε_n is formed by the optimal values F_2^I and F_2^N of problems $\mathrm{P}^{\mathrm{I}}_{\mathrm{F}_2}$ and $\mathrm{P}^{\mathrm{N}}_{\mathrm{F}_2}$. Furthermore, two Pareto points are found as (F_1^I, F_2^N) and (F_1^N, F_2^I). Note that the step size is set as 1, i.e., $\Delta = 1$ (the smallest unit of CO_2 emission). Then, the single-objective model P3(ε_n) is solved by calling IBD. Figure 5.2 summarizes the framework of the ε-constrained based method.

5.2.2 Improved Benders decomposition

Benders decomposition (BD) proposed by Benders (1962) is an iterative method and has been widely studied in the past few decades. In the BD procedure, at each iteration, the complex variables (usually integer variables) and corresponding constraints form a master problem (MP) (Rah-

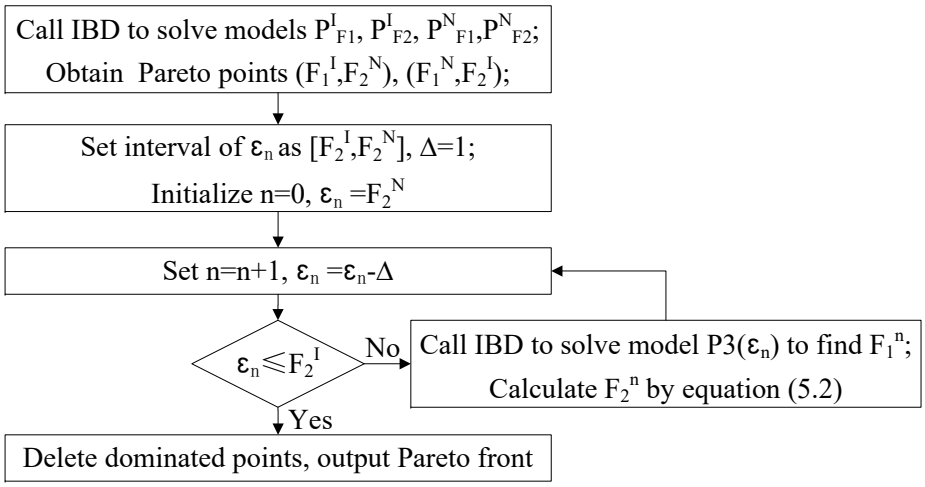


Figure 5.2: The framework of ε-constrained based method

maniani et al., 2017), which is exactly solved to obtain an upper bound (UB) (for a maximization problem). With the solution of MP, the remaining constraints and variables form a sub-problem (SP) and corresponding dual problem (DSP). Then, the DSP is exactly solved to update the lower bound (LB) that corresponds to a feasible solution for a maximization problem. The newly obtained solutions from DSP and generated optimal cuts are used to form a new MP for the next iteration. The process continues until the termination criteria are met. The obtained optimal solution is output.

Compared with the classical BD, the proposed improved Benders decomposition (IBD) includes four accelerate strategies: 1) at each iteration, IBD proposes several valid inequalities, while BD has no valid inequalities; 2) the MP of IBD contains a dynamic updated subset of scenarios, whereas the MP of BD is often a deterministic problem; 3) the IBD generates optimal cuts based on subsets of scenarios, but BD generates one optimal cut for each scenario; 4) IBD proposes additional optimal cuts compared to the traditional BD.

Figure 5.3 depicts the framework of IBD. Firstly, we decompose the sets of scenarios into S_n^I and S_n^{II}, and construct valid inequalities based on problem properties of model P3(ε_n). With the first-stage of model P3(ε_n), S_n^I and valid inequalities, MP$_n$ is formed and exactly solved to update UB. Then, SP$_n$ is formed based on the second-stage of model P3(ε_n), solution of MP$_n$, and S_n^{II}. Then, the corresponding DSP$_n$ is constructed and exactly solved to update LB. Next, scenarios of S_n^{II} are divided into $|G|$ groups based on the non-increasing order of components demand.

Based on the optimal solution of DSP_n, an optimal cut is formed for each group. If UB is larger than LB, an auxiliary problem (AP_n) is formed based on DSP_n, and exactly solved to generate additional optimal cuts. Set $n = n + 1$, then S_n^I and S_n^{II} are updated based on the scenarios in which the demand is not satisfied, and valid inequalities are updated based on the LB. MP_n is updated with the obtained optimal, and additional optimal cuts, S_n^I, and valid inequalities. The process repeats until UB is smaller than or equal to LB. Then, the solution of LB is output as an optimal solution.

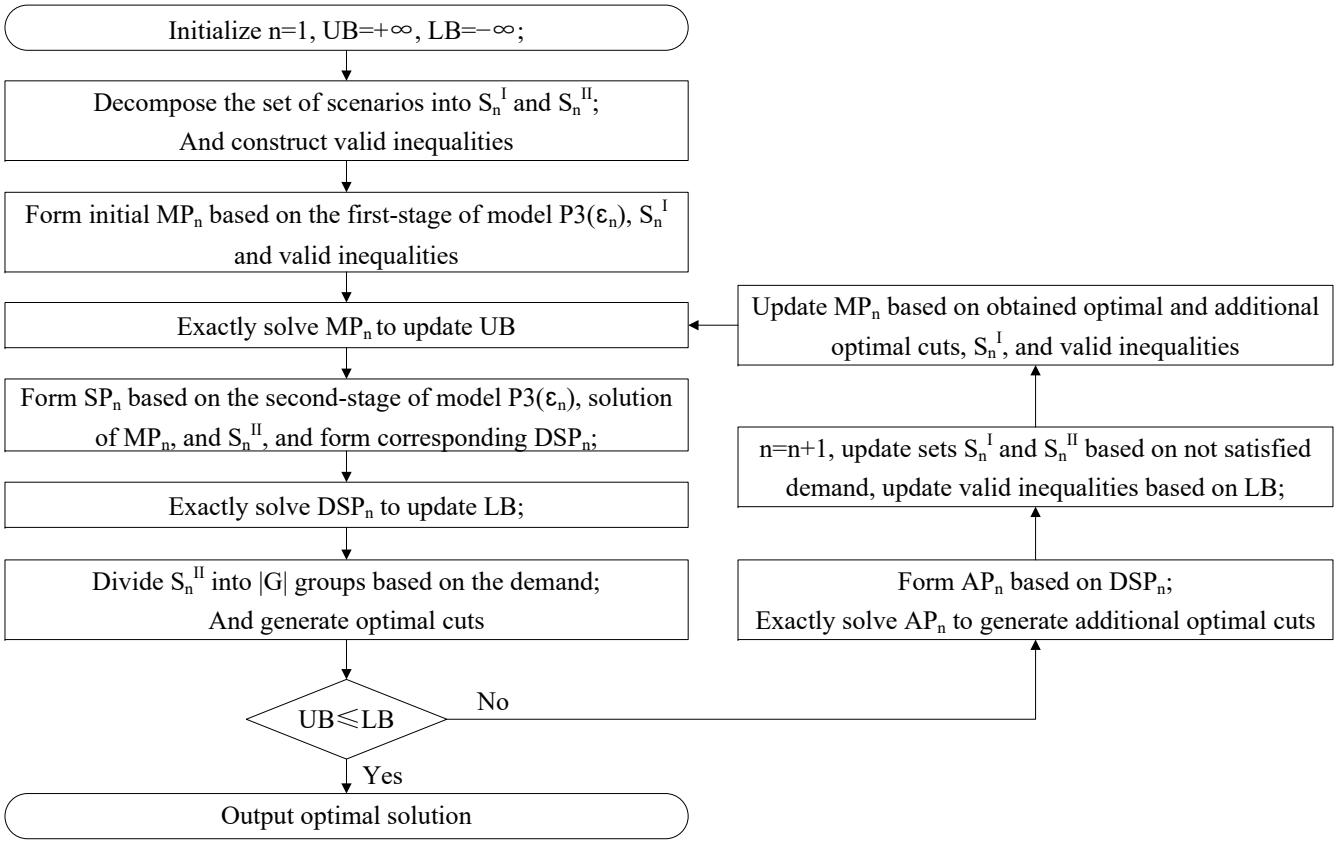


Figure 5.3: The framework of improved Benders decomposition

In the following, we detail the formulations of MP, SP and DSP, valid inequalities construction, S_n^I and S_n^{II} formation, scenario grouping related to S_n^{II} and the additional optimal cuts.

The master problem

At the n$-th$ iteration ($n \geq 1$), MP_n is formed with the first stage of model P3(ε_n), S_n^I, two valid inequalities, and obtained optimal and additional optimal cuts. Accordingly, MP_n can be presented

as follows:

$$\textbf{MP}_{\boldsymbol{n}}: \quad \max\ F_1 = \sum_{k\in K}\left(-c_k^S x_k - \sum_{m\in M}\sum_{w\in W_k} c_m^P y_{kmw} - \sum_{w\in W_k} c_k^W z_{kw}\right) + \sum_{s\in S_n^I}\rho_s\xi_s + \sum_{g\in G}\xi_g$$

$$s.t. \quad (5.3)-(5.11), (5.13), (5.26)-(5.27)$$

$$\xi_s \le \xi_s^{max},\ \forall s \in S \tag{5.28}$$

$$\sum_{k\in K}\left(c_k^S x_k + \sum_{m\in M}\sum_{w\in W_k} c_m^P y_{kmw} + \sum_{w\in W_k} c_k^W z_{kw}\right) \le \sum_{s\in S}\rho_s\xi_s^{max} - LB_n \tag{5.29}$$

$$\xi_g \le \sum_{s\in S_g}\rho_s\left(\sum_{i\in I}\sum_{p\in P}\sum_{t\in T}\theta_{ip}O_{ipts}\tilde{\psi}_{ipts}^{15} + \sum_{j\in J}\sum_{t\in T}V_j\tilde{\psi}_{jts}^{16} + \sum_{k\in K}\sum_{t\in T}E_k\tilde{\psi}_{kts}^{19} + \sum_{k\in K}\sum_{p\in P}\sum_{t\in T}E_k\tilde{\psi}_{kpts}^{45}v_{kp}\right.$$
$$\left. + \sum_{k\in K}\sum_{t\in T}U_k\tilde{\psi}_{kts}^{46}x_k + \sum_{l\in L}\sum_{c\in C}\sum_{t\in T}D_{lcts}\tilde{\psi}_{lcts}^{24}\right),\ \forall g \in G \tag{5.30}$$

$$\xi_g \le \sum_{s\in S_g}\rho_s\left(\sum_{i\in I}\sum_{p\in P}\sum_{t\in T}\theta_{ip}O_{ipts}\tilde{\psi}_{ipts}^{15*} + \sum_{j\in J}\sum_{t\in T}V_j\tilde{\psi}_{jts}^{16*} + \sum_{k\in K}\sum_{t\in T}E_k\tilde{\psi}_{kts}^{19*} + \sum_{k\in K}\sum_{p\in P}\sum_{t\in T}E_k\tilde{\psi}_{kpts}^{45*}v_{kp}\right.$$
$$\left. + \sum_{k\in K}\sum_{t\in T}U_k\tilde{\psi}_{kts}^{46*}x_k + \sum_{l\in L}\sum_{c\in C}\sum_{t\in T}D_{lcts}\tilde{\psi}_{lcts}^{24*}\right),\ \forall g \in G \tag{5.31}$$

$$\xi_s = \sum_{k\in K}\sum_{l\in L}\sum_{c\in C}\sum_{t\in T}\sigma_c^R q_{klcts}^{D-R} - \sum_{i\in I}\sum_{j\in J}\sum_{p\in P}\sum_{t\in T}(c_{ijp}^{S-C} + c_p^C)q_{ijpts}^{S-C} - \sum_{j\in J}\sum_{p\in P}\sum_{t\in T}c_p^{IP}\tau_{jpts}^{IC}$$
$$- \sum_{j\in J}\sum_{k\in K}\sum_{p\in P}\sum_{t\in T}(c_{jkp}^{C-D} + c_p^D)q_{jkpts}^{C-D} - \sum_{k\in K}\sum_{l\in L}\sum_{c\in C}\sum_{t\in T}c_{klc}^{D-R}q_{klcts}^{D-R} - \sum_{k\in K}\sum_{c\in C}\sum_{t\in T}c_c^{IC}\tau_{kcts}^{ID},\ \forall s \in S_n^I \tag{5.32}$$

$$\sum_{j\in J} q_{ijpts}^{S-C} \le \theta_{ip}O_{ipts},\ \forall s \in S_n^I, i \in I, p \in P, t \in T \tag{5.33}$$

$$\sum_{p\in P}\iota_p^P\tau_{jpts}^{IC} \le V_j,\ \forall s \in S_n^I, j \in J, t \in T \tag{5.34}$$

$$\tau_{jpts}^{IC} = \tau_{jp(t-1)s}^{IC} + \sum_{i\in I}\theta_{ip}q_{ijpts}^{S-C} - \sum_{k\in K}q_{jkpts}^{C-D},\ \forall s \in S_n^I, j \in J, p \in P, t \in T \tag{5.35}$$

$$\tau_{jp0s}^{IC} = 0,\ \forall s \in S_n^I, j \in J, p \in P \tag{5.36}$$

$$\sum_{j\in J}\sum_{p\in P}\omega_p q_{jkpts}^{C-D} \le E_k,\ \forall s \in S_n^I, k \in K, t \in T \tag{5.37}$$

$$\sum_{j\in J}\omega_p q_{jkpts}^{C-D} \le E_k v_{kp},\ \forall s \in S_n^I, k \in K, p \in P, t \in T \tag{5.38}$$

$$\sum_{c\in C} \iota_c^C \tau_{kcts}^{ID} \leq U_k x_k,\ \forall s \in S_n^I, k \in K, t \in T \tag{5.39}$$

$$\tau_{kcts}^{ID} = \tau_{kc(t-1)s}^{ID} + \sum_{j\in J}\sum_{p\in P} \delta_{cp} q_{jkpts}^{C-D} - \sum_{l\in L} q_{klcts}^{D-R},\ \forall s \in S_n^I, k \in K, c \in C, t \in T \tag{5.40}$$

$$\tau_{kc0s}^{ID} = 0,\ \forall s \in S_n^I, k \in K, c \in C \tag{5.41}$$

$$\sum_{k\in K} q_{klcts}^{D-R} \leq D_{lcts},\ \forall s \in S_n^I, l \in L, c \in C, t \in T \tag{5.42}$$

$$q_{ijpts}^{S-C}, q_{jkpts}^{C-D}, q_{klcts}^{D-R}, \tau_{jpts}^{IC}, \tau_{kcts}^{ID} \geq 0,\ \forall s \in S_n^I, i \in I, j \in J, k \in K, l \in L, p \in P, c \in C, t \in T \tag{5.43}$$

$$\xi_s, \xi_g \in R,\ \forall s \in S, s \in G \tag{5.44}$$

where constraints (5.28) and (5.29) are valid inequalities, and constraints (5.30) and (5.31) are optimal and additional optimal cuts that will detailed later.

At the (n+1)$-th$ iteration, constraint (5.29), S_{n+1}^I and S_{n+1}^{II} will be updated, and new generated optimal and additional optimal cuts (constraints (5.30) and (5.31)) will be added to MP$_n$ to form MP$_{n+1}$. Note that the initial MP, i.e., MP$_1$ does not contain constraints (5.30) and (5.31). By exactly solving MP$_n$, we can obtain an UB_n, and the corresponding solutions are denoted by $(\tilde{x}_k^n, \tilde{y}_{kmw}^n, \tilde{z}_{kw}^n, \tilde{v}_{kp}^n, \tilde{u}_{kmwr}^n)$.

The sub-problem

As presented above, the SP decides the transportation and inventory levels of EOL products and components, which can be solved individually under each scenario. In model P3(ε_n), constraints (5.20) and (5.21) connecting MP and SP. Moreover, the two decision variables (x_k^n, v_{kp}^n) are fixed when solving SP at the n-th iteration, which are denoted by $\tilde{x}_k^n$ and $\tilde{v}_{kp}^n$, respectively. Then, for each scenario $s \in S_n^{II}$, the SP at n-th iteration is presented as follows:

$$\begin{aligned}\textbf{SP}_{\boldsymbol{n}}^{\boldsymbol{s}}: \quad \max & \sum_{k\in K}\sum_{l\in L}\sum_{c\in C}\sum_{t\in T} \sigma_c^R q_{klcts}^{D-R} - \sum_{i\in I}\sum_{j\in J}\sum_{p\in P}\sum_{t\in T} (c_{ijp}^{S-C} + c_p^C) q_{ijpts}^{S-C} \\ & - \sum_{j\in J}\sum_{k\in K}\sum_{p\in P}\sum_{t\in T} (c_{jkp}^{C-D} + c_p^D) q_{jkpts}^{C-D} - \sum_{k\in K}\sum_{l\in L}\sum_{c\in C}\sum_{t\in T} c_{klc}^{D-R} q_{klcts}^{D-R}\end{aligned}$$

$$- \sum_{j\in J}\sum_{p\in P}\sum_{t\in T} c_p^{IP}\tau_{jpts}^{IC} - \sum_{k\in K}\sum_{c\in C}\sum_{t\in T} c_c^{IC}\tau_{kcts}^{ID}$$

$$s.t. \quad (5.15)-(5.19), (5.22)-(5.25)$$

$$\sum_{j\in J}\omega_p q_{jkpts}^{C-D} \le E_k\tilde{v}_{kp}^n,\ \forall k\in K, p\in P, t\in T \tag{5.45}$$

$$\sum_{c\in C}\iota_c^C\tau_{kcts}^{ID} \le U_k\tilde{x}_k^n,\ \forall k\in K, t\in T \tag{5.46}$$

In order to generate optimal cuts, the DSP_n^s can be solved instead of directly solving SP_n^s.

The corresponding dual sub-problem

Since the SP_n^s belongs to continuous linear programming, according to the linear programming duality (Tyndall, 1965), all constraints of SP_n^s correspond to variables of DSP_n^s, and all variables of SP_n^s correspond to constraints of DSP_n^s.

To formulate the DSP, we first introduce the dual variables. Let ψ_{ipts}^{15}, ψ_{jts}^{16}, ψ_{jpts}^{17}, ψ_{jps}^{18}, ψ_{kts}^{19}, ψ_{kcts}^{22}, ψ_{kcs}^{23}, ψ_{lcts}^{24}, ψ_{kpts}^{45} and ψ_{kts}^{46} denote the dual variables of constraints (5.15)-(5.19), (5.22)-(5.24) and (5.45)-(5.46), where $s\in S_n^{II}$. Then, we construct dual constraints corresponding $q_{ijpts}^{S-C}, q_{jkpts}^{C-D}, q_{klcts}^{D-R}$, $\tau_{jpts}^{IC}, \tau_{kcts}^{ID}$. Thus, for each scenario $s\in S_n^{II}$, the DSP at n-th iteration can be written as follows:

$$\begin{aligned}\textbf{DSP}_{\boldsymbol{n}}^{\boldsymbol{s}}: \quad \min & \sum_{i\in I}\sum_{p\in P}\sum_{t\in T}\theta_{ip}O_{ipts}\psi_{ipts}^{15} + \sum_{j\in J}\sum_{t\in T}V_j\psi_{jts}^{16} + \sum_{k\in K}\sum_{t\in T}E_k\psi_{kts}^{19} \\ & + \sum_{k\in K}\sum_{p\in P}\sum_{t\in T}E_k\tilde{v}_{kp}^n\psi_{kpts}^{45} + \sum_{k\in K}\sum_{t\in T}U_k\tilde{x}_k^n\psi_{kts}^{46} + \sum_{l\in L}\sum_{c\in C}\sum_{t\in T}D_{lcts}\psi_{lcts}^{24} \end{aligned} \tag{5.47}$$

$$s.t. \quad \psi_{ipts}^{15} - \psi_{jpts}^{17} \ge -c_{ijp}^{S-C} - c_p^C,\ \forall i\in I, j\in j, p\in P, t\in T \tag{5.48}$$

$$\psi_{jpts}^{17} + \omega_p\psi_{kts}^{19} + \omega_p\psi_{kpts}^{45} - \sum_{c\in C}\delta_{cp}\psi_{kcts}^{22} \ge -c_{jkp}^{C-D} - c_p^D,\ \forall j\in J, k\in K, p\in P, t\in T \tag{5.49}$$

$$\psi_{kcts}^{22} + \psi_{lcts}^{24} \ge -c_{klc}^{D-R} + \sigma_c^R,\ \forall k\in K, l\in L, c\in C, t\in T \tag{5.50}$$

$$\iota_p^P\psi_{jts}^{16} + \psi_{jpts}^{17} - \psi_{jp(t+1)}^{17} \ge -c_p^{IP},\ \forall j\in J, p\in P, t\in T\setminus\{0,|T|\} \tag{5.51}$$

$$\iota_p^P\psi_{jts}^{16} + \psi_{jps}^{18} - \psi_{jp(t+1)}^{17} \ge -c_p^{IP},\ \forall j\in J, p\in P, t=0 \tag{5.52}$$

$$\iota_p^P\psi_{jts}^{16} + \psi_{jpts}^{17} \ge -c_p^{IP},\ \forall j\in J, p\in P, t=|T| \tag{5.53}$$

$$\iota_c^C \psi_{kts}^{46} + \psi_{kcts}^{22} - \psi_{kc(t+1)}^{22} \geq -c_c^{IC},\ \forall k \in K, c \in C, t \in T \setminus \{0, |T|\} \tag{5.54}$$

$$\iota_c^C \psi_{kts}^{46} + \psi_{kcs}^{23} - \psi_{kc(t+1)}^{22} \geq -c_c^{IC},\ \forall k \in K, c \in C, t = 0 \tag{5.55}$$

$$\iota_c^C \psi_{kts}^{46} + \psi_{kcts}^{22} \geq -c_c^{IC},\ \forall k \in K, c \in C, t = |T| \tag{5.56}$$

$$\psi_{ipts}^{15}, \psi_{jts}^{16}, \psi_{kts}^{19}, \psi_{lcts}^{24}, \psi_{kpts}^{45}, \psi_{kts}^{46} \geq 0 \tag{5.57}$$

$$\psi_{jpts}^{17}, \psi_{jps}^{18}, \psi_{kcts}^{22}, \psi_{kcs}^{23} \in R \tag{5.58}$$

Since our model possesses complete recourse (i.e., all the SP_n^s are feasible and all the DSP_n^s are bounded), only optimal cuts are generated (Birge and Louveaux, 2011). Denote the solution of DSP_n^s as $\tilde{\psi}_{ipts}^{15}$, $\tilde{\psi}_{jts}^{16}$, $\tilde{\psi}_{jpts}^{17}$, $\tilde{\psi}_{jps}^{18}$, $\tilde{\psi}_{kts}^{19}$, $\tilde{\psi}_{kcts}^{22}$, $\tilde{\psi}_{kcs}^{23}$, $\tilde{\psi}_{lcts}^{24}$, $\tilde{\psi}_{kpts}^{45}$ and $\tilde{\psi}_{kts}^{46}$. Then, the optimal function value of the second-stage problem in the next iteration cannot exceed that of the current iteration. Therefore, the following optimality cuts can be obtained and used to form MP_{n+1}:

$$\begin{aligned}\xi_s \leq & \sum_{i\in I}\sum_{p\in P}\sum_{t\in T}\theta_{ip}O_{ipts}\tilde{\psi}_{ipts}^{15} + \sum_{j\in J}\sum_{t\in T}V_j\tilde{\psi}_{jts}^{16} + \sum_{k\in K}\sum_{t\in T}E_k\tilde{\psi}_{kts}^{19} + \sum_{k\in K}\sum_{p\in P}\sum_{t\in T}E_k\tilde{\psi}_{kpts}^{45}v_{kp} \\ & + \sum_{k\in K}\sum_{t\in T}U_k\tilde{\psi}_{kts}^{46}x_k + \sum_{l\in L}\sum_{c\in C}\sum_{t\in T}D_{lcts}\tilde{\psi}_{lcts}^{24},\ \forall s \in S\end{aligned} \tag{5.59}$$

Valid inequalities

According to model P3(ε_n), the objective value of second-stage problem ξ_s can vary from $-\infty$ to $+\infty$ (see page 80). The first valid inequality ((5.28)) attempts to narrow the range of ξ_s, which might accelerate the convergence speed of the algorithm.

Note that only x_k and v_{kp} associate the first-stage and second-stage problems. Therefore, we assume that all disassembly plants are open and every disassembly plant can handle all EOL products, i.e., $x_k = 1$ and $v_{kp} = 1$ for all $k \in K$, $p \in P$. Then, we can obtain the corresponding optimal objective value ξ_s^{max} of the second-stage problem, where $s \in S$. Naturally, the value of ξ_s in an optimal solution of model P3(ε_n) can not exceed ξ_s^{max}, where $s \in S$. Then, the proposition is presented as follows.

Proposition 5. *Let ξ_s^{max} denote the optimal objective value of second-stage problem with $x_k = 1$ and*

$v_{kp} = 1$ *for all* $k \in K$, $p \in P$ *under scenario* s*, then the following inequality:*

$$\xi_s \leq \xi_s^{max},\ \forall s \in S$$

is a valid inequality for P3(ε_n)*.*

Denote LB as the function value of the best solution of P3 found in all $(n-1)$-th iterations. We hope the function objective value is not smaller than LB. Thus we have the following valid inequality, i.e., $\sum_{k\in K}(-c_k^S x_k - \sum_{m\in M}\sum_{w\in W_k} c_m^P y_{kmw} - \sum_{w\in W_k} c_k^W z_{kw}) + \sum_{s\in S}\rho_s\xi_s \geq LB$. This inequality can be equivalently transformed to the following one: $\sum_{k\in K}(c_k^S x_k + \sum_{m\in M}\sum_{w\in W_k} c_m^P y_{kmw} + \sum_{w\in W_k} c_k^W z_{kw}) \leq \sum_{s\in S}\rho_s\xi_s - LB$. As $\xi_s \leq \xi_s^{max}$, $\forall s \in S$ is true, and $\rho_s \geq 0$, $\forall s \in S$, we also have $\sum_{s\in S}\rho_s\xi_s \leq \sum \rho_s\xi_s^{max}$. Thus, we have the second valid inequality ((5.29)):

Proposition 6. *The following inequality:*

$$\sum_{k\in K}\left(c_k^S x_k + \sum_{m\in M}\sum_{w\in W_k} c_m^P y_{kmw} + \sum_{w\in W_k} c_k^W z_{kw}\right) \leq \sum_{s\in S}\rho_s\xi_s^{max} - LB_n$$

is a valid inequality for P3(ε_{n+1})*.*

The formation of S_n^I and S_n^{II}

To improve the performance of BD, Crainic et al. (2016, 2021) decompose the set of scenarios into two sub-sets S_n^I and S_n^{II} at the beginning of BD to participate in the formations of MP and SP. In the study. we propose a new strategy to dynamically form S_n^I and S_n^{II} at each iteration of IBD.

The strategy consists of the first iteration ($n = 1$), S_1^I is formed by the scenario with the greatest demand in the set of scenario S, and S_1^{II} is formed by the rest of scenarios of S. At n-th iteration, based on the solution of DSP$_{n-1}$, the set of scenarios S_n with non-satisfied demands is formed, S_n^I is updated by adding the scenario with the greatest demand in S_n, i. e. $S_n^I = S_{n-1}^I \bigcup sm$, where sm is the scenario with the biggest demand in S_n, and S_n^{II} is formed by the rest of scenarios of S_n.

Scenario grouping and cut generation

At the n-th ($n > 1$) iteration, MP$_n$ is formed based on obtained optimal and additional optimal cuts, S_n^I, and valid inequalities. In the traditional BD, $|S_n^{II}|$ cuts are used to form MP$_n$. If $|S_n^{II}|$ is large, the formed MP$_n$ is huge and difficult to be solved. In this chapter, the $|S_n^{II}|$ is decomposed into G groups. As $G < |S_n^{II}|$, the formed MP$_n$ is relatively small and can be easily solved.

Before decomposing S_n^{II}, sort the scenarios with a non-increasing order of component demands. Given the parameter G, the first $\lceil |S_n^{II}|/G \rceil$ scenarios are assigned to group 1, and the $\lceil (g-1)|S_n^{II}|/G+1 \rceil$ to $\lceil g|S_n^{II}|/G \rceil$scenarios are assigned to group g, where $\lceil x \rceil$ denotes the smallest integer that is larger than or equal to x. Accordingly, the auxiliary variables ξ_s of set S_n^{II} need to be updated as ξ_g. Then, the optimality cut (5.59) can be rewritten as follows (i.e. (5.30)):

$$\xi_g \leq \sum_{s \in S_g} \rho_s \left(\sum_{i \in I} \sum_{p \in P} \sum_{t \in T} \theta_{ip} O_{ipts} \tilde{\psi}_{ipts}^{15} + \sum_{j \in J} \sum_{t \in T} V_j \tilde{\psi}_{jts}^{16} + \sum_{k \in K} \sum_{t \in T} E_k \tilde{\psi}_{kts}^{19} + \sum_{k \in K} \sum_{p \in P} \sum_{t \in T} E_k \tilde{\psi}_{kpts}^{45} v_{kp} \right.$$
$$\left. + \sum_{k \in K} \sum_{t \in T} U_k \tilde{\psi}_{kts}^{46} x_k + \sum_{l \in L} \sum_{c \in C} \sum_{t \in T} D_{lcts} \tilde{\psi}_{lcts}^{24} \right), \ \forall g \in G$$

Additional optimal cut

When the DSP has multiple optimal solutions, each solution can generate a cut. To this end, we need to generate the strongest cut not dominated by other cuts (Magnanti and Wong, 1981). To generate the additional optimal cut, we first form an auxiliary model AM$_n$ based on MP and solve AM$_n$ to get a core point. Then, we form another auxiliary model AP$_n^s$ based on the information of core point and DSP, and solve AP$_n^s$ to obtain the additional cut.

We adopt a similar way of Adulyasak et al. (2015) to form AM$_n$. Let $\hat{o}$ denote a small positive number, and two auxiliary variables are introduced, i.e., $\hat{ax}_k$ and $\hat{av}_{kp}$. The AM$_n$ can be presented as:

$$\textbf{AM}_{\boldsymbol{n}}: \quad \max \sum_{k \in K} \hat{ax}_k + \sum_{k \in K} \sum_{p \in P} \hat{av}_{kp} \tag{5.60}$$

$$s.t. \quad (5.3) - (5.11), (5.13), (5.26) - (5.29), (5.33) - (5.43)$$

$$x_k - \hat{o}\hat{a}x_k \geq 0,\ \forall k \in K \tag{5.61}$$

$$x_k + \hat{o}\hat{a}x_k \leq 1,\ \forall k \in K \tag{5.62}$$

$$v_{kp} - \hat{o}\hat{a}v_{kp} \geq 0,\ \forall k \in K, p \in P \tag{5.63}$$

$$v_{kp} + \hat{o}\hat{a}v_{kp} \geq 1,\ \forall k \in K, p \in P \tag{5.64}$$

$$\hat{a}x_k, \hat{a}v_{kp} \in \{0,1\},\ \forall k \in K, p \in P \tag{5.65}$$

where Constraints (5.33)-(5.43) only contain one scenario with maximal demand. In addition, this problem is not solved optimally. A feasible solution is enough to get the core point. Therefore, it is easy to solve this problem within one second.

Furthermore, the value of the core point is dynamically updated via $x_k^0 = (1-\lambda)x_k^0 + \lambda\tilde{x}_k^n$ and $v_{kp}^0 = (1-\lambda)v_{kp}^0 + \lambda\tilde{v}_{kp}^n$, where $\tilde{x}_k^n$ and $\tilde{v}_{kp}^n$ are the current solution of MP_n, and usually λ takes the value of 0.5 (de Sá et al., 2013).

With the obtained core point (x_k^0, v_{kp}^0) and the information of DSP_n^s, for $s \in S_n^{II}$, the following AP_n^s can be formed:

$$\begin{aligned}
\textbf{AP}_{\boldsymbol{n}}^{\boldsymbol{s}}\textbf{:}\quad & \min \sum_{i\in I}\sum_{p\in P}\sum_{t\in T}\theta_{ip}O_{ipts}\psi_{ipts}^{15} + \sum_{j\in J}\sum_{t\in T}V_j\psi_{jts}^{16} + \sum_{k\in K}\sum_{t\in T}E_k\psi_{kts}^{19} \\
& + \sum_{k\in K}\sum_{p\in P}\sum_{t\in T}E_k v_{kp}^0\psi_{kpts}^{45} + \sum_{k\in K}\sum_{t\in T}U_k x_k^0\psi_{kts}^{46} + \sum_{l\in L}\sum_{c\in C}\sum_{t\in T}D_{lcts}\psi_{lcts}^{24} \qquad (5.66) \\
s.t.\quad & (5.48) - (5.58) \\
& \sum_{i\in I}\sum_{p\in P}\sum_{t\in T}\theta_{ip}O_{ipts}\psi_{ipts}^{15} + \sum_{j\in J}\sum_{t\in T}V_j\psi_{jts}^{16} + \sum_{k\in K}\sum_{t\in T}E_k\psi_{kts}^{19} \\
& + \sum_{k\in K}\sum_{p\in P}\sum_{t\in T}E_k\tilde{v}_{kp}^n\psi_{kpts}^{45} + \sum_{k\in K}\sum_{t\in T}U_k\tilde{x}_k^n\psi_{kts}^{46} + \sum_{l\in L}\sum_{c\in C}\sum_{t\in T}D_{lcts}\psi_{lcts}^{24} = DSP_n^{s*} \qquad (5.67)
\end{aligned}$$

where DSP_n^{s*} represents the optimal objective value of DSP_n^s. Constraints (5.48)-(5.58) are dual constraints, and Constraint (5.67) forces the solution of AP_n^s is still optimal for DSP_n^s.

By exactly solving AP_n^s, we can obtain the optimal solution that is denoted as $\tilde{\psi}_{ipts}^{15*}$, $\tilde{\psi}_{jts}^{16*}$, $\tilde{\psi}_{jpts}^{17*}$, $\tilde{\psi}_{jps}^{18*}$, $\tilde{\psi}_{kts}^{19*}$, $\tilde{\psi}_{kcts}^{22*}$, $\tilde{\psi}_{kcs}^{23*}$, $\tilde{\psi}_{lcts}^{24*}$, $\tilde{\psi}_{kpts}^{45*}$ and $\tilde{\psi}_{kts}^{46*}$. Then, the following additional optimal cut ((5.31)) can

be obtained and used to form MP_{n+1}:

$$\xi_g \leq \sum_{s\in S_g} \rho_s \left(\sum_{i\in I} \sum_{p\in P} \sum_{t\in T} \theta_{ip} O_{ipts} \tilde{\psi}^{15*}_{ipts} + \sum_{j\in J} \sum_{t\in T} V_j \tilde{\psi}^{16*}_{jts} + \sum_{k\in K} \sum_{t\in T} E_k \tilde{\psi}^{19*}_{kts} + \sum_{k\in K} \sum_{p\in P} \sum_{t\in T} E_k \tilde{\psi}^{45*}_{kpts} v_{kp} \right.$$
$$\left. + \sum_{k\in K} \sum_{t\in T} U_k \tilde{\psi}^{46*}_{kts} x_k + \sum_{l\in L} \sum_{c\in C} \sum_{t\in T} D_{lcts} \tilde{\psi}^{24*}_{lcts} \right), \ \forall g \in G$$

5.3 Numerical Experiments

This section conducts numerical experiments, including one case study and randomly generated instances to demonstrate the usage of our proposed model and the efficiency of proposed methods. All the experiments are conducted on a personal laptop with Core I5 and 3.20 GHz CPU with 12 GB RAM. The program is coded using C++ programming language in Microsoft Visual Studio 2019, and all the models are solved by ILOG CPLEX 22.10 solver with the default setting.

5.3.1 Case study

In this part, a case based on Alumur et al. (2012) is used to illustrate the studied problem. The RSC network comprises 40 cities in Germany, including 26 suppliers, 6 collection centers, 5 candidates of disassembly plants, 3 remanufacturing plants, and the locations of these facilities are shown in Figure 5.4. In this instance, two kinds of EOL products (washing machines and tumble dryers) are considered, and each of them has 4 kinds of most profitable components. Specifically, a washing machine includes one frame (steel), one motor, two ABS parts and one washing tube, and a tumble dryer contains one frame (steel), one motor, two ABS parts, and one air blower. Since Alumur et al. (2012) do not study the disassembly line for EOL products, we generate the disassembly task precedence schemes referring to Koc et al. (2009), which are depicted in Figure 5.5.

The distances between different facilities are obtained using the highway transportation network data from Google Maps, and the transportation costs per kilometer per unit product and component are taken as 0.005 and 0.003 EUR, respectively. The costs to preprocess and disas-

Figure 5.4: The facility locations of the case study

semble one EOL product are 1 and 2 EUR, respectively. The inventory costs of one EOL product and component are 1 and 0.2 EUR, respectively. The revenues of one unit of frame, motor, ABS parts, washing tube and air blower are 17.5, 5.5, 1.4, 1.4 and 2.1 EUR, respectively (Alumur et al., 2012). The mean task processing times are randomly generated in $[10, 50]$, the risk level α, $E[Z_r^2]$ and b_r are set as 0.05, 0.05 and 0.2, respectively, and the cycle time is set as 90 (He et al., 2021). A three-year planning horizon is considered.

The amounts of EOL washing machines and tumble dryers are estimated based on the population of cities in line with Alumur et al. (2012). Specifically, one person generates 19 kilograms of electronic waste per year, and 27.7% of them are considered large household appliances, and

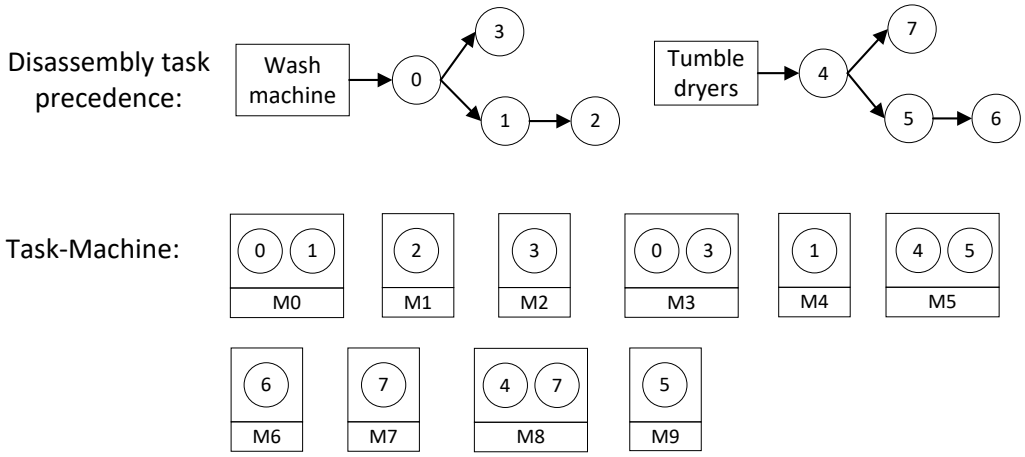


Figure 5.5: The task precedence relationships and machines

30% and 10% are considered washing machines and tumble dryers. The average weights of a washing machine and a tumble dryer are assumed to be 60 kg and 45 kg, and 20% of them can be collected. The demands for components are randomly generated based on the total supply of EOL products (Alumur et al., 2012). The capacity of EOL products at the collection center, capacity of components and production capacity at disassembly plants are randomly generated from the interval $[5000, 10000]$. The inventory capacities needed for one EOL product and one component, and the production capacity needed for one EOL product are randomly generated from the interval $[5, 30]$. The costs to set up a disassembly plant, procure a machine and open a workstation are randomly generated from the intervals $[190000, 210000]$, $[1000, 5000]$, and $[1000, 2000]$ respectively (Haase and Müller, 2014). The CO_2 emissions of machines are randomly generated from the interval $[2000, 5000]$.

The obtained Pareto front is reported in Figure 5.6. We can see that there are 7 non-dominated points, where the total excepted profit increases from 0 to 6.97E+05 EUR, and the corresponding CO_2 emission varies from 0 to 1.56E+04 Kg. For example, point A has the largest profit and the highest CO_2 emission. This is because all the disassembly plants purchase low-tech machines with high CO_2 emissions but with a lower procurement price. In contrast, point D has a profit and CO_2 emission of 0, indicating that no disassembly plants have been set up. Furthermore, the profit of point B is much higher than that of point C. This is because point B set up more disassembly plants that can meet the demand for components.

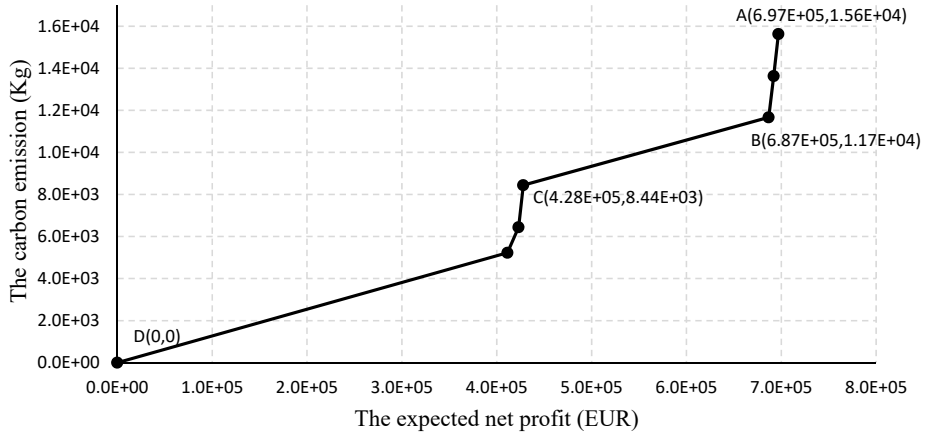


Figure 5.6: The Pareto front of the case study

Considering the solution of Pareto point A, two disassembly plants are set up and located in Aachen and Erfurt (see Figure 5.7), where EOL tumble dryers are dismantled in Aachen, and EOL washing machines are dismantled in Erfurt. The transportation flows of EOL products and components (in period 1) are illustrated in Figure 5.7, where the EOL tumble dryers are transported from collection centers Duisburg and Bonn to Aachen, the EOL washing machines are transported from collection centers Leipzig and Hannover to Erfurt. The components in Aachen are shipped to remanufacturing plants in Mainz and Dortmund, and the components in Erfurt are shipped to the remanufacturing plant in Magdeburg. We can find that most EOL products and components are shipped between facilities that are close to each other, which can reduce transportation costs.

In the disassembly plant located in Aachen, machines 5, 6 and 7 are procured, and two workstations are opened. In Erfurt, machines 0, 1 and 2 are procured, and two workstations are opened. The disassembly line configurations are depicted in Figure 5.8.

Furthermore, we compare the traditional top-down decision-making process with our joint decision-making process (integrated model). Specifically, in the traditional top-down decision-making process, the locations of disassembly plants and the procurement of disassembly equipment are first decided, then the disassembly line balancing, transportation and inventory levels are determined. Table 5.1 reports the detailed costs of two decision processes under Pareto point A. Compared to the top-down decision-making, the joint decision-making spends a larger cost to

Figure 5.7: The opened disassembly plants and optimal EOL products and components transportation

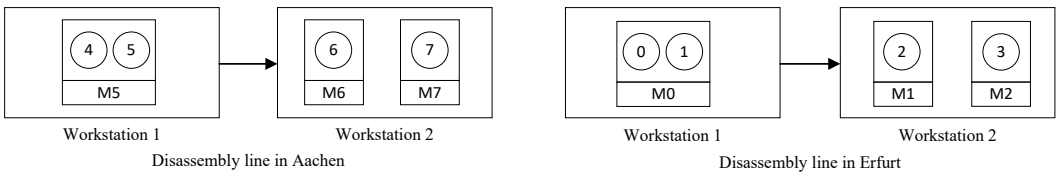


Figure 5.8: The line configurations of opened disassembly plants

set up disassembly plants, while smaller costs to open workstations and transport EOL products and components, and the profit increases from 6.81E+05 EUR to 6.97E+05 EUR. This is because the joint decision-making needs fewer workstations, and the locations of disassembly plants are different, leading to lower transportation costs. Thus, we have the following managerial insight.

Managerial Insight 1. *The joint decision-making process coordinating different levels of decisions can improve the profitability of the whole RSC.*

Table 5.1: The detail costs of top-down and joint decision-making processes

	Cost						Revenue	Net profit
	Disassembly plants	Machines	Workstations	Transportation	Prepossess and Disassembly	Inventory		
Top-down decision	4.00E+05	1.67E+04	5.00E+03	2.25E+05	1.62E+05	2.21E+02	1.49E+06	6.81E+05
Joint decision	4.20E+05	1.67E+04	2.50E+03	1.91E+05	1.62E+05	2.21E+02	1.49E+06	6.97E+05

5.3.2 Experiments on randomly generated instances

In this part, the proposed IBD approach is tested on randomly generated instances, and compared with the sample average approximation (SAA) method, which is widely used to solve the scenario-based stochastic programming model (Long et al., 2012) and has been used to solve the integrated SC and ALB problem (Hamta et al., 2015).

Instances generation

There are no standard benchmark instances available, so we randomly generate instances based on previous studies. According to the random instances reported by Yolmeh and Saif (2021), where the maximal number of components, customers, manufacturers and periods are set as $\{10, 25, 20, 4\}$, we set the maximal numbers of supply points, collection centers, disassembly plants and remanufacturing plants as follows: $|I| = 25, |J| = 20, |K| = 10, |L| = 8$, and the maximal numbers of machines, workstations, tasks, components, products and periods are set as $|M| = 20, |W_k| = 12, |R| = 32, |C| = 10, |P| = 4, |T| = 4$. The product information is adopted from Kalayci and Gupta (2013b). The generated instances are illustrated in Table 5.2.

The number of scenarios is an important parameter for our problem, where a larger number of scenarios may improve the solution accuracy but require much more computational time. To this end, we test 20 problem combinations under two numbers of scenarios, i.e., $S = 10$ and $S = 100$, respectively, where the maximal number of scenarios is set as 100 according to Haase and Müller (2014). Furthermore, each combination contains five randomly generated instances and the average results are reported.

The locations of supply points, collection centers, disassembly plants and remanufacturing plants are randomly generated in a square with a side length of 100 units, and the Euclidean distances are used to calculate the shipping costs (Haase and Müller, 2014). The supply of EOL

Table 5.2: The parameters of generated instances

Combinations	Supply points ($\|I\|$)	Collection centers ($\|J\|$)	Disassembly plants ($\|K\|$)	Remanufacturing plants ($\|L\|$)	Periods ($\|T\|$)	Products ($\|P\|$)	Tasks ($\|R\|$)	Components ($\|C\|$)	machines ($\|M\|$)
1	10	5	4	2	2	2	16	5	10
2	10	5	4	2	2	3	24	5	15
3	10	5	4	2	3	2	16	5	10
4	10	5	4	3	2	2	16	6	10
5	10	5	4	3	2	3	24	6	15
6	15	10	6	3	3	2	16	6	10
7	15	10	6	4	2	2	16	7	10
8	15	10	6	4	2	3	24	7	15
9	15	10	6	4	3	2	16	7	10
10	15	10	6	5	2	2	16	7	10
11	20	15	8	5	2	3	24	8	15
12	20	15	8	5	3	2	16	8	10
13	20	15	8	6	2	2	16	8	10
14	20	15	8	6	2	3	24	8	15
15	20	15	8	6	3	2	16	9	10
16	25	20	10	7	2	2	16	9	10
17	25	20	10	7	2	3	24	9	15
18	25	20	10	7	3	3	24	9	15
19	25	20	10	8	4	3	24	10	15
20	25	20	10	8	3	4	32	10	20

products and demand for components are generated from the uniform distributions $U(500, 3000)$ and $U(5000, 30000)$, respecting the real supply and demand of the case study. Other parameters are set in line with those of the case study. Moreover, the numbers of components obtained from one EOL product are randomly generated in $[1, 5]$ (He et al., 2022). Table 5.3 summarizes the input parameters.

Table 5.3: Input parameters of random instances

Parameters	Value	Parameters	Value
V_j	$U(5000, 10000)$	c_k^W	$U(1000, 2000)$
ι_p^P	$U(5, 30)$	c_p^C	1
U_k	$U(5000, 10000)$	c_p^D	2
ι_c^C	$U(5, 30)$	c_p^{IP}	1
E_k	$U(5000, 10000)$	c_c^{IC}	0.2
ω_p	$U(5, 30)$	σ_c^R	$u(5, 20)$
O_{ipts}	$U(500, 3000)$	δ_{cp}	$[1, 5]$
D_{lcts}	$U(5000, 30000)$	pt_r	$[10, 50]$
c_k^S	$U(190000, 210000)$	CT_k	90
c_m^P	$U(1000, 5000)$	α	0.05
e_m	$U(2000, 5000)$	$(E[Z_r^2], b_r)$	$(0.05, 0.2)$

Computational results

Table 5.4 reports the experimental results of 20 different problem combinations under scenario $|S| = 10$, where N_P^{SAA} and N_P^{IBD} represent the average numbers of Pareto points found by SAA and IBD, respectively. T_{SAA} and T_{IBD} denote the average computational times of SAA and IBD, respectively. N_{opt}^{SAA} and N_{opt}^{IBD} represent the number of instances in each combination where all the Pareto points are found by SAA and IBD in 36000 seconds, respectively. Moreover, in terms of computational times, the improvements of IBD over SAA are calculated by $(T_{SAA} - T_{IBD}/T_{SAA}) \times 100\%$ and reported in Column 8.

Table 5.4: Experimental results under scenario number $|S| = 10$

Combinations	SAA			IBD			Improvements
	N_P^{SAA}	$T_{SAA}(s)$	N_{opt}^{SAA}	N_P^{IBD}	$T_{IBD}(s)$	N_{opt}^{IBD}	
1	3.4	3.0	5	3.4	5.5	5	-81.7%
2	4.4	6.6	5	4.4	17.7	5	-166.6%
3	6.0	11.5	5	6.0	23.8	5	-106.9%
4	6.0	8.9	5	6.0	19.6	5	-119.3%
5	4.6	10.9	5	4.6	47.7	5	-338.3%
6	6.0	33.9	5	6.0	60.8	5	-79.6%
7	6.0	19.9	5	6.0	49.3	5	-148.0%
8	7.6	910.4	5	7.6	1189.9	5	-30.7%
9	6.4	41.7	5	6.4	75.8	5	-81.6%
10	6.0	30.5	5	6.0	66.1	5	-116.9%
11	9.8	1118.7	5	9.8	1362.7	5	-21.8%
12	7.6	234.9	5	7.6	257.1	5	-9.4%
13	6.0	106.9	5	6.0	102.9	5	3.8%
14	9.6	1757.9	5	9.6	1625.4	5	7.5%
15	9.4	426.8	5	9.4	272.8	5	36.1%
16	6.0	220.6	5	6.0	184.7	5	16.3%
17	9.4	2158.5	5	9.4	1991.1	5	7.8%
18	9.9	4155.1	5	9.9	2662.7	5	35.9%
19	9.0	12851.2	5	9.0	8410.4	5	34.6%
20	8.0	28770.0	3	10.6	19143.9	5	-
Average	7.0	2643.9	4.8	7.1	1878.5	5	28.9%

From Table 5.4, it can be observed that IBD successfully obtains all the Pareto points for all problem combinations. On the other hand, SAA finds all the Pareto points except for problem combination 20, where 3 out of 5 instances are found. The computational times of SAA vary from 3.0 to 28770.0 seconds, and the average value is 2643.9 seconds. For IBD, the computational

times vary from 5.5 to 19143.9 seconds, and the average value is 1878.5 seconds. In particular, IBD outperforms SAA in terms of computational time for problem combinations 14 to 20, while for problem combinations 1 to 13, IBD needs more computational time than SAA. This is probably because it is easy to solve the small-scale instances directly, while IBD needs to iterate many times, resulting in more computational times.

The experimental results of 20 different problem combinations under scenario $|S| = 100$ are reported in Table 5.5. It can be found that the computational times of SAA sharply increase with an average value of 11658.5 seconds. In contrast, the average computational time of IBD increases from 1878.5 to 4462.8 seconds. Compared to SAA, IBD can save about 61.7$\%$ computational times on average. For problem combinations 38, 39 and 40, SAA only finds Pareto points 3 out of 5, 1 out of 5 and 0 out of 5 instances, respectively, while IBD can find all Pareto points.

Table 5.5: Experimental results under scenario number $|S| = 100$

Combinations	SAA			IBD			Improvements
	N_P^{SAA}	$T_{SAA}(s)$	N_{opt}^{SAA}	N_P^{IBD}	$T_{IBD}(s)$	N_{opt}^{IBD}	
21	3.2	61.5	5	3.2	34.8	5	43.4%
22	5.2	358.8	5	5.2	77.1	5	78.5%
23	5.0	422.8	5	5.0	80.3	5	81.0%
24	6.0	293.5	5	6.0	115.4	5	60.7%
25	4.6	327.2	5	4.6	145.6	5	55.5%
26	6.6	2178.8	5	6.6	322.6	5	85.2%
27	5.8	838.2	5	5.8	287.7	5	65.7%
28	9.0	10092.1	5	9.0	1910.7	5	81.1%
29	9.0	4550.9	5	9.0	696.7	5	84.7%
30	6.0	1593.9	5	6.0	288.3	5	81.9%
31	7.8	15490.3	5	7.8	2781.1	5	82.0%
32	8.0	13268.5	5	8.0	2264.4	5	82.9%
33	7.4	5951.7	5	7.4	647.9	5	89.1%
34	6.0	13528.6	5	6.0	3634.1	5	73.1%
35	6.8	20407.1	5	6.8	2201.0	5	89.2%
36	8.0	14660.2	5	8.0	2325.3	5	84.1%
37	8.2	26530.6	5	8.2	7325.0	5	72.4%
38	6.6	31572.2	3	8.0	9163.9	5	-
39	3.8	35042.7	1	9.2	22249.3	5	-
40	0.4	36000.0	0	10.0	32704.0	5	-
Average	6.2	11658.5	4.5	7.0	4462.8	5	61.7%

In summary, the proposed IBD approach can effectively solve our problem, find all the optimal Pareto points, and reduce over 55$\%$ computational times compared to SAA.

5.3.3 Sensitivity analysis

In this part, the sensitivity analysis is applied to the studied problem. Since we need to set up the disassembly plants, the effect of the number of candidate locations for disassembly plants is first explored. We test 10 instances by increasing the number of candidate locations for disassembly plants from 2 to 20, and other parameters remain unchanged. The results are shown in Figure 5.9, in which the computational time increases with the number of candidate locations for disassembly plants. The number of Pareto points and objective values have no strict relationship with the number of candidate locations for disassembly plants.

Managerial Insight 2. *The number of candidate locations for disassembly plants directly influences the solution time needed for the decision-maker.*

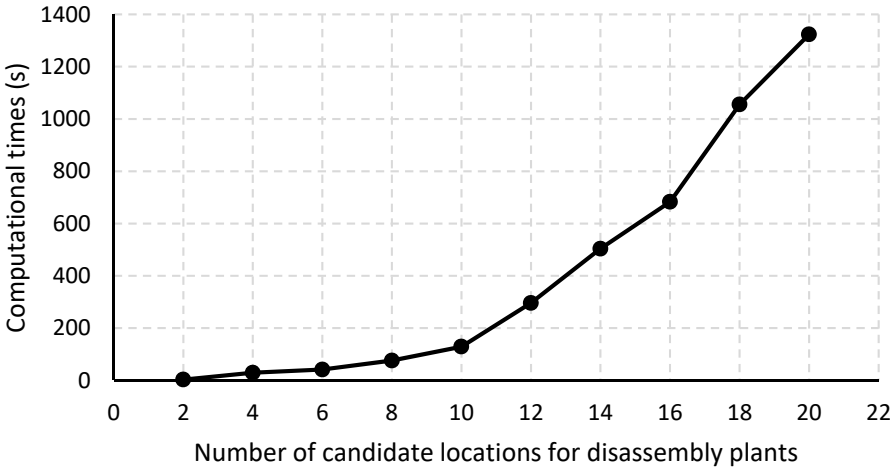


Figure 5.9: Sensitivity analysis of the number of candidate locations for disassembly plants

In addition, the impact of component demand is analyzed. To this end, we maintain the supplies of EOL products unchanged and gradually increase the component demand. Specifically, we use the demand ratio, i.e., the demand divided by the supply, to reflect the relative change of the component demand over the EOL product supply (Ramezanian and Khalesi, 2021). Figure 5.10 illustrates the trend of the expected profit (consider the Pareto point with maximum profit) with different component demand ratios. We observe that the expected profit increases with the ratio when the demand ratio is smaller than 1. In contrast, when the demand ratio is not smaller than 1, the profit no longer increases. This is attributed to the fact that the available amount

of EOL products to be dismantled has reached its limit, and some of the demands are unmet. Therefore, it can be summarized by the following managerial insight.

Managerial Insight 3. *The profit of the RSC increases with the rise of component demand until the component demand is larger than the supply of EOL products.*

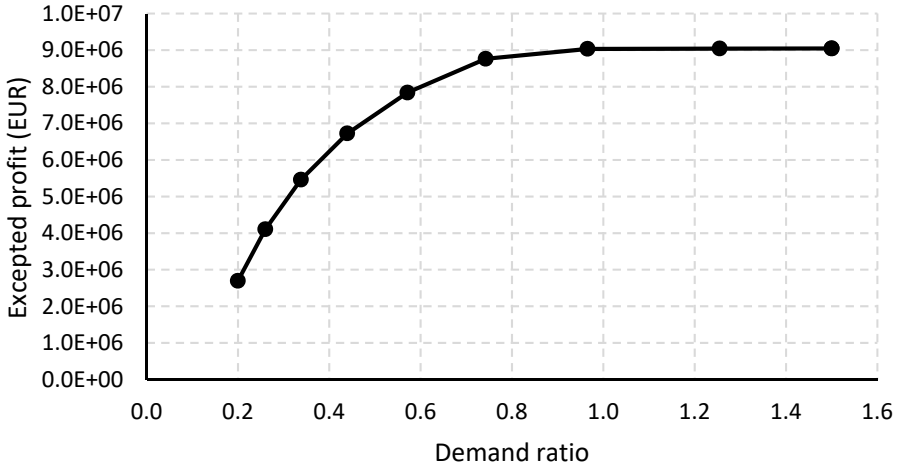


Figure 5.10: Sensitivity analysis of component demands

Furthermore, we investigate the impact of the number of EOL product types because we study the multi-product problem. As the type of EOL product increases, the corresponding numbers of tasks and machines are increased. The type of EOL product $|P|$ increases from 1 to 4, and the obtained Pareto fronts are depicted in Figure 5.11. We can see that there are 3 Pareto points under $|P| = 1$, 7 Pareto points under $|P| = 2$, 9 Pareto points under $|P| = 3$, and 13 Pareto points under $|P| = 4$. It implies that there are more alternative solutions when the types of EOL product increase. On the other hand, the corresponding computational times increase from 1.4 to 893.1 seconds. Hence, we have the following managerial insight.

Managerial Insight 4. *The more types of EOL products are dismantled, the more alternative solutions for the decision-maker, and the longer solution time needed.*

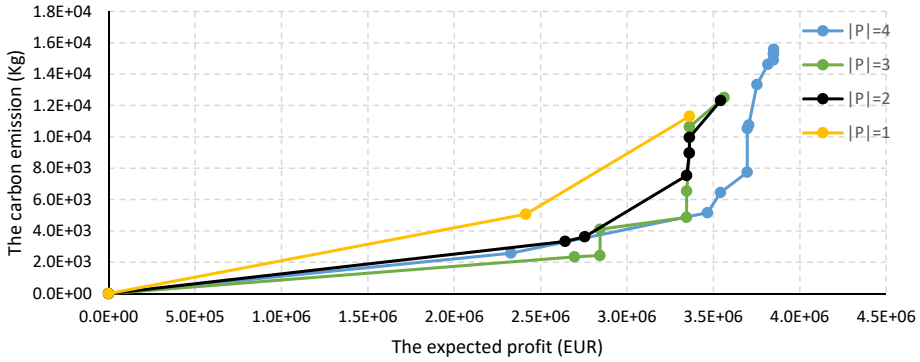


Figure 5.11: Sensitivity analysis of EOL products

5.4 Conclusions

This chapter studies an integrated multi-product RSC design and DLBP with stochastic supply of EOL products, demand for components and task processing times. The problem is formulated by a two-stage stochastic programming model and then transformed into a distribution-free model. ε-constrained method is used to convert the bi-objective problem into a set of single-objective problems, and an IBD approach is devised to solve the single-objective problems efficiently. Computational results of randomly generated instances demonstrate the efficiency of our proposed IBD approach. Moreover, sensitivity analyses are conducted and some managerial insights are discussed.

Within a given period, the production capacity is highly dependent on the cycle time. While, the cycle time of each disassembly plant is predetermined in this chapter, which means that the maximal production capacity of the disassembly plant is fixed. To increase the flexibility of the RSC, we extend the previous study by additionally considering the cycle time as a decision variable.

Chapter 6

Conclusions and Perspectives

Contents

6.1 Thesis Conclusions

This thesis studies three novel stochastic disassembly line balancing related problems. Firstly, we research a new single product DLBP with partial information of task processing time, and construct a new distribution-free formulation and a second-order cone program approximation-based formulation for it. Considering that the traditional single-product disassembly line may be inappropriate and uneconomical to handle these increasing EOL product variants, we then study a novel multi-product DLBP, where identical parts of EOL products and uncertainty task times are simultaneously considered, and propose a lifted cut-and-solve method to solve the problem. Finally, a multi-product disassembly line balancing related RSC design problem is investigated, where strategic and tactical decisions are jointly considered, and an improved Benders decomposition is designed for it.

In Chapter 3, we study a single product DLBP with uncertain task processing time, where only the mean, lower and upper bounds of task processing times are known. The objective is

to minimize the disassembly cost. For the studied problem, a joint chance-constrained model is proposed. Then, a new distribution-free formulation is constructed based on problem property analysis. Successively, a second-order cone program approximation-based formulation is developed. Last, experimental results on benchmark instances and randomly generated instances demonstrate the effectiveness and efficiency of the proposed formulation.

In Chapter 4, we address a stochastic multi-product DLBP, where identical parts of EOL products and uncertainty task times are considered together. The objective is to minimize the disassembly cost. For this problem, a new joint chance-constrained model is formulated. Based on the problem analysis, the joint chance-constrained model is approximately transformed into a distribution-free model. Then, several valid inequalities are provided and an exact lifted cut-and-solve method is designed to solve the problem efficiently. Numerical experiments on an illustrative example, 10 instances based on realistic products and 480 randomly generated instances are conducted to evaluate the performances of the proposed model, valid inequalities and solution method.

In Chapter 5, we investigate a novel multi-product disassembly line balancing related RSC design problem, where EOL products supply, components demand and task processing times are uncertain. The objectives are to maximize the expected net profit and minimize CO_2 emissions, simultaneously. For the problem, a bi-objective nonlinear two-stage stochastic programming model is formulated. In particular, the first-stage problem determines the locations of disassembly plants, machines procurement, assignments of machines and tasks, and open workstations. The second-stage problem decides the transportation of EOL products and components, and inventory levels of EOL products and components. Subsequently, the nonlinear model is approximately transformed into a linear distribution-free model based on problem properties. Then, an exact ε-constrained based method is proposed, and an improved Benders decomposition is developed to solve the transformed single objective problems in the ε-constrained method. Numerical experiments, including one case study and 200 randomly generated instances, are conducted to evaluate the performance of the proposed methods. Furthermore, sensitivity analysis is made to draw managerial insights.

6.2 Future Research Directions

For stochastic disassembly line balancing related problems, many interesting research directions remain to be investigated. They are summarized as follows:

1) Considering the disruptions of the disassembly line and RSC. This thesis studies the stochastic DLBP and disassembly line balancing related RSC design, where the task processing times, products supply and components demand are assumed to be uncertain. However, there may exist some extraordinary events that will disrupt the disassembly line and RSC, which has an unpredictable scaling impact. Such as cutting electricity will interrupt the disassembly line, and natural disasters may cause interruptions in RSC. Therefore, we can study a more resilient disassembly line and RSC, where the stochastic disruption events are considered.

2) Considering developing an integrated disassembly chain for multiple products. Chapters 3 and 4 of this thesis investigate DLBP, which belongs to the tactical level decision, and Chapter 5 studies disassembly line balancing related RSC design, which considers strategic and tactical decisions together. However, operational-level decisions, such as vehicle routing problems (including collecting EOL products and distributing components), also affect the whole performance of the disassembly chain. How to coordinate activities between different partners of the chain remains a challenge. Therefore, in future studies, we can investigate a disassembly chain that integrates strategic, tactical, and operational levels of decisions.

3) Social and ethical considerations need to receive more attention. This thesis focuses on economic and environmental objectives. However, there are many other objectives, for instance, fair labor practices, gender equality, and human rights, that can be coordinated into the joint RSC and disassembly line design problem. Therefore, we can design a disassembly line with economic, environmental, social and ethical objectives.

4) Extend the application of the proposed distribution-free model. This thesis studies stochas-

tic optimization problems, where only partial information of uncertain parameters is known. Based on different information of uncertain parameters, we approximately transform stochastic models into linear distribution-free models for better resolution. Hence, it is encouraging to extend such approximated transformation methods to solve other stochastic problems with known partial information.

5) Developing more efficient solution methods to solve studied problems. In Chapters 4 and 5, we devise two different exact algorithms to solve our studied problems. Due to the NP-hardness nature of the studied problems, it is still time-consuming when the problem scale is large. Therefore, more advanced methods, such as meta-heuristics can be developed. Moreover, design problem-specific heuristics may be another promising direction to reduce computational times.

Publications

Journal papers

[1] **Peng Hu**, Chu Feng, Liu Ming, Wang Shijin, Wu Peng. An integrated approach for a new flexible multi-product disassembly line balancing problem. *Computers & Operations Research*, 2022, 148: 105932.

[2] **Peng Hu**, Chu Feng, Fang Yunfei, Wu Peng. Novel distribution-free model and method for stochastic disassembly line balancing with limited distributional information. *Journal of Combinatorial Optimization*, 2022: 1-24.

[3] **Peng Hu**, Chu Feng, Alexandre Dolgui, Chengbin Chu, Ming Liu. Integrated multi-product reverse supply chain design and disassembly line balancing under uncertainty. *Omega-International Journal of Management Science*. (Under revision)

Conference papers

[4] **Peng Hu**, Chu Feng. Bi-objective optimization for an integrated facility location and disassembly line balancing problem. *In Proceedings of the 10th IFAC Conference on Manufacturing Modelling, Management and Control* (MIM2022). Jun. 22-24th, 2022, Nantes, France.

[5] **Peng Hu**, Chu Feng. Study on a new multi-product stochastic disassembly line balancing problem. *In Proceedings of 23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision* (ROADEF 2022). Feb. 23-25th, 2022, Lyon, France.

[6] **Peng Hu**, Chu Feng. Integrated optimization for flexible disassembly line. *In Proceedings of Rencontres Académie Industrie AFIS 2022*. Dec. 6-8th, 2022, Toulouse, France.

Bibliography

Trends in solid waste management, 2018. URL https://datatopics.worldbank.org/what-a-waste/trends_in_solid_waste_management.html. Accessed on August 25, 2023.

Y. Adulyasak, J.-F. Cordeau, and R. Jans. Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867, 2015.

S. Agrawal et al. A collaborative ant colony algorithm to stochastic mixed-model u-shaped disassembly line balancing and sequencing problem. *International Journal of Production Research*, 46(6):1405–1429, 2008.

F. T. Altekin. A comparison of piecewise linear programming formulations for stochastic disassembly line balancing. *International Journal of Production Research*, 55(24):7412–7434, 2017.

F. T. Altekin, L. Kandiller, and N. E. Ozdemirel. Profit-oriented disassembly-line balancing. *International Journal of Production Research*, 46(10):2675–2693, 2008.

S. A. Alumur, S. Nickel, F. Saldanha-da Gama, and V. Verter. Multi-period reverse logistics network design. *European Journal of Operational Research*, 220(1):67–78, 2012.

S. Avikal, R. Jain, and P. Mishra. A heuristic for u-shaped disassembly line balancing problems. *MIT International Journal of Mechanical Engineering*, 3(1):51–56, 2013.

A. Aydemir-Karadag and O. Turkbey. Multi-objective optimization of stochastic disassembly line balancing with station paralleling. *Computers & Industrial Engineering*, 65(3):413–425, 2013.

C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.

J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

A. Bensmaine, M. Dahane, and L. Benyoucef. A non-dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment. *Computers & Industrial Engineering*, 66(3):519–524, 2013.

M. L. Bentaha, O. Battaïa, and A. Dolgui. A stochastic formulation of the disassembly line balancing problem. In *IFIP International Conference on Advances in Production Management Systems*, pages 397–404. Springer, 2012.

M. L. Bentaha, O. Battaïa, and A. Dolgui. Chance constrained programming model for stochastic profit–oriented disassembly line balancing in the presence of hazardous parts. In *IFIP International Conference on Advances in Production Management Systems*, pages 103–110. Springer, 2013a.

M. L. Bentaha, O. Battaïa, and A. Dolgui. A decomposition method for stochastic partial disassembly line balancing with profit maximization. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 404–409. IEEE, 2013b.

M. L. Bentaha, O. Battaïa, and A. Dolgui. A sample average approximation method for disassembly line balancing problem under uncertainty. *Computers & Operations Research*, 51:111–122, 2014.

M. L. Bentaha, O. Battaïa, and A. Dolgui. An exact solution approach for disassembly line balancing problem under uncertainty of the task processing times. *International Journal of Production Research*, 53(6):1807–1818, 2015a.

M. L. Bentaha, O. Battaïa, A. Dolgui, and S. J. Hu. Second order conic approximation for disassembly line design with joint probabilistic constraints. *European Journal of Operational Research*, 247 (3):957–967, 2015b.

M. L. Bentaha, A. Dolgui, O. Battaïa, R. J. Riggs, and J. Hu. Profit-oriented partial disassembly line design: dealing with hazardous parts and task processing times uncertainty. *International Journal of Production Research*, 56(24):7220–7242, 2018.

J. R. Birge and F. Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

M. A. Boschetti, V. Maniezzo, M. Roffilli, and A. Bolufé Röhler. Matheuristics: Optimization, simulation and control. In *International workshop on hybrid metaheuristics*, pages 171–177. Springer, 2009.

A. Budak. Sustainable reverse logistics optimization with triple bottom line approach: An integration of disassembly line balancing. *Journal of Cleaner Production*, 270:122475, 2020.

A. Charnes and W. W. Cooper. Chance-constrained programming. *Management science*, 6(1): 73–79, 1959.

J. Cheng and A. Lisser. A second-order cone programming approach for linear programs with joint probabilistic constraints. *Operations Research Letters*, 40(5):325–328, 2012.

M. Chica, O. Cordon, S. Damas, and J. Bautista. Multiobjective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. *Information Sciences*, 180(18):3465–3487, 2010.

M. Chitsaz, J.-F. Cordeau, and R. Jans. A unified decomposition matheuristic for assembly, production, and inventory routing. *INFORMS Journal on Computing*, 31(1):134–152, 2019.

Z. A. Çil, S. Mete, and F. Serin. Robotic disassembly line balancing problem: A mathematical model and ant colony optimization approach. *Applied Mathematical Modelling*, 86:335–348, 2020.

S. Climer and W. Zhang. Cut-and-solve: An iterative search strategy for combinatorial optimization problems. *Artificial Intelligence*, 170(8-9):714–738, 2006.

C. A. C. Coello and G. B. Lamont. *Applications of multi-objective evolutionary algorithms*, volume 1. World Scientific, 2004.

Y. Collette and P. Siarry. *Multiobjective optimization: principles and case studies*. Springer Science & Business Media, 2004.

T. G. Crainic, W. Rei, M. Hewitt, and F. Maggioni. *Partial Benders decomposition strategies for two-stage stochastic integer programs*. In Publication CIRRELT-2016-37, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport. Université de Montréal, Montréal, QC, Canada., 2016.

T. G. Crainic, M. Hewitt, F. Maggioni, and W. Rei. Partial Benders decomposition: general methodology and application to stochastic network design. *Transportation Science*, 55(2):414–435, 2021.

F. Cucchiella, I. D'Adamo, P. Rosa, and S. Terzi. Scrap automotive electronics: A mini-review of current management practices. *Waste Management & Research*, 34(1):3–10, 2016.

E. M. de Sá, R. S. de Camargo, and G. de Miranda. An improved Benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research*, 226(2):185–202, 2013.

K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary computation*, 7(3):205–230, 1999.

K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings 6*, pages 849–858. Springer, 2000.

Z. Diri Kenger, Ç. Koç, and E. Özceylan. Integrated disassembly line balancing and routing problem. *International Journal of Production Research*, 58(23):7250–7268, 2020.

T. Doi, T. Nishi, and S. Voß. Two-level decomposition-based matheuristic for airline crew rostering problems with fair working time. *European Journal of Operational Research*, 267(2):428–438, 2018.

E. B. Edis, R. S. Edis, and M. A. Ilgin. Mixed integer programming approaches to partial disassembly line balancing and sequencing problem. *Computers & Operations Research*, 138:105559, 2022.

M. Ehrgott and X. Gandibleux. *Multi-objective Combinatorial Optimization - Theory, Methodology, and Applications*, pages 369–444. Springer US, 2002.

Y. Fang, F. Chu, S. Mammar, and A. Che. An optimal algorithm for automated truck freight transportation via lane reservation strategy. *Transportation Research Part C: Emerging Technologies*, 26:170–183, 2013.

Y. Fang, Q. Liu, M. Li, Y. Laili, and D. T. Pham. Evolutionary many-objective optimization for mixed-model disassembly line balancing with multi-robotic workstations. *European Journal of Operational Research*, 276(1):160–174, 2019.

Y. Fang, H. Ming, M. Li, Q. Liu, and D. T. Pham. Multi-objective evolutionary simulated annealing optimisation for mixed-model multi-robotic disassembly line balancing with interval processing time. *International Journal of Production Research*, 58(3):846–862, 2020a.

Y. Fang, H. Xu, Q. Liu, and D. T. Pham. Evolutionary optimization using epsilon method for resource-constrained multi-robotic disassembly line balancing. *Journal of Manufacturing Systems*, 56:392–413, 2020b.

M. W. Friske, L. S. Buriol, and E. Camponogara. A relax-and-fix and fix-and-optimize algorithm for a maritime inventory routing problem. *Computers & Operations Research*, 137:105520, 2022.

S. L. Gadegaard, A. Klose, and L. R. Nielsen. An improved cut-and-solve algorithm for the single-source capacitated facility location problem. *EURO Journal on Computational Optimization*, 6(1): 1–27, 2018.

S. García, D. Molina, M. Lozano, and F. Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec' 2005 special session on real parameter optimization. *Journal of Heuristics*, 15(6):617–644, 2009.

A. M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10:237–260, 1972.

A. Gharaei, M. Karimi, and S. A. Hoseini Shekarabi. Joint economic lot-sizing in multi-product multi-level integrated supply chains: Generalized benders decomposition. *International Journal of Systems Science: Operations & Logistics*, 7(4):309–325, 2020.

E. Goksoy Kalaycilar, S. Batun, and M. Azizoğlu. A stochastic programming approach for the disassembly line balancing with hazardous task failures. *International Journal of Production Research*, 60(10):3237–3262, 2022.

A. Gungor and S. M. Gupta. Disassembly line balancing. 1999.

A. Gungor and S. M. Gupta. A solution approach to the disassembly line balancing problem in the presence of task failures. *International Journal of Production Research*, 39(7):1427–1467, 2001.

A. Güngör and S. M. Gupta. Disassembly line in product recovery. *International Journal of Production Research*, 40(11):2569–2589, 2002.

H. Guo, L. Zhang, Y. Ren, Y. Li, Z. Zhou, and J. Wu. Optimizing a stochastic disassembly line balancing problem with task failure via a hybrid variable neighborhood descent-artificial bee colony algorithm. *International Journal of Production Research*, 61(7):2307–2321, 2023.

K. Haase and S. Müller. A comparison of linear reformulations for multinomial logit choice probabilities in facility location models. *European Journal of Operational Research*, 232(3):689–691, 2014.

M. K. Habibi, O. Battaïa, V.-D. Cung, and A. Dolgui. Collection-disassembly problem in reverse supply chain. *International Journal of Production Economics*, 183:334–344, 2017.

M. K. K. Habibi, O. Battaïa, V.-D. Cung, A. Dolgui, and M. K. Tiwari. Sample average approximation for multi-vehicle collection–disassembly problem under uncertainty. *International Journal of Production Research*, 57(8):2409–2428, 2019.

N. Hamta, M. Akbarpour Shirazi, S. Fatemi Ghomi, and S. Behdad. Supply chain network optimization considering assembly line balancing and demand uncertainty. *International Journal of Production Research*, 53(10):2970–2994, 2015.

J. He, F. Chu, F. Zheng, M. Liu, and C. Chu. A multi-objective distribution-free model and method for stochastic disassembly line balancing problem. *International Journal of Production Research*, pages 1–17, 2019.

J. He, F. Chu, F. Zheng, and M. Liu. A green-oriented bi-objective disassembly line balancing problem with stochastic task processing times. *Annals of Operations Research*, pages 1–23, 2020a.

J. He, F. Chu, F. Zheng, M. Liu, and C. Chu. A multi-objective distribution-free model and method for stochastic disassembly line balancing problem. *International Journal of Production Research*, 58(18):5721–5737, 2020b.

J. He, F. Chu, F. Zheng, and M. Liu. A green-oriented bi-objective disassembly line balancing problem with stochastic task processing times. *Annals of Operations Research*, 296(1-2):71–93, 2021.

J. He, F. Chu, A. Dolgui, F. Zheng, and M. Liu. Integrated stochastic disassembly line balancing and planning problem with machine specificity. *International Journal of Production Research*, 60(5): 1688–1708, 2022.

A. Heching, J. N. Hooker, and R. Kimura. A logic-based benders approach to home healthcare delivery. *Transportation Science*, 53(2):510–522, 2019.

S. Hezer and Y. Kara. A network-based shortest route model for parallel disassembly line balancing problem. *International Journal of Production Research*, 53(6):1849–1865, 2015.

W. Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

J. N. Hooker and G. Ottosson. Logic-based benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.

K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi. Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review*, 52:2191–2233, 2019.

M. A. Ilgin, H. Akçay, and C. Araz. Disassembly line balancing using linear physical programming. *International Journal of Production Research*, 55(20):6108–6119, 2017.

S. Jiang and S. Yang. A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 21(3):329–346, 2017.

C. B. Kalayci and S. M. Gupta. Ant colony optimization for sequence-dependent disassembly line balancing problem. *Journal of Manufacturing Technology Management*, 24(3):413–427, 2013a.

C. B. Kalayci and S. M. Gupta. Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem. *Expert Systems with Applications*, 40(18):7231–7241, 2013b.

C. B. Kalayci and S. M. Gupta. A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem. *The International Journal of Advanced Manufacturing Technology*, 69:197–209, 2013c.

C. B. Kalayci, A. Hancilar, A. Gungor, and S. M. Gupta. Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm. *Journal of Manufacturing Systems*, 37:672–682, 2015.

C. B. Kalayci, O. Polat, and S. M. Gupta. A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem. *Annals of Operations Research*, 242(2):321–354, 2016.

Z. D. Kenger, Ç. Koç, and E. Özceylan. Integrated disassembly line balancing and routing problem with mobile additive manufacturing. *International Journal of Production Economics*, 235:108088, 2021.

A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

A. Koc, I. Sabuncuoglu, and E. Erel. Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an and/or graph. *IIE Transactions*, 41 (10):866–881, 2009.

I. Kucukkoc. Balancing of two-sided disassembly lines: Problem definition, MILP model and genetic algorithm approach. *Computers & Operations Research*, 124:105064, 2020.

A. Lambert. Linear programming in disassembly/clustering sequence generation. *Computers & Industrial Engineering*, 36(4):723–738, 1999.

M. Laumanns, L. Thiele, and E. Zitzler. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3):932–942, 2006.

E. L. Lawler and D. E. Wood. Branch-and-bound methods: A survey. *Operations Research*, 14(4): 699–719, 1966.

J. Li, X. Chen, Z. Zhu, C. Yang, and C. Chu. A branch, bound, and remember algorithm for the simple disassembly line balancing problem. *Computers & Operations Research*, 105:47–57, 2019a.

Z. Li, I. Kucukkoc, and Z. Zhang. Iterated local search method and mathematical model for sequence-dependent u-shaped disassembly line balancing problem. *Computers & Industrial Engineering*, 137:106056, 2019b.

Z. Li, Z. A. Çil, S. Mete, and I. Kucukkoc. A fast branch, bound and remember algorithm for disassembly line balancing problem. *International Journal of Production Research*, 58(11):3220–3234, 2020.

W. Liang, Z. Zhang, Y. Zhang, P. Xu, and T. Yin. Improved social spider algorithm for partial disassembly line balancing problem considering the energy consumption involved in tool switching. *International Journal of Production Research*, 61(7):2250–2266, 2023.

S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.

M. Lindahl, M. Sørensen, and T. R. Stidsen. A fix-and-optimize matheuristic for university timetabling. *Journal of Heuristics*, 24:645–665, 2018.

J. Liu, Z. Zhou, D. T. Pham, W. Xu, C. Ji, and Q. Liu. Collaborative optimization of robotic disassembly sequence planning and robotic disassembly line balancing problem using improved discrete bees algorithm in remanufacturing. *Robotics and Computer-Integrated Manufacturing*, 61:101829, 2020a.

K. Liu and Z.-H. Zhang. Capacitated disassembly scheduling under stochastic yield and demand. *European Journal of Operational Research*, 269(1):244–257, 2018.

M. Liu, X. Liu, F. Chu, F. Zheng, and C. Chu. Robust disassembly line balancing with ambiguous task processing times. *International Journal of Production Research*, pages 1–30, 2019.

M. Liu, X. Liu, F. Chu, F. Zheng, and C. Chu. An exact method for disassembly line balancing problem with limited distributional information. *International Journal of Production Research*, pages 1–18, 2020b.

X. Liu, F. Chu, A. Dolgui, F. Zheng, and M. Liu. Service-oriented bi-objective robust collection-disassembly problem with equipment selection. *International Journal of Production Research*, 59(6):1676–1690, 2021.

X. Liu, F. Chu, F. Zheng, C. Chu, and M. Liu. Distributionally robust and risk-averse optimisation for the stochastic multi-product disassembly line balancing problem with workforce assignment. *International Journal of Production Research*, 60(6):1973–1991, 2022.

Y. Long, L. H. Lee, and E. P. Chew. The sample average approximation method for empty container repositioning with uncertainties. *European Journal of Operational Research*, 222(1):65–75, 2012.

J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.

Y.-S. Ma, H.-B. Jun, H.-W. Kim, and D.-H. Lee. Disassembly process planning algorithms for end-of-life product recovery and environmentally conscious disposal. *International Journal of Production Research*, 49(23):7007–7027, 2011.

T. L. Magnanti and R. T. Wong. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981.

R. T. Marler and J. S. Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 41:853–862, 2010.

S. McGovern and S. Gupta. Combinatorial optimization analysis of the unary np-complete disassembly line balancing problem. *International Journal of Production Research*, 45(18-19):4485–4511, 2007a.

S. M. McGovern and S. M. Gupta. 2-opt heuristic for the disassembly line balancing problem. In *Environmentally conscious manufacturing iii*, volume 5262, pages 71–84. International Society for Optics and Photonics, 2004.

S. M. McGovern and S. M. Gupta. Local search heuristics and greedy algorithm for balancing a disassembly line. *International Journal of Operations and Quantitative Management*, 11(2):91, 2005.

S. M. McGovern and S. M. Gupta. A balancing method and genetic algorithm for disassembly line balancing. *European Journal of Operational Research*, 179(3):692–708, 2007b.

S. Mutlu and B. Güner. A memetic algorithm for mixed-model two-sided disassembly line balancing problem. *Procedia CIRP*, 98:67–72, 2021.

A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2007.

M. Ng. Distribution-free vessel deployment for liner shipping. *European Journal of Operational Research*, 238(3):858–862, 2014.

I. H. Osman and J. P. Kelly. Meta-heuristics theory and applications. *Journal of the Operational Research Society*, 48(6):657–657, 1997.

E. Özceylan and T. Paksoy. Reverse supply chain optimisation with disassembly line balancing. *International Journal of Production Research*, 51(20):5985–6001, 2013.

E. Özceylan and T. Paksoy. Fuzzy mathematical programming approaches for reverse supply chain optimization with disassembly line balancing problem. *Journal of Intelligent & Fuzzy Systems*, 26(4):1969–1985, 2014.

E. Özceylan, T. Paksoy, and T. Bektaş. Modeling and optimizing the integrated problem of closed-loop supply chain network design and disassembly line balancing. *Transportation Research Part E: Logistics and Transportation Review*, 61:142–164, 2014.

E. Özceylan, C. B. Kalayci, A. Güngör, and S. M. Gupta. Disassembly line balancing problem: a review of the state of the art and future directions. *International Journal of Production Research*, 57(15-16):4805–4827, 2019.

T. Paksoy, A. Güngör, E. Özceylan, and A. Hancilar. Mixed model disassembly line balancing problem with fuzzy goals. *International Journal of Production Research*, 51(20):6082–6096, 2013.

D. A. Paterson, W. L. Ijomah, and J. F. Windmill. End-of-life decision tool with emphasis on remanufacturing. *Journal of Cleaner production*, 148:653–664, 2017.

G. Perakis and G. Roels. Regret in the newsvendor model with partial information. *Operations Research*, 56(1):188–203, 2008.

F. Qiu and J. Wang. Chance-constrained transmission switching with guaranteed wind power utilization. *IEEE Transactions on Power Systems*, 30(3):1270–1278, 2014.

R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

R. Ramezanian and S. Khalesi. Integration of multi-product supply chain network design and assembly line balancing. *Operational Research*, 21(1):453–483, 2021.

Y. Ren, D. Yu, C. Zhang, G. Tian, L. Meng, and X. Zhou. An improved gravitational search algorithm for profit-oriented partial disassembly line balancing problem. *International Journal of Production Research*, 55(24):7302–7316, 2017.

R. J. Riggs, O. Battaïa, and S. J. Hu. Disassembly line balancing under high variety of end of life states using a joint precedence graph approach. *Journal of Manufacturing Systems*, 37:638–648, 2015.

A. Ruszczyński and A. Shapiro. Stochastic programming models. *Handbooks in Operations Research and Management Science*, 10:1–64, 2003.

Q. Sun, H. Chen, R. Long, and J. Yang. Who will pay for the "bicycle cemetery"? evolutionary game analysis of recycling abandoned shared bicycles under dynamic reward and punishment. *European Journal of Operational Research*, 305(2):917–929, 2023.

Y. Tang, M. Zhou, E. Zussman, and R. Caudill. Disassembly modeling, planning, and application. *Journal of Manufacturing Systems*, 21(3):200–217, 2002.

M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005.

G. Tian, C. Zhang, X. Zhang, Y. Feng, G. Yuan, T. Peng, and D. T. Pham. Multi-objective evolutionary algorithm with machine learning and local search for an energy-efficient disassembly line balancing problem in remanufacturing. *Journal of Manufacturing Science and Engineering*, 145 (5):051002, 2023.

W. F. Tyndall. A duality theorem for a class of continuous linear programming problems. *Journal of the Society for Industrial and Applied Mathematics*, 13(3):644–666, 1965.

C. Von Lücken, B. Barán, and C. Brizuela. A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, 58:707–756, 2014.

K. Wang, X. Li, and L. Gao. A multi-objective discrete flower pollination algorithm for stochastic

two-sided partial disassembly line balancing problem. *Computers & Industrial Engineering*, 130: 634–649, 2019.

K. Wang, X. Li, L. Gao, and P. Li. Modeling and balancing for green disassembly line using associated parts precedence graph and multi-objective genetic simulated annealing. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 8:1597–1613, 2021a.

K. Wang, X. Li, L. Gao, P. Li, and S. M. Gupta. A genetic simulated annealing algorithm for parallel partial disassembly line balancing problem. *Applied Soft Computing*, 107:107404, 2021b.

Y. Wang, N. Wang, S. Cheng, X. Zhang, H. Liu, J. Shi, Q. Ma, and M. Zhou. Optimization of disassembly line balancing using an improved multi-objective genetic algorithm. *Advances in Production Engineering & Management*, 16(2):240–252, 2021c.

P. Wu, A. Che, F. Chu, and Y. Fang. Exact and heuristic algorithms for rapid and station arrival-time guaranteed bus transportation via lane reservation. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):2028–2043, 2017.

Q. Xiao, X. Guo, and D. Li. Partial disassembly line balancing under uncertainty: robust optimisation models and an improved migrating birds optimisation algorithm. *International Journal of Production Research*, 59(10):2977–2995, 2021.

Z. Yang, F. Chu, and H. Chen. A cut-and-solve based algorithm for the single-source capacitated facility location problem. *European Journal of Operational Research*, 221(3):521–532, 2012.

Ö. F. Yılmaz et al. Tactical level strategies for multi-objective disassembly line balancing problem with multi-manned stations: an optimization model and solution approaches. *Annals of Operations Research*, 319(2):1793–1843, 2022.

T. Yin, Z. Zhang, Y. Zhang, T. Wu, and W. Liang. Mixed-integer programming model and hybrid driving algorithm for multi-product partial disassembly line balancing problem with multi-robot workstations. *Robotics and Computer-Integrated Manufacturing*, 73:102251, 2022.

A. Yolmeh and U. Saif. Closed-loop supply chain network design integrated with assembly and disassembly line balancing under uncertainty: an enhanced decomposition approach. *International Journal of Production Research*, 59(9):2690–2707, 2021.

L. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE transactions on Automatic Control*, 8(1):59–60, 1963.

Y. Zeng, Z. Zhang, T. Yin, and H. Zheng. Robotic disassembly line balancing and sequencing problem considering energy-saving and high-profit for waste household appliances. *Journal of Cleaner Production*, 381:135209, 2022.

Y. Zhang, Z. Zhang, C. Guan, and P. Xu. Improved whale optimisation algorithm for two-sided disassembly line balancing problems considering part characteristic indexes. *International Journal of Production Research*, 60(8):2553–2571, 2022.

Z. Zhang, K. Wang, L. Zhu, and Y. Wang. A pareto improved artificial fish swarm algorithm for solving a multi-objective fuzzy disassembly line balancing problem. *Expert Systems with Applications*, 86:165–176, 2017.

F. Zheng, J. He, F. Chu, and M. Liu. A new distribution-free model for disassembly line balancing problem with stochastic task processing times. *International Journal of Production Research*, 56 (24):7341–7353, 2018.

L. Zhu, Z. Zhang, and C. Guan. Multi-objective partial parallel disassembly line balancing problem using hybrid group neighbourhood search algorithm. *Journal of Manufacturing Systems*, 56: 252–269, 2020.

E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK report*, 103, 2001.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Optimisation stochastique et multicritère pour l'industrie remanufacturière
Mots clés: équilibrage de ligne de désassemblage, logistique inverse, optimisation stochastique, optimisation multicritère, couper et résoudre, décomposition de Benders

Résumé: Le désassemblage des produits en fin de vie (EOL) dans la remanufacturation a attiré une attention considérable ces dernières années en raison de ses avantages en matière d'économie de ressources non renouvelables, de protection de l'environnement et de promotion de la croissance économique. Dans la littérature existante, 1) la plupart des problèmes stochastiques d'équilibrage de la ligne de désassemblage supposent que les distributions de probabilité des paramètres incertains soient connues ; 2) la majorité des problèmes d'équilibrage de la ligne de désassemblage se concentrent sur un seul produit ; 3) peu de travaux portent sur les problèmes de conception de la chaîne logistique inversée (RSC) liés à l'équilibrage de la ligne de désassemblage. En réalité, plusieurs RSC liées au désassemblage de produits EOL existent dans les industries de la remanufacturation, telles que l'automobile, les téléphones mobiles, etc. Pour combler ces lacunes dans la littérature, trois nouveaux problèmes liés à l'équilibrage de la ligne de désassemblage sont étudiés dans cette thèse.

Tout d'abord, une DLBP à produit unique avec des informations partielles sur les temps de traitement des tâches est étudiée, où seules la moyenne, la borne inférieure et la borne supérieure des temps de traitement des tâches sont connues. L'objectif est de minimiser le coût de désassemblage. Pour le problème étudié, un modèle conjoint à contraintes de probabilités est proposé. Ensuite, une nouvelle formulation sans distribution et une formulation basée sur une approximation de programme de cônes de second ordre sont construites en fonction des propriétés du problème. Les résultats expérimentaux sur 7 instances de référence et sur 81 instances générées aléatoirement montrent l'efficacité de l'approche proposée.

Deuxièmement, une nouve DLBP stochastique multi-produits avec un temps de traitement de tâche incertain est abordée, où seules la moyenne, l'écart type et la limite supérieure des temps de tâche sont disponibles. L'objectif est de minimiser le coût de désassemblage. Pour le problème, un modèle conjoint à contraintes de probabilités est formulé. Ensuite, sur la base de l'analyse du problème, le modèle conjoint à contraintes de probabilités est approximativement transformé en un modèle sans distribution. Ensuite, plusieurs inégalités valides et une méthode exacte de coupe et de résolution sont conçues pour résoudre efficacement le problème. Les résultats des expériences sur un exemple illustratif et sur 490 instances générées aléatoirement démontrent les bonnes performances du modèle proposé, des inégalités valides et de la méthode de résolution.

Enfin, un nouveau problème de conception de la RSC lié à l'équilibrage de la ligne de désassemblage multi-produits est étudié, où l'approvisionnement en produits EOL, la demande en composants et les temps de traitement des tâches sont supposés incertains. Les objectifs sont de maximiser le profit attendu et de minimiser simultanément les émissions de CO_2. Pour le problème, un modèle bi-objectif de programmation stochastique à deux étapes et non linéaire est formulé, et approximativement transformé en un modèle sans distribution linéaire en fonction des propriétés du problème. Ensuite, une méthode basée sur des contraintes ε-construites est proposée, dans laquelle une décomposition de Benders améliorée est conçue pour résoudre les problèmes transformés à objectif unique. Des expériences numériques comprenant une étude de cas et sur 200 instances générées aléatoirement sont menées pour évaluer les performances des méthodes proposées. De plus, une analyse de sensibilité est réalisée pour tirer des enseignements en matière de gestion.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Stochastic and multi-criteria optimization for remanufacturing industry

Keywords: disassembly line balancing, reverse supply chain, stochastic optimization, multi-objective optimization, cut-and-solve, Benders decomposition

Abstract: End-of-Life (EOL) products disassembly in remanufacturing has received extensive attention in recent years owing to their advantages in saving non-renewable resources, protecting the environment and promoting economic growth. In the existing literature, 1) most of stochastic disassembly line balancing problems assume that the probability distributions of uncertain parameters are known; 2) majority of disassembly line balancing problems focus on single product; 3) few works study the disassembly line balancing related reverse supply chain (RSC) design problems. In reality, multiple EOL products disassembly related RSC exist in remanufacturing industries, such as automobile, mobile phone, etc. To bring these research gaps, three new disassembly line balancing related problems are investigated in this thesis.

Firstly, a single product disassembly line balancing problem (DLBP) with partial information of task processing times is studied, where only the mean, lower and upper bounds of task processing times are known. The objective is to minimize the disassembly cost. For the studied problem, a joint chance-constrained model is proposed. Then, a new distribution-free formulation and a second-order cone program approximation-based formulation are constructed based on problem properties. Experimental results on 7 benchmark instances and 81 randomly generated instances show the effectiveness and efficiency of the proposed approach.

Secondly, a new stochastic multi-product DLBP with uncertain task processing time is addressed, where only the mean, standard deviation and upper bound of task times are available. The objective is to minimize the disassembly cost. For the problem, a joint chance-constrained model is formulated. Then, based on problem analysis, the joint chance-constrained model is approximately transformed into a distribution-free model. Subsequently, several valid inequalities and an exact lifted cut-and-solve method are designed to efficiently solve the problem. Experiments results on an illustrative example and 490 randomly generated instances demonstrate the good performances of the proposed model, valid inequalities and solution method.

Finally, a novel multi-product disassembly line balancing related RSC design problem is investigated, where EOL products supply, components demand and task processing times are assumed to be uncertain. The objectives are to maximize the expected profit and minimize CO_2 emissions, simultaneously. For the problem, a bi-objective nonlinear two-stage stochastic programming model is formulated and approximately transformed to a linear distribution-free model based on problem properties. Then, an exact ε-constrained based method is proposed, in which an improved Benders decomposition is designed to solve the transformed single objective problems. Numerical experiments including one case study and 200 randomly generated instances are conducted to evaluate the performance of proposed methods. Moreover, sensitivity analysis is made to draw managerial insights.