Integrated stochastic disassembly line balancing and planning problem with machine specificity

Junkai He, Feng Chu, Alexandre Dolgui, Feifeng Zheng, Ming Liu

Integrated stochastic disassembly line balancing and planning problem with machine specifcity

Junkai He[a,b], Feng Chu[a,c], Alexandre Dolgui[d], Feifeng Zheng[b], Ming Liu[e]

[a] *Laboratoire IBISC, Univ-Évry, Université Paris-Saclay, 91025 Évry, France*

[b] *Glorious Sun School of Business & Management, Donghua University, Shanghai 200051, China*

[c] *School of Economics & Management, Fuzhou University, Fuzhou 350116, China*

[d] *Department of Automation, Production and Computer Science, IMT Atlantique, LS2N-CNRS, Nantes 44300, France*

[e] *School of Economics & Management, Tongji University, Shanghai 200092, China*

Abstract

The disassembly is a fundamental basis in converting End-of-Life (EOL) products into useful components. Related research becomes a hot topic in the last decades due to the increasing awareness of environmental protection and energy conservation. However, there are many opening questions needed to be investigated, especially the efficient coordination of different-level decisions is a big challenge under uncertainty. In this paper, a novel integrated stochastic disassembly line balancing and planning problem is studied to minimize the system cost, where component yield ratios and component demands are assumed to be uncertain. In this work, available machines are assumed to have different prices, abilities, and capacities for task processing. For the problem, a two-stage non-linear stochastic programming model is first constructed. Then, it is further transformed into a linear formulation. Based on problem property analysis, a valid inequality is proposed to reduce the search space of optimal solutions. Finally, a sample average approximation (SAA) and a L-shaped algorithm are adopted to solve the problem. Numerical experiments on randomly generated instances demonstrate that the valid inequality can save around 11% of average computation time, and the L-shaped algorithm can save around 64% of average computation time compared with the SAA algorithm, without a big sacrifice of the solution quality.

Keywords: Integrated disassembly line balancing and planning; uncertainty, machine specificity, two-stage stochastic programming; valid inequality; SAA and L-shaped.

Email address: feng.chu@univ-evry.fr (Feng Chu[a,c])

1. Introduction

Nowadays, the development of technology and the ever-changing demand of customers sharply increase the creation and consumption ratio of products (Kazancoglu and Ozturkoglu, 2018). Although this brings a lot of economic benefits, plenty of End-of-Life (EOL) products are generated as well. As reported by the World Bank, global annual waste generation is expected to jump to 3.4 billion tonnes over the next 30 years (World bank report, 2018). Inappropriate management on EOL products may further aggravate environment pollution and irreversible resource waste (Tian and Zhang, 2019). Therefore, EOL products should be properly managed by remanufacturing industries to save non-renewable resources, decrease waste, and reduce pollution (Polotski et al., 2017). As shown in Figure 1, remanufacturing plays an essential role to recycle EOL products and it consists of a series of operations. Among them, disassembly is the key process to decompose EOL products into reusable components for downstream manufacturers (Wu et al., 2016). However, Zikopoulos (2017) indicates that there is a lack of mature disassembly systems and a lot of potential values of EOL products have not been discovered. Consequently, how to appropriately design a disassembly system and to efficiently dispose EOL products are important research topics in remanufacturing.

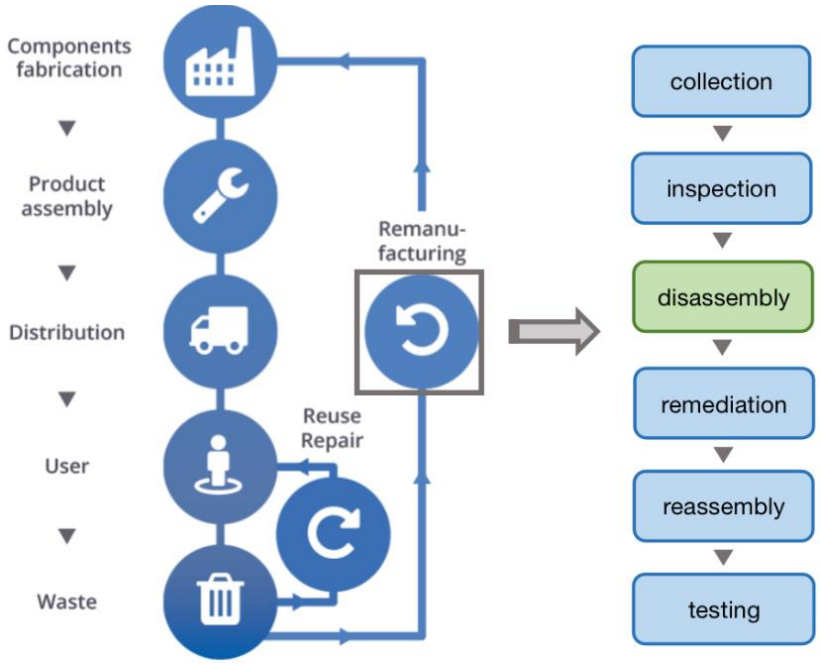

Figure 1: The roles of remanufacturing and disassembly.

Different kinds of disassembly line problems have been studied in the literature, including disassembly line balancing problems (DLBP), disassembly line planning problems (DLPP), disassembly line sequencing problems, collection-disassembly line problems, disassembly-supply chain problems (Lee et al., 2001; Kannan et al, 2017; Liu et al., 2020b; Özceylan et al., 2014). Among them, the DLBP and DLPP are two widely studied disassembly line problems. To the best of our knowledge, these two problems are usually studied separately because they belong to different decision levels. Yet, considering disassembly line balancing and planning in an integrated manner may save the overall cost of the disassembly line system. In this paper, we take the first attempt to research an integrated disassembly line balancing and planning problem. The characteristics of the considered problem are as follows.

The DLBP, belonging to a tactical level, is to select an optimal disassembly process and assign tasks to a set of workstations to form a disassembly line (Gügör and Gupta, 1999). In the most previous DLBP works, the machine specificity that is an important part of line balancing has seldom been taken. In this paper, machine specificities such as price, capacity, and ability are considered. On the other hand, the DLPP, corresponding to an operational level, appropriately determines the procurement amount of EOL products to be separated to satisfy uncertain component demand caused by diverse factors, such as market fluctuation (Kim et al., 2007). Besides, the unpredictable damage of EOL products leading to uncertain component yield ratio that impacts line efficiency merits to be investigated (Kim et al., 2007; Tian and Zhang, 2019).

In summary, we study an *integrated stochastic disassembly line balancing and planning problem* (ISDLBPP) in this work. The objective is to minimize the system cost, including the acquisition cost of machines, the configuration cost of workstations, the procurement cost of EOL products, and the inventory cost of components. The problem is firstly formulated as a non-linear *two-stage stochastic mixed-integer programming* (TSMIP). Based on problem property analysis, the nonlinear model is transformed into a linear one and a valid inequality is proposed to reduce the search space of optimal solutions. Then, a *sample average approximation* (SAA) and a L-shaped algorithm are proposed to efficiently solve the problem. The contributions of this study are threefold:

i. An integrated disassembly line balancing and line planning problem is first addressed, in which component demands and component yield ratios are assumed to be uncertain.

ii. Machine specificities, such as price, ability, capacity, are considered.

iii. The studied problem is formulated by a TSMIP model.

iv. Efficient valid inequality and solution methods are proposed based on problem property analysis.

The rest of this paper is organized as follows. A literature review is presented in Section 2. In Section 3, the studied ISDLBPP is described in detail and is formulated by a TSMIP model. The valid inequality and solution methods are proposed in Section 4. Numerical experiments are conducted and reported in Section 5. Finally, the management insights from this research, the conclusions to this work, and the suggestions on future research directions are established.

2. Literature review

In this section, we firstly review the existing literature for the DLBP and DLPP. Then, we review the previous works on integrated disassembly line problems to position our research.

2.1. The DLBP and DLPP

Özceylan et al. (2018) indicate in their review article that more than 120 papers about DLBP have been published before. Although different characteristics of disassembly lines have been considered, the machine specificity is seldom taken, at least not be well defined. To be specific, the previous DLBP works usually focus on the assignment of disassembly tasks to workstations from a macro perspective. However, in reality, disassembly tasks are processed by machines executed in workstations. Therefore, it is important to consider the machine specificity and assignment that may greatly impact disassembly performance. Recently, He et al. (2020) first consider different-technology workstations and their impacts on the system cost and CO_2 emission in a bi-objective stochastic DLBP. The authors assume that a high-technology workstation with a higher price may generate less contaminant emission during task processing, compared with the cheaper ones. However, a limitation of He et al. (2020) is that each machine is supposed to process only one dedicated task, regardless of a set of tasks. In this paper, we assume that specific machines can have different prices, abilities, and capacities for task processing.

Compared to the widely studied DLBP, only a small number of articles have considered DLPP, most of which focus on deterministic cases (Hojato, 2016). To be specific, Kim et al. (2005; 2006) and Prakash et al. (2012) investigate deterministic DLPP with machine capacity. Integer programming models are formulated to minimize the total cost, including the disassembly line setup, product procurement, and

inventory costs. For solving the problems, a series of solution methods including a Lagrangian relaxation-based heuristic, a two-stage heuristic, and a constraint-based simulated annealing algorithm are designed. Ji et al. (2016) introduce the yield ratios of components into a deterministic DLPP. A *mixed-integer programming* (MIP) model is constructed and a Lagrangian relaxation-based two-stage heuristic is devised for the problem. Tian and Zhang (2019) further consider that the yield ratios of components depend on the procurement price of returned products. A MIP model and a particle swarm-based dynamic programming algorithm are proposed to solve the problem. For non-deterministic settings, Kim and Xirouchakis (2010) suppose that component demands are uncertain. A stochastic programming model is formulated to minimize the sum of the expected value of setup, inventory, and back-order penalty costs. A Lagrangian relaxation-based heuristic and a fast heuristic are proposed to solve the problem. Liu and Zhang (2018) consider simultaneously uncertain component demands and yield ratios. For the problem, a chance-constrained stochastic programming model is proposed to minimize the setup, procurement, and inventory costs. A problem-based approximation heuristic is designed to solve the problem.

For easier comparison, the main differences between the literature and this work are listed in Table 1.

Table 1: Differences between the literature and this work.

Research	Machine specificity	Multi-uncertainty	Integrated DLBP & DLPP
DLBP	×	×	×
DLPP	×	only one	×
This work	✓	✓	✓

Based on the review of DLBP and DLPP literature, we conclude in Table 1 that (i) machine specificity is seldom considered, and an exception is the workstation diversity in He et al. (2020). However, there may exist different types of machines with different prices, abilities, and capacities for processing a set of tasks. (ii) Uncertain component demand and component yield ratio are simultaneously considered only in Liu and Zhang (2018), but appropriately estimating these uncertainties together may ensure systematic stability. (iii) The two related disassembly line problems (DLBP and DLPP) have not been optimized in an integrated manner. However, the efficient coordination of disassembly line balancing and planning decisions can without a doubt improve system performance.

2.2. Integrated disassembly line problems

Although some existing works study integrated assembly line balancing and planning problems (Esmaeilbeigi et al., 2016; Naderi et al., 2019; Özcan, 2019, etc.), the characteristics of assembly problems cannot be directly used in disassembly ones (Özceylan et al., 2019). To the best of our knowledge, only the following works consider integrated disassembly line-related problems.

Özceylan et al. (2014) jointly optimize the strategic and tactical decisions of a closed-loop supply chain, in which the latter concerns disassembly line balancing in the reverse chain. The objective is to minimize the overall cost. A nonlinear MIP formulation is constructed for the problem. Numerical examples are presented and tested for the proposed model. Liu et al. (2020b) study a collection-disassembly problem to minimize the system cost and to maximize the service level, with partial distribution information of component demands. For the problem, a distributionally robust bi-objective formulation is proposed. A SAA-based model and an approximated MIP model are constructed and solved via the ε-constraint method. Especially, Ehm (2019) considers a typical disassembly line balancing and a single-period disassembly line scheduling to minimize the makespan. For the problem, a MIP model is developed and evaluated by randomly generated instances.

It can be seen from the above literature that although some papers investigate integrated disassembly line problems, the DLBP and DLPP have not been studied before. Besides, some specific characteristics mentioned in Section 2.1 have not been taken into account. Compared with the previous literature, we novelly address an ISDLBPP integrating a machine-specific DLBP and a multi-cycle stochastic DLPP.

3. Problem statement and formulation

In this section, the considered ISDLBPP is first described in detail, then a two-stage stochastic programming model is formulated for it.

3.1. Description of the considered ISDLBPP

Given a type of EOL product, the corresponding disassembly scheme set $\mathcal{L}$ and disassembly task set $\mathcal{J}$ are revealed. The potential disassembly schemes have the same function for separating EOL products into components but may contain different tasks. The following Figure 2 shows three disassembly schemes for separating an EOL hand light (Tang et al., 2002), denoted by L1, L2, and L3, respectively. In this figure, each rectangle indicates a product status and each circle represents a disassembly task. For example, disassembly scheme L1 is formed by tasks 1, 3, 6, 7, 9, and 10. Specifically, task 1 can decompose the product (with 7 components) into

two parts: one consists of components 2, 3, 4, and 5; the other one is of components 1, 6, and 7. Then, task 9 can segregate component 1 from part $\{1, 6, 7\}$. Respecting the *task precedence constraint*, tasks 3 and 9 cannot be processed before task 1, tasks 6 and 7 should be started after tasks 3, and task 10 follows after task 9.

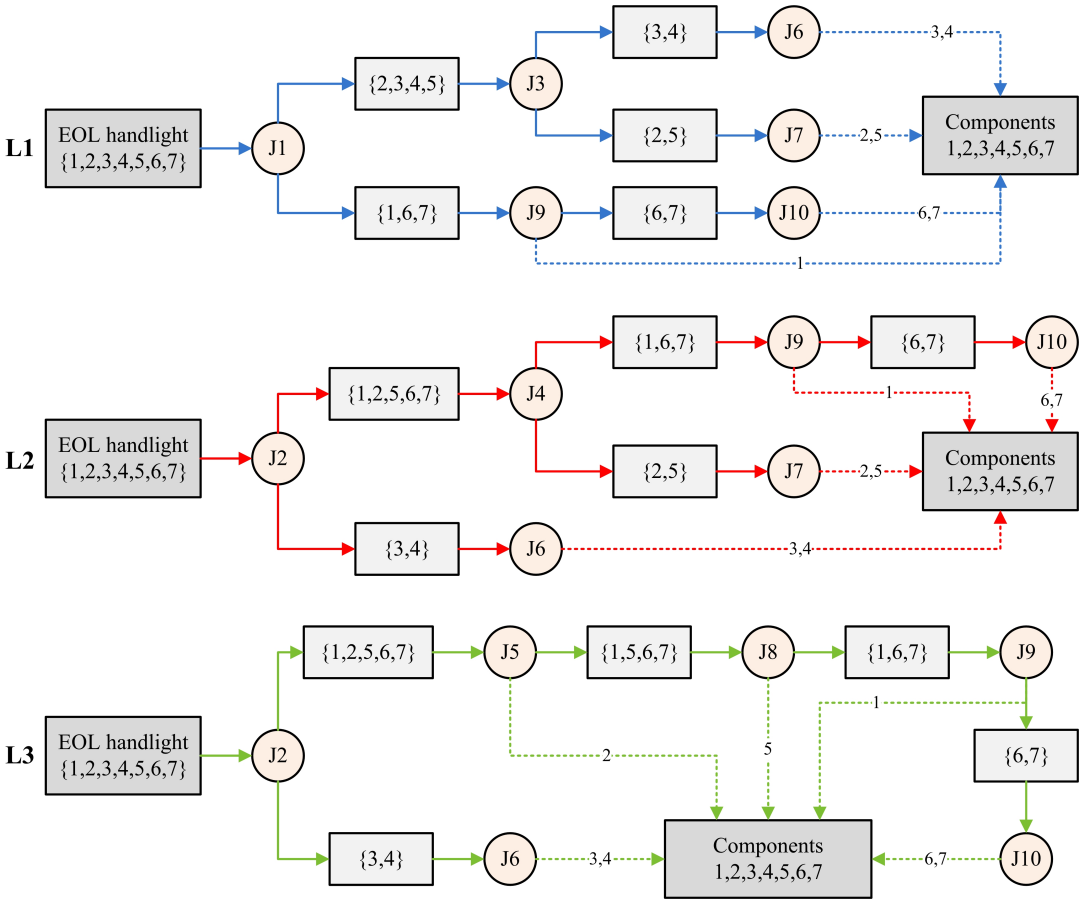

Figure 2: Three different disassembly schemes with disassembly tasks.

The disassembly tasks can be accomplished by specific machines from a machine set $\mathcal{M}$, and each machine is able to process a set of tasks. In Figure 3, there are five available machines for the previously mentioned hand-light instance. For example, machine 1 can process tasks 1, 3, and 9; machine 2 can only process task 1; machine 3 is available for tasks 3 and 9; machine 4 can accomplish tasks 6 and 7; machine 5 can serve task 10. To formulate a disassembly line based on scheme L1, one option is to buy machines 1, 4, and 5. While the other option is to acquire machines 2, 3, 4, and 5. Note that machines 1 and 2 (or machines 1 and 3) cannot be selected simultaneously for a disassembly scheme since they have an overlapping task 1.

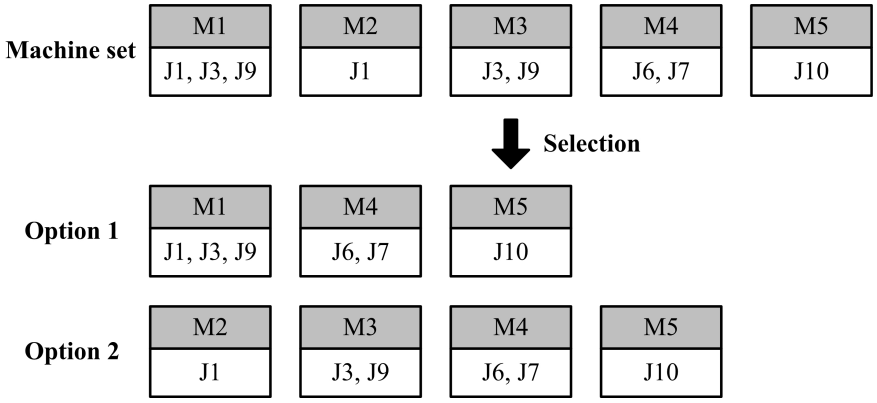

Figure 3: Two machine acquisition options for disassembly scheme L1.

The purchased machines are assigned into workstations, and configuring a workstation for allocating these machines require a workstation configuration cost. Note that machine assignment to workstations must respect the *cycle time constraint* and *task precedence relation*. Specifically, the cycle time denotes the maximal total task processing times among the configured workstations. On the other hand, the following Figure 4 illustrates a disassembly line that violates the precedence relation in scheme L1, i.e., tasks 3 and 9 processed by machine 3 cannot be served after tasks 6 and 7 processed by machine 4. A potential adjustment is to change the positions of machines 3 and 4. Then, the disassembly line can be feasible for separating EOL products if the cycle time is respected.

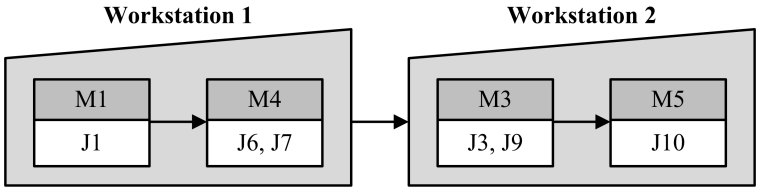

Figure 4: A disassembly line violating the precedence constraint.

Consider a time horizon $\mathcal{T}$, a decomposition planning determines the number of EOL products to be purchased and decomposed in each period. Suppose that a physical-complete EOL product contains a set of components, i.e., $\mathcal{I}$. The demand of each component is uncertain over the horizon and the number of component i obtained from an EOL product depends on a stochastic yield ratio $r_i \in [0, 1]$. For example, if the EOL product contains 4 units of component 1 whose yield ratio r_i is 0.8, then $4 \times 0.8 = 3.2$ units of component 1 can be obtained from a product.

In summary, the following assumptions should be respected in this work:

i. Potential disassembly schemes are known for separating a kind of EOL product into components.

ii. Each disassembly scheme (line) has a dedicated task sequence that has to be respected, and different schemes may have task overlaps.

iii. The total task processing time in each workstation should be less than the given cycle time.

iv. The candidate machines can process a set of tasks, and they have different acquisition prices, abilities, and capacities.

v. A task can only be accomplished by one machine.

vi. The component demand and yield ratio are stochastic and scenario-based.

The considered ISDLBPP consists of (i) selecting a disassembly line for a kind of EOL product, purchasing appropriate specific machines, and depositing them into configured workstations; (ii) determining EOL-product decomposition planning to satisfy component demands over the planning horizon. The objective is to minimize the overall system cost, including the machine acquisition cost, workstation configuration cost, EOL-product procurement cost, and component inventory cost. In the following, a non-linear TSMIP model is formulated for the studied problem.

3.2. Two-stage stochastic mixed-integer programming formulation

Before presenting the TSMIP model, the related parameters and decision variables are first introduced as follows.

Parameters:

- l : Index of disassembly schemes, and $l \in \mathcal{L}$.

- j : Index of disassembly tasks, and $j \in \mathcal{J}$. Note that $\mathcal{J}_l \in \mathcal{J}$ denotes the task set in the l-th disassembly scheme.

- m, m' : Indices of machines, and $m, m' \in \mathcal{M}, m \neq m'$.

- w : Index of workstations, and $w \in \mathcal{W}$.

- i : Index of components, and $i \in \mathcal{I}$.

- t : Index of time periods, and $t \in \mathcal{T}$.

- $\boldsymbol{\omega}$: Index of scenarios, and $\boldsymbol{\omega} \in \boldsymbol{\Omega}$.

- $a_{j,m}$: Binary parameter, equaling to 1 if task j can be accomplished by machine m, 0 otherwise, where $j \in \mathcal{J}$ and $m \in \mathcal{M}$.

- $b_{m,m'}$: Binary parameter, equaling to 1 if machines m and m' can process the same task(s), 0 otherwise, where $m, m' \in \mathcal{M}$ and $m \neq m'$.

- $c_{m,m'}$: Binary parameter, equaling to 1 if the task on machines m and m' respect the precedence relation, 0 otherwise, where $m, m' \in \mathcal{M}$ and $m \neq m'$.

- p_m : Total task processing time in machine m, where $m \in \mathcal{M}$.

- q_i : Number of component i obtained from one unit of product, where $i \in \mathcal{I}$.

- $r_i(\boldsymbol{\omega})$: yield ratio of component i under scenario $\boldsymbol{\omega}$, where $i \in \mathcal{I}$ and $\boldsymbol{\omega} \in \boldsymbol{\Omega}$.

- $d_{i,t}(\boldsymbol{\omega})$: demand of component i in period t under scenario $\boldsymbol{\omega}$, where $i \in \mathcal{I}$, $t \in \mathcal{T}$ and $\boldsymbol{\omega} \in \boldsymbol{\Omega}$.

- CT : Cycle time.

- $CP_{m,t}$: Capacity of machine m in period t, where $m \in \mathcal{M}$ and $t \in \mathcal{T}$.

- C_m^{I} : Acquisition cost of machine m, where $m \in \mathcal{M}$.

- C^{II} : Configuration cost of a workstation.

- C^{III} : Procurement cost of an EOL product.

- C_i^{IV} : Inventory cost of component i, where $i \in \mathcal{I}$.

- $\overline{\mathcal{G}}$: a sufficiently large number.

First-stage decision variables:

- z_l: Binary variable, equaling to 1 if disassembly scheme l is selected for the disassembly line, 0 otherwise, where $l \in \mathcal{L}$.

- $x_{j,m}$: Binary variable, equaling to 1 if task j is assigned to machine m, 0 otherwise, where $j \in \mathcal{J}$ and $m \in \mathcal{M}$.

- y_m: Binary variable, equaling to 1 if machine m is purchased for the disassembly line, 0 otherwise, where $m \in \mathcal{M}$.

- $u_{m,w}$: Binary variable, equaling to 1 if machine m is assigned to workstation w, 0 otherwise, where $m \in \mathcal{M}$ and $w \in \mathcal{W}$.

- v_w: Binary variable, equaling to 1 if workstation w is configured for locating the acquired machines, 0 otherwise, where $w \in \mathcal{W}$.

Second-stage decision variables:

- $\beta_t(\boldsymbol{\omega})$: Real-number variable, representing the amount of disassembled EOL products in period t, where $t \in \mathcal{T}$ and $\boldsymbol{\omega} \in \boldsymbol{\Omega}$.

- $\gamma_{i,t}(\boldsymbol{\omega})$: Real-number variable, indicating the inventory amount of component i at the end of period t, where $i \in \mathcal{I}$, $t \in \mathcal{T}$ and $\boldsymbol{\omega} \in \boldsymbol{\Omega}$.

Based on Birge and Louveaux (2011), in two-stage stochastic programming, deterministic decisions are taken in the first stage, after which the occurred random events can affect the outcome of the first-stage decision. To compensate for the potential effects in the first-stage, recourse decisions are made in the second stage for each scenario. Therefore, two decision stages have interacted relation (Küçükyavuz and Sen, 2017). For the considered ISDLBPP, the first-stage deterministic decisions consist of machine acquisition and workstation configuration. And the second-stage decisions optimize the numbers of purchased EOL products and component inventory in each period for each realization of component demand and yield ratio (i.e., scenario $\boldsymbol{\omega}$). The TSMIP model for the considered problem is established as follows.

TSMIP Model [P1]:

$$\min \left(\sum_{m \in \mathcal{M}} C_m^{\mathrm{I}} \cdot y_m + C^{\mathrm{II}} \cdot \sum_{w \in \mathcal{W}} v_w \right) + \mathbb{E}_{\boldsymbol{\Omega}} \left[\mathcal{Q} \left(y_m, \boldsymbol{\omega} \right) \right] \tag{1}$$

Subject to:

$$\sum_{l \in \mathcal{L}} z_l = 1 \tag{2}$$

$$\sum_{m \in \mathcal{M}} x_{j,m} \leq 1 + \overline{\mathcal{G}} \cdot (1 - z_l), \qquad \forall j \in \mathcal{J}_l \tag{3}$$

$$\sum_{m \in \mathcal{M}} x_{j,m} \geq 1 - \overline{\mathcal{G}} \cdot (1 - z_l), \qquad \forall j \in \mathcal{J}_l \tag{4}$$

$$x_{j,m} \leq a_{j,m}, \qquad \forall j \in \mathcal{J}, \forall m \in \mathcal{M} \tag{5}$$

$$\sum_{j \in \mathcal{J}} x_{j,m} = \sum_{j \in \mathcal{J}} a_{j,m} \cdot y_m, \qquad \forall m \in \mathcal{M} \tag{6}$$

$$y_m + y_{m'} \leq 2 - b_{m,m'}, \qquad \forall m, m' \in \mathcal{M}, m \neq m' \tag{7}$$

$$u_{m,w} \leq y_m, \qquad \forall m \in \mathcal{M}, \forall w \in \mathcal{W} \tag{8}$$

$$\sum_{w\in\mathcal{W}} u_{m,w} \le 1+\overline{\mathcal{G}}\cdot(1-y_m), \qquad \forall m\in\mathcal{M} \tag{9}$$

$$\sum_{w\in\mathcal{W}} u_{m,w} \ge 1-\overline{\mathcal{G}}\cdot(1-y_m), \qquad \forall m\in\mathcal{M} \tag{10}$$

$$u_{m,w}+u_{m',w} \le 1+c_{m,m'}, \qquad \forall m,m'\in\mathcal{M}, m\neq m', \forall w\in\mathcal{W} \tag{11}$$

$$\sum_{m\in\mathcal{M}} p_m\cdot u_{m,w} \le CT, \qquad \forall w\in\mathcal{W} \tag{12}$$

$$\sum_{m\in\mathcal{M}} u_{m,w} \le |\mathcal{M}|\cdot v_w, \qquad \forall w\in\mathcal{W} \tag{13}$$

$$z_l, x_{j,m}, y_m, u_{m,w}, v_w \in \{0,1\}, \qquad \forall l\in\mathcal{L}, \forall j\in\mathcal{J}, \forall m\in\mathcal{M}, w\in\mathcal{W} \tag{14}$$

$$\mathcal{Q}\left(y_m,\boldsymbol{\omega}\right)=\min\sum_{t\in\mathcal{T}\setminus 0}\left(C^{\mathrm{III}}\cdot\beta_t(\boldsymbol{\omega})+\sum_{i\in\mathcal{I}}C_i^{\mathrm{IV}}\cdot\gamma_{i,t}(\boldsymbol{\omega})\right) \tag{15}$$

$$\gamma_{i,0}(\boldsymbol{\omega})=0, \qquad \forall i\in\mathcal{I}, t=0, \forall\boldsymbol{\omega}\in\boldsymbol{\Omega} \tag{16}$$

$$\gamma_{i,t}(\boldsymbol{\omega})=\gamma_{i,t-1}(\boldsymbol{\omega})+r_i(\boldsymbol{\omega})q_i\cdot\beta_t(\boldsymbol{\omega})-d_{i,t}(\boldsymbol{\omega}), \qquad \forall i\in\mathcal{I}, \forall t\in\mathcal{T}\setminus 0, \forall\boldsymbol{\omega}\in\boldsymbol{\Omega} \tag{17}$$

$$\beta_t(\boldsymbol{\omega})\cdot y_m \le CP_{m,t}, \qquad \forall t\in\mathcal{T}, \forall m\in\mathcal{M}, \forall\boldsymbol{\omega}\in\boldsymbol{\Omega} \tag{18}$$

$$\beta_t(\boldsymbol{\omega}), \gamma_{i,t}(\boldsymbol{\omega}) \ge 0, \qquad \forall i\in\mathcal{I}, \forall t\in\mathcal{T}, \forall\boldsymbol{\omega}\in\boldsymbol{\Omega} \tag{19}$$

The objective function (1) consists of two parts: (i) minimizing machine acquisition and workstation configuration costs for line balancing, and (ii) minimizing product procurement and component inventory costs for line planning, i.e., $\mathcal{Q}\left(y_m,\boldsymbol{\omega}\right)$ calculated in formula (15), known as the *recourse objective*. In model **P1**, constraints (2) to (14) are related to the first-stage decisions, while constraints (16) to (19) associate the second-stage decisions. Especially, $\mathbb{E}[\cdot]$ in objective function (1) means the mathematical expectation.

Constraint (2) indicates that only one disassembly scheme can be selected. Constraints (3) and (4) guarantee that if a scheme is selected, each task in it can only be assigned to one machine. Constraints (5) and (6) ensure that each task can only be assigned to the machine that is acquired and can handle it (task processing ability). Constraints (7) assure that any two machines with overlap cannot be purchased together. Constraints (8) to (10) ensure that each machine can only be allocated in one workstation if it is purchased. Constraints (11) state that any two machines respecting task precedence relation can be placed in the same workstation. Constraints (12) ensure that the total processing time in each workstation must respect the cycle time. Constraints (13) guarantee that is a workstation is configured, the

number of machines in it must be smaller than $|\mathcal{M}|$. In constraints (14), the ranges of decision variables in the first stage are presented.

In terms of the second-stage formulas, constraint (15) defines product purchase cost and component inventory cost. Constraints (16) regulate the initial inventory. Constraints (17) assume the flow conservation of each component in each period. Constraints (18) guarantee that the capacity of each machine in each period should be respected. Finally, constraints (19) mark variable ranges in the second stage.

The above model **P1** is non-linear since two decision variables $\beta_t(\boldsymbol{\omega})$ and y_m multiply in constraints (18). Therefore, it cannot be directly solved by calling commercial optimization solvers, for example, CPLEX. For easier solving model **P1**, we linearize it as follows.

3.3. Model linearization

To linearize constraints (18), an auxiliary variable $\kappa_{m,t}(\boldsymbol{\omega}) = \beta_t(\boldsymbol{\omega}) \cdot y_m$ is introduced. It is obvious that if $y_m = 1$, then $\kappa_{m,t}(\boldsymbol{\omega}) = \beta_t(\boldsymbol{\omega})$ means that if a machine is purchased, its capacity has to be respected for each disassembly period; otherwise $y_m = 0$, and $\kappa_{m,t}(\boldsymbol{\omega}) = 0$ means that if a candidate machine has not been selected, the disassembly quantity of this machine is zero. Then, constraints (18) can be linearized by a group of constraints.

$$\kappa_{m,t}(\boldsymbol{\omega}) \leq \beta_t(\boldsymbol{\omega}), \qquad \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \forall \boldsymbol{\omega} \in \boldsymbol{\Omega} \tag{20}$$

$$\kappa_{m,t}(\boldsymbol{\omega}) \geq \beta_t(\boldsymbol{\omega}) - \overline{\mathcal{G}} \cdot (1 - y_m), \qquad \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \forall \boldsymbol{\omega} \in \boldsymbol{\Omega} \tag{21}$$

$$\kappa_{m,t}(\boldsymbol{\omega}) \leq CP_{m,t}, \qquad \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \forall \boldsymbol{\omega} \in \boldsymbol{\Omega} \tag{22}$$

$$\kappa_{m,t}(\boldsymbol{\omega}) \geq 0, \qquad \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \forall \boldsymbol{\omega} \in \boldsymbol{\Omega} \tag{23}$$

To be specific, if $y_m = 1$, constraints (20) and (21) together guarantee that $\kappa_{m,t}(\boldsymbol{\omega}) = \beta_t(\boldsymbol{\omega})$. Otherwise $y_m = 0$, all constraints are verified. Consequently, constraints (20) to (23) jointly insure that constraints (18) are established. Now a linear formulation of model **P1** can be constructed.

Linear TSMIP model [P2]:

$$\min \left(\sum_{m \in \mathcal{M}} C_m^{\mathrm{I}} \cdot y_m + C^{\mathrm{II}} \sum_{w \in \mathcal{W}} v_w \right) + \mathbb{E}_{\boldsymbol{\Omega}} \left[\mathcal{Q}\left(y_m, \boldsymbol{\omega}\right) \right]$$

s.t.: $(2) - (17), (19),$ **(20)-(23)**

In the following, problem properties are analyzed and solution methods are proposed to efficiently solve linear TSMIP model **P2**.

4. Solution methods

In this section, a valid inequality is first proposed to tighten model **P2** in Section 3 such that the search space of the optimal solution can be reduced. Then, the *SAA* method and the *L-shaped* method are adopted to solve the studied problem.

4.1. Model improvement

In this part, a valid inequality is proposed with the purpose to pre-determine the maximum number of purchased machines for a disassembly line.

For the EOL product shown in Figure 5, there are two potential disassembly schemes (i.e., $|\mathcal{L}| = 2$) for product separation, each of which has a known task set $\mathcal{J}_l$ ($l \in \mathcal{L}$). For each disassembly scheme, the available machine combinations for accomplishing the tasks can be computed. For example, we have two machine combinations 1a and 1b for disassembly scheme 1; and combinations 2a and 2b for scheme 2. Among them, it is observed that the number of machines included in combination **2b** is the largest, which can be denoted as the *upper bound* of purchased machines for this EOL product. To this respect, a parameter $\mathcal{M}_{upper}$ is introduced to guarantee that the number of purchased machines is no more than the worst case. The valid inequality is established in the following formula, denoted as VI.

$$\textbf{VI:} \quad \sum_{m \in \mathcal{M}} y_m \leq \mathcal{M}_{upper} \tag{24}$$

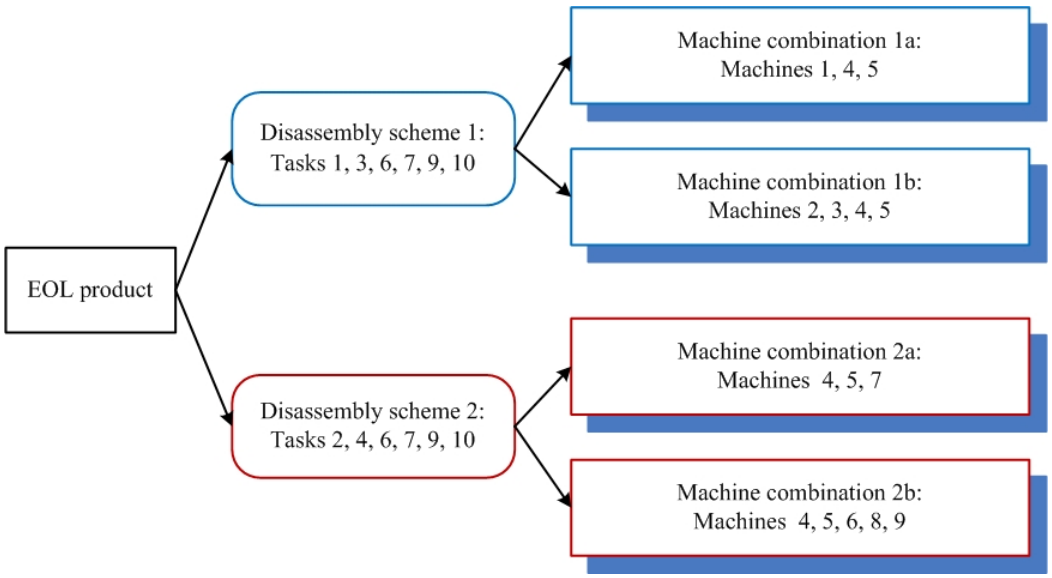

Figure 5: The main idea of the valid inequality.

This VI is redundant for defining the studied problem, but it may enhance the search efficiency of the optimal solution. Experimental results in Section 5 also prove its effectiveness. Therefore, the proposed VI is added into model **P2**, denoted as model **P3** below.

Linear TSMIP model with the VI [P3]:

$$\min \left(\sum_{m \in \mathcal{M}} C_m^{\mathrm{I}} \cdot y_m + C^{\mathrm{II}} \sum_{w \in \mathcal{W}} v_w \right) + \mathbb{E}_{\mathbf{\Omega}} \left[\mathcal{Q}\left(y_m, \boldsymbol{\omega}\right) \right]$$
$$\textbf{s.t.:}\ (2)-(17), (19)-(23), \textbf{(24)}$$

In the following, two widely used scenario-based solution methods, including the SAA method and the L-shaped method, are applied to solve model **P3**.

4.2. The SAA method

The SAA method, based on *Monte Carlo Sampling* (MCS) technique, generates a random sample (consisting of $|\mathbf{S}|$ scenarios) from scenario set $\mathbf{\Omega}$ and approximates the objective value by the *sample average formulation* (Emelogu et al., 2016). Without loss of generality, the approximate sample average formulation of model **P3** is proposed as follows, where $\mathbf{S} \in \mathbf{\Omega}$.

Sample average formulation [P4]:

$$\mathbf{f_S} = \min \left(\sum_{m \in \mathcal{M}} C_m^{\mathrm{I}} \cdot y_m + C^{\mathrm{II}} \sum_{w \in \mathcal{W}} v_w \right) + \frac{1}{|\mathbf{S}|} \sum_{s=1}^{|\mathbf{S}|} \mathcal{Q}\left(y_m, s\right)$$
$$\textbf{s.t.:}\ (2)-(14), (24)$$
$$\mathcal{Q}\left(y_m, s\right) = \min \sum_{t \in \mathcal{T}} \left(C^{\mathrm{III}} \cdot \beta_t(s) + \sum_{i \in \mathcal{I}} C_i^{\mathrm{IV}} \cdot \gamma_{i,t}(s) \right), \quad \forall s \in \mathbf{S} \tag{25}$$
$$\gamma_{i,0}(s) = 0, \quad \forall i \in \mathcal{I}, t = 0, \forall s \in \mathbf{S} \tag{26}$$
$$\gamma_{i,t}(s) = \gamma_{i,t-1}(s) + r_i(s) \cdot q_i \cdot \beta_t(s) - d_{i,t}(s), \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \setminus 0, \forall s \in \mathbf{S} \tag{27}$$
$$\kappa_{m,t}(s) \leq \beta_t(s), \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \forall s \in \mathbf{S} \tag{28}$$
$$\kappa_{m,t}(s) \geq \beta_t(s) - \overline{\mathcal{G}} \cdot (1 - y_m), \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \forall s \in \mathbf{S} \tag{29}$$
$$\kappa_{m,t}(s) \leq CP_{m,t}, \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \forall s \in \mathbf{S} \tag{30}$$
$$\beta_t(s), \gamma_{i,t}(s), \kappa_{m,t}(s) \geq 0, \quad \forall m \in \mathcal{M}, \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \forall s \in \mathbf{S} \tag{31}$$

Exactly solving model **P4** can obtain an approximate solution to the original problem. However, it is hard to guarantee that this solution is good since only one

sample with $|\mathbf{S}|$ scenarios is tested until now. Based on Beltran-Royo (2017), Liu et al. (2019), etc., the SAA method can test more than one random samples (denoted as $|\mathbf{K}|$ samples) and obtain their sample average objective values, respectively. Then, computing the mean objective value of all samples can get an approximate solution. That is, model **P4** is solved for $|\mathbf{K}|$ times to obtain the objective value for each sample k, denoted as $\mathbf{f}_{\mathbf{S}}^{k}$, and the mean value is calculated via $\frac{1}{|\mathbf{K}|}\sum_{k=1}^{|\mathbf{K}|}\mathbf{f}_{\mathbf{S}}^{k}$.

There may be a tradeoff between the size of $|\mathbf{S}|$ and $|\mathbf{K}|$. To be specific, if $|\mathbf{S}|$ is big enough, the one-sample test may be sufficient to get a near-optimal solution; Otherwise, if $|\mathbf{S}|$ is quite small, the $|\mathbf{K}|$-sample test may ensure that more instances can be tested, which may be more accurate than the one-sample test.

Most existing works simply test $|\mathbf{K}|$ random samples with replacement, however, there is a risk that two samples contain totally the same scenarios. To avoid this, we regulate that the number of the repeated scenarios between any two samples should be no more than $\bar{U}$, such that there are less repetition-tested scenarios. The framework of the SAA algorithm for model **P3** is presented in *Algorithm 1*.

Algorithm 1: The SAA algorithm for model **P3**.

Input: Model **P4** and related parameters, $|\mathbf{K}|$ and $|\mathbf{S}|$.

Let $k = 1$, obtain $|\mathbf{S}|$ scenarios from $\mathbf{\Omega}$;
Solve exactly model **P4** to obtain $\mathbf{f}_{\mathbf{S}}^{1}$;
while $k < |\boldsymbol{K}|$ **do**
    Set $k = k + 1$;
    Obtain $|\mathbf{S}|$ scenarios for the k-th sample such that the number of the same scenarios between the k-th sample and any previously tested sample is no more than $\bar{\boldsymbol{U}}$. Solve exactly model **P4** to obtain $\mathbf{f}_{\mathbf{S}}^{k}$;
end
Compute the average objective value of $|\mathbf{K}|$ samples via $\frac{1}{|\mathbf{K}|}\sum_{k=1}^{|\mathbf{K}|}\mathbf{f}_{\mathbf{S}}^{k}$;

Output: The approximate objective value.

Although this algorithm is easy to be implemented to find an approximate objective value of model **P3**, its computational efficiency may have a sharp decrease with the growth of the numbers of samples $|\mathbf{K}|$ and scenarios $|\mathbf{S}|$. In the following, the L-shaped method is adapted to resolve model **P3**.

4.3. The L-shaped method

The L-shaped method, based on *Benders decomposition*, is a widely used scenario-based iterative solution method for two-stage stochastic programming (Birge and Louveaux, 2011). The main basis of the L-shaped method is to split the original two-stage stochastic problem into the master problem **MP** (first-stage problem) and

the recourse problem **RP** (second-stage problem). In each iteration of the method, a **MP** is exactly solved to obtain the solution that will be used to form a **RP**. Then, for each scenario, the optimal solution of the dual problem of the **RP** is obtained to generate a new cut that is added to the current **MP** to form a new **MP**. In this way, the **MP** and **RP** are iteratively solved until a stop criterion is met.

For the considered ISDLBPP, the first stage is considered as the **MP** since the disassembly line selection, machine acquisition and workstation configuration decisions are tactical decisions for a disassembly line. Among them, the capacities of the acquired machines determine the upper limits of EOL products that can be decomposed in each period, so it also indirectly affects the number of products to be purchased. The second stage is seen as the **RP**, in which the product procurement and component inventory decisions are made for each scenario, based on the decisions in the **MP**. Then the decisions obtained from the **RP** compensate for the effects of uncertainty and urge the **MP** to adjust its decisions. In the following, we introduce the formulations of the **MP** and **RP** of the considered problem and explain in detail the iterative procedures of the L-shaped method.

Define n as the iteration index. In the initial iteration (i.e., $n = 1$), **MP**(1) can be formulated as follows.

$$\begin{aligned}
&\mathbf{MP}(1)\mathbf{:}\ \min \sum_{m\in\mathcal{M}} C_m^{\mathrm{I}} \cdot y_m + C^{\mathrm{II}} \sum_{w\in\mathcal{W}} v_w \\
&\mathbf{s.t.:}\ \ (2)-(14),(24)
\end{aligned}$$

Exactly solving **MP**(1) can obtain its optimal solution $(z_l^1, y_m^1, v_w^1, x_{j,m}^1, u_{m,w}^1)$. Among them, solution vector y_m^1 will be used to form the initial recourse problem since machine capacity is directly related to the number of EOL products to be separated. For each scenario $\boldsymbol{\omega} \in \boldsymbol{\Omega}$, the initial recourse problem can be formulated as follows.

$$\mathbf{RP}(1,\boldsymbol{\omega})\mathbf{:}\ \min \sum_{t\in\mathcal{T}} \left(C^{\mathrm{III}} \cdot \beta_t(\boldsymbol{\omega}) + \sum_{i\in\mathcal{I}} C_i^{\mathrm{IV}} \cdot \gamma_{i,t}(\boldsymbol{\omega}) \right)$$

$$\mathbf{s.t.:}\ \gamma_{i,0}(\boldsymbol{\omega}) = 0, \quad \forall i \in \mathcal{I}, t = 0 \tag{32}$$

$$\gamma_{i,t}(\boldsymbol{\omega}) = \gamma_{i,t-1}(\boldsymbol{\omega}) + r_i(\omega) \cdot q_i \cdot \beta_t(\boldsymbol{\omega}) - d_{i,t}(\boldsymbol{\omega}), \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \setminus 0 \tag{33}$$

$$\kappa_{m,t}(\boldsymbol{\omega}) \le \beta_t(\boldsymbol{\omega}), \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \tag{34}$$

$$\kappa_{m,t}(\boldsymbol{\omega}) \ge \beta_t(\boldsymbol{\omega}) - \overline{\mathcal{G}} \cdot (1 - \boldsymbol{y}_m^1), \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \tag{35}$$

$$\kappa_{m,t}(\boldsymbol{\omega}) \le CP_{m,t}, \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \tag{36}$$

$$\beta_t(\boldsymbol{\omega}), \gamma_{i,t}(\boldsymbol{\omega}), \kappa_{m,t}(\boldsymbol{\omega}) \ge 0, \quad \forall m \in \mathcal{M}, \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \tag{37}$$

Then, optimal commercial softwares, for example CPLEX, can be called to obtain the optimal solution of the dual problem of $\mathbf{RP}(1,\boldsymbol{\omega})$ for each scenario $\boldsymbol{\omega} \in \boldsymbol{\Omega}$, denoted as $\boldsymbol{\pi}^1_{\boldsymbol{\omega}}$. Next, the following $\mathbf{MP}(2)$ can be formed by adding a new cut into $\mathbf{MP}(1)$, i.e., constraints (38), where $\boldsymbol{\theta}^2$ is a decision variable that is related to the stop criterion, and $\boldsymbol{E}_1$ and $\boldsymbol{e}_1$ are parameters that can be calculated by formulas $\boldsymbol{E}_1 = \sum_{\boldsymbol{\omega}=1}^{|\boldsymbol{\Omega}|} \boldsymbol{p}_{\boldsymbol{\omega}} (\boldsymbol{\pi}^1_{\boldsymbol{\omega}})^{\mathrm{T}} \boldsymbol{T}_{\boldsymbol{\omega}}$, and $\boldsymbol{e}_1 = \sum_{\boldsymbol{\omega}=1}^{|\boldsymbol{\Omega}|} \boldsymbol{p}_{\boldsymbol{\omega}} (\boldsymbol{\pi}^1_{\boldsymbol{\omega}})^{\mathrm{T}} \boldsymbol{h}_{\boldsymbol{\omega}}$, respectively. Note that vectors $\boldsymbol{T}_{\boldsymbol{\omega}}$ and $\boldsymbol{h}_{\boldsymbol{\omega}}$ represent the coefficients of $\boldsymbol{y}^1_m$ and the constant term in $\mathbf{RP}(1,\boldsymbol{\omega})$ under scenario $\boldsymbol{\omega}$.

$$
\begin{aligned}
&\mathbf{MP}(2)\textbf{:}\ \min \sum_{m\in\mathcal{M}} C^{\mathrm{I}}_m \cdot y_m + C^{\mathrm{II}} \sum_{w\in\mathcal{W}} v_w + \boldsymbol{\theta}^2 \\
&\textbf{s.t.:}\ \ (2)-(14),(24) \\
&\qquad \boldsymbol{E}_1 \cdot y_m + \boldsymbol{\theta}^2 \geq \boldsymbol{e}_1, \quad \forall m \in \mathcal{M} && (38)\\
&\qquad \boldsymbol{\theta}^2 \in \mathbb{R} && (39)
\end{aligned}
$$

Without losing of generality, the generic form $\mathbf{MP}(n)$ is established below, in which constraints (40) represent the set of added cuts after each iteration, and $(n-1)$ represents the number of cuts.

$$
\begin{aligned}
&\mathbf{MP}(n)\textbf{:}\ \min \sum_{m\in\mathcal{M}} C^{\mathrm{I}}_m \cdot y_m + C^{\mathrm{II}} \sum_{w\in\mathcal{W}} v_w + \boldsymbol{\theta}^n \\
&\textbf{s.t.:}\ \ (2)-(14),(24) \\
&\qquad \boldsymbol{E}_b \cdot y_m + \boldsymbol{\theta}^n \geq \boldsymbol{e}_b, \quad b = 1,2,...,n-1, \forall m \in \mathcal{M} && (40)\\
&\qquad \boldsymbol{\theta}^n \in \mathbb{R} && (41)
\end{aligned}
$$

The solution vector $\boldsymbol{y}^n_m$ from the optimal solution of $\mathbf{MP}(n)$ is used to build the generic form $\mathbf{RP}(n,\boldsymbol{\omega})$ of the considered problem.

$$
\begin{aligned}
&\mathbf{RP}(n,\boldsymbol{\omega})\textbf{:}\ \min \sum_{t\in\mathcal{T}} \left(C^{\mathrm{III}} \cdot \beta_t(\boldsymbol{\omega}) + \sum_{i\in\mathcal{I}} C^{\mathrm{IV}}_i \cdot \gamma_{i,t}(\boldsymbol{\omega}) \right) \\
&\textbf{s.t.:}\ \ (32)-(34),(36)-(37) \\
&\qquad \kappa_{m,t}(\boldsymbol{\omega}) \geq \beta_t(\boldsymbol{\omega}) - \overline{\mathcal{G}} \cdot (1 - \boldsymbol{y}^n_m), \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} && (42)
\end{aligned}
$$

Similar to the procedures for $\mathbf{RP}(1,\boldsymbol{\omega})$, we calculate that $\boldsymbol{E}_n = \sum_{\boldsymbol{\omega}=1}^{|\boldsymbol{\Omega}|} \boldsymbol{p}_{\boldsymbol{\omega}} (\boldsymbol{\pi}^n_{\boldsymbol{\omega}})^{\mathrm{T}} \boldsymbol{T}_{\boldsymbol{\omega}}$ and $\boldsymbol{e}_n = \sum_{\boldsymbol{\omega}=1}^{|\boldsymbol{\Omega}|} \boldsymbol{p}_{\boldsymbol{\omega}} (\boldsymbol{\pi}^n_{\boldsymbol{\omega}})^{\mathrm{T}} \boldsymbol{h}_{\boldsymbol{\omega}}$, where $\boldsymbol{T}_{\boldsymbol{\omega}}$ and $\boldsymbol{h}_{\boldsymbol{\omega}}$ represent the coefficients of $\boldsymbol{y}^n_m$ and the constant term in $\mathbf{RP}(n,\boldsymbol{\omega})$. Let $\boldsymbol{o}^n = \boldsymbol{e}_n - \boldsymbol{E}_n \cdot \boldsymbol{y}^n_m$. In fact, $\boldsymbol{\theta}^n$ express the approximation of the objective of the recourse problem (Laporte and Louveaux, 1993). Then $\boldsymbol{\theta}^n < \boldsymbol{o}^n$ means that the obtained objective is smaller than the current one,

we add a new cut into $\mathbf{MP}(n)$ to form $\mathbf{MP}(n+1)$ for further test. Otherwise, the L-shaped algorithm stops and outputs the approximate solution and objective value. The framework of the L-shaped algorithm is detailed in *Algorithm 2* and Figure 6.

Algorithm 2: The L-shaped algorithm for model **P3**.

Input: $\mathbf{MP}(1)$, related parameters, $|\boldsymbol{\Omega}|$ scenarios and probability $\boldsymbol{p_\omega}$.

Initialize $n = 1$; Set $\boldsymbol{\theta}^1 = -\infty$;
Solve exactly $\mathbf{MP}(1)$ and obtain its optimal solution $(z_l^1, y_m^1, v_w^1, x_{j,m}^1, u_{m,w}^1)$, and use y_m^1 to form $\mathbf{RP}(1, \boldsymbol{\omega})$;
for $\boldsymbol{\omega} = 1, 2, ..., |\boldsymbol{\Omega}|$ **do**
    Obtain the optimal solution of the dual problem of $\mathbf{RP}(1, \boldsymbol{\omega})$ as $\boldsymbol{\pi}_{\boldsymbol{\omega}}^1$;
end
Calculate $\boldsymbol{E}_1 = \sum_{\boldsymbol{\omega}=1}^{|\boldsymbol{\Omega}|} \boldsymbol{p_\omega} (\boldsymbol{\pi}_{\boldsymbol{\omega}}^1)^{\mathrm{T}} \boldsymbol{T_\omega}$ and $\boldsymbol{e}_1 = \sum_{\boldsymbol{\omega}=1}^{|\boldsymbol{\Omega}|} \boldsymbol{p_\omega} (\boldsymbol{\pi}_{\boldsymbol{\omega}}^1)^{\mathrm{T}} \boldsymbol{h_\omega}$;
Let $\boldsymbol{o}^1 = \boldsymbol{e}_1 - \boldsymbol{E}_1 \cdot \boldsymbol{y}_m^1$;
while $\boldsymbol{\theta}^n < \boldsymbol{o}^n$ **do**
    Set $n = n + 1$;
    Add a cut $\boldsymbol{E}_{n-1} \cdot y_m + \boldsymbol{\theta}^n \geq \boldsymbol{e}_{n-1}$ into $\mathbf{MP}(n-1)$ to form $\mathbf{MP}(n)$;
    Solve exactly $\mathbf{MP}(n)$ to obtain its optimal solution $(z_l^n, y_m^n, v_w^n, x_{j,m}^n, u_{m,w}^n)$ and $\boldsymbol{\theta}^n$; use y_m^n to form $\mathbf{RP}(n, \boldsymbol{\omega})$;
    for $\boldsymbol{\omega} = 1, 2, ..., |\boldsymbol{\Omega}|$ **do**
        Obtain the optimal solution of the dual problem of $\mathbf{RP}(n, \boldsymbol{\omega})$ as $\boldsymbol{\pi}_{\boldsymbol{\omega}}^n$;
    end
    Calculate $\boldsymbol{E}_n = \sum_{\boldsymbol{\omega}=1}^{|\boldsymbol{\Omega}|} \boldsymbol{p_\omega} (\boldsymbol{\pi}_{\boldsymbol{\omega}}^n)^{\mathrm{T}} \boldsymbol{T_\omega}$ and $\boldsymbol{e}_n = \sum_{\boldsymbol{\omega}=1}^{|\boldsymbol{\Omega}|} \boldsymbol{p_\omega} (\boldsymbol{\pi}_{\boldsymbol{\omega}}^n)^{\mathrm{T}} \boldsymbol{h_\omega}$;
    Let $\boldsymbol{o}^n = \boldsymbol{e}_n - \boldsymbol{E}_n \cdot \boldsymbol{y}_m^n$;
end

Output: The approximate solution and objective value.

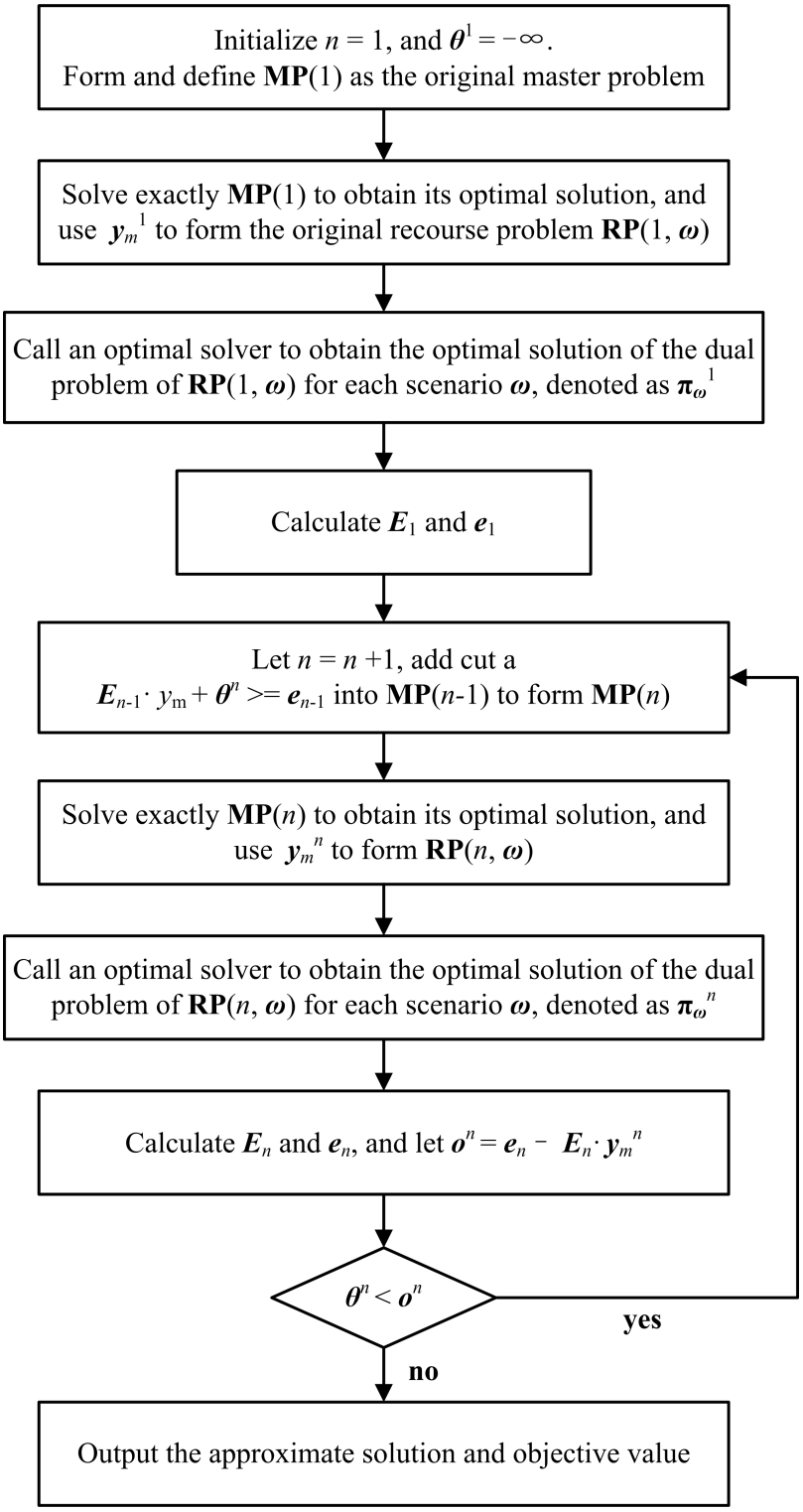

Figure 6: The framework of the L-shaped algorithm.

5. Numerical experiment and analysis

In this section, numerical experiments are conducted on 24 randomly generated instance sets to evaluate the performance of the proposed valid inequality and solution methods. Each instance set contains 5 equal-size instances, therefore, 120 instances are tested in total. The proposed models and solution approaches are coded in MATLAB 2018a by calling commercial CPLEX 12.7. Numerical experiments are conducted on a personal computer with Core i5, 1.8GHz processor, and 8GB RAM in Windows 10 Operating System. Before presenting the computational results, the input information and performance indicators are first presented, respectively.

5.1. Instance settings and performance indicators

The following Table 2 collects the input information in this section. The parameter generation refers to the previous related works and follows the assumptions of our problem (please see the column 'Remarks' in Table 2).

Table 2: Input information of parameters.

Items	Settings	Remarks
$\|\mathcal{J}\|$	[10, 20, 30, 40, 50, 60]	Bentaha et al. (2015) test up to 37
$\|\mathcal{I}\|$	[5, 10, 15, 20, 25, 30]	He et al. (2020) test up to 30
$\|\mathcal{M}\|$	[4, 8, 16, 24, 32, 48, 64]	He et al. (2020) test up to 55
CT	[100, 200, 300, 400]	Respect He et al. (2020)
p_j	randomly generated in [10, 50]	Respect He et al. (2020)
$CP_{m,t}$	randomly generated in [100, 400]	Respect assumption iv
q_i	randomly generated in [1, 5]	Respect Tian and Zhang (2019)
r_i	randomly generated in [0.6, 1]	Respect assumption vi
d_{it}	randomly generated in [10, 50]	Respect assumption vi
C_m^{I}	[100, 120, 140, 160, 180, 200]	Respect assumption iv
$C^{\mathrm{II}}, C^{\mathrm{III}}$	500, 2	-
C_i^{IV}	randomly generated in [1, 3]	Respect Kim and Xirouchakis (2010)

The performance indicators we use in this section are presented below:

- **Obj**: objective value.
- **Time**(s): computation time in seconds.
- $\mathbf{Gap}_{\mathrm{obj}}$: objective value gap (%) between different models or methods.
- $\boldsymbol{\Delta}$(s): computation time deviation between different models or methods.
- **Avg.**: average value.

5.2. Effect of valid inequality

To facilitate operation and inspection, we consider a special case of our problem, in which there exists *only one scenario*. This means that the original problem can be specially transformed into a deterministic problem. To help the readers follow our idea, the following notations are used to conclude the model transformation.

- [**P1**]: original non-linear TSMIP model.
- [**P2**]: transformed linear TSMIP model.
- [**P3**]: transformed linear TSMIP model with the VI.
- [$\mathbf{P2}^{\#}$]: the case that model **P2** includes only one scenario (used for the test).
- [$\mathbf{P3}^{\#}$]: the case that model **P3** includes only one scenario (used for the test).

Without loss of generality, *deterministic models* $\mathbf{P2}^{\#}$ and $\mathbf{P3}^{\#}$ can be established as follows for verifying the effectiveness of the VI, in which only one scenario of component yield ratios and demands is considered.

Deterministic model [$\mathbf{P2}^{\#}$]:

$$\min \sum_{m\in\mathcal{M}} C_m^{\mathrm{I}} \cdot y_m + C^{\mathrm{II}} \sum_{w\in\mathcal{W}} v_w + C^{\mathrm{III}} \sum_{t\in\mathcal{T}} \beta_t + \sum_{i\in\mathcal{I}} \sum_{t\in\mathcal{T}} C_i^{\mathrm{IV}} \cdot \gamma_{i,t} \tag{43}$$

s.t.: (2) − (14)

$$\gamma_{i,0} = 0, \quad \forall i \in \mathcal{I}, t = 0 \tag{44}$$

$$\gamma_{i,t} = \gamma_{i,t-1} + r_i \cdot q_i \cdot \beta_t - d_{i,t}, \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \setminus 0 \tag{45}$$

$$\kappa_{m,t} \leq CP_{m,t}, \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \tag{46}$$

$$\kappa_{m,t} \leq \beta_t, \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \tag{47}$$

$$\kappa_{m,t} \geq \beta_t - \overline{\mathcal{G}} \cdot (1 - y_m), \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \tag{48}$$

$$\beta_t, \gamma_{i,t}, \kappa_{m,t} \geq 0, \quad \forall m \in \mathcal{M}, \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \tag{49}$$

Deterministic model [$\mathbf{P3}^{\#}$]:

$$\min \sum_{m\in\mathcal{M}} C_m^{\mathrm{I}} \cdot y_m + C^{\mathrm{II}} \sum_{w\in\mathcal{W}} v_w + C^{\mathrm{III}} \sum_{t\in\mathcal{T}} \beta_t + \sum_{i\in\mathcal{I}} \sum_{t\in\mathcal{T}} C_i^{\mathrm{IV}} \cdot \gamma_{i,t}$$

s.t.: (2) − (14), (44) − (49), **(24)**

One can recall Subsection 3.2 for understanding the meaning of formulas in models $\mathbf{P2}^{\#}$ and $\mathbf{P3}^{\#}$. In the following, these two deterministic models are optimally solved and computational results are presented in Table 3.

Table 3: Comparison between models $\mathbf{P2}^{\#}$ and $\mathbf{P3}^{\#}$

Instance sets	Input information				$\mathbf{P2}^{\#}$	$\mathbf{P3}^{\#}$	
	$\lvert\mathcal{J}\rvert(\lvert\mathcal{I}\rvert)$	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{M}\rvert$	Obj	Time(s)	Time(s)	Δ^*
1	10(5)	2	4	40980	0.74	0.72	-0.02
2		2	8	41290	1.46	1.43	-0.03
3		4	8	14200	1.48	1.42	-0.06
4		4	16	52570	5.95	5.88	-0.07
5	20(10)	2	4	85810	0.92	0.87	-0.05
6		2	8	56450	1.70	1.66	-0.04
7		4	8	75830	1.78	1.70	-0.08
8		4	16	70420	6.62	6.44	-0.18
9	30(15)	4	16	159720	6.95	6.06	-0.77
10		4	24	117720	18.75	15.93	-2.82
11		6	24	193770	20.81	16.73	-4.08
12		6	36	112590	54.92	49.12	-5.80
13	40(20)	4	16	217570	7.09	6.37	-0.72
14		4	24	225720	18.77	16.26	-2.51
15		6	24	112360	18.87	15.57	-3.30
16		6	36	200930	62.55	56.36	-6.19
17	50(25)	6	36	274670	58.12	53.26	-0.486
18		6	48	293160	160.03	137.36	-22.67
19		8	48	337520	158.75	129.72	-29.03
20		8	64	225570	587.70	535.66	-52.04
21	60(30)	6	36	334580	66.23	58.16	-6.07
22		6	48	459210	175.09	152.81	-22.28
23		8	48	306250	194.24	163.79	-30.45
24		8	64	304500	626.74	567.41	-59.33
Avg.	-	-	-	**179730**	94.0	83.4	**-11.2%**

* Δ = Time($\mathbf{P3}^{\#}$) − Time($\mathbf{P2}^{\#}$)

From Table 3, it can be seen in column 'Obj' that the two models output the same objective values for each instance set. However, their computation times have a big difference. Specifically, the computation time of model $\mathbf{P2}^{\#}$ changes from 0.74s to 626.74s and has an average value of 94.01s. While the computation time of model $\mathbf{P3}^{\#}$ varies from 0.72s to 567.41s with an average value of 83.4s. The computation time deviation of the two models in column 'Δ' demonstrates that model $\mathbf{P3}^{\#}$ can reduce averagely 11.2% computation time compared with model $\mathbf{P2}^{\#}$ for the tested instances. With this observation, we then draw the average computation time deviation along with the change of the number of machines (other parameters unchanged) in Figure 7. We find that the time difference between the two models

is becoming larger especially when the number of machines is very big (for example from 36 to 64 in Figure 7). Therefore, it can be concluded that embedding the VI can reduce the computation time and reduce the search space.

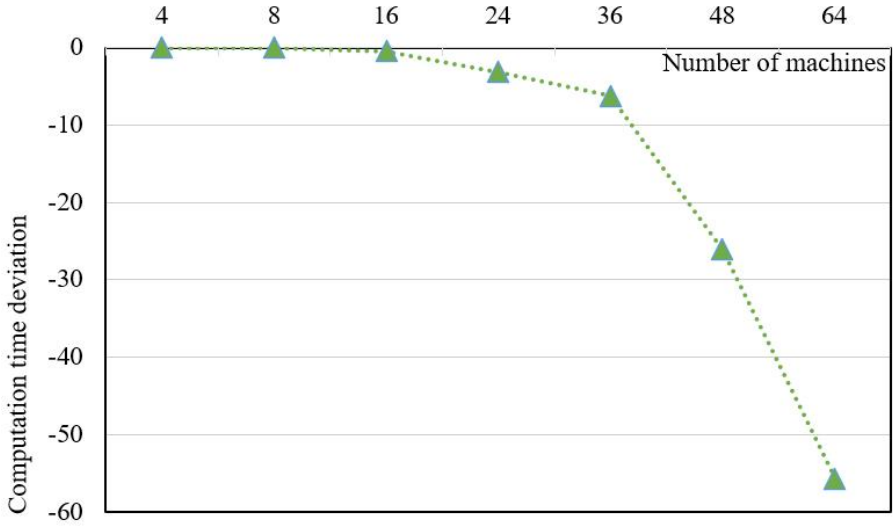

Figure 7: Computation time deviation with the number of machines.

5.3. Comparison of solution methods

In this part, model **P3** that integrates the VI is used to compare the performance of the SAA and L-shaped algorithms. Before conducting numerical experiments, preliminary analysis is conducted to adjust the scenario size and the sample parameters for the SAA algorithm (please see Appendix A and Appendix B).

5.3.1. Computational results

The computational results of the two algorithms are reported in Table 4. In this table, the objective values of the SAA algorithm vary from 21480 to 340150 with an average value of 170537. While the L-shaped algorithm outputs objective values from 23400 to 335570, whose mean value is 172426. Comparing their objective value gap in column 'Gap_{obj}', we conclude that the two algorithms have a similar performance in terms of objective values and the average objective value gap between them is only 1.67%. Especially, in some instance sets (please see the bolded values in column 'Gap_{obj}'), the L-shaped algorithm has smaller objective values than those of the SAA algorithm. The reason may be that the SAA algorithm cannot always guarantee the tested samples cover enough scenarios in set $\boldsymbol{\Omega}$.

In terms of the computation time, it is observed in Table 4 that the SAA algorithm has an average time of 5038.4s, ranging from 11.2s to 37983.7s. On the other

hand, the computation time of the L-shaped algorithm varies from 321.3s to 5398.2s, whose mean value is only 1772.9s. Observing their computation time deviation in column 'Δ', we identify that the L-shaped algorithm can save averagely 64.8% of computation time for the tested instances. Especially, the L-shaped algorithm shows a big superiority when the number of machines reaches 48 and 64, which saves more than 5000s and 30000s, respectively (please see the bolded values in column 'Δ'). In summary, the L-shaped algorithm can save more than 60% of the computation time compared with the SAA algorithm, without sacrificing the solution quality.

Table 4: Comparison between the SAA and L-shaped algorithms

Sets	Input information			**SAA**		**L-shaped**		**Comparison**	
	$\|\mathcal{J}\|(\|\mathcal{I}\|)$	$\|\mathcal{L}\|$	$\|\mathcal{M}\|$	Obj	Time(s)	Obj	Time(s)	$\text{Gap}^*_{\text{obj}}$	Δ^{**}
1	10(5)	2	4	51040	11.2	51500	321.3	0.90	309.7
2		2	8	40090	29.8	41340	425.6	3.12	395.8
3		4	8	21480	32.1	23400	419.8	8.94	387.7
4		4	16	55120	146.2	51970	688.9	**-5.71**	542.7
5	20(10)	2	4	80970	22.1	82370	365.7	1.73	344.5
6		2	8	37860	51.3	40290	523.6	6.42	472.3
7		4	8	79030	52.6	86410	635.1	9.34	582.5
8		4	16	78070	176.8	78810	836.6	0.95	659.8
9	30(15)	4	16	125010	168.5	125890	1011.6	0.70	843.1
10		4	24	109170	503.6	110860	1548.1	1.55	1044.5
11		6	24	175500	511.5	170850	1497.3	**-2.65**	985.8
12		6	36	103670	2125.6	105500	1870.2	1.77	-255.4
13	40(20)	4	16	215020	162.4	213760	1167.7	**-0.59**	1005.3
14		4	24	224900	499.4	228180	1498.2	1.46	998.8
15		6	24	163620	509.6	168230	1532.9	2.82	1023.3
16		6	36	202180	1908.7	200790	2037.8	**-0.69**	129.1
17	50(25)	6	36	267150	2005.6	270410	1823.2	1.22	-182.4
18		6	48	282490	7846.4	295470	2612.5	4.59	**-5233.9**
19		8	48	300080	7878.1	304350	2598.7	1.42	**-5279.4**
20		8	64	251210	36890.6	253960	5108.9	1.09	**-31781.7**
21	60(30)	6	36	290560	2478.3	296610	2250.9	2.08	-227.4
22		6	48	313110	9682.5	317050	3178.8	1.26	**-6503.7**
23		8	48	340150	9245.1	335570	3198.7	**-1.35**	**-6046.4**
24		8	64	285410	37983.7	284650	5398.2	**-0.27**	**-32585.5**
average	-	-	-	170537	5038.4	172426	1772.9	**1.67%**	**64.8%**

* Gap_{obj} = [Obj(L-shaped) − Obj(SAA)]/Obj(SAA)

** Δ = Time(L-shaped) − Time(SAA)

5.3.2. Sensitivity analysis

Firstly, we discuss the impact of product complexity on computational results, that is, we focus on the number of tasks. To be specific, comparing sets 1-4 with sets 5-8, sets 9-12, etc., we cannot find a strict relationship between the number of tasks and the computation time. However, we observe that objective values increase in general with the growth of the number of tasks, while other parameters remain unchanged. Drawing the average objective values along with the number of tasks for the two approaches in Figure 8, we see clearly that (i) the objective values and the number of tasks have a positive correlation. (ii) The two algorithms have a similar performance on objective values.

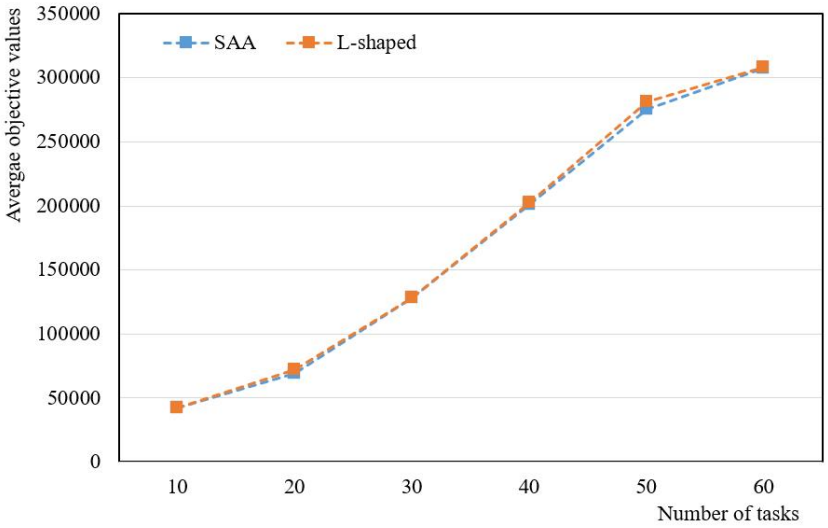

Figure 8: The trend of objective values with the number of tasks.

Secondly, we explore the effect of the available range of equipment on computational results. For easier illustration, we calculate the average objective values and average computation time along with the number of machines, respectively. It is observed that the number of machines cannot regularly influence the objective values. On the contrary, we can see that the computation time increases with the growth of the number of machines. We draw the computation times of the two algorithms along with the number of machines in Figure 9. This figure shows obviously that (i) the computation times of the two algorithms increase with the growth of the number of machines; (ii) The trend of the computation time of the L-shaped algorithm is more stable, which varies much less slowly compared with the SAA algorithm.

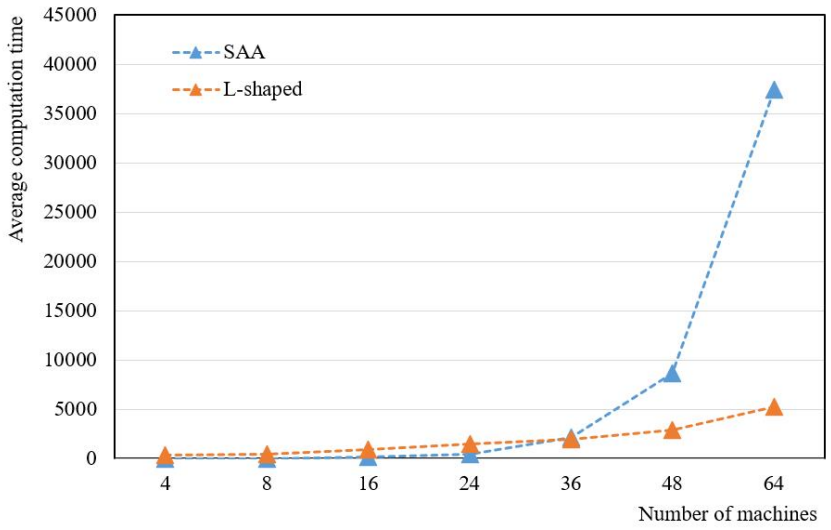

Figure 9: The trend of computation time with the number of machines.

5.3.3. Management insights

There are some management insights that we can learn from the proposed problem and solution methods.

- If disassembly line balancing and planning are not optimized integrally, there is a potential risk that the decisions made in balancing maybe not feasible in planning, at least not a perfect match. While the overall optimization in this paper pre-eliminates this case and may enhance disassembly performance and reduce system cost.

- The complexity of product structure may have a big influence on the system cost. This consultation can assist managers to make better decisions.

- The number of available equipment in markets may directly affect the time required to make a reasonable decision. This is also what managers need to consider when making decisions.

6. Conclusions and perspectives

In this paper, a novel integrated stochastic disassembly line balancing and planning problem is investigated, where the demands and yield ratios of components are uncertain. The objective of this integrated problem is to minimize the system cost. For the problem, a non-linear two-stage stochastic mixed-integer programming model is constructed, which is further transformed into a linear one. Based on problem

property analysis, a valid inequality is proposed and proved to be effective to reduce the search space of an optimal solution, especially when the problem size is bigger. Then, the SAA algorithm and L-shaped algorithm are proposed to solve the linear two-stage stochastic programming. Computational results demonstrate that (i) the number of disassembly task that relies on the structure of EOL products generally impacts the objective value; (ii) The number of available machines is considerably related to the computation time; (iii) The two algorithms have a similar performance in terms of objective values. However, the L-shaped algorithm can save 64.8% of computation time on average, compared with the SAA algorithm, especially when the number of machines is bigger. In the end, several discussions on the benefits of our problem and solution methods are provided.

Potential future research directions may include: (i) the disassembly of multi-type EOL products can be considered, the reason is that decomposing similar EOL products in a common disassembly line may save limited resources and enhance disassembly efficiency; (ii) Other kinds of disassembly lines can be adopted into integrated disassembly line problems, for example, the parallel line, U-shape line, etc.; (iii) Some old disassembly lines may not be suitable anymore for a new EOL product, and forming a new disassembly line may be quite expensive. Consequently, a reconfigurable disassembly line with higher-technology equipment can be designed for decomposing new types of EOL products; (iv) More efficient heuristics are expected to be developed for solving each sample in the SAA algorithm, and systematic evaluation criterion is needed to identify the approximation of stochastic methods.

Acknowledgements

This work was partly supported by the National Natural Science Foundation of China [grant number: 71832001, 71771048, 71571134].

Appendix A

With respect to Emelogu et al. (2016), we examine six scenario sizes including $|\mathbf{\Omega}| = 3, 5, 10, 20, 30, 40$ based on the instances with 10, 20, 30, 40 tasks, respectively (referred as tests 1-4). The parameter information respects the instance settings in Subsection 5.1. The computational results under different scenario sizes are reported in Table 5 and their corresponding average results are illustrated in Figure 10, where the numbers on the horizontal axis denote the tested scenario sizes.

It is observed from Table 5 that the obtained objective values under $|\mathbf{\Omega}| = 3, 5, 10, 20, 30, 40$ are very close to each other. More precisely, it can be observed in

Figure 10 that the average objective values under $|\mathbf{\Omega}| = 40$ and $|\mathbf{\Omega}| = 20$ outperform those of other cases. As for the average computation time, it is concluded from Figure 10 that the computation time has an obvious and sharp increase with the growth of the scenario size. Although the objective value under $|\mathbf{\Omega}| = 40$ is the best, the corresponding computation time is almost three times than that of $|\mathbf{\Omega}| = 20$. Considering that the objective values under $|\mathbf{\Omega}| = 20$ is good and the computation time is smaller than that of $|\mathbf{\Omega}| = 40$ (Liu et al., 2020a), therefore, the scenario size is determined as 20 for the further experiments.

Table 5: Scenario-size adjusting for the proposed algorithms.

	SAA under $\|\mathbf{\Omega}\| = 3$		SAA under $\|\mathbf{\Omega}\| = 5$		SAA under $\|\mathbf{\Omega}\| = 10$	
Tests	Obj	Time(s)	Obj	Time(s)	Obj	Time(s)
1	51540	1.89	51460	2.84	51820	5.51
2	84870	2.44	90840	3.93	74640	7.89
3	118600	22.63	113200	38.75	114940	88.83
4	198800	27.33	197240	47.24	209560	102.59
Avg.	113453	13.57	113185	23.19	112740	51.20
	SAA under $\|\mathbf{\Omega}\| = 20$		SAA under $\|\mathbf{\Omega}\| = 30$		SAA under $\|\mathbf{\Omega}\| = 40$	
Tests	Obj	Time(s)	Obj	Time(s)	Obj	Time(s)
1	51260	11.16	49049	16.19	48935	21.94
2	86140	16.75	80518	23.81	79431	32.87
3	113600	228.63	115350	374.48	112980	603.19
4	198790	249.45	209760	415.89	205750	668.83
Avg.	112448	126.49	113669	207.59	111774	331.71

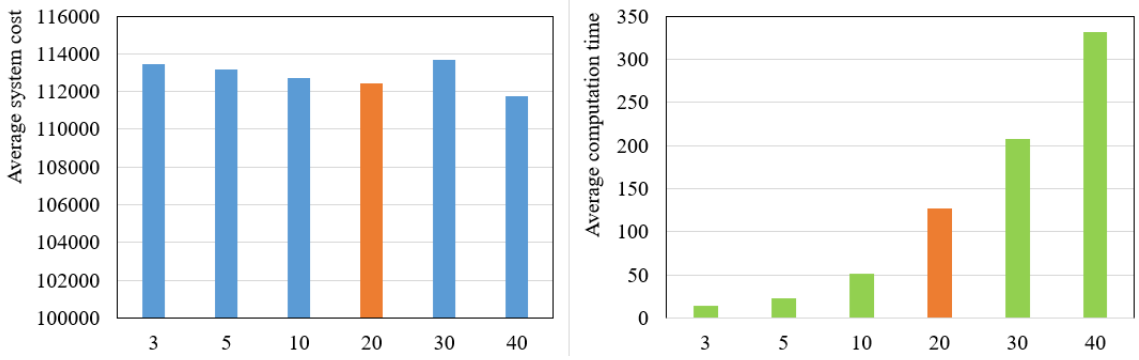

Figure 10: The impact of the scenario size on the objective and computation time.

Appendix B

After deciding $|\mathbf{\Omega}| = 20$, the tradeoff between $|\mathbf{K}|$ and $|\mathbf{S}|$ should be well defined for the SAA algorithm, as discussed in Subsection 4.2. In this respect, six different combinations of $|\mathbf{K}|$ and $|\mathbf{S}|$ that guarantee $|\mathbf{K}| \cdot |\mathbf{S}| = 20$ have been tested, based on the instances with 10, 20, 30, 40 tasks (denoted as tests 5-8). The computational results for these instances are collected in Table 6 and the corresponding average values are presented in Figure 11.

It is observed from Table 6 that the average objective values under different combinations slightly increase from 112448 to 117750, with the increase of $|\mathbf{K}|$ or decrease of $|\mathbf{S}|$. While the average computation times reduce from 126.49s to 91.81s. Although the objective value under $|\mathbf{K}| = 1$ and $|\mathbf{S}| = 20$ is better than other ones, its computation time is the biggest among different combinations. According to the corresponding average results in Figure 11, we observe that the objective values under $|\mathbf{K}| = 4$, $|\mathbf{S}| = 5$ and $|\mathbf{K}| = 5$, $|\mathbf{S}| = 4$ outperform other ones, and their objective values is very close to each other. However, the computation time under $|\mathbf{K}| = 5$, $|\mathbf{S}| = 4$ is less than that of $|\mathbf{K}| = 4$, $|\mathbf{S}| = 5$. Comprehensively considering the objective value and computation time, we use $|\mathbf{K}| = 5$, $|\mathbf{S}| = 4$ for the SAA algorithm because it can efficiently solve the problem with only a small sacrifice of solution quality.

Table 6: Parameter adjusting for the SAA algorithm.

	$\lvert\mathbf{K}\rvert = 1, \lvert\mathbf{S}\rvert = 20, \bar{U} = 0$		$\lvert\mathbf{K}\rvert = 2, \lvert\mathbf{S}\rvert = 10, \bar{U} = 5$		$\lvert\mathbf{K}\rvert = 4, \lvert\mathbf{S}\rvert = 5, \bar{U} = 3$	
Tests	Obj	Time(s)	Obj	Time(s)	Obj	Time(s)
5	51260	11.16	53460	11.86	53500	12.07
6	86140	16.75	86590	15.92	86140	16.38
7	113600	228.63	118640	184.92	115500	179.93
8	198790	249.45	206880	220.87	203650	198.68
Avg.	112448	126.49	116392	108.39	114687	101.77
	$\lvert\mathbf{K}\rvert = 5, \lvert\mathbf{S}\rvert = 4, \bar{U} = 2$		$\lvert\mathbf{K}\rvert = 10, \lvert\mathbf{S}\rvert = 2, \bar{U} = 1$		$\lvert\mathbf{K}\rvert = 20, \lvert\mathbf{S}\rvert = 1, \bar{U} = 1$	
Tests	Obj	Time(s)	Obj	Time(s)	Obj	Time(s)
5	54200	11.56	51380	12.05	53870	11.78
6	86770	15.24	86840	14.27	87650	15.66
7	113920	168.13	114840	171.99	122360	165.12
8	204790	179.49	209180	171.26	207120	174.67
Avg.	114920	93.61	115560	92.39	117750	91.81

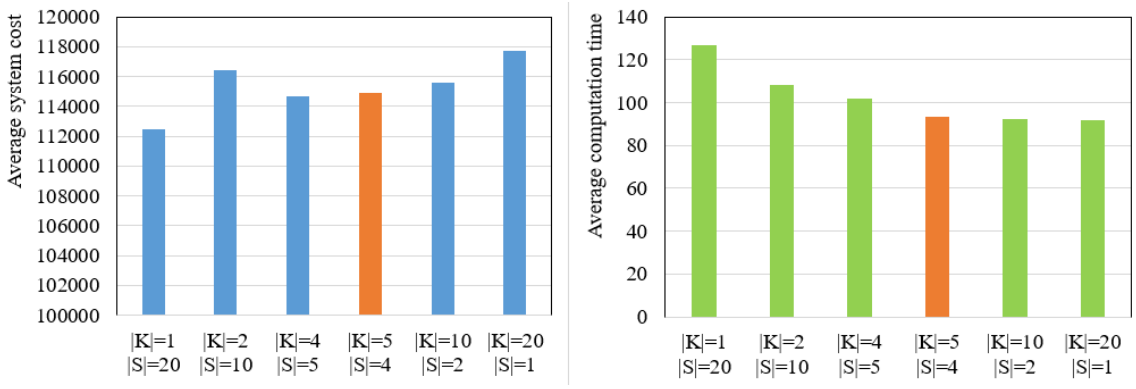

Figure 11: The impact of $|\mathbf{K}|$ and $|\mathbf{S}|$ on the objective and computation time.

In summary, each generated instance contains 20 scenarios for further comparing the SAA algorithm and the L-shaped algorithm. Moreover, the SAA algorithm tests 4 scenarios each time and repeats 5 times, i.e., $|\mathbf{S}| = 4$ and $|\mathbf{K}| = 5$. Therefore, it tests finally 20 scenarios. The upper limit of the task repetition of any two samples is 2. Differently, the L-shaped algorithm directly tests all the 20 scenarios.

References

[1] https://www.worldbank.org/en/news/press-release/2018/09/20/global-waste-to-grow-by-70-percent-by-2050-unless-urgent-action-is-taken-world-bank-report.

[2] Beltran-Royo, C. 2017. "Two-stage stochastic mixed-integer linear programming: The conditional scenario approach." *Omega*, 70: 31-42.

[3] Bentaha, M. L., O. Battaïa, A. Dolgui. 2015. "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.

[4] Birge, J. R. and F. Louveaux. 2011. "Introduction to stochastic progamming." *Springer (New York), 2rd Edition.*

[5] Ehm, F. 2019. "A data-driven modeling approach for integrated disassembly planning and scheduling." *Journal of Remanufacturing*, 9(2): 89-107.

[6] Emelogu, A., S. Chowdhury, M. Marufuzzaman, L. Bian, and B. Eksioglu. 2016. "An enhanced sample average approximation method for stochastic optimization." *International Journal of Production Economics*, 182: 230-252.

[7] Esmaeilbeigi, R., B. Naderi, and P. Charkhgard. 2016. "New formulations for the setup assembly line balancing and scheduling problem." *OR spectrum*, 38(2): 493-518.

[8] Güngör, A., and S. M. Gupta. 1999. "Disassembly Line Balancing." *In Proceedings of the Annual Meeting of the Northeast Decision Science Institute*, 193-195. Newport, RI.

[9] He, J., F. Chu, F. Zheng, and M. Liu. 2020. "A green-oriented bi-objective disassembly line balancing problem with stochastic task processing times." *Annals of Operations Research*, doi:10.1007/s10479-020-03558-z.

[10] Ji, X., Z. Zhang, S. Huang, and L. Li. 2016. "Capacitated disassembly scheduling with parts commonality and start-up cost and its industrial application." *International Journal of Production Research*, 54(4): 1225-1243.

[11] Kannan, D., K. Garg, P. C. Jha, and A. Diabat. 2017. "Integrating disassembly line balancing in the planning of a reverse logistics network from the perspective of a third party provider." *Annals of Operations Research*, 253(1): 353-376.

[12] Kazancoglu, Y., and Y. Ozturkoglu. 2018. "Integrated framework of disassembly line balancing with green and business objectives using a mixed MCDM." *Journal of Cleaner Production*, 191: 179-191.

[13] Kim, H. J., D. H. Lee, and P. Xirouchakis. 2005. "A Lagrangean heuristic algorithm for disassembly scheduling with capacity constraints." *Journal of the Operational Research Society*, 57(10): 1231-1240.

[14] Kim, J. G., H. B. Jeon, H. J. Kim, D. H. Lee, and P. Xirouchakis. 2006. "Disassembly scheduling with capacity constraints: minimizing the number of products disassembled." *Journal of Engineering Manufacture*, 220(9): 1473-1481.

[15] Kim, H. J., D. H. Lee, and P. Xirouchakis. 2007. "Disassembly scheduling: literature review and future research directions." *International Journal of Production Research*, 45(18-19): 4465-4484.

[16] Kim, H. J., and P. Xirouchakis. 2010. "Capacitated disassembly scheduling with random demand." *International Journal of Production Research*, 48(23): 7177-7194.

[17] Küçükyavuz, S., and S. Sen. 2017 "An introduction to two-stage stochastic mixed-integer programming." *Leading Developments from INFORMS Communities. INFORMS*, 1-27.

[18] Laporte, G., and F. V. Louveaux. 1993. "The integer L-shaped method for stochastic integer programs with complete recourse." *Operations research letters* 13(3): 133-142.

[19] Lee, D. H., J. G. Kang, and P. Xirouchakis. 2001. "Disassembly planning and scheduling: review and further research." *Journal of Engineering Manufacture*, 215(5): 695-709.

[20] Liu, M., B. Liang, F. Zheng, and F. Chu. 2019. "Stochastic airline fleet assignment with risk aversion." *IEEE Transactions on Intelligent Transportation Systems*, 20(8): 3081-3090.

[21] Liu, M., X. Liu, F. Chu, F. Zheng, C. Chu. 2020a. "Profit-oriented distributionally robust chance constrained flowshop scheduling considering credit risk." *International Journal of Production Research*, 58(8): 2527-2549.

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

[23] Liu, X., F. Chu, A. Dolgui, F. Zheng, and M. Liu. 2020b. "Service-oriented bi-objective robust collection-disassembly problem with equipment selection." *International Journal of Production Research*, doi:10.1080/00207543.2020.1723815.

[24] Naderi, B., A. Azab, and K. Borooshan. 2019. "A realistic multi-manned five-sided mixed-model assembly line balancing and scheduling problem with moving workers and limited workspace." *International Journal of Production Research*, 57(3): 643-661.

[25] Özcan, U. 2019. "Balancing and scheduling tasks in parallel assembly lines with sequence-dependent setup times." *International Journal of Production Economics*, 213: 81-96.

[26] Özceylan, E., T. Paksoy, and T. Bektaş. 2014. "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.

[27] Özceylan, E., C. B. Kalayci, A. Güngör, and S. M. Gupta. 2019. "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.

[28] Prakash, P. K. S., D. Ceglarek, and M. K. Tiwari. 2012. "Constraint-based simulated annealing (CBSA) approach to solve the disassembly scheduling problem." *The International Journal of Advanced Manufacturing Technology*, 60(9-12): 1125-1137.

[29] Polotski, V., J. P. Kenne, and A. Gharbi. 2017. "Set-up and production planning in hybrid manufacturing-remanufacturing systems with large returns." *International Journal of Production Research*, 55(13): 3766-3787.

[30] Tang, Y., M. Zhou, E. Zussman, and R. Caudill. 2002. "Disassembly modeling, planning, and application." *Journal of Manufacturing Systems*, 21(3): 200-17.

[31] Tian, X. and Z. Zhang. 2019. "Capacitated disassembly scheduling and pricing of returned products with price-dependent yield." *Omega*, 84: 160-174.

[32] Wu, Z., C. K. Kwong, C. K. M. Lee, and J. Tang. 2016. "Joint decision of product configuration and remanufacturing for product family design." *International Journal of Production Research*, 54(15): 4689-4702.

[33] Zikopoulos, C. 2017. "Remanufacturing lotsizing with stochastic leadtime resulting from stochastic quality of returns." *International Journal of Production Research*, 55(6): 1565-1587.