International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: www.tandfonline.com/journals/tprs20

Distributionally robust and risk-averse optimisation for the stochastic multi-product disassembly line balancing problem with workforce assignment

Xin Liu, Feng Chu, Feifeng Zheng, Chengbin Chu & Ming Liu

To cite this article: Xin Liu, Feng Chu, Feifeng Zheng, Chengbin Chu & Ming Liu (2022) 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, DOI: 10.1080/00207543.2021.1881648

To link to this article: https://doi.org/10.1080/00207543.2021.1881648

Published online: 08 Feb 2021.

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
2022, VOL. 60, NO. 6, 1973–1991
https://doi.org/10.1080/00207543.2021.1881648

Taylor & Francis
Taylor & Francis Group

Distributionally robust and risk-averse optimisation for the stochastic multi-product disassembly line balancing problem with workforce assignment

Xin Liu[a], Feng Chu[b], Feifeng Zheng[a], Chengbin Chu[c] and Ming Liu[d]

[a]Glorious Sun School of Business & Management, Donghua University, Shanghai, People's Republic of China; [b]IBISC, Univ Évry, University of Paris-Saclay, Évry, France; [c]Laboratoire d'Informatique, University of Paris-Est, Champs-sur-Marne, France; [d]School of Economics & Management, Tongji University, Shanghai, People's Republic of China

ABSTRACT

Existing works usually focus on the single-product disassembly line balancing problem (DLBP). In practice, end-of-life (EOL) products to be disassembled may be heterogeneous, and the actual processing time of each task may vary with its assigned worker. This work studies a stochastic multi-product DLBP with workforce assignment, to minimise the system cost. Due to historical data scarcity, we assume that only partial distributional information of uncertain task processing times is known. Exceeding the preset cycle time may lead to a disassembly performance reduction, thus we control the cycle time violation via conditional Value-at-Risk (CVaR) constraints, i.e. in a risk-averse fashion. For the problem, we first propose a novel formulation with distributionally robust CVaR constraints. Then some valid inequalities are proposed, leading to an improved model. Two solution approaches, i.e. an exact cutting-plane method and an approximation method, are further proposed and compared, via numerical experiments. Some managerial insights are also drawn.

ARTICLE HISTORY
Received 28 February 2020
Accepted 10 January 2021

KEYWORDS
Disassembly line balancing; multiple products; workforce assignment; distributionally robust; risk-averse

1. Introduction

Due to the excessive resource consumption and environmental damage, environmentally friendly production has received a large amount of attention from government and public (Zhang et al. "Progress in Enhancement of CO2 Absorption by Nanofluids," 2018; Cai et al. 2019). In industry, recycling and remanufacturing of the EOL products are becoming increasingly important (Zhao et al. 2014). For example, with the development of the new energy automobile industry, China now has the largest electric vehicle fleet all over the world (Zhou et al. 2019), leading to battery waste problems. Thus, the government has adopted concrete measures to handle the recycling of electric vehicle battery (MIIT Ministry of Industry (2018; Tang et al. 2018).

Disassembly process, which handles EOL products and extract reusable components or materials, plays an important role (Ren et al. 2017; Özceylan et al. 2019). Disassembly line balancing problem (DLBP), i.e. to select a disassembly process from candidate ones and assign selected disassembly tasks to opened workstations for each EOL product, is a crucial issue and thus has attracted great attention of practitioners and researchers (Ilgin, Akçay, and Araz 2017; Ren et al. 2018; Deniz and Ozcelik 2019). In a disassembly system, an EOL product and its subassemblies are moved from workstation to workstation until all components are removed. The cycle time, i.e. time span between consecutive release of subassemblies on any two adjacent workstations, is strictly dependent on market demand and customer orders (Braglia, Frosolini, and Zammori 2008). A workstation's working time exceeding the given cycle time may lead to the blockage on previous workstations and starvation on following workstations.

Existing works usually study the DLBPs for a single product under deterministic settings (Kalayci et al. 2015; Kalaycılar, Azizoğlu, and Yeralan 2016; Ren et al. 2017; Deniz and Ozcelik 2019). However, in practice, uncertainties cannot be neglected, especially the uncertainty in disassembly task processing times, due to changeable EOL products' conditions (Bentaha, Battaïa, and Dolgui 2014), different workers' skills, and volatile disassembly environments (Liu et al. 2020a). In addition, the complete probability distribution may be unavailable and only partial information is known, due to the lack of historical data (Delage and Ye 2010).

In deterministic settings, the working time of each workstation is usually bounded by the cycle time (Altekin, Kandiller, and Ozdemirel 2008; Hezer and Kara 2015), but the restriction is not applicable under

CONTACT Ming Liu mingliu@tongji.edu.cn School of Economics & Management, Tongji University, Shanghai 200092, People's Republic of China

stochastic environment with limited distributional information. Related works usually focus on respecting the cycle time via chance constraints (Zheng et al. 2018; Liu et al. 2020a; He et al. 2020). Compared with chance constraints, the CVaR constraints provide more information, i.e. it can not only restrict the cycle time violation probability but can offer guarantees on the expected cycle time excess for each workstation. Therefore, this work is motivated to employ CVaR constraints, i.e. restricting the CVaR of each workstation's working time no larger than the cycle time, from a worst-case perspective.

In addition, EOL products to be disassembled may be heterogeneous and can be categorised into several product types, such that the practical disassembly systems are usually more complicated (Ilgin and Gupta 2010; Hrouga, Godichaud, and Amodeo 2016). For instance, in the real-world disassembly system of electric vehicle batteries, batteries used in different vehicles are typically different (Wegener et al. 2015; Ciez and Whitacre 2016; Yan et al. 2016). The disassembly of each EOL product requires specific operations or workstations. For example, during the disassembly process of the Audi Q5 hybrid battery system, the handling operations often require high manoeuvrability and varying forces and techniques. Besides, workers have different ages, skills and work experiences (Bruecker et al. 2015; Gérard, Clautiaux, and Sadykov 2016), thus the actual processing time of a disassembly task may be different if it is handled by different workers.

Motivated by the above observations, to meet real-world disassembly processes, this work considers a stochastic multi-product DLBP with workforce assignment. In particular, the disassembly task processing times are uncertain, and only limited information on the probability distribution, i.e. the first two moments and unimodality, is available. The problem is to (i) assign workstations to products, (ii) select disassembly process from candidate ones for each product, (iii) assign the disassembly tasks to opened workstations, and (iv) assign workers to disassembly tasks. The objective is to minimise the total system cost, which is the sum of (i) the cost for assigning workstations and workers, and (ii) the cost for operating workstations and handling hazardous tasks. The working time of each workstation should not exceed the given cycle time from a risk-averse perspective (i.e. subject to distributionally robust CVaR constraints).

The contribution of this paper mainly includes:

(1) A new stochastic multi-product disassembly line balancing problem with workforce assignment, under moment and structural distributional information on task processing times, is studied.
(2) For the problem, a novel stochastic programming formulation with distributionally robust CVaR constraints is proposed, in which each workstation's working time should respect the given cycle time in a risk-averse fashion.
(3) Several problem properties are discussed and analysed. Then valid inequalities based on the problem properties are proposed, to enhance the original model.
(4) Two solution methods, i.e. an exact cutting-plane algorithm and an approximation method, which are based on equivalent and approximated transformations of the CVaR constraints, are proposed.

Different from our seminar work (Liu, Liu, and Chu 2020) minimising the CVaR of the cycle time, where disassembly task processing times are normally distributed, this paper considers (i) multiple EOL products, (ii) workforce assignment, (iii) distributionally robust CVaR constraints, and (iv) develops an exact cutting-plane algorithm and approximation method for the problem. Besides, this work focuses on a more general case where the uncertain task processing times can be correlated.

The rest of this work is organised as follows. Section 2 gives a brief related literature review. Section 3 describes the problem and proposes a novel stochastic programming formulation with distributionally robust CVaR constraints. Section 4 develops an exact and an approximation solution methods. In Section 5, computational experiments are conducted and the numerical results are reported. In Section 6, this work is summarised and the future research directions are suggested.

2. Literature review

Most existing researches on stochastic DLBPs assume that the probability distribution of the task processing times is known or can be exactly estimated (Bentaha, Battaïa, and Dolgui 2012; Aydemir-Karadag and Turkbey 2013; Bentaha, Battaïa, and Dolgui 2013b, 2014; Bentaha et al. "Dealing with Uncertainty in Disassembly Line Design," 2014; Bentaha, Battaïa, and Dolgui 2015; Bentaha et al. "Second Order Conic Approximation," 2015; Altekin 2017). This work investigates the stochastic multi-product DLBP with workforce assignment under limited distributional information, from a risk-averse perspective. Thus, in the following, we only review the most related literature, i.e.DLBPs under partial distributional information, DLBPs considering multiple products and stochastic assembly line balancing problems (ALBP).

2.1. Single-product DLBPs under partial distributional information

There are some works considering stochastic single-product DLBPs under partial distributional information, and most of them focus on employing chance constraints. The related works are reported in Table 1.

Zheng et al. (2018) consider a stochastic single-product DLBP, and they assume that task processing times are uncertain and only the mean, standard deviation and upper bounds are given. They improve the classic AND/OR graph, by drawing a decomposition colour graph, to describe candidate disassembly processes. A chance-constrained formulation is constructed for the problem, and an inequality is proposed to approximate the chance constraints, leading to an approximated model that can be solved by calling commercial solvers (e.g. CPLEX). Liu et al. (2020a) investigate a stochastic single-product DLBP, where disassembly task processing times may be correlated and only the mean and covariance matrix of task processing times are known. A distributionally robust chance-constrained formulation is proposed, and an approximated second-order cone programming model is then constructed. He et al. (2020) study a bi-objective stochastic single-product DLBP, under only the mean, standard deviation and upper bound of task processing times. The objectives are to minimise the system cost and to minimise the cycle time. For the problem, a bi-objective chance-constrained formulation is first proposed, which is then approximated and solved via the ϵ-constraint method. He et al. (2019) investigate a bi-objective stochastic single-product DLBP, to minimise the system cost and to minimise the total toxic emission. Only the means, standard deviations and change-rate upper bounds of the uncertain task processing times are given. For the problem, a bi-objective chance-constrained model is proposed and solved by the ϵ-constraint method. Liu et al. (2020b) investigate the problem from a two-stage perspective, i.e. determining the disassembly process and the task-to-workstation assignment in the first stage and the penalty cost for exceeding the cycle time is estimated in the second stage. A two-stage distributional robust formulation is devised, to minimise the worst-case expected system cost. An exact cutting-plane algorithm is developed.

In sum, distributionally robust DLBP with CVaR constraints, to control the worst-case cycle time excess expectation, has not been considered.

2.2. Multi-product DLBPs

In practice, products to be disassembled can be categorised into different types. Existing works usually consider multi-product DLBPs under deterministic settings.

Ilgin, Akçay, and Araz (2017) study a deterministic multi-product DLBP, in which the cycle time cannot be violated. Several objectives are considered, i.e. minimising the cycle time deviation, the number of workstations, the difference between the slack times of different products in a workstation, and the total idle time. In their work, it is assumed that the disassembly process is preset and fixed, the problem only determines the task-to-workstation assignment. Kannan et al. (2017) consider a deterministic multi-product DLBP, which integrates the

Table 1. Difference between this paper with the literature (in problem setting, modelling approach, and solution method).

	Problem setting		Modeling approach		Solution method	
	DLBP characteristics	Restriction of the cycle time excess expectation	Modeling methodology	Valid inequalities	Approximated transformation	Exact approach
Zheng et al. (2018)	Single product		Chance-constrained programming		Approximated mixed-integer models	None[a]
He et al. (2019)	Single product		Chance-constrained programming		Approximated mixed-integer models	None[a]
He et al. (2020)	Single product		Chance-constrained programming		Approximated mixed-integer models	None[a]
Liu, Liu, and Chu (2020)	Single product		(1) Chance-constrained programming; (2) Scenario-based mixed-integer programming		Approximated mixed-integer models	None[a]
Liu et al. (2020b)	Single product		Two-stage stochastic programming		Cutting-plane algorithm[b]	
Liu et al. (2020a)	Single product		Chance-constrained programming		Approximated second-order cone (SOC) model	None[a]
Our work	Multiple products; workforce	✓	CVaR-constrained programming	✓	Approximated SOC model	Cutting-plane algorithm[c]

[a] Models are solved by calling commercial solver CPLEX.
[b] Adding cuts to check whether the worst-case second-stage penalty cost is reached.
[c] Adding cuts to perform the CVaR constraints.

collection and disassembly of EOL products. In their work, the disassembly line is selected, to determine the disassembly capacity, and the precedence relationship between disassembly tasks are not considered. A MIP model is proposed for the problem.

Concluding, literature on the multi-product DLBPs is very rare.

2.3. Stochastic assembly line balancing

The assembly line balancing problem (ALBP) is related to the considered problem. Thus literature on ALBPs with uncertain task processing times is reviewed in this subsection.

Some researches focus on stochastic ALBPs with task processing times within given intervals. Gurevsky, Battaïa, and Dolgui (2012) restrict the maximal working time of workstations no larger than the cycle time, to minimise the maximum working time of workstations. They discuss the stability radius for feasible solutions and propose a polynomial time algorithm computing the stability radius. Gurevsky et al. (2013) also restrict the maximal working time of workstations no larger than the cycle time. They propose a branch-and-bound algorithm, to minimise the number of workstations. HazıR and Dolgui (2013) consider the cycle time minimisation and propose a tight upper bound for the problem. A decomposition-based solution approach is developed. Moreira et al. (2015) restrict the maximal working time of workstations no larger than the cycle time. They propose two mixed-integer programming formulations and a construction heuristic, to minimise the number of workstations. Pereira and Alvarez-Miranda (2018) restrict the maximal working time of workstations no larger than the cycle time, to minimise the number of workstations. For the problem, several lower bounds, a heuristic and an exact branch-and-bound method are developed. Pereira (2018) minimises the maximal regret of the cycle time.

There are some researches on the stochastic ALBPs under known probability distribution. Cakir, Altiparmak, and Dengiz (2011) consider a multi-objective ALBP with normally distributed task processing times, in which the probability of exceeding the cycle time is restricted to be no larger than a preset level. The objective is to minimise the difference between workload of workstations and minimise the design cost. A hybrid simulated annealing algorithm is developed. Zhang, Xu, and Gen (2014) investigate a bi-objective ALBP with task processing times following a uniform distribution, to minimise the cycle time and the total cost. An evolutionary algorithm is proposed for the problem. Pınarbaşı, Yüzükırmızı, and Roklu (2016) study an ALBP with normally distributed task processing times, to optimise the workload balance. A constraint programming approach is proposed. Dolgui and Kovalev (2012) study a robust line balancing problem with uncertain task processing times described via a limited set of scenarios, to minimise the cycle time. Several special cases are discussed, and NP-hardness proofs are provided.

To the best of our knowledge, (i) there is no result for the stochastic DLBP jointly considering multiple products and workforce assignment, and (ii) we are the first to introduce the distributionally robust CVaR constraints. Compared with chance constraints, CVaR constraints not only restrict the cycle time violation probability, but restrict the cycle time excess expectation. Thus we fill this research gap in this paper.

3. Problem description and formulation

In this section, the problem description is first given and a new stochastic programming formulation with distributionally robust CVaR constraints is then proposed.

3.1. Problem description

This work studies a stochastic multi-product disassembly line balancing problem with workforce assignment. The problem involves a set $\mathcal{J}$ of ordered workstations and a set $\mathcal{R}$ of workers. In the problem, there are $|\mathcal{L}|$ types of EOL products. For simplicity and brevity, we use a product $l \in \mathcal{L}$ to represent a type of products with similar characteristics in the following. For each EOL product $l \in \mathcal{L}$, (i) there is a set $\mathcal{T}_l$ of disassembly tasks and $\cup_{l\in\mathcal{L}}\mathcal{T}_l = \mathcal{T}$, where $\mathcal{T}$ denotes the set of all disassembly tasks; (ii) there is a set $\mathcal{S}_l$ of disassembly states (i.e. subassembly nodes) and $\cup_{l\in\mathcal{L}}\mathcal{S}_l = \mathcal{S}$, where $\mathcal{S}$ denotes the set of all disassembly states; (iii) for each disassembly state $k \in \mathcal{S}_l$, its predecessor tasks and its successor tasks are collected in sets P_k and Q_k, respectively.

An illustrative example including two products, expressed by a decomposition colour graph introduced by Zheng et al. (2018), is shown in Figures 1 and 2. The candidate disassembly processes for product 1 and product 2 are shown in Figure 1(a,b), respectively. For example, product 1 can be disassembled into 5 components, and there are two alternative disassembly processes coloured in red and purple, respectively. There are two types of vertices in Figure 1, denoted by $\bigcirc$ and $\square$. A vertex of type $\bigcirc$ implies a disassembly task, and we have $\mathcal{T}_1 = \{\bigcirc\} = \{T_1, T_2, \ldots, T_5\}$. A vertex of type $\square$ represents a disassembly state, and $\mathcal{S}_1 = \{\square\} = \{S_0, S_1, S_2, S_3, S_e\}$, where S_0 denotes the beginning state with no component disassembled and S_e represents the ending state with all required components separated. For

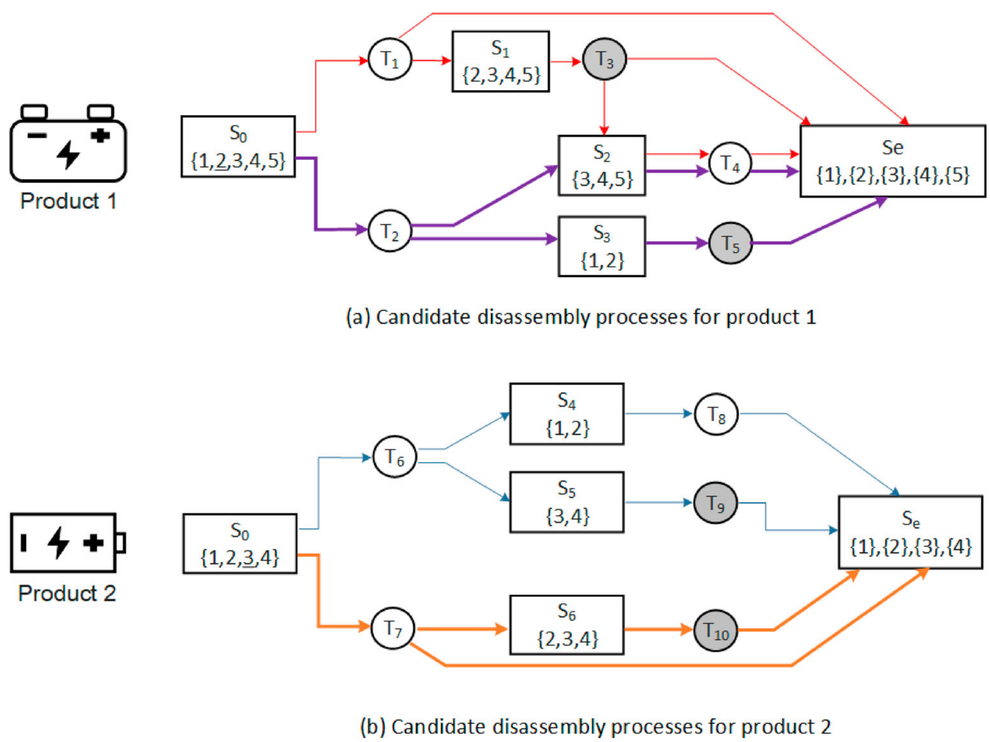

Figure 1. An illustrative example.

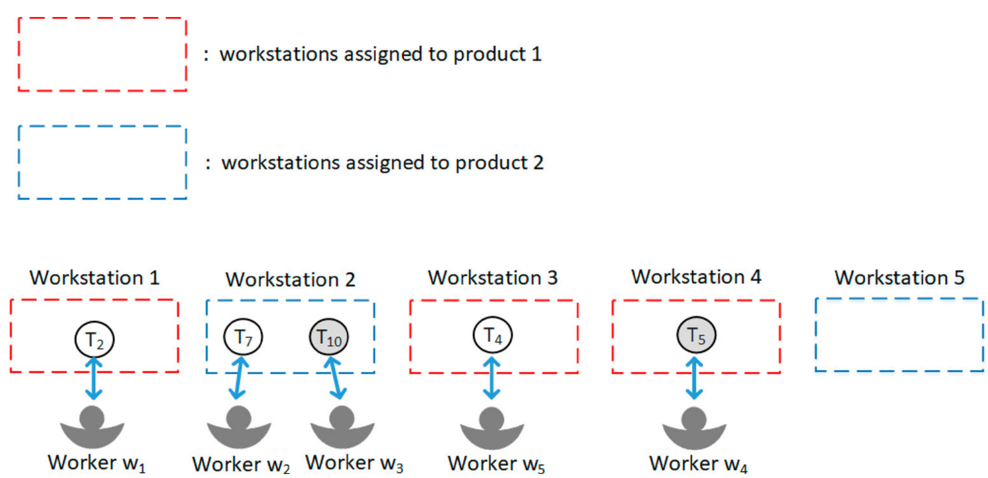

Figure 2. Disassembly line selection, workstation-to-product assignment, task-to-workstation assignments, worker-to-task assignment.

example, we can observe that for product 1 in Figure 1(a), the predecessor tasks of disassembly state S_2 are collected in set $P_2 = \{T_2, T_3\}$, and its successor task is T_4. Especially, we assume that component 2 in product 1 is hazardous, and shaded nodes denoting tasks T_3 and T_5 that separate component 2 are hazardous tasks. For product 2, component 3 is hazardous, and shaded nodes denoting tasks T_9 and T_{10} that separate component 3 are hazardous tasks.

Disassembly line selection, workstation-to-product assignment, task-to-workstation assignment, and worker-to-task assignment are reported in Figure 2. In Figure 2, workstations are illustrated by rectangles with dotted lines, and workstations in red are assigned to product 1 and the blue ones are assigned to product 2. For product 1, it can be obtained from Figure 2 that (i) workstations 1, 3 and 4 are assigned to it, (ii) disassembly line that includes tasks T_2, T_4 and T_5 is selected, (iii) tasks T_2, T_4 and T_5 are assigned to workstations 1, 3 and 4, respectively, and (iv) tasks T_2, T_4 and T_5 are handled by workers w_1, w_5 and w_4, respectively.

This work follows the basic assumptions:

(1) For each EOL product $l \in \mathcal{L}$, the precedence relationship between tasks and subassembly states must be respected. Only one disassembly process should be selected for each EOL product.

(2) A multi-function workstation can potentially handle the disassembly processes of some specific products. A preset binary parameter δ_{jl} is used to describe whether workstation $j \in \mathcal{J}$ is capable to handle the disassembly process of product $l \in \mathcal{L}$.
A fixed assignment cost c^P_{jl} is considered, if workstation $j \in \mathcal{J}$ is assigned to product $l \in \mathcal{L}$. During each time unit, there is an operating cost c^F for an opened workstation $j \in \mathcal{J}$, and handling the hazardous tasks leads to extra cost, denoted by c^H.
(3) The original processing time of each disassembly task $i \in \mathcal{T}$ is stochastic and denoted by ξ_i, and $\boldsymbol{\xi} = [\xi_1, \xi_2, \dots, \xi_{|\mathcal{T}|}]^\top$. The true probability distribution (denoted by $\mathbb{P}_{\boldsymbol{\xi}}$) of $\boldsymbol{\xi}$ is unavailable, due to data scarcity. The given distributional information includes the first two moments (i.e. $\mathbb{E}[\boldsymbol{\xi}] = \boldsymbol{\mu}$ and $\mathbb{E}[\boldsymbol{\xi}\boldsymbol{\xi}^\top] = \boldsymbol{\Gamma}$) and that the probability distribution is unimodal in a generalised sense:
We consider that $\mathbb{P}_{\boldsymbol{\xi}}$ possesses an α-unimodality, which is a general structural information (Li, Jiang, and Mathieu 2019). Such an α-unimodality ensures that the increasing rate of the density of the probability distribution can increase when α is larger than the size of random parameters, and the increasing rate is controlled by α. To ensure the above moment and structural information, the assumption $(\frac{\alpha+2}{\alpha})\boldsymbol{\Gamma} \succ (\frac{\alpha+1}{\alpha})^2 \boldsymbol{\mu}\boldsymbol{\mu}^\top$ should be followed.
(4) Any selected disassembly task for an EOL product should be handled by exactly one workstation and one worker. A worker can be assigned to at most one task during the given cycle time.Workers have different ages, skill levels and work experiences. Thus the actual processing time of a task varies with its assigned worker, and a workforce coefficient ζ_{ri} is used to calculate the actual processing time of task $i \in \mathcal{T}$ handled by worker $r \in \mathcal{R}$. That is, if task $i \in \mathcal{T}$ is handled by worker $r \in \mathcal{R}$, the actual processing time of task i is calculated as $\zeta_{ri}\xi_i$.
If worker $r \in \mathcal{R}$ is assigned to handle task $i \in \mathcal{T}$, a fixed (workforce) assignment cost c^W_{ri} is considered.
(5) The given cycle time is respected in a risk-averse fashion, i.e. the CVaR of each workstation's working time is controlled and should not be larger than a given cycle time. CVaR is a popular risk measure, to estimate the right tail of the uncertain parameter. Compared with chance constraints, CVaR constraints not only restrict the cycle time violation probability, but also estimate the cycle time excess expectation (Li, Jiang, and Mathieu 2019).

The problem is to select a disassembly process for each product, determine the workstation-to-product assignment, task-to-workstation assignment and worker-to-task assignment. The pursued objective is to minimise the total cost, including (i) the cost for assigning workstations to products and assigning workers to tasks, and (ii) the cost for operating workstations and handling hazardous tasks.

3.2. Formulation

In this part, a stochastic programming formulation **[P1]** with distributionally robust CVaR constraints is proposed. Before introducing the mathematical formulation for the problem, we first provide the definition of CVaR (Rockafellar and Uryasev 2000; Sarykalin, Serraino, and Uryasev 2008).

We use a binary variable π_{rij} denoting whether task $i \in \mathcal{T}$ is assigned to workstation $j \in \mathcal{J}$ and handled by worker $r \in \mathcal{R}$, workstation j's working time is denoted by $\sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}} \pi_{rij}\zeta_{ri}\xi_i$. The cycle time excess of workstation j is calculated as $\sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}} \pi_{rij}\zeta_{ri}\xi_i - C$, which is stochastic due to the random variable ξ_i. Given partially known distribution $\mathbb{P}_{\boldsymbol{\xi}}$ of $\boldsymbol{\xi}$, $F_{\mathbb{P}_{\boldsymbol{\xi}}}(z) = \text{Prob}\{\sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}} \pi_{rij}\zeta_{ri}\xi_i - C \le z\}$ denotes the cumulative distribution function of $(\sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}} \pi_{rij}\zeta_{ri}\xi_i - C)$, and the VaR of workstation j's cycle time excess with confidence level $\epsilon \in (0,1)$ is the value defined as:

$$\text{VaR}^{\epsilon}_{\mathbb{P}_{\boldsymbol{\xi}}}\left(\sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}} \pi_{rij}\zeta_{ri}\xi_i - C\right) = \min\left\{z | F_{\mathbb{P}_{\boldsymbol{\xi}}}(z) \ge \epsilon\right\},$$

which is the lower ϵ-percentile of random variable $(\sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}} \pi_{rij}\zeta_{ri}\xi_i - C)$. Then we have the CVaR of workstation j's cycle time excess with confidence level ϵ:

$$\begin{aligned}
&\text{CVaR}^{\epsilon}_{\mathbb{P}_{\boldsymbol{\xi}}}\left(\sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}} \pi_{rij}\zeta_{ri}\xi_i - C\right)\\
&\quad = \mathbb{E}\Bigg[\sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}} \pi_{rij}\zeta_{ri}\xi_i - C \Bigg| \sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}} \pi_{rij}\zeta_{ri}\xi_i - C\\
&\quad \ge \text{VaR}^{\epsilon}_{\mathbb{P}_{\boldsymbol{\xi}}}\left(\sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}} \pi_{rij}\zeta_{ri}\xi_i - C\right)\Bigg].
\end{aligned}$$

To provide guarantees on the cycle time excess, we apply the distributionally robust CVaR constraints:

$$\sup_{\mathbb{P}_{\boldsymbol{\xi}}} \text{CVaR}^{\epsilon}_{\mathbb{P}_{\boldsymbol{\xi}}}\left(\sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}} \pi_{rij}\zeta_{ri}\xi_i - C\right) \le 0, \quad \forall j \in \mathcal{J},$$

where $\sup_{\mathbb{P}_{\boldsymbol{\xi}}}$ denotes the worst-case scenario in terms of partially known probability distribution $\mathbb{P}_{\boldsymbol{\xi}}$, i.e. the distributional robustness. As the CVaR criterion obeys the subadditivity law (Rockafellar and Uryasev, 2000), i.e.$\text{CVaR}_{\epsilon}(X + a) = \text{CVaR}_{\epsilon}(X) + a$, where X is a random variable and $a \in \mathbb{R}$ is a fixed scalar, the above constraints can be equivalently rewritten as:

$$\sup_{\mathbb{P}_\xi} \mathrm{CVaR}^{\epsilon}_{\mathbb{P}_\xi}\left(\sum_{i\in\mathcal{T}}\sum_{r\in\mathcal{R}}\pi_{rij}\zeta_{ri}\xi_i\right) \le C, \quad \forall j \in \mathcal{J}. \quad (1)$$

In the following, we describe basic notations, define decision variables, and then introduce a stochastic programming formulation.

Parameters

- $\mathcal{L}$: Set of EOL products, indexed by l, and $\mathcal{L} = \{1, 2, \dots, |\mathcal{L}|\}$.
- $\mathcal{J}$: Set of workstations, indexed by j, and $\mathcal{J} = \{1, 2, \dots, |\mathcal{J}|\}$.
- $\mathcal{R}$: Set of workers, indexed by r, and $\mathcal{R} = \{1, 2, \dots, |\mathcal{R}|\}$.
- $\mathcal{H}$: Set of hazardous tasks.
- $\mathcal{T}_l$: Set of tasks indexed by i, to disassemble an EOL product $l \in \mathcal{L}$, and $\mathcal{T} = \cup_{l\in\mathcal{L}}\mathcal{T}_l$.
- $\mathcal{S}_l$: Set of subassembly states, indexed by k, for an EOL product $l \in \mathcal{L}$. The beginning and ending states for product l are denoted by S^l_0 and S^l_e.
- P_k: Set of preceding tasks of subassembly state $k \in \mathcal{S}_l$, $l \in \mathcal{L}$.
- Q_k: Set of successor tasks of subassembly state $k \in \mathcal{S}_l$, $l \in \mathcal{L}$.
- δ_{jl}: Binary parameter, equal to 1 if workstation j is capable to disassemble product $l \in \mathcal{L}$, and 0 otherwise.
- C: Given cycle time.
- c^P_{jl}: Cost for assigning workstation j to EOL product $l \in \mathcal{L}$, over one period of the cycle-time length.
- c^W_{ri}: Cost for assigning worker $r \in \mathcal{R}$ to handle disassembly task $i \in \mathcal{T}$, over one period of the cycle-time length.
- c^F: Unit-time cost for operating a workstation.
- c^H: Unit-time cost for a workstation handling hazardous tasks.
- ξ_i: Stochastic original processing time of disassembly task $i \in \mathcal{T}$.
- ζ_{ri}: Workforce coefficient, i.e. a preset scalar to be multiplied by the original processing time of task $i \in \mathcal{T}$ if it is handled by worker $r \in \mathcal{R}$.
- $\mathcal{M}$: A large enough number.

Decision variables

- η_{jl}: Binary variable, equal to 1 if workstation $j \in \mathcal{J}$ is assigned to EOL product $l \in \mathcal{L}$, and 0 otherwise.
- x_{ij}: Binary variable, equal to 1 if disassembly task $i \in \mathcal{T}$ is assigned to workstation $j \in \mathcal{J}$, and 0 otherwise.
- y_{jl}: Binary variable, equal to 1 if workstation $j \in \mathcal{J}$ is opened, to disassemble EOL product $l \in \mathcal{L}$, and 0 otherwise.
- λ_{jl}: Binary variable, equal to 1 if the last opened workstation for product $l \in \mathcal{L}$ is workstation $j \in \mathcal{J}$, and 0 otherwise.
- z_j: Binary variable, equal to 1 if a hazardous task is handled by workstation $j \in \mathcal{J}$, 0 otherwise.
- ν_{ri}: Binary variable, equal to 1 if worker $r \in \mathcal{R}$ is assigned to handle disassembly task $i \in \mathcal{T}$, and 0 otherwise.
- π_{rij}: Binary variable to linearise $x_{ij} \cdot \nu_{ri}$, equal to 1 if task $i \in \mathcal{T}$ is assigned to workstation $j \in \mathcal{J}$ (i.e. $x_{ij} = 1$) and handled by worker $r \in \mathcal{R}$ (i.e. $\nu_{ri} = 1$), and 0 otherwise.

A stochastic programming formulation [**P1**] is:

$$\min f = \left\{\sum_{l\in\mathcal{L}}\sum_{j\in\mathcal{J}} c^P_{jl}\eta_{jl} + \sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{T}} c^W_{ri}\nu_{ri} + C\left(\sum_{l\in\mathcal{L}}\sum_{j\in\mathcal{J}} c^F y_{jl} + c^H\sum_{j\in\mathcal{J}} z_j\right)\right\}$$

s.t. Constraints (1)

$$\sum_{l\in\mathcal{L}}\eta_{jl} = 1, \quad j \in \mathcal{J} \quad (2)$$

$$\eta_{jl} \le \delta_{jl}, \quad \forall j \in \mathcal{J}, l \in \mathcal{L} \quad (3)$$

$$y_{jl} \le \eta_{jl}, \quad \forall j \in \mathcal{J}, l \in \mathcal{L} \quad (4)$$

$$x_{ij} \le y_{jl}, \quad \forall i \in \mathcal{T}_l, j \in \mathcal{J}, l \in \mathcal{L} \quad (5)$$

$$\sum_{j\in\mathcal{J}} x_{ij} \le 1, \quad \forall i \in \mathcal{T} \quad (6)$$

$$\sum_{i\in Q_k}\sum_{j\in\mathcal{J}} x_{ij} = 1, \quad k = S^l_0, \forall l \in \mathcal{L} \quad (7)$$

$$\sum_{i\in Q_k}\sum_{j\in\mathcal{J}} x_{ij} = \sum_{i\in P_k}\sum_{j\in\mathcal{J}} x_{ij}, \quad k \in \mathcal{S}_l\backslash\{S^l_0, S^l_e\}, l \in \mathcal{L} \quad (8)$$

$$\sum_{i\in Q_k} x_{ij} \le \sum_{i\in P_k}\sum_{j'=1}^{j} x_{ij'}, \quad j \in \mathcal{J}, k \in \mathcal{S}_l\backslash\{S^l_0, S^l_e\}, l \in \mathcal{L} \quad (9)$$

$$\sum_{j\in\mathcal{J}}\lambda_{jl} = 1, \quad l \in \mathcal{L} \quad (10)$$

$$\lambda_{jl} \le y_{jl}, \quad j \in \mathcal{J}, l \in \mathcal{L} \quad (11)$$

$$\sum_{j\in\mathcal{J}} j x_{ij} \le \sum_{j\in\mathcal{J}} j\lambda_{jl}, \quad \forall i \in \mathcal{T}_l, l \in \mathcal{L} \quad (12)$$

$$\sum_{j''\in\mathcal{J}} y_{j''l} \geq \sum_{j'=1}^{j} \eta_{j'l} - \mathcal{M}(1-\lambda_{jl}), \quad \forall j \in \mathcal{J}, l \in \mathcal{L} \tag{13}$$

$$\sum_{j''\in\mathcal{J}} y_{j''l} \leq \sum_{j'=1}^{j} \eta_{j'l} + \mathcal{M}(1-\lambda_{jl}), \quad \forall j \in \mathcal{J}, l \in \mathcal{L} \tag{14}$$

$$z_j \geq x_{ij}, \quad \forall i \in \mathcal{H}, j \in \mathcal{J} \tag{15}$$

$$\sum_{i\in\mathcal{T}} v_{ri} \leq 1, \quad \forall r \in \mathcal{R} \tag{16}$$

$$\sum_{r\in\mathcal{R}} v_{ri} = \sum_{j\in\mathcal{J}} x_{ij}, \quad \forall i \in \mathcal{T} \tag{17}$$

$$\pi_{rij} \leq x_{ij}, \quad \forall r \in \mathcal{R}, i \in \mathcal{T}, j \in \mathcal{J} \tag{18}$$

$$\pi_{rij} \leq v_{ri}, \quad \forall r \in \mathcal{R}, i \in \mathcal{T}, j \in \mathcal{J} \tag{19}$$

$$\pi_{rij} \geq x_{ij} + v_{ri} - 1, \quad \forall r \in \mathcal{R}, i \in \mathcal{T}, j \in \mathcal{J} \tag{20}$$

$$x_{ij}, y_{jl}, z_j, \eta_{jl}, \lambda_{jl}, v_{rj}, \pi_{rij} \in \{0,1\}, \quad \forall r \in \mathcal{R}, i \in \mathcal{T}, j \in \mathcal{J}, l \in \mathcal{L} \tag{21}$$

The objective is to minimise the total cost in one period of the cycle-time length, including (i) the cost for assigning workstations to products, i.e.$\sum_{l\in\mathcal{L}}\sum_{j\in\mathcal{J}} c^P_{jl}\eta_{jl}$, (ii) the cost for assigning workers to disassembly tasks, i.e.$\sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{T}} c^W_{ri} v_{ri}$, (iii) the workstation operating cost, i.e.$C \cdot \sum_{l\in\mathcal{L}}\sum_{j\in\mathcal{J}} c^F y_{jl}$, and (iv) the cost for handling hazardous tasks, i.e.$C \cdot \sum_{j\in\mathcal{J}} c^H z_j$.

Constraint (2) ensures that each workstation is assigned to exactly one product. Constraint (3) ensures that workstation $j \in \mathcal{J}$ cannot be assigned to a product that is beyond its handling capability. Constraint (4) means that workstation $j \in \mathcal{J}$ can be opened for product $l \in \mathcal{L}$, only if workstation $j \in \mathcal{J}$ is assigned to product $l \in \mathcal{L}$. Constraint (5) determines the opened workstations. Constraint (6) ensures that a disassembly task $i \in \mathcal{T}$ can be assigned to at most one workstation. Constraint (7) implies that for a product $l \in \mathcal{L}$, there must be exactly one disassembly process is selected. Constraint (8) ensures the flow conservation, i.e. for each disassembly state of a product, the numbers of its preceding tasks and successor tasks are the same (one or none). Constraint (9) respects the precedence relationship of tasks, i.e. for a disassembly state of product $l \in \mathcal{L}$, its preceding tasks should be handled by workstations with indices no larger than those handling its successor tasks. Constraint (10) ensures that there is exactly one last opened workstation for each product $l \in \mathcal{L}$. Constraint (11) restricts that if workstation $j \in \mathcal{J}$ is not opened for product $l \in \mathcal{L}$, it cannot be product l's last workstation. Constraint (12) respects the given order of workstations, i.e. if workstation j is the last opened workstations for product $l \in \mathcal{L}$, disassembly tasks cannot be assigned to workstations with indices larger than j. Constraints (13)–(14) ensure that if workstation $j \in \mathcal{J}$ is the last opened workstation for product $l \in \mathcal{L}$, workstations with indices no larger than j must be opened. As the right-hand value of Constraint (13) should be nonnegative, we set the value of $\mathcal{M}$ to be $|J|$, which is the maximal value of $\sum_{j\in\mathcal{J}} \eta_{jl}$. Constraint (15) determines workstations handling hazardous tasks. Constraint (16) ensures that a worker $r \in \mathcal{R}$ can handle at most one disassembly task. Constraint (17) restricts that there should be a worker handling disassembly task $i \in \mathcal{T}$ if it is selected. Constraints (18)–(20) are to linearise $\pi_{rij} = x_{ij} \cdot v_{ri}$. Constraint (21) gives ranges of decision variables.

The deterministic single-product DLBP, to minimise the total number of opened workstations, is proved to be NP-hard (Mcgovern and Gupta 2007). Thus our problem is also NP-hard.

4. Solution approaches

In this section, we analyse the problem properties to enhance the formulation, leading to an improved model [**P2**]. To tackle the distributionally robust CVaR constraints, an ambiguity set characterised by the given distributional information is constructed. Based on the ambiguity set, an equivalent reformulation of model [**P2**] is proposed, and an exact cutting-plane method is proposed. To efficiently solve large-scale problem instances, an approximation method is further developed.

4.1. Property analyses

We first present a problem property by Proposition 4.1, for a special case where there are only two products (i.e. $|\mathcal{L}| = 2$) and all workstations are capable to handle both of the two products (i.e.$\delta_{j1} = \delta_{j2} = 1, \forall j \in \mathcal{J}$).

Proposition 4.1: *In an optimal solution, the workstation* $j^* = \arg\max_{j\in\mathcal{J}}\{|c^P_{j1} - c^P_{j2}|\}$ *is assigned to product* $l^* = \arg\min_{l=\{1,2\}}\{c^P_{j^*,l}\}$:

$$\eta_{j^*,l^*} = 1, \quad \textit{where } j^* = \arg\min_{j\in\mathcal{J}}\{|c^P_{j1} - c^P_{j2}|\},$$
$$l^* = \arg\min_{l=\{1,2\}}\{c^P_{j^*,l}\}$$

Proof: Suppose that in the optimal solution, the numbers of workstations assigned to the two products are n_1 and n_2 ($n_1 + n_2 = |\mathcal{J}|$), and the workstation sets assigned to the two products are denoted by $\mathcal{J}_1 = \{j_1, j_2, \ldots, j_{n_1}\}$

and $\mathcal{J}_2 = \{j_{n_1+1}, j_{n_1+2}, \ldots, j_{n_1+n_2}\}$. Without loss of generality, we assume that j^* ($j^* = \arg\max_{j\in\mathcal{J}}\{|c^P_{j1} - c^P_{j2}|\}$) should be assigned to the first product according to the above proposition (i.e.$c^P_{j^*,1} \le c^P_{j^*,2}$). Suppose to the contrary that in an optimal solution j^* is not assigned to the first product (i.e.$\eta_{j^*,2} = 1$, and $j^* \in \mathcal{J}_2$). We randomly select a workstation (denoted by j') from those assigned to the first product, and exchange j' and j^*, i.e. assign j' to the second product and j^* to the first product. If f_0 and f_1 are used to denote the objective values of the original solution and the solution after exchange, we can have

$$\begin{aligned} f_1 - f_0 &= c^P_{j',2} + c^P_{j^*,1} - c^P_{j',1} - c^P_{j^*,2} \\ &= c^P_{j^*,1} - c^P_{j^*,2} + c^P_{j',2} - c^P_{j',1}. \end{aligned}$$

As $|c^P_{j^*1} - c^P_{j^*2}|$ is maximal over all workstations and $c^P_{j^*,l} \le c^P_{j^*,2}$, we have $f_1 - f_0 \le 0$. That is, the objective value decreases via the exchange. It contradicts to the fact that an optimal solution assigns workstation j^* to the second product. Thus in the optimal solution, workstation $j^* = \arg\max_{j\in\mathcal{J}}\{|c^P_{j1} - c^P_{j2}|\}$ is assigned to product $l^* = \arg\min_{l=\{1,2\}}\{c^P_{j^*,l}\}$. ■

In the following, we provide several valid inequalities for the general case, which can tighten a part of solution space.

Proposition 4.2: *For any feasible solution, the following relation holds:*

$$y_{1l} = \eta_{1l}, \quad \forall l \in \mathcal{L} \tag{22}$$

Proof: Suppose that for each product $l \in \mathcal{L}$, workstations assigned to l are collected in set $\mathcal{S}_l$. From Constraints (2) and (3), it can be observed that each workstation must be assigned to a product. Suppose workstation $j = 1$ is assigned to product $l' \in \mathcal{L}$. There must be exactly one disassembly process is selected for a product j' (i.e. Constraint (7)), thus at least one workstation is opened for product j', to handle the selected disassembly tasks. Suppose from the contrary that workstation 1 is assigned to but not opened for product j'. According to Constraint (5), if $y_{1l'} = 0$, $x_{i1} = 0$ for all $i \in \mathcal{T}$, and $x_{ij} = 0$ for all $i \in \mathcal{T}$ and $j \in \mathcal{J}$ (i.e. Constraint (9)). It contradicts to the fact that at least one workstation is opened for each product. Thus, in any feasible solution, $y_{1l} = \eta_{1l}$, $\forall l \in \mathcal{L}$. ■

Proposition 4.3: *Constraints* (12)–(14) *can be replaced via:*

$$y_{j',l} \ge \eta_{j',l} - 1 + \lambda_{jl}, \quad \forall j \in \mathcal{J}, j' = \{1, \ldots, j\}, l \in \mathcal{L} \tag{23}$$

$$y_{j',l} \le 1 - \lambda_{jl}, \quad \forall j \in \mathcal{J}\backslash\{|\mathcal{J}|\}, j' \in \{j+1, \ldots, |\mathcal{J}|\}, l \in \mathcal{L} \tag{24}$$

Proof: Constraints (12)–(14), which use *big-M* values, guarantee that if workstation $j \in \mathcal{J}$ is the last opened workstation for product $l \in \mathcal{L}$ (i.e. $\lambda_{jl} = 1$), $\sum_{j\in\mathcal{J}} y_{jl} = \sum_{j'=1}^{j} \eta_{j'l}$. That is, (i) workstations with indices no larger than j, which are assigned to product l, must be all opened for product l, and (ii) workstations with indices larger than j cannot be opened. Inequalities (24)–(25) are valid and can replace Constraints (12)–(14), to tighten a part of solution space. ■

Proposition 4.4: *For any feasible solution, the following inequalities should be satisfied:*

$$\sum_{j\in\mathcal{J}} \eta_{jl} \le |\mathcal{J}| - 1, \quad \forall l \in \mathcal{L} \tag{25}$$

Proof: For each product $l \in \mathcal{L}$, one disassembly process should be finalised, and thus at least one workstation is assigned to product l to handle the selected disassembly tasks. Suppose to the contrary that there exists a product $l' \in \mathcal{L}$, such that $\sum_{j\in\mathcal{J}} \eta_{jl} = |\mathcal{J}|$. There is no workstation assigned to other products $l \in \mathcal{L}\backslash\{l'\}$, which contradicts to the fact that at least one workstation is assigned to each product. ■

Observation 4.1: For any feasible solution, to respect the given order of workstations, the following inequalities should be satisfied:

$$y_{j_1,l} \ge y_{j_2,l} - (2 - \eta_{j_1,l} - \eta_{j_2,l}), \quad j_1, j_2 \in \mathcal{J}, j_1 \le j_2 - 1 \tag{26}$$

The above valid inequalities are appended to the original problem formulation [**P1**], leading to a new model [**P2**]:

$$\begin{aligned} [\mathbf{P2}]: \min f = \Bigg\{ &\sum_{l\in\mathcal{L}}\sum_{j\in\mathcal{J}} c^P_{jl}\eta_{jl} + \sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{T}} c^W_{ri} v_{ri} \\ &+ C \cdot \left(\sum_{l\in\mathcal{L}}\sum_{j\in\mathcal{J}} c^F y_{jl} + c^H \sum_{j\in\mathcal{J}} z_j \right) \Bigg\} \\ \text{s.t.} \quad &(1)\text{–}(11), (15)\text{–}(26) \end{aligned}$$

4.2. Ambiguity set

Due to the CVaR constraints under partially known probability distribution $\mathbb{P}_\xi$, it is still difficult to solve model [**P2**]. To address the challenge, a natural and popular way is to employ an ambiguity set characterised via the given information, including a family of plausible probability distributions (Delage and Ye 2010; Jiang and Guan 2016; Li, Jiang, and Mathieu 2019).

Given the first two moments (i.e. $\mathbb{E}[\boldsymbol{\xi}] = \boldsymbol{\mu}$ and $\mathbb{E}[\boldsymbol{\xi}\boldsymbol{\xi}^\top] = \boldsymbol{\Gamma}$) of uncertain vector $\boldsymbol{\xi}$ and the α-unimodality of the probability distribution $\mathbb{P}_{\xi}$, following Li, Jiang, and Mathieu (2019), an ambiguity set $\mathcal{P}(\boldsymbol{\mu}, \boldsymbol{\Gamma}, \alpha)$ is defined:

$$\mathcal{P}(\boldsymbol{\mu}, \boldsymbol{\Gamma}, \alpha) = \left\{ \mathbb{P}_{\xi} : \begin{array}{l} \mathbb{E}_{\mathbb{P}_{\xi}}[\boldsymbol{\xi}] = \boldsymbol{\mu}, \\ \mathbb{E}_{\mathbb{P}_{\xi}}[\boldsymbol{\xi}\boldsymbol{\xi}^{\mathsf{T}}] = \boldsymbol{\Gamma}, \\ \mathbb{P}_{\xi} \text{ is } \alpha\text{-unimodal about } 0, \end{array} \right\},$$

where $\mathbb{E}_{\mathbb{P}_{\xi}}[\cdot]$ denotes the expected value of the expression in brackets, under distribution $\mathbb{P}_{\xi}$. We use $\mathcal{P} = \mathcal{P}(\boldsymbol{\mu}, \boldsymbol{\Gamma}, \alpha)$ for short in the following. In order to ensure that $\mathcal{P} \neq 0$, the assumption $(\frac{\alpha+2}{\alpha})\boldsymbol{\Gamma} \succ (\frac{\alpha+1}{\alpha})^2 \boldsymbol{\mu}\boldsymbol{\mu}^\top$ should be followed.

Then constraints (1) can be written as:

$$\sup_{\mathbb{P}_{\xi} \in \mathcal{P}} \text{CVaR}_{\epsilon} \left(\sum_{i \in \mathcal{T}} \sum_{r \in \mathcal{R}} \pi_{rij} \cdot \zeta_{ri} \cdot \xi_i \right) \leq C, \quad \forall j \in \mathcal{J} \tag{27}$$

Based on ambiguity set $\mathcal{P}$, formulation **[P2]** under partial known distribution $\mathbb{P}_{\xi}$ can be written as model **[P3]**:

$$\begin{aligned} &[\mathbf{P3}] : \min f \\ &\qquad \text{s.t. } (2)\text{–}(11), (15)\text{–}(27) \end{aligned}$$

4.3. Equivalent reformulation

In this subsection, based on Theorem 3 in Li, Jiang, and Mathieu (2019), an equivalent reformulation of model **[P3]** is established.

We first introduce a new decision variable vector with length $|\mathcal{T}|$, denoted by $\boldsymbol{\chi}_j = [\sum_{r \in \mathcal{R}} \zeta_{r1}\pi_{r1j}, \sum_{r \in \mathcal{R}} \zeta_{r2}\pi_{r2j}, \ldots, \sum_{r \in \mathcal{R}} \zeta_{r|\mathcal{T}|}\pi_{r|\mathcal{T}|j}]^\top, \forall j \in \mathcal{J}$. Then the distributionally robust CVaR constraints (27) in **[P3]** can be rewritten as:

$$\sup_{\mathbb{P}_{\xi} \in \mathcal{P}} \text{CVaR}_{\epsilon} \left(\boldsymbol{\chi}_j^\top \boldsymbol{\xi} \right) \leq C, \quad \forall j \in \mathcal{J}. \tag{28}$$

Constraints (28) are equivalent to a set of second-order cone (SOC) constraints:

$$\begin{aligned} &\left\| \begin{bmatrix} (1 - \gamma_j^{-\alpha})\beta_j - (1 - \gamma_j^{(-1-\alpha)})\boldsymbol{\mu}^\top \boldsymbol{\chi}_j \\ (\frac{\alpha}{\alpha+1})(1 - \gamma_j^{(-\alpha-1)}) A \boldsymbol{\chi}_j \end{bmatrix} \right\| \\ &\leq 2\epsilon \cdot C - (1 - \gamma_j^{(-\alpha-1)})\boldsymbol{\mu}^\top \boldsymbol{\chi}_j \\ &\quad + (1 - \gamma_j^{-\alpha} - 2\epsilon)\beta_j, \quad \forall j \in \mathcal{J} \end{aligned} \tag{29}$$

$$\begin{aligned} &\left\| \begin{bmatrix} (1 - \gamma_j^{-\alpha})\beta_j - (1 - \gamma_j^{(-1-\alpha)})\boldsymbol{\mu}^\top \boldsymbol{\chi}_j \\ (\frac{\alpha}{\alpha+1})(1 - \gamma_j^{(-\alpha-1)}) A \boldsymbol{\chi}_j \end{bmatrix} \right\| \\ &\leq 2\epsilon \cdot C - (1 + \gamma_j^{(-\alpha-1)})\boldsymbol{\mu}^\top \boldsymbol{\chi}_j \\ &\quad + (1 + \gamma_j^{-\alpha} - 2\epsilon)\beta_j, \quad \forall j \in \mathcal{J} \end{aligned} \tag{30}$$

for all $\gamma_j \geq 1$, where (i) $\epsilon \in (0, 1)$ is the preset risk level, (ii) $\beta_j \in \mathbb{R}$ are new introduced decision variables, (iii) $\| \cdot \|$ denotes the 2-norm of a vector, and (iv) matrix $A = [(\frac{\alpha+2}{\alpha})\boldsymbol{\Gamma} - (\frac{\alpha+1}{\alpha})^2 \boldsymbol{\mu}\boldsymbol{\mu}^\top]^{1/2}$ (Li, Jiang, and Mathieu 2019), in which $[\cdot]^{1/2}$ denotes the square-rooting matrix of the matrix included in the brackets.

Accordingly, formulation **[P3]** can be equivalently transformed into:

$$\begin{aligned} &[\mathbf{P4}] : \min f \\ &\qquad \text{s.t. } (2)\text{–}(11), (15)\text{–}(26), (29), (30) \\ &\qquad\qquad \beta_j \in \mathbb{R}, \quad \forall j \in \mathcal{J} \end{aligned} \tag{1}$$

However, it is still computationally intractable to solve formulation **[P4]**, as the number of SOC constraints is infinite, i.e. for all $\gamma_j \geq 1$ and $j \in \mathcal{J}$. Thus in the following, an exact cutting-plane algorithm is proposed.

4.4. Cutting-plane algorithm

In this subsection, an exact cutting-plane algorithm is developed, as shown in Algorithm 1. Instead of solving formulation **[P4]** with infinite number of SOC constraints (or infinite number of γ_j), the cutting-plane algorithm focuses on solving a relaxed master problem and finding a series of $\hat{\gamma}_j$ for all $j \in \mathcal{J}$ (i.e. $\hat{\gamma}_j$ such that SOC constraints (29) and (30) are violated by the obtained relaxation solution). When no such $\hat{\gamma}_j$ can be found, then the problem is solved to optimality; otherwise, SOC constraints related to $\hat{\gamma}_j$, serving as cuts, will be added to the master problem.

For the considered problem, formation **[P4]** is decomposed into a relaxed master problem **[MP]**, in which a set of cuts are to be iteratively added, to perform the restrictions of SOC Constraints (29) and (30):

$$\begin{aligned} &[\mathbf{MP}] : \min f \\ &\qquad \text{s.t. } (2)\text{–}(11), (15)\text{–}(26) \end{aligned}$$

with a set of SOC cuts:

$$\begin{aligned} &\left\| \begin{bmatrix} (1 - \hat{\gamma}_j^{-\alpha})\beta_j - (1 - \hat{\gamma}_j^{(-1-\alpha)})\boldsymbol{\mu}^\top \boldsymbol{\chi}_j \\ (\frac{\alpha}{\alpha+1})(1 - \hat{\gamma}_j^{(-\alpha-1)}) A \boldsymbol{\chi}_j \end{bmatrix} \right\| \\ &\leq 2\epsilon \cdot C - (1 - \hat{\gamma}_j^{(-\alpha-1)})\boldsymbol{\mu}^\top \boldsymbol{\chi}_j \\ &\quad + (1 - \hat{\gamma}_j^{-\alpha} - 2\epsilon)\beta_j, \quad \forall j \in \mathcal{J}, \end{aligned} \tag{32}$$

$$\left\| \begin{bmatrix} (1 - \hat{\gamma}_j^{-\alpha})\beta_j - (1 - \hat{\gamma}_j^{(-1-\alpha)})\boldsymbol{\mu}^\top \boldsymbol{\chi}_j \\ (\frac{\alpha}{\alpha+1})(1 - \hat{\gamma}_j^{(-\alpha-1)}) A \boldsymbol{\chi}_j \end{bmatrix} \right\|$$

$$\leq 2\epsilon \cdot C - (1+\hat{\gamma}_j^{(-\alpha-1)})\boldsymbol{\mu}^\top \boldsymbol{\chi}_j + (1+\hat{\gamma}_j^{-\alpha} - 2\epsilon)\beta_j, \quad \forall j \in \mathcal{J}, \tag{33}$$

where β_j for all $j \in \mathcal{J}$ are decision variables, and $\hat{\gamma}_j$ is obtained by a search method introduced by Li, Jiang, and Mathieu (2019), as shown in Algorithm 2. In Algorithm 2, there are 3 cases, where $\hat{\beta}_j = 0$ in case 1, $\hat{\beta}_j \neq 0$ and $\hat{\Sigma}_0 = \hat{\mu}_0^2$ in case 2, and $\hat{\beta}_j \neq 0$ and $\hat{\Sigma}_0 > \hat{\mu}_0^2$ in case 3. The reason why $\hat{\Sigma}_0 < \hat{\mu}_0^2$ doesn't exist is explained as follows.

To ensure Assumption (3) in Section 3.1, $(\frac{\alpha+2}{\alpha})\boldsymbol{\Gamma} \succ (\frac{\alpha+1}{\alpha})^2 \boldsymbol{\mu}\boldsymbol{\mu}^\top$ should be satisfied. Given $\hat{\mu}_0 = (\frac{\alpha+1}{\alpha})\boldsymbol{\mu}^\top \hat{\boldsymbol{\chi}}_j$ and $\hat{\Sigma}_0 = (\frac{\alpha+2}{\alpha})\hat{\boldsymbol{\chi}}_j^\top \boldsymbol{\Gamma} \hat{\boldsymbol{\chi}}_j$, we have

$$\begin{aligned}
\hat{\mu}_0^2 &= \hat{\mu}_0 \hat{\mu}_0 \\
&= \left(\left(\frac{\alpha+1}{\alpha}\right)\boldsymbol{\mu}^\top \hat{\boldsymbol{\chi}}_j\right)\left(\left(\frac{\alpha+1}{\alpha}\right)\boldsymbol{\mu}^\top \hat{\boldsymbol{\chi}}_j\right) \\
&= \left(\frac{\alpha+1}{\alpha}\right)^2 \boldsymbol{\mu}^\top \hat{\boldsymbol{\chi}}_j \boldsymbol{\mu}^\top \hat{\boldsymbol{\chi}}_j \\
&= \left(\frac{\alpha+1}{\alpha}\right)^2 \hat{\boldsymbol{\chi}}_j^\top \boldsymbol{\mu}\boldsymbol{\mu}^\top \hat{\boldsymbol{\chi}}_j
\end{aligned}$$

and

$$\begin{aligned}
\hat{\Sigma}_0 - \hat{\mu}_0^2 &= \left(\frac{\alpha+2}{\alpha}\right)\hat{\boldsymbol{\chi}}_j^\top \boldsymbol{\Gamma}\hat{\boldsymbol{\chi}}_j - \left(\frac{\alpha+1}{\alpha}\right)^2 \hat{\boldsymbol{\chi}}_j^\top \boldsymbol{\mu}\boldsymbol{\mu}^\top \hat{\boldsymbol{\chi}}_j \\
&= \hat{\boldsymbol{\chi}}_j^\top \left(\left(\frac{\alpha+2}{\alpha}\right)\boldsymbol{\Gamma} - \left(\frac{\alpha+1}{\alpha}\right)^2 \boldsymbol{\mu}\boldsymbol{\mu}^\top\right)\hat{\boldsymbol{\chi}}_j
\end{aligned}$$

As matrix $(\frac{\alpha+2}{\alpha})\boldsymbol{\Gamma} - (\frac{\alpha+1}{\alpha})^2\boldsymbol{\mu}\boldsymbol{\mu}^\top \succ 0$ is positive definite, and $\boldsymbol{\chi}_j \geq 0$, we have $\hat{\boldsymbol{\chi}}_j^\top((\frac{\alpha+2}{\alpha})\boldsymbol{\Gamma} - (\frac{\alpha+1}{\alpha})^2\boldsymbol{\mu}\boldsymbol{\mu}^\top)\hat{\boldsymbol{\chi}}_j \geq 0$. That is, $\hat{\Sigma}_0 - \hat{\mu}_0^2 \geq 0$.

We observe from the numerical experiments that the exact cutting-plane algorithm is quite time-consuming. Therefore, to efficiently solve the practical large-scale problems, we develop an approximated formulation for model [**P3**] in the following.

4.5. Approximation method

Following Proposition 9 in Li, Jiang, and Mathieu (2019) (please see Appendix), distributionally robust CVaR constraints (27) in [**P3**] can be conservatively approximated by a limited set of SOC constraints:

$$\left\| \begin{bmatrix} \beta_j - (\frac{\epsilon+1}{\epsilon})\boldsymbol{\mu}^\top \boldsymbol{\chi}_j \\ A\boldsymbol{\chi}_j \end{bmatrix} \right\| \leq \left[\frac{2\epsilon(\alpha+1)}{\alpha}\right] C - \left[\frac{2\epsilon(\alpha+1)}{\alpha} - 1\right]\beta_j - \left(\frac{\alpha+1}{\alpha}\right)\boldsymbol{\mu}^\top \boldsymbol{\chi}_j, \quad \forall j \in \mathcal{J} \tag{34}$$

Algorithm 1: The cutting-plane algorithm

Input: Parameters for the problem: $\mathcal{T}_l$, $\mathcal{S}_l$, P_k, Q_k, δ_{jl}, C, c^F, c^H, c^P_{jl}, c^W_{ri}, ζ_{ri}, $\boldsymbol{\mu}$, $\boldsymbol{\Gamma}$, $\forall i \in \mathcal{T}, j \in \mathcal{J}, r \in \mathcal{R}, l \in \mathcal{L}$;

while *true* **do**
- Solve [**MP**] by calling CPLEX;
- Given $\hat{\beta}_j$ and $\hat{\boldsymbol{\chi}}_j$ for all $j \in \mathcal{J}$ obtained before, find a series of $\hat{\gamma}_j$ for all $j \in \mathcal{J}$ according to the method in Algorithm 2;
- **if** *$\hat{\gamma}_j$ cannot be found* **then**
  - Break;
- **else**
  - Add Constraints (32)- -(33) with the series of new $\hat{\gamma}_j$ for all $j \in \mathcal{J}$ to [**MP**];
- **end**

end

end

Output: An optimal solution (i.e., (i) workstation-to-product assignment, (ii) disassembly line selection for each product, (iii) task-to-workstation assignment and (iv) worker-to-task assignment) with its corresponding objective f^*.

$$\left\| \begin{bmatrix} \beta_j - (\frac{\epsilon+1}{\epsilon})\boldsymbol{\mu}^\top \boldsymbol{\chi}_j \\ Aa(\boldsymbol{x}) \end{bmatrix} \right\| \leq \left[\frac{2\epsilon(\alpha+1)}{\alpha}\right] C - \left[\frac{(2\epsilon-1)(\alpha+1)-1}{\alpha}\right]\beta_j - \left(\frac{\alpha+1}{\alpha}\right)\boldsymbol{\mu}^\top \boldsymbol{\chi}_j, \quad \forall j \in \mathcal{J} \tag{35}$$

where (i) $\beta_j \in \mathbb{R}$ for all $j \in \mathcal{J}$ are decision variables, and (ii) matrix $A = [(\frac{\alpha+2}{\alpha})\boldsymbol{\Gamma} - (\frac{\alpha+1}{\alpha})^2\boldsymbol{\mu}\boldsymbol{\mu}^\top]^{1/2}$, in which $[\cdot]^{1/2}$ denotes the square-rooting matrix of the matrix included in the brackets.

Accordingly, the formulation [**P3**] can be conservatively approximated by:

$$\begin{aligned}
[\mathbf{P5}] : \min\ & f \\
\text{s.t. } & (2)\text{–}(11),(15)\text{–}(26),(34),(35) \\
& \beta_j \in \mathbb{R}, \quad \forall j \in \mathcal{J}
\end{aligned}$$

Note that formulation [**P5**] can be solved by calling commercial solvers, such as CPLEX.

5. Computational experiments

In this part, numerical experiments are conducted, to evaluate the proposed cutting-plane algorithm and the

Algorithm 2: Algorithm for finding parameter γ_j

Input: Parameters for the problem, and $\hat{\beta}_j$, $\hat{\boldsymbol{\chi}}_j$ for all $j \in \mathcal{J}$ obtained by solving [**MP**];

for $j = 1 : |\mathcal{J}|$ **do**

  Define $\hat{\mu}_0 = (\frac{\alpha+1}{\alpha})\boldsymbol{\mu}^\top \hat{\boldsymbol{\chi}}_j$, and $\hat{\Sigma}_0 = (\frac{\alpha+2}{\alpha})\hat{\boldsymbol{\chi}}_j^\top \boldsymbol{\Gamma} \hat{\boldsymbol{\chi}}_j$;

  if $\hat{\beta}_j = 0$ **then**

    Constraints (29)- -(30) are violated at $\hat{\gamma}_j = \infty, \forall j \in \mathcal{J}$;

  end

  if $\hat{\beta}_j \neq 0$ *and* $\hat{\Sigma}_0 = \hat{\mu}_0^2$ **then**

    Constraints (29)- -(30) are violated at $\hat{\gamma}_j = \max\{\hat{\mu}_0/\hat{\beta}_j, 1\}, \forall j \in \mathcal{J}$;

  end

  if $\hat{\beta}_j \neq 0$ *and* $\hat{\Sigma}_0 > \hat{\mu}_0^2$ **then**

    Constraints (29)-(30) are violated at the unique root of equation:

    $$2\left[\left(\frac{\alpha+1}{\alpha}\right)\left(\frac{1-\gamma_j^{-\alpha}}{1-\gamma_j^{-\alpha-1}} - \mu_\beta\right)\right] = (\gamma_j - \mu_{\beta_j}) - \frac{\Gamma_{\beta_j}}{\gamma_j - \mu_{\beta_j}}$$

    with γ_j in interval

    $$\left[1 + \sqrt{(1-\mu_{\beta_j})^2 + \Gamma_\beta}, 1 + 1/\alpha + \sqrt{(1-\mu_\beta + 1/\alpha)^2 + \Gamma_{\beta_j}}\right],$$

    $\forall j \in \mathcal{J}$, where $\mu_{\beta_j} = \hat{\mu}_0/\hat{\beta}_j$ and $\Gamma_{\beta_j} = (\hat{\Sigma}_0 - \hat{\mu}_0^2)/\hat{\beta}_j^2$.

  end

end

Output: $\hat{\gamma}_j$ for all $j \in \mathcal{J}$.

approximation method. There is no existing solution method in literature for stochastic DLBP with distributionally robust CVaR constraints. The SAA method is employed as a benchmark approach (see Appendix), due to its wide applications to solve general stochastic programs. All methods are coded in MATLAB_2016a and formulations are solved by calling CPLEX solver of version 12.10.0. Numerical experiments are conducted on a personal computer with Core i5 and 3.00Hz processor and 8.00GB RAM under Windows 10 Operating System. For each approach, each instance is tested 5 times, to obtain the average results. The computational time for each approach is limited to 3600 seconds. During the numerical experiments, due to the conservative approximation of the approximation method, a larger ϵ may be required to obtain feasible solutions, thus if the problem is infeasible, $\epsilon = \epsilon + \Delta$, where Δ is set to be 0.05.

Data of the tested instances are available on https://www.researchgate.net/publication/344363955_Input_data_of_the_tested_instances_in_our_work. As in line with Li, Jiang, and Mathieu (2019), the parameter α for the cutting-plane algorithm and the approximation method is set to be 1. Preliminary analyse has been conducted to adjust the maximal iteration number of the cutting-plane algorithm, and the maximal iteration number is set to be 20.

5.1. Out-of-sample test

In addition to the objective value and the computational time, we also employ a performance criterion, i.e. the out-of-sample safety level (Xie and Ahmed 2018). The out-of-sample safety level is obtained by the out-of-sample test, i.e.testing the obtained solutions in a large set of scenarios, where each scenario corresponds to a vector of randomly generated original disassembly task processing times. We set the number of tested scenarios to be 10,000, as in line with Zhang, Shen, and Erdogan (2017). Such a set of scenarios is used to represent the realizations of original disassembly task processing times. Thus the known distributional information on the uncertain original disassembly task processing times, i.e. the given mean vector, covariance matrix and the unimodality of their probability distribution should be satisfied during the generation process of the 10,000 scenarios.

A multi-variable lognormal distribution that is unimodal, with the given mean vector and covariance matrix as input parameters, satisfies the given distributional information thus it is employed to generate the original disassembly task processing times (Zhang, Shen, and Erdogan 2017; Zhang, Jiang, and Shen 2018). The out-of-sample safety level is to estimate the probability of each workstation's working time no larger than the given cycle time.

For each approach, based on its obtained disassembly line selection, task-to-workstation assignment and worker-to-task assignment:

(1) Under each scenario out of the tested 10,000 ones, the actual working time of each workstation can be easily calculated, and the cycle time violation can be estimated.
(2) Then we evaluate the out-of-sample safety level by $\frac{n}{10{,}000\times|\mathcal{I}|} \times 100\%$, in which n denotes the number of workstations that respect the given cycle time under all tested 10,000 scenarios.

5.2. An illustrative example

An illustrative example, based on the decomposition colour graph for two products in Figure 1, is studied in this subsection, to compare the solutions obtained

Table 2. Input data of the illustrative example.

Parameter	Value
Number of tasks ($\lvert\mathcal{T}\rvert$)	10
Maximal number of workstations ($\lvert\mathcal{J}\rvert$)	6
Cycle time (C)	15
Hazardous tasks	[3,5,9,10]
Unit-time cost for each workstation (c^H)	3
Unit-time cost for handling hazardous operation (c^H)	2
Preset risk level (ϵ)	20%

by different approaches. For the instance, there are 2 products, 10 tasks, 6 workstations and 10 workers. Input parameters of the illustrative instance are reported in Table 2. The given first two moments of the original task processing times, and the covariance matrix which can be calculated accordingly, are shown in Table 3, and more specific data is available on the website mentioned above. For the SAA method, we randomly generate 1000 scenarios of random task processing times following the same procedures as above.

Table 4 reports the computational results obtained by the tested solution methods, where we use 'safety level' for short, denoting the out-of-sample safety level. In each row $i \in \{1, 2, \dots, 10\}$, if there is a symbol '$w_r$', it means that (1) task i is assigned to workstation $j \in \{1, 2, 3, 4, 5\}$ (i.e. $x_{ij} = 1$), and (2) task i is handled by worker r (i.e. $v_{ri} = 1$). We can observe from Table 4 that the cutting-plane algorithm and the SAA method obtain the same task-to-workstation assignment, i.e. tasks 2, 4 and 5 are assigned to the first workstation and tasks 7 and 10 are assigned to the second workstation. However, via the approximation method, workstations 1, 2, 3 and 5 are opened, and they handle task 7, tasks 2, 4, task 5, task 10, respectively. In addition, the worker-to-task assignments obtained by the tested methods are different: tasks

Table 3. First two moments and the covariance matrix of the illustrative example.

		Second moment matrix Γ									
Task (i)	μ	1	2	3	4	5	6	7	8	9	10
1	6	51.08	32.52	64.53	56.54	39.95	79.53	23.07	25.74	55.94	70.93
2	4	32.52	26.27	42.71	37.65	28.04	52.03	13.67	17.51	38.75	47.55
3	8	64.53	42.71	87.71	73.50	54.77	105.70	31.39	33.70	74.48	96.79
4	7	56.54	37.65	73.50	68.11	44.20	92.67	27.50	27.69	64.58	83.58
5	5	39.95	28.04	54.77	44.20	38.55	66.32	20.12	20.76	49.12	59.87
6	10	79.53	52.03	105.70	92.67	66.32	135.71	41.11	39.54	92.63	119.49
7	3	23.07	13.67	31.39	27.50	20.12	41.11	14.46	10.23	27.47	35.94
8	3	25.74	17.51	33.70	27.69	20.76	39.54	10.23	16.03	28.81	35.60
9	7	55.94	38.75	74.48	64.58	49.12	92.63	27.47	28.81	69.07	82.64
10	9	70.93	47.55	96.79	83.58	59.87	119.49	35.94	35.60	82.64	110.75
		Covariance matrix Σ									
Task (i)		1	2	3	4	5	6	7	8	9	10
1		15.08	8.52	16.53	14.54	9.95	19.53	5.07	7.74	13.94	16.93
2		8.52	10.27	10.71	9.65	8.04	12.03	1.67	5.51	10.75	11.55
3		16.53	10.71	23.71	17.50	14.77	25.70	7.39	9.70	18.48	24.79
4		14.54	9.65	17.50	19.11	9.20	22.67	6.50	6.69	15.58	20.58
5		9.95	8.04	14.77	9.20	13.55	16.32	5.12	5.76	14.12	14.87
6		19.53	12.03	25.70	22.67	16.32	35.71	11.11	9.54	22.63	29.49
7		5.07	1.67	7.39	6.50	5.12	11.11	5.46	1.23	6.47	8.94
8		7.74	5.51	9.70	6.69	5.76	9.54	1.23	7.03	7.81	8.60
9		13.94	10.75	18.48	15.58	14.12	22.63	6.47	7.81	20.07	19.64
10		16.93	11.55	24.79	20.58	14.87	29.49	8.94	8.60	19.64	29.75

Table 4. Task-to-workstation, worker-to-task and workstation-to-product assignments for the illustrative example.

	Cutting-plane algorithm						Approximation model						The SAA					
Task (i)\ Workstation (j)	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1																		
2	w_7							w_{10}					w_{10}					
3																		
4	w_4							w_4					w_9					
5	w_5								w_5				w_5					
6																		
7					w_6		w_6										w_6	
8																		
9																		
10					w_9						w_9						w_2	
Objective	228						349						224					
Safety level	98.53%						99.54%						95.93%					
Time (s)	256						20						435					

2, 4 and 5 are processed by workers 7, 4, 5 via the cutting-plane algorithm, by workers 10, 4, 5 via the approximation method, and by workers 10, 9, 5 via the SAA method, and tasks 7 and 10 are processed by workers 6, 9 via the cutting-plane algorithm and the approximation method by workers 6, 2 via the SAA method.

Moreover, it can be also observed from Table 4 that (i) the safety level obtained by the approximation method is 99.54%, about 1.03% and 3.76% larger than those of the cutting-plane algorithm and the SAA method; (ii) the computational time of the approximation method is 20 seconds, about 94.5% and 95.4% smaller than those of the cutting-plane algorithm and the SAA method; (iii) but the objective obtained by the approximation method is 349, about 53% and 55.8% larger than the cutting-plane algorithm and the SAA method; (iv) though the SAA method obtains the smallest cost, its obtained safety level is the smallest among the tested methods; (v) the cutting-plane algorithm can obtain a smaller total cost compared with the approximation method, and a higher safety level compared with the SAA method.

In sum, for the illustrative example, (i) the approximation method can obtain the highest safety level within a smallest computational time, with sacrifice of the total cost; (ii) though the total cost obtained by the SAA is slightly smaller, the safety level is lower as well; (iii) the cutting-plane algorithm can obtain a smaller system cost, with a little sacrifice of the safety level.

5.3. Numerical experiments on randomly generated instances

In this subsection, numerical experiments on instances of different scales are conducted and reported. The tested instances are generated based on several benchmark instances, and the input data of the benchmark instances are reported in Table 5. Note that multiple products and workforce assignment are not considered in the benchmark instances. In the numerical experiments, we consider two products, and the candidate disassembly processes (i.e. the precedence relationship between tasks and subassembly states) of each product follow those of one benchmark instance. That is, we generate each tested instance, by merging two benchmark instances. Taking instance 2 in Table 6 for example, the candidate disassembly processes of the first product follow instance 'BBD12' and those of the second product follow instance 'TZC02'.

The scales of the tested instances are reported in Table 6, where (i) columns 2 and 3 denote the benchmark instances based on which the candidate disassembly processes of product 1 and 2 are generated, respectively; (ii) volumes 4-7 represent the problem scale of each instance.

Table 5. Data set of the benchmark instances.

Instance	Product	Literature	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{J}\rvert$	$\lvert\mathcal{S}\rvert$
BBD12	Compass	Bentaha, Battaïa, and Dolgui (2012)	10	3	7
TZC02	Hand light	Tang et al. (2002)	10	5	9
L99b	Ball-point pen	Lambert (1999)	20	9	15
KSE09	Sample product	Koc, Sabuncuoglu, and Erel (2009)	23	6	15
BBD13b	Piston and connecting rod	Bentaha, Battaïa, and Dolgui (2013a)	25	4	13
L99a	Radio set	Lambert (1999)	30	9	20
BAO2015	Rigid caster	Bentaha, Battaïa, and Dolgui (2015)	32	4	16
MJKL11	Automatic pencil	Ma et al. (2011)	37	10	24

Table 6. Scales of tested instances.

Set	Product 1	Product 2	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{J}\rvert$	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{R}\rvert$
1	BBD12	BBD12	20	6	14	20
2	BBD12	TZC02	20	8	16	20
3	BBD13	L99b	30	12	22	30
4	BBD14	KSE09	33	9	22	33
5	BBD15	BBD13b	35	7	20	35
6	BBD16	L99a	40	12	27	40
7	BBD17	BAO2015	42	7	23	42
8	BBD18	MJKL11	47	13	31	47
9	TZC02	TZC02	20	10	18	20
10	TZC03	L99b	30	14	24	30
11	TZC04	KSE09	33	11	24	33
12	TZC05	BBD13b	35	9	22	35
13	TZC06	L99a	40	14	29	40
14	TZC07	BAO2015	42	9	25	42
15	TZC08	MJKL11	47	15	33	47
16	L99b	L99b	40	18	30	40
17	L99b	KSE09	43	15	30	43
18	L99b	BBD13b	45	13	28	45
19	KSE09	KSE09	46	12	30	46
20	KSE09	BBD13b	48	10	28	48

Numerical results for instances of different scales are reported in Table 7. Since the SAA method loses its power to obtain feasible solutions within 3600 s for instances 11–20, the average results of the first 10 instances and all tested instances are reported in the last two raws, respectively. We can obtain that for the first ten instances, (i) the average objective value of the approximation method (under [**P1**] and [**P2**]) is 359.8, about 82.45% and 95.64% larger than those of the cutting-plane algorithm and the SAA method; (ii) the average safety level obtained by the approximation method is 99.61%, about 5.32% and 4.6% higher than those obtained by the cutting-plane algorithm and the SAA method; (iii) the average computational time of the approximation method is 2236 seconds, about 33.27% and 33.87% smaller than those of the cutting-plane algorithm and the SAA method; (iv) the average objective value obtained by the SAA method is 183.5, about 6.99% smaller than that obtained by the cutting-plane algorithm; (v) the average safety level obtained by the SAA method is 95.23%, slightly higher than that of the cutting-plane algorithm; but (vi) the computational time of the SAA method is 0.88% larger than that of the cutting-plane algorithm.

In addition, for the 20 tested instances, (1) from the results of the cutting-plane algorithm, it can be obtained that model [**P2**] slightly improve [**P1**], in terms of the average objective value, the average safety level and the computational time; (ii) the average objective value obtained by the approximation method based on [**P2**] is 384.1, about 11.46% smaller than that based on [**P1**]; (iii) the average safety level obtained by the approximation method based on [**P2**] is 99.41%, slightly higher than that based on [**P1**]; (iv) the average computational time of the approximation method based on [**P2**] is 2674, about 7.51% smaller than that based on [**P1**].

In sum, we can obtain from Table 7 that (i) the improved model [**P2**] outperforms [**P1**], in terms of the objective value, safety level and the computational time; (ii) the approximation method outperforms the cutting-plane algorithm and the SAA method, in terms of the safety level and computational time; but (iii) the objective value obtained by the approximation method is larger; (iv) the average objective obtained by the SAA method is smaller than the cutting-plane algorithm, but it fails to find feasible solutions within 3600s for larger-scale instances (i.e. instances 11–20); (v) compared with the approximation method, the cutting-plane algorithm can obtain solutions with smaller objective value, with little sacrifice of the safety level; and (v) the cutting-plane algorithm can find solutions with high quality, even for large-scale instances.

5.4. Sensitivity analyses

In this subsection, sensitivity analyses, based on the illustrative example, are conducted. Impact of the cycle time is first examined, and six values of C are tested, i.e. $15, 20, \ldots, 40$. Numerical results are shown in Table 8, where we can observe that compared with the approximation method and the SAA method, the cutting-plane algorithm is more unstable under different cycle times, in terms of the computational time. With the increase of the cycle time, the total costs obtained by the tested

Table 7. Numerical results of instances of different scales.

	Cutting-plane algorithm based on model [**P1**]			Cutting-plane algorithm based on model [**P2**]			Approximation method based on [**P1**]			Approximation method based on [**P2**]			The SAA		
	Obj	Safety level (%)	Time (s)	Obj	Safety level (%)	Time (s)	Obj	Safety level (%)	Time (s)	Obj	Safety level (%)	Time (s)	Obj	Safety level (%)	Time (s)
1	166	98.36	1133	166	98.36	1102	214	99.71	112	214	99.71	144	161	94.22	1413
2	236	94.65	3600	239	95.25	3600	307	99.55	197	307	99.55	227	179	95.35	3600
3	183	91.63	3600	183	91.58	3600	449	99.51	3600	442	99.50	3600	188	93.64	3600
4	215	98.91	3600	215	98.75	3600	282	99.73	1352	282	99.73	1873	215	96.93	3600
5	205	98.89	3600	161	93.68	3600	246	99.68	2650	246	99.68	1766	161	96.55	3600
6	178	99.50	3600	178	99.55	3600	181	99.85	3600	181	99.85	3600	177	99.49	3600
7	160	91.62	3600	161	93.57	3600	227	99.54	3307	227	99.54	2314	160	96.96	3600
8	217	92.30	3600	217	95.17	3600	665	99.56	3600	500	99.71	1353	219	96.80	3600
9	236	90.75	3600	236	89.37	3600	426	99.41	2839	426	99.42	1394	179	89.37	3600
10	197	90.97	3600	197	90.74	3600	759	99.48	3600	615	99.55	3600	196	92.96	3600
11	220	94.64	3600	220	95.56	3600	405	99.55	1054	405	99.57	3600	–	–	–
12	208	95.26	3600	208	95.38	3600	330	99.47	3600	327	99.47	2795	–	–	–
13	261	92.63	3600	261	92.83	3600	721	98.37	3600	599	99.01	2021	–	–	–
14	194	93.62	3600	194	96.52	3600	408	99.37	3105	369	99.50	3600	–	–	–
15	236	92.19	3600	236	91.86	3600	827	99.50	3600	593	99.49	3601	–	–	–
16	237	89.29	3600	237	89.37	3600	917	98.02	3600	679	98.31	3600	–	–	–
17	230	92.22	3600	230	92.35	3600	426	98.82	3600	383	98.91	3600	–	–	–
18	169	88.23	3600	169	88.23	3600	368	98.44	3600	368	98.44	3600	–	–	–
19	252	98.60	3600	252	98.60	3600	255	99.72	3600	255	99.72	3600	–	–	–
20	203	99.09	3600	203	99.09	3600	263	99.50	3600	263	99.50	3600	–	–	–
Average (instances 1–10)	199.3	94.76	3353	195.3	94.60	3350	375.6	99.60	2485	344	99.62	1987	183.5	95.23	3381.3
Average (instances 1–20)	210.2	94.17	3477	208.2	94.29	3475	433.8	99.34	2891	384.1	99.41	2674	–	–	–

'–' denotes that no feasible solution can be found within 3600 s.

Table 8. The impact of cycle time C.

	Cutting-plane algorithm			Approximation method			The SAA		
Cycle time (C)	Objective	Safety level (%)	Time (s)	Objective	Safety level (%)	Time (s)	Objective	Safety level (%)	Time (s)
15	183	88.04	229	365	99.37	19	174	91.72	429
20	228	98.53	356	349	99.54	19	224	95.93	435
25	276	99.22	24	351	99.55	18	274	98.03	426
30	324	98.96	209	328	99.75	16	324	98.96	436
35	374	99.11	31	376	99.84	16	373	97.70	440
40	424	99.49	57	424	99.92	16	423	98.51	429
Average	301.5	97.23	151	365.5	99.66	17	298.7	96.81	432

Table 9. The impact of the risk level ϵ.

	Cutting-plane algorithm			Approximation method			The SAA		
Risk level (ϵ)	Objective	Safety level (%)	Time (s)	Objective	Safety level (%)	Time (s)	Objective	Safety level (%)	Time (s)
0.05	234	98.2	238	411	99.67	64	224	93.8	471
0.1	231	91.65	230	411	99.67	46	224	93.8	486
0.15	229	98.2	251	411	99.67	21	224	94.8	482
0.2	228	98.53	256	349	99.54	20	224	95.8	483
0.25	223	91.65	240	286	99.14	18	224	96.8	466
0.3	223	92.65	221	228	98.53	21	224	97.8	478

three methods cutting-plane algorithm increases, and the safety level increases first and then changes slightly. The average safety level of the approximation method is 99.66%, about 2.5% and 2.94% larger than those of the cutting-plane algorithm and the SAA method. The average total cost obtained by the approximation method is 365.5, about 21.23% and 22.36% larger than those of the cutting-plane algorithm and the SAA method. Though the total cost obtained by the cutting-plane algorithm is slightly smaller than that of the SAA method, its obtained safety level is higher and the computational time is smaller.

The impact of the risk level ϵ is then examined, and ϵ is set from $0.05, 0.1, \ldots, 0.3$, as shown in Table 9. For the approximation method, with the increase of ϵ, the computational time first increases and then barely changes. The reason is that under smaller values of ϵ, the problem described by the approximation method is infeasible, due to the conservation, thus a larger ϵ is required. The solutions obtained by the SAA method are the same under different tested values of ϵ. The objective values of the cutting-plane algorithm and the approximation method decrease. The reason may be that when ϵ increases, less workstations are required, leading to a smaller system cost.

From the above observation, we provide the following suggestions for decision makers:

(1) Trade-offs between the cycle time and the system cost are significant for decision making. A larger cycle time results in a higher safety level, while it also leads to larger cost for operating workstation and handling hazardous tasks, thus the system cost increases.
(2) When the practitioners and managers place more weight on the safety level or the computational time, the approximation method is recommended.
(3) When the practitioners and managers place more weight on the total cost with a reasonable safety level, the cutting-plane algorithm is recommended.

6. Conclusion

This paper investigates a stochastic multi-product DLBP with workforce assignment, in which the task processing times are uncertain. Only partial information is given, i.e. the first two moments and the unimodality of task processing times. For the problem, a new stochastic optimisation formulation with distributionally robust CVaR constraints is first proposed. To solve the problem, an exact cutting-plane method and an approximation method are developed. Numerical experiments on instances of different scales are conducted, to evaluate the proposed solution approaches.

In this work, the workload balance between opened workstations is regulated via the CVaR constraints combining with the minimisation of the opened workstation number. However, the widely used criterion of optimising balance measures, such as minimising the total idle times and minimising the maximal workload difference between opened workstations, may have more advantages in balancing workstations' workloads (Kalayci and Gupta 2014; McGovern and Gupta 2015). Besides, workers' preference on their assigned tasks is not considered in this work, which may lead to the reduction of the fairness of employees. In addition, during the disassembly process, directions of the EOL product while removing different components may be different. For example, an EOL refrigerator may be laid down from vertical position to the horizontal position. Direction changes may be required between two consecutive tasks, leading to a higher cost. Therefore, our future research directions may include: (1) investigating the bi-objective optimisation program, to minimise the total cost and the total idle times simultaneously; (2) studying the problem where workers' dissatisfaction is included in the objective; (3) incorporating the direction changes between consecutive tasks into objective function.

Acknowledgments

The authors would like to thank the anonymous referees for their constructive comments. This work was supported by the National Science Foundation of China (NSFC) under Grants 71531011, 71771048, 71432007, 71832001, 71871159 and 71571134.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Science Foundation of China (NSFC) [grant number 71531011], [grant number 71771048], [grant number 71432007], [grant number 71832001], [grant number 71871159], [grant number 71571134].

Notes on contributors

Xin Liu received the Ph.D. degree in management science and engineering from Tongji University, Shanghai, China, in 2020. She is a Lecturer in Donghua University. Her research interests include production scheduling and supply chain optimisation.

Feng Chu received the B.S. degree in electrical engineering from Hefei University of Technology, Hefei, China, in 1986; the M.S. degree in metrology, automatic control, and electrical engineering from National Polytechnic Institute of Lorraine, Lorraine, France, in 1991; and the Ph.D. degree in automatic control, computer science, and production management from University of Metz, Metz, France, in 1995. She is currently a Full Professor of Operations Research at Univ Evry, University of Paris Saclay, Evry, France and co leader of the Algorithmic, Operations Research, Bioinformatics and Statistical learning group. Her research interests include the modelling, analysis, and optimisation of complex systems, such as intelligent transportation systems and logistic and production systems based on combinatorial optimisation, operations research, and Petri nets. Dr. Chu was an Associate Editor for IEEE T-SMC, Part C from 2010 to 2013. She is currently an Associate Editor for IEEE T-ITS and IEEE TASE. She is an IPC member for over 70 international conferences.

Feifeng Zheng received the B.S. degree in information management, the M.S. degree in management science and engineering and the Ph.D. degree in management science and engineering from Xi'an Jiaotong University, Xi'an, China, in 1998, 2003 and 2006. He is a Professor in Donghua University, Shanghai. His research interests include production scheduling and container terminal resource scheduling.

Chengbin Chu received the B.S. degree in electrical engineering from the Hefei University of Technology, Hefei, China, in 1985, and the Ph.D. degree in computer science from the University of Metz, Metz, France, in 1990. He was with the National Research Institute in Computer Science and Automation, Metz, from 1987 to 1996. He was a Professor with the University of Technology of Troyes, Troyes, France, from 1996 to 2008, where he was the Founding Director of the Industrial Systems Optimization Laboratory. He is currently a Professor with ESIEE Paris, Université Paris-Est, France. He is interested in operations research and modelling, analysis, and optimisation of supply chain and production systems. Dr. Chu is currently an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.

Ming Liu received the B.S. degree in management science and engineering and the Ph.D. degree in management science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2005 and 2010. He is an Associate Professor in Tongji University, Shanghai. His research interests include logistics optimisation and production scheduling.

References

Altekin, F. T. 2017. "A Comparison of Piecewise Linear Programming Formulations for Stochastic Disassembly Line Balancing." *International Journal of Production Research* 55: 7412–7434.

Altekin, F. T., L. Kandiller, and N. E. Ozdemirel. 2008. "Profit-oriented Disassembly-line Balancing." *International Journal of Production Research* 46: 2675–2693.

Aydemir-Karadag, A., and O. Turkbey. 2013. "Multi-objective Optimization of Stochastic Disassembly Line Balancing with Station Paralleling." *Computers & Industrial Engineering* 65: 413–425.

Bentaha, M. L., O. Battaïa, and A. Dolgui. 2012. "A Stochastic Formulation of the Disassembly Line Balancing Problem." In *IFIP International Conference on Advances in Production Management Systems*, 397–404. Berlin, Heidelberg: Springer.

Bentaha, M. L., O. Battaïa, and A. Dolgui. 2013a. "Chance Constrained Programming Model for Stochastic Profit–Oriented Disassembly Line Balancing in the Presence of Hazardous Parts." In *IFIP International Conference on Advances in Producti Management Systems*, 103–110. Berlin, Heidelberg: Springer.

Bentaha, M. L., O. Battaïa, and A. Dolgui. 2013b. "L-shaped Algorithm for Stochastic Disassembly Line Balancing Problem." *IFAC Proceedings Volumes* 46: 407–411.

Bentaha, M. L., O. Battaïa, and A. Dolgui. 2014. "A Sample Average Approximation Method for Disassembly Line Balancing Problem Under Uncertainty." *Computers & Operations Research* 51: 111–122.

Bentaha, M. L., O. Battaïa, and 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: 1807–1818.

Bentaha, M. L., O. Battaïa, A. Dolgui, and S. J. Hu. 2014. "Dealing with Uncertainty in Disassembly Line Design." *CIRP Annals – Manufacturing Technology* 63: 21–24.

Bentaha, M. L., O. Battaïa, A. Dolgui, and S. J. Hu. 2015. "Second Order Conic Approximation for Disassembly Line Design with Joint Probabilistic Constraints." *European Journal of Operational Research* 247: 957–967.

Braglia, M., M. Frosolini, and F. Zammori. 2008. "Overall Equipment Effectiveness of a Manufacturing Line (OEEML): An Integrated Approach to Assess Systems Performance." *Journal of Manufacturing Technology Management* 20: 8–29.

Bruecker, P. D., J. V. D. Bergh, J. Beliën, and E. Demeulemeester. 2015. "Workforce Planning Incorporating Skills: State of the Art." *European Journal of Operational Research* 243: 1–16.
Cai, W., C. Liu, K.h. Lai, L. Li, J. Cunha, and L. Hu. 2019. "Energy Performance Certification in Mechanical Manufacturing Industry: A Review and Analysis." *Energy Conversion and Management*186: 415–432.
Cakir, B., F. Altiparmak, and B. Dengiz. 2011. "Multi-objective Optimization of a Stochastic Assembly Line Balancing: A Hybrid Simulated Annealing Algorithm." *Computers & Industrial Engineering* 60: 376–384.
Ciez, R. E., and J. Whitacre. 2016. "Comparative Techno-economic Analysis of Hybrid Micro-grid Systems Utilizing Different Battery Types." *Energy Conversion and Management* 112: 435–444.
Delage, E., and Y. Ye. 2010. "Distributionally Robust Optimization Under Moment Uncertainty with Application to Data-driven Problems." *Operations Research* 58: 595–612.
Deniz, N., and F. Ozcelik. 2019. "An Extended Review on Disassembly Line Balancing with Bibliometric & Social Network and Future Study Realization Analysis." *Journal of Cleaner Production* 225: 697–715.
Dolgui, A., and S. Kovalev. 2012. "Scenario Based Robust Line Balancing: Computational Complexity." *Discrete Applied Mathematics* 160: 1955–1963.
Gérard, M., F. Clautiaux, and R. Sadykov. 2016. "Column Generation Based Approaches for a Tour Scheduling Problem with a Multi-skill Heterogeneous Workforce." *European Journal of Operational Research* 252: 1019–1030.
Gurevsky, E., O. Battaïa, and A. Dolgui. 2012. "Balancing of Simple Assembly Lines Under Variations of Task Processing Times." *Annals of Operations Research* 201: 265–286.
Gurevsky, E., Ö. Hazır, O. Battaıa, and A. Dolgui. 2013. "Robust Balancing of Straight Assembly Lines with Interval Task Times." *Journal of the Operational Research Society* 64: 1607–1613.
HazıR, Ö., and A. Dolgui. 2013. "Assembly Line Balancing Under Uncertainty: Robust Optimization Models and Exact Solution Method." *Computers & Industrial Engineering* 65: 261–267.
He, J., F. Chu, F. Zheng, M. Liu, and C. Chu. 2019. "A Green-oriented Bi-objective Disassembly Line Balancing Problem with Stochastic Task Processing Times." *Annals of Operations Research* 296: 71–93.
He, J., F. Chu, F. Zheng, M. Liu, and C. Chu. 2020. "A Multi-objective Distribution-free Model and Method for Stochastic Disassembly Line Balancing Problem." *International Journal of Production Research* 58: 5721–5737.
Hezer, S., and Y. Kara. 2015. "A Network-based Shortest Route Model for Parallel Disassembly Line Balancing Problem." *International Journal of Production Research* 53: 1849–1865.
Hrouga, M., M. Godichaud, and L. Amodeo. 2016. "Heuristics for Multi-product Capacitated Disassembly Lot Sizing with Lost Sales." *IFAC-PapersOnLine* 49: 628–633.
Ilgin, M. A., H. Akçay, and C. Araz. 2017. "Disassembly Line Balancing Using Linear Physical Programming." *International Journal of Production Research* 55: 6108–6119.
Ilgin, M. A., and S. M. Gupta. 2010. "Comparison of Economic Benefits of Sensor Embedded Products and Conventional Products in a Multi-product Disassembly Line." *Computers & Industrial Engineering*59: 748–763.
Jiang, R., and Y. Guan. 2016. "Data-driven Chance Constrained Stochastic Program." *Mathematical Programming* 158: 291–327.
Kalayci, C. B., and S. M. Gupta. 2014. "A Tabu Search Algorithm for Balancing a Sequence-dependent Disassembly Line." *Production Planning & Control* 25: 149–160.
Kalayci, C. B., A. Hancilar, A. Gungor, and S. M. Gupta. 2015. "Multi-objective Fuzzy Disassembly Line Balancing Using a Hybrid Discrete Artificial Bee Colony Algorithm." *Journal of Manufacturing Systems* 37: 672–682.
Kalaycılar, E. G., M. Azizoğlu, and S. Yeralan. 2016. "A Disassembly Line Balancing Problem with Fixed Number of Workstations." *European Journal of Operational Research* 249: 592–604.
Kannan, D., K. Garg, P. 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: 353–376.
Koc, A., I. Sabuncuoglu, and E. Erel. 2009. "Two Exact Formulations for Disassembly Line Balancing Problems with Task Precedence Diagram Construction Using An AND/OR Graph." *IIE Transactions*41: 866–881.
Lambert, A. J. D. 1999. "Linear Programming in Disassembly/clustering Sequence Generation." *Computers & Industrial Engineering* 36: 723–738.
Li, B., R. Jiang, and J. L. Mathieu. 2019. "Ambiguous Risk Constraints with Moment and Unimodality Information." *Mathematical Programming* 173: 151–192.
Liu, M., R. Liu, and F. Chu. 2020. "Distribution-Free And Risk-Averse Disassembly Line Balancing Problem." In *2019 International Conference on Industrial Engineering and Systems Management (IESM)*, IEEE. doi:10.1109/IESM45758.2019.8948079.
Liu, M., X. Liu, F. Chu, F. Zheng, and C. Chu. 2020a. "Robust Disassembly Line Balancing with Ambiguous Task Processing Times." *International Journal of Production Research* 58: 5806–5835.
Liu, M., X. Liu, F. Chu, F. Zheng, and C. Chu. 2020b. "An Exact Method for Disassembly Line Balancing Problem with Limited Distributional Information." *International Journal of Production Research*. doi:10.1080/00207543.2019.1704092.
Ma, Y. S., H. B. Jun, H. W. Kim, and D. H. Lee. 2011. "Disassembly Process Planning Algorithms for End-of-life Product Recovery and Environmentally Conscious Disposal." *International Journal of Production Research* 49: 7007–7027.
Mcgovern, S. M., and S. M. Gupta. 2007. "Combinatorial Optimization Analysis of the Unary Np-complete Disassembly Line Balancing Problem." *International Journal of Production Research* 45: 4485–4511.
McGovern, S. M., and S. M. Gupta. 2015. "Unified Assembly- and Disassembly-line Model Formulae." *Journal of Manufacturing Technology Management* 26: 195–212.
MIIT (Ministry of Industry and Information Technology of the People's Republic of China). 2018. "Interim Measures for the Management of Recycling and Utilization of Power Batteries for New Energy Vehicles." Http://www.miit.gov.cn/n1146295/n1652858/n1652930/n3757016/c6068 823/content.html.
Moreira, M. C. O., J. F. Cordeau, A. M. Costa, and G. Laporte. 2015. "Robust Assembly Line Balancing with Heterogeneous Workers." *Computers & Industrial Engineering* 88: 254–263.
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: 4805–4827.
Pereira, J. 2018. "The Robust (minmax Regret) Assembly Line Worker Assignment and Balancing Problem." *Computers & Operations Research* 93: 27–40.
Pereira, J., and E. Alvarez-Miranda. 2018. "An Exact Approach for the Robust Assembly Line Balancing Problem." *Omega* 78: 85–98.
Pınarbaşı, M., M. Yüzükırmızı, and B. Roklu. 2016. "Variability Modelling and Balancing of Stochastic Assembly Lines." *International Journal of Production Research* 54: 5761–5782.
Ren, Y., D. Yu, C. Zhang, G. Tian, L. Meng, and X. Zhou. 2017. "An Improved Gravitational Search Algorithm for Profit-oriented Partial Disassembly Line Balancing Problem." *International Journal of Production Research* 55: 7302–7316.
Ren, Y., C. Zhang, F. Zhao, G. Tian, W. Lin, L. Meng, and H. Li. 2018. "Disassembly Line Balancing Problem Using Interdependent Weights-based Multi-criteria Decision Making and 2-optimal Algorithm." *Journal of Cleaner Production* 174: 1475–1486.
Rockafellar, R. T., and S. Uryasev. 2000. "Optimization of Conditional Value-at-risk." *Journal of Risk*2: 21–42.
Sarykalin, S., G. Serraino, and S. Uryasev. 2008. "Value-at-Risk vs. Conditional Value-at-Risk in Risk Management and Optimization." In *INFORMS*, 2008.
Tang, Y., Q. Zhang, Y. Li, G. Wang, and Y. Li. 2018. "Recycling Mechanisms and Policy Suggestions for Spent Electric Vehicles' Power Battery-A Case of Beijing." *Journal of Cleaner Production* 186: 388–406.
Tang, Y., M. C. Zhou, E. Zussman, and R. Caudill. 2002. "Disassembly Modeling, Planning, and Application." *Journal of Manufacturing Systems* 21: 200–217.
Wegener, K., W. H. Chen, F. Dietrich, K. Dröder, and S. Kara. 2015. "Robot Assisted Disassembly for the Recycling of Electric Vehicle Batteries." *Procedia CIRP* 29: 716–721.
Xie, W., and S. Ahmed. 2018. "On Deterministic Reformulations of Distributionally Robust Joint Chance Constrained Optimization Problems." *SIAM Journal on Optimization* 28: 1151–1182.
Yan, D., L. Lu, Z. Li, X. Feng, M. Ouyang, and F. Jiang. 2016. "Durability Comparison of Four Different Types of High-power Batteries in Hev and Their Degradation Mechanism Analysis." *Applied Energy*179: 1123–1130.
Zhang, Z., J. Cai, F. Chen, H. Li, W. Zhang, and W. Qi. 2018. "Progress in Enhancement of CO2 Absorption by Nanofluids: A Mini Review of Mechanisms and Current Status." *Renewable Energy* 118: 527–535.
Zhang, Y., R. Jiang, and S. Shen. 2018. "Ambiguous Chance-constrained Binary Programs Under Mean-covariance Information." *SIAM Journal on Optimization* 28: 2922–2944.
Zhang, Y., S. Shen, and S. A. Erdogan. 2017. "Solving 0-1 Semidefinite Programs for Distributionally Robust Allocation of Surgery Blocks." *Optimization Letters* 12: 1503–1521.
Zhang, W., W. Xu, and M. Gen. 2014. "Hybrid Multiobjective Evolutionary Algorithm for Assembly Line Balancing Problem with Stochastic Processing Time." *Procedia Computer Science* 36: 587–592.
Zhao, S. E., Y. L. Li, R. Fu, and W. Yuan. 2014. "Fuzzy Reasoning Petri Nets and Its Application to Disassembly Sequence Decision-making for the End-of-life Product Recycling and Remanufacturing." *International Journal of Computer Integrated Manufacturing* 27: 415–421.
Zheng, F., J. He, F. Chu, and M. Liu. 2018. "A New Distribution-free Model for Disassembly Line Balancing Problem with Stochastic Task Processing Times." *International Journal of Production Research* 56: 7341–7353.
Zhou, F., M. K. Lim, Y. He, Y. Lin, and S. Chen. 2019. "End-of-life Vehicle (ELV) Recycling Management: Improving Performance Using An Ism Approach." *Journal of Cleaner Production* 228: 231–243.

Appendix. The SAA method

The SAA method has been widely and successfully used for solving stochastic programs (Pagnoncelli et al., 2009; Bentaha et al., 2014). The SAA method is adapted for our problem, by replacing the partial known distribution $\mathbb{P}_{\xi}$ with an empirical one (e.g. Normal distribution), corresponding to a limited set Ω of randomly generated scenarios. Rockafellar and Uryasev (2000) show that the CVaR of a random variable X can be equivalently described by

$$\mathrm{CVaR}_{\epsilon}(X) = \inf_{\beta\in\mathbb{R}} \left\{ \beta + \frac{1}{1-\epsilon}\mathbb{E}\left[\max\{X-\beta, 0\}\right] \right\}.$$

We adapt the SAA method based on the above equation: new decision variables $q_j \in \mathbb{R}$ and $U_j(\omega) \geq 0$, and a recourse variable $R_j \geq 0$ denoting the CVaR of workstation j's working time beyond the cycle time are introduced, and an SAA-based MIP formulation is established in the following:

$$\begin{aligned}
[\mathbf{P6}]: \min \; & \sum_{l\in\mathcal{L}}\sum_{j\in\mathcal{J}} c^{P}_{jl}\eta_{jl} + \sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{T}} c^{W}_{ri}\nu_{ri} \\
& + C\cdot\left(\sum_{l\in\mathcal{L}}\sum_{j\in\mathcal{J}} c^{F} y_{jl} + c^{H}\sum_{j\in\mathcal{J}} z_j\right) \qquad \text{(A1)} \\
& + \sum_{j\in\mathcal{J}} R_j
\end{aligned}$$

$$\text{s.t.} \quad (2)\text{–}(12),(15)\text{–}(20),(22)\text{–}(27) \qquad \text{(A2)}$$

$$q_j + \frac{1}{1-\epsilon}\sum_{\omega\in\Omega} U_j(\omega) - R_j \leq C, \quad \forall j\in\mathcal{J} \qquad \text{(A3)}$$

$$U_j(\omega) \geq \sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{T}} \pi_{rij}\zeta_{ri}\xi_i(\omega) - q_j, \; \forall j\in\mathcal{J}, \omega\in\Omega \qquad \text{(A4)}$$

$$q_j \in \mathbb{R}, \quad \forall j\in\mathcal{J} \qquad \text{(A4)}$$

$$U_j(\omega), R_j \geq 0, \quad \forall j\in\mathcal{J}, \omega\in\Omega \qquad \text{(A5)}$$