



**HAL**  
open science

# A new and general stochastic parallel machine ScheLoc problem with limited location capacity and customer credit risk

Ming Liu, Tao Lin, Feng Chu, Feifeng Zheng, Chengbin Chu

► **To cite this version:**

Ming Liu, Tao Lin, Feng Chu, Feifeng Zheng, Chengbin Chu. A new and general stochastic parallel machine ScheLoc problem with limited location capacity and customer credit risk. *RAIRO - Operations Research*, 2023, 57 (3), pp.1179-1193. 10.1051/ro/2023016 . hal-04488659

**HAL Id: hal-04488659**


**<https://hal.science/hal-04488659v1>**

Submitted on 4 Apr 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## A NEW AND GENERAL STOCHASTIC PARALLEL MACHINE SCHELOC PROBLEM WITH LIMITED LOCATION CAPACITY AND CUSTOMER CREDIT RISK

MING LIU<sup>1</sup>, TAO LIN<sup>1</sup>, FENG CHU<sup>2,\*</sup>, FEIFENG ZHENG<sup>3</sup> AND CHENGBIN CHU<sup>4</sup>

**Abstract.** Scheduling-Location (ScheLoc) problem considering machine location and job scheduling simultaneously is a relatively new and hot topic. The existing works assume that only one machine can be placed at a location, which may not be suitable for some practical applications. Besides, the customer credit risk which largely impacts the manufacturer's profit has not been addressed in the ScheLoc problem. Therefore, in this work, we study a new and general stochastic parallel machine ScheLoc problem with limited location capacity and customer credit risk. The problem consists of determining the machine-to-location assignment, job acceptance, job-to-machine assignment, and scheduling of accepted jobs on each machine. The objective is to maximize the worst-case probability of manufacturer's profit being greater than or equal to a given profit (referred to as the profit likelihood). For the problem, a distributionally robust chance-constrained (DRCC) programming model is proposed. Then, we develop two model-based approaches: (i) a sample average approximation (SAA) method; (ii) a model-based constructive heuristic. Numerical results of 300 instances adapted from the literature show the average profit likelihood proposed by the constructive heuristic is 9.43% higher than that provided by the SAA, while the average computation time of the constructive heuristic is only 4.24% of that needed by the SAA.

**Mathematics Subject Classification.** 90.

Received February 24, 2022. Accepted February 10, 2023.

### 1. INTRODUCTION

Recent information and automation technologies provide foundations for Industry 4.0 [2, 15]. The core idea of Industry 4.0 is to use the emerging technologies in a way that business and engineering processes are deeply integrated [24]. Scheduling problem and location problem play important roles in manufacturing, and both of them have received much attentions from academia [1, 14, 26, 27]. In practice, these two problems closely intertwine and impact the production performance together. For example, in container port operations, decision

---

*Keywords.* parallel machine ScheLoc problem, limited location capacity, customer credit risk, distributionally robust optimization.

<sup>1</sup> School of Economics & Management, Tongji University, Shanghai, P.R. China.

<sup>2</sup> School of Economics & Management, Fuzhou University, Fujian, P.R. China.

<sup>3</sup> Glorious Sun School of Business & Management, Donghua University, Shanghai, P.R. China.

<sup>4</sup> Université Gustave-Eiffel, ESIEE Paris, COSYS-GRETTIA, F-77454 Marne-la-Vallée, France.

\*Corresponding author: [feng.chu@univ-evry.fr](mailto:feng.chu@univ-evry.fr)

makers must determine the positions of ships in berth (*i.e.*, location problem) and the schedule for loading and unloading containers (*i.e.*, scheduling problem), simultaneously [11]. Another example comes from the mining industry. The planners must jointly determine which candidate locations should be selected to deploy crushers (*i.e.*, space problem) and how to sequence minerals on the crushers at those locations (*i.e.*, scheduling problem) [10]. These facts indicate the necessity of investigating the integrated scheduling-location problem.

The scheduling-location problem is referred to as ScheLoc problem which is first introduced by Hennes and Hamacher [9]. The deterministic single machine ScheLoc problem is well addressed by Hennes and Hamacher [9] and Kalsch and Drezner [11]. Since parallel machine manufacturing environment is quite common in practice, recent works focus on the deterministic parallel machine ScheLoc problem [8, 13, 28]. In reality, unexpected events often occur. Especially, job processing time can be uncertain as it may be affected by various factors such as machine and equipment conditions, manufacturing environment and operator skills [3]. Therefore, Liu *et al.* [19] study a stochastic parallel machine ScheLoc problem with uncertain job processing time. Moreover, Liu and Liu [16] address the parallel machine ScheLoc problem with only mean vector and covariance matrix of job processing time known. These works promote the studies of ScheLoc problems. However, all of them assume that only one machine can be placed at a location, which may not be suitable for some practical applications. In reality, some locations are larger to accommodate more machines than the others. Therefore, this study extends the existing works and investigate a new and general parallel machine ScheLoc problem. For this novel problem, we assume a location can place several machines and its holding capacity is limited.

Usually, manufacturers complete required jobs and get the corresponding payment timely. However, according to Liu *et al.* [20], many manufacturers have experienced late payment from their customers. Manufacturers prefer jobs from customers with low credit risk to curb the adverse consequences of high customer late payment probability and maintain competitiveness. Tsosie and Nicastrò [22] point out that customer credit has been applied to evaluate the customer payment probability. Liu *et al.* [20] indicate further that around 34% manufacturers have started to assess the customer payment probability based on their credit history. They are also the first to consider customer credit risk for flow shop scheduling problem. However, there is no research investigating the ScheLoc problem under customer credit risk.

Motivated by the above facts, this work addresses a novel and general stochastic parallel machine ScheLoc problem with limited location capacity and customer credit risk. The customer credit risk and job processing time are both assumed to be ambiguous, and only their partial distributional information is given. The studied problem consists of determining the machine-to-location assignment, job acceptance, job-to-machine assignment, and scheduling of accepted jobs on each machine. The objective is to maximize the profit likelihood, *i.e.*, the worst-case probability of manufacturer's profit being greater than or equal to a given level. Besides, the worst-case probability of job completion time respecting the due date (referred to as the service likelihood) is also guaranteed. For the problem, a distributionally robust chance-constrained (DRCC) programming model is constructed. Then, we develop two model-based approaches: (i) a sample average approximation (SAA) method; (ii) a constructive heuristic. The contributions of this paper mainly include:

- (1) A new and general stochastic parallel machine ScheLoc problem with limited location capacity and customer credit risk is investigated. Especially, the assumption in the literature that only one machine can be placed at a location is relaxed.
- (2) For the problem, a DRCC programming model is proposed.
- (3) Two model-based approaches, *i.e.*, a SAA method and a constructive heuristic, are developed to efficiently solve the problem.

The remainder of this paper is laid out as follows. A literature review is presented in Section 2. In Section 3, we describe the studied problem in detail and propose a new DRCC programming model. In Section 4, a SAA method and a model-based constructive heuristic are developed. Numerical experiments are conducted in Section 5. In Section 6, this work is concluded and future research directions are suggested.

## 2. LITERATURE REVIEW

This paper addresses a general stochastic parallel machine ScheLoc problem with limited location capacity and customer credit risk. For the problem, a DRCC programming model is constructed. In the following, we review the related works about ScheLoc problems and distributionally robust (DR) approaches.

The ScheLoc problem, first introduced by Hennes and Hamacher [9], has received much attention from academia [8, 12, 13, 16, 19, 25, 28]. Most existing works concern the deterministic ScheLoc problems [8, 9, 12, 13, 25, 28]. Hennes and Hamacher [9] first study a deterministic single machine ScheLoc problem, to minimize the makespan. A polynomial-time algorithm based on earliest release date (ERD) first rule is developed to solve the problem. Since parallel machine environment is quite common in practice, Heßler and Deghdak [8] investigate a deterministic parallel machine ScheLoc problem. The objective is to minimize the makespan. For the problem, a mixed integer programming (MIP) model is first proposed. Then, several constructive heuristics are developed for it. 1450 instances are tested to demonstrate the effectiveness of the developed algorithms. The same problem is then studied by Wang *et al.* [25], Kramer and Kramer [12] and Li *et al.* [13]. Wang *et al.* [25] first propose a new MIP model. Then, three constructive heuristics are designed to solve the problem. Numerical experiment results on benchmark instances proposed by Heßler and Deghdak [8] show that both the new MIP model and the proposed heuristics perform better than that of Heßler and Deghdak [8]. Kramer and Kramer [12] propose a new MIP model from the perspective of arc-flow formulation. Then, a column generation method, two constructive heuristics and an iterated local search metaheuristic are designed. Numerical experiments show that the optimal solutions of the same benchmark instances can be obtained. Li *et al.* [13] propose three new mixed integer linear programming (MILP) models and a novel logic-based Benders decomposition (LBBDD) method. Numerical results demonstrate the efficiency of the new formulations and the algorithm. Zhang *et al.* [28] investigate a new variant of the parallel machine ScheLoc problem considering delivery times and due dates. The work aims to minimizing the total cost. They first develop MILP models. A tailored LBBDD method and a matheuristic are then developed to solve the problem.

In practice, unexpected events such as machine breakdown often occur. Therefore, some recent works focus on the ScheLoc problem with uncertain job processing time. Liu *et al.* [19] first study a stochastic parallel machine ScheLoc problem. The objective is to minimize the weighted sum of the expected total completion time and the location cost. For the problem, a two-stage stochastic programming model is established. Then, a SAA method, a genetic algorithm and a scenario-based heuristic are developed to solve the model. Liu and Liu [16] consider the stochastic parallel machine ScheLoc problem, in which only the mean and covariance matrix of job processing time are known. The objective is to minimize the machine location cost. The worst-case probability of job completion time respecting the due date is also guaranteed. For the problem, a distributionally robust (DR) formulation is first proposed. Then, an approximated mixed integer second order cone programming model is developed for it.

The above works provide beneficial support for decision-making in the ScheLoc problem. However, all of them restrict that only one machine can be placed at a location, and ignore the customer credit risk which has a non-negligible impact on the manufacturers' profit [20]. Note that Liu *et al.* [20] are the first one to study a stochastic flow shop scheduling problem with customer credit risk. Nevertheless, there is no research work integrating it into the ScheLoc problem, to the best of our knowledge.

Besides, due to data scarcity, it is usually impractical to accurately estimate the probability distribution of random parameters [5]. DR optimization, first introduced by Scarf [21], is an appropriate tool to handle uncertainty in the case where its distributional knowledge cannot be fully obtained and optimize the worst-case performance measure over an uncertainty set, namely ambiguity set. Wagner [23] and Delage and Ye [5] use moment-based ambiguity sets for deriving tractable reformulations of DR models. Referring to their works, we also consider a moment-based ambiguity set which is constructed based on the given mean vector and covariance matrix of uncertain parameters for this study.

Based on above facts, in the work, we study a new and general stochastic parallel machine ScheLoc problem with limited location capacity and customer credit risk. Especially, we assume that customer credit risk and job

TABLE 1. Comparison of ScheLoc related works.

Literature	Problem	Deter.	Stoch.	CCR	Capacity	Objective	Approach
Hennes and Hamacher [9]	Single	✓			1	Min makespan	ERD
Heßler and Deghdak [8]	Parallel	✓			1	Min makespan	MIP and CH
Wang <i>et al.</i> [25]	Parallel	✓			1	Min makespan	MIP and CH
Kramer and Kramer [12]	Parallel	✓			1	Min makespan	MIP, CG, CH and ILS
Li <i>et al.</i> [13]	Parallel	✓			1	Min makespan	MILP, LBBD
Zhang <i>et al.</i> [28]	Parallel	✓			1	Min total cost	MILP, LBBD, matheuristic
Li <i>et al.</i> [19]	Parallel		✓		1	Min weighted sum of LSC and expected TCT	SAA, GA and CH
Liu <i>et al.</i> [16]	Parallel		✓		1	Min LSC	MI-SOCP
This paper	Parallel		✓	✓	$\mathbb{N}_+$	Max profit likelihood	SAA, model-based CH

**Notes.** Deter.: deterministic environment; Stoch.: stochastic environment; CCR: customer credit risk;  $\mathbb{N}_+$ : the location-dependent capacity; Single: single machine; Parallel: parallel machine; TCT: total completion time; CH: constructive heuristic; CG: column generation; ILS: iterated local search; LSC: location selection cost.

processing time are ambiguous and only their partial distributional information is given due to data scarcity. For the problem, a DRCC programming model is constructed. Table 1 summarizes main differences between our work and the closely related literature.

### 3. PROBLEM DESCRIPTION AND FORMULATION

In the section, the detail of the studied problem is first described. Then, a new mathematical programming model is constructed. Besides, a complexity analysis is conducted for the studied problem.

#### 3.1. Problem description

For the studied problem, consider a set of candidate locations  $K$ , a set of machines  $N$  and a set of jobs  $J$ . A location  $k \in K$  can be selected to place at most  $\delta_k$  machines, where  $\delta_k$  is called the capacity of location  $k \in K$ . When location  $k \in K$  is selected, there is a location selection cost  $c_k^L$ . A cost  $c_{nk}^M$  is incurred when a machine  $n \in N$  is assigned to a location  $k \in K$ . A job  $j \in J$  with a customer payment probability  $l_j \in [0, 1]$  corresponding to a credit risk can be rejected, but results in penalty  $e_j$ . If job  $j \in J$  is accepted to process, the manufacturer can obtain an expected revenue  $f_j \cdot l_j$ , where  $f_j$  denotes the revenue without credit risk [20]. In addition, job  $j \in J$  is located initially at a storage area and has a release time  $r_{jk}$  that corresponds to its moving time from the storage area to a location  $k \in K$ . Each job  $j \in J$  has a due date  $d_j$ . The uncertain processing time of job  $j \in J$  is denoted as  $p_j$ , for which only the mean and covariance are known. Besides, this work follows basic assumptions:

- (1) A location can place several machines, and its capacity is limited.
- (2) All parallel machines are identical.
- (3) Job  $j \in J$  can only be assigned to exactly one machine if it is accepted to process.
- (4) A machine can process at most one job at a time.
- (5) Any two uncertain parameters are independent of each other.

To illustrate the studied problem, an example with 3 candidate locations, 2 machines and 8 jobs is presented. The job storage areas and candidate locations are dispersed on a network in Figure 1. The release time of job  $j$  to location  $k$  is set to be their Euclidean distance (*e.g.*,  $r_{21} = 2$ ) where we assume that the common moving speed is 1 per time unit. The means of job processing times and customer payment probabilities are set to be [5, 4, 6, 3, 3, 2, 9, 7] and [0.6, 0.8, 0.3, 0.7, 0.5, 0.8, 0.1, 0.3], respectively. The job rejection penalties, the revenues for processing job without credit risk, and the due dates are set to be [4, 2, 4, 2, 1, 5, 1, 2], [8, 10, 7, 11, 12, 14, 6, 10], and [12, 7, 19, 6, 18, 8, 23, 19], respectively. The location selection costs are set to be [18,

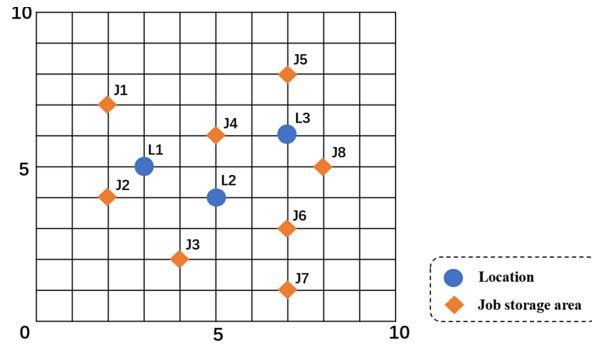


FIGURE 1. An illustrated example for the problem.

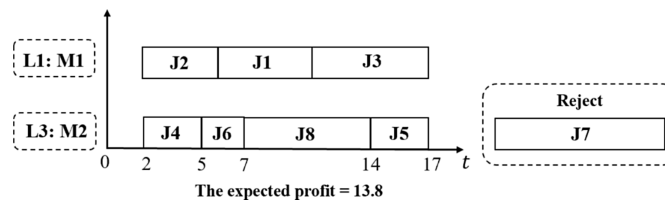


FIGURE 2. A feasible solution of the example.

30, 10], and the machine-to-location assignment costs and the capacity of each location are uniformly set to be 1. The problem consists of determining the machine-to-location assignment, job acceptance, job-to-machine assignment and scheduling of accepted jobs on each machine. A feasible solution is shown in Figure 2. It can be observed that machines 1 and 2 are assigned to locations 1 and 3, respectively. Job 7 is rejected. Jobs 2, 1 and 3 are assigned to the machine 1 at location 1, while jobs 4, 6, 8 and 5 are assigned to the machine 2 at location 3. The expected profit is calculated as 13.8.

### 3.2. A new DRCC programming model

In the section, problem parameters and decision variables are first defined. Then, a DRCC programming model is constructed.

#### Input parameters

- $K$  Set of candidate locations, which is indexed by  $k$ ;
- $N$  Set of machines, which is indexed by  $n$ ;
- $J$  Set of jobs, which is indexed by  $i, j$ ;
- $\delta_k$  The capacity of location  $k \in K$ , which denotes the maximum number of machines that can be placed;
- $c_k^L$  The cost of selecting a location  $k \in K$ ;
- $c_{nk}^M$  The cost of placing machine  $n \in N$  at location  $k \in K$ ;
- $l_j$  Customer payment probability of job  $j \in J$ ;
- $e_j$  Rejection penalty of job  $j \in J$ ;
- $f_j$  Revenue for processing job  $j \in J$  without credit risk;
- $r_{jk}$  Release time of job  $j \in J$  to location  $k \in K$  that corresponds to its moving time from  $j$ 's storage area to location  $k$ ;
- $d_j$  Due date of job  $j \in J$ ;
- $p_j$  Uncertain processing time of job  $j \in J$ ;

- $\zeta$  The given profit level;
- $\alpha$  The risk of at least one job's completion time exceeds its due date;
- $\mathcal{H}$  Equals  $\max_{j \in J, k \in K} \{r_{jk}\} + \sum_{j \in J} p_j + \max_{j \in J} \{p_j\}$ , which is taken as a large enough positive number;
- $\mathcal{P}$  The moment-based ambiguity set, *i.e.*,

$$\mathcal{P} = \left\{ \mathbb{P} : \begin{array}{l} \mathbb{E}_{\mathbb{P}}[\mathbf{U}] = \boldsymbol{\mu}, \\ \mathbb{E}_{\mathbb{P}}[(\mathbf{U} - \boldsymbol{\mu})(\mathbf{U} - \boldsymbol{\mu})^T] = \boldsymbol{\Gamma} \end{array} \right\}$$

where  $\mathbf{U}$  is a column vector which collects uncertain parameters (*i.e.*,  $\mathbf{U} = [\mathbf{l}, \mathbf{p}]^T$ , where  $\mathbf{l} = [l_1, \dots, l_{|J|}]^T$  and  $\mathbf{p} = [p_1, \dots, p_{|J|}]^T$ ),  $\mathbb{P}$  denotes the unknown probability distribution of  $\mathbf{U}$ , and  $\mathbb{E}_{\mathbb{P}}[\cdot]$  indicates the expected value with probability distribution  $\mathbb{P}$ .  $\boldsymbol{\mu}$  and  $\boldsymbol{\Gamma}$  are the expectation and covariance matrix of  $\mathbf{U}$ , which are assumed to be known in advance.

*Decision variables*

- $z_k$  Equals 1 if location  $k \in K$  is selected, and 0 otherwise;
- $h_j$  Equals 1 if job  $j \in J$  is accepted to process, and 0 otherwise;
- $u_{nk}$  Equals 1 if machine  $n \in N$  is placed at location  $k \in K$ , and 0 otherwise;
- $x_{jnk}$  Equals 1 if job  $j \in J$  is assigned to machine  $n \in N$  at location  $k \in K$ , and 0 otherwise;
- $y_{ijnk}$  Equals 1 if job  $j \in J$  is processed after job  $i \in J$  on machine  $n \in N$  at location  $k \in K$ , and 0 otherwise;
- $C_j$  The completion time of job  $j \in J$ .

In the following, we present the DRCC programming model, named as **P1**.

$$[\mathbf{P1}]: \quad \max_{\mathbb{P} \in \mathcal{P}} \inf_{\mathbb{P} \in \mathcal{P}} \mathbb{P} \left\{ \begin{array}{l} \sum_{j \in J} f_j \cdot l_j \cdot h_j - \sum_{j \in J} e_j \cdot (1 - h_j) \\ - \sum_{k \in K} c_k^L \cdot z_k - \sum_{n \in N} \sum_{k \in K} c_{nk}^M \cdot u_{nk} \geq \zeta \end{array} \right\} \tag{1}$$

$$\text{s.t.} \quad \sum_{n \in N} u_{nk} \leq \delta_k \cdot z_k, \quad \forall k \in K \tag{2}$$

$$\sum_{k \in K} u_{nk} \leq 1, \quad \forall n \in N \tag{3}$$

$$\sum_{n \in N} \sum_{k \in K} x_{jnk} = h_j, \quad \forall j \in J \tag{4}$$

$$\sum_{j \in J} x_{jnk} \leq |J| \cdot u_{nk}, \quad \forall n \in N, k \in K \tag{5}$$

$$y_{ijnk} + y_{jink} \leq \frac{1}{2} \cdot (x_{ink} + x_{jnk}), \quad \forall i, j \in J, i \neq j, n \in N, k \in K \tag{6}$$

$$y_{ijnk} + y_{jink} \geq x_{ink} + x_{jnk} - 1, \quad \forall i, j \in J, i \neq j, n \in N, k \in K \tag{7}$$

$$\inf_{\mathbb{P} \in \mathcal{P}} \mathbb{P} \left\{ \begin{array}{l} \max_{i \in J, i \neq j, n \in N, k \in K} \left\{ \sum_{n \in N} \sum_{k \in K} r_{jk} \cdot x_{jnk} \right. \\ \left. + p_j, C_i + p_j - \mathcal{H} \cdot (1 - y_{ijnk}) \right\} \leq d_j, j \in J \\ \geq 1 - \alpha \end{array} \right\} \tag{8}$$

$$z_k, h_j, u_{nk}, x_{jnk}, y_{ijnk} \in \{0, 1\}, C_j \geq 0, \quad \forall i, j \in J, i \neq j, n \in N, k \in K. \tag{9}$$

The objective function (1) is to maximize the profit likelihood, *i.e.*, the worst-case probability of manufacturer's profit being greater than or equal to the given level  $\zeta$ . Notably, the manufacturer's profit consists of four parts, *i.e.*, the revenue by processing accepted jobs, the job rejection penalty, the cost of selecting locations, and the cost of assigning machines to these selected locations.

Constraint (2) denotes the number of machines placed at location  $k$  cannot exceed its capacity. Constraint (3) indicates a machine cannot be placed at multiple locations. Constraint (4) implies that if a job has been accepted for processing, it must be assigned to a machine. Constraint (5) denotes a machine can process jobs only when it has been placed at a location. Constraint (6) indicates there is no processing precedence relationship between job  $i$  and job  $j$  if they are not assigned to the same machine  $n$  at location  $k$ . Constraint (7) expresses that either job  $i$  is processed before job  $j$ , or job  $j$  is processed before job  $i$  provided that they are both assigned to the same machine  $n$  at location  $k$ . Constraint (8) ensures that the service likelihood is greater than or equal to a certain level  $1 - \alpha$ . Constraint (9) denotes the domain of decision variables.

### 3.3. The complexity analysis

In this subsection, we conduct a complexity analysis for the parallel machine ScheLoc problem with limited location capacity and customer credit risk.

**Theorem 1.** *The concerned parallel machine ScheLoc problem with limited location capacity and customer credit risk is NP-complete.*

*Proof.* The proof is based on the following fact that PARTITION reduces to the parallel machine ScheLoc problem with limited location capacity and customer credit risk. Given a set of integers  $\tau_1, \tau_2, \dots, \tau_{|J|}$  and an auxiliary integer  $\rho$ , such that  $\rho = \frac{1}{2} \sum_{j=1}^{|J|} \tau_j$ . The PARTITION problem is to decide whether the set can be partitioned into two subsets such that the sum of the integers of the two subsets are both equal to  $\rho$ . It is known that the PARTITION problem is NP-complete, which is a decision problem in terms of complexity theory.

In the following, we construct a special case of the studied problem (*i.e.*, a decision problem). The number of candidate locations, machines and jobs are set to be 2, 2,  $|J|$ , respectively. The capacities of both locations are set to be 1. The location selection cost, machine-to-location assignment cost, and job rejection penalty are uniformly set to be 0. The customer payment probability of each job and the revenue for processing each job are uniformly set to be 1. The release time of each job from its storage area to any location is set to be 0. The processing time of job  $j \in J$  is set to be  $\tau_j$ , and all jobs' due dates are uniformly set to be  $\rho$ . To sum up,

$$\delta_k = 1, c_k^I = 0, c_{nk}^M = 0, e_j = 0, l_j = 1, f_j = 1, r_{jk} = 0, p_j = \tau_j, d_j = \rho, \quad k \in \{1, 2\}, \quad j \in J.$$

In the case, a schedule with profit of  $|J|$  exists if and only if all jobs have been accepted and processed no later than the common due date  $\rho$ . This can be done if and only if the jobs can be partitioned into two sets with equal total processing time, which is the PARTITION problem.  $\square$

## 4. SOLUTION METHODS

Due to the objective function (1) in probabilistic form and DR chance Constraint (8), model **P1** cannot be directly solved *via* commercial solvers. Thus, we first propose a widely applied SAA method to solve the problem approximately. However, from the computation results in Section 5, we can see that no feasible solution can be found within the time limit for some large-scale instances. Therefore, we further propose a model-based constructive heuristic to solve the problem efficiently.

### 4.1. The SAA method

The principle of the SAA method is to replace the unknown probability distribution by an empirical distribution, which corresponds to a set of scenarios. In recent years, it has been widely applied to transform the DRCC programming model into a deterministic one [7, 18, 20]. Referring to these works, in the following, we apply this method to convert the model **P1** into a tractable model **P2**.



*New input parameters*

- $\Omega$  Set of scenarios, which is indexed by  $\omega$ ;
- $l_{j\omega}$  Customer payment probability of job  $j \in J$  in scenario  $\omega \in \Omega$ ;
- $p_{j\omega}$  Processing time of job  $j \in J$  in scenario  $\omega \in \Omega$ ;
- $\mathcal{H}_\omega$  Equals  $\max_{j \in J, k \in K} \{r_{jk}\} + \sum_{j \in J} p_{j\omega} + \max_{j \in J} \{p_{j\omega}\}$ , which is taken as a large enough positive number in scenario  $\omega \in \Omega$ .

*New decision variables*

- $m_\omega$  Equals 1 if manufacturer’s profit being greater than or equal to a given level  $\zeta$  in scenario  $\omega \in \Omega$ , and 0 otherwise;
- $\psi_{j\omega}$  The completion time of job  $j \in J$  in scenario  $\omega \in \Omega$ ;
- $q_\omega$  Equals 1 if at least one job’s completion time exceeds its due date in scenario  $\omega \in \Omega$ , and 0 otherwise.

$$[\mathbf{P2}]: \quad \max \frac{1}{|\Omega|} \sum_{\omega \in \Omega} m_\omega \tag{10}$$

$$\text{s.t.} \quad (2) - (7), (9)$$

$$\zeta - \left( \sum_{j \in J} f_j \cdot l_{j\omega} \cdot h_j - \sum_{j \in J} e_j \cdot (1 - h_j) - \sum_{k \in K} c_k^L \cdot z_k - \sum_{n \in N} \sum_{k \in K} c_{nk}^M \cdot u_{nk} \right) \leq \zeta \cdot (1 - m_\omega), \quad \forall \omega \in \Omega \tag{11}$$

$$\sum_{k \in K} \sum_{n \in N} r_{jk} \cdot x_{jnk} + p_{j\omega} \leq \psi_{j\omega}, \quad \forall j \in J, \omega \in \Omega \tag{12}$$

$$\psi_{i\omega} + p_{i\omega} - \mathcal{H}_\omega \cdot (1 - y_{ijnk}) \leq \psi_{j\omega}, \quad \forall i, j \in J, i \neq j, n \in N, k \in K \tag{13}$$

$$\psi_{j\omega} - \mathcal{H}_\omega \cdot q_\omega \leq d_j, \quad \forall j \in J, \omega \in \Omega \tag{14}$$

$$\sum_{\omega \in \Omega} q_\omega \leq |\Omega| \cdot \alpha \tag{15}$$

$$z_k, h_j, u_{nk}, x_{jnk}, y_{ijnk}, q_\omega, m_\omega \in \{0, 1\}, \psi_{j\omega}, C_j \geq 0, \quad \forall j \in J, \omega \in \Omega. \tag{16}$$

The objective function (10) is to maximize the proportion of scenarios in which the manufacturer’s profit is greater than or equal to the given level  $\zeta$ . Constraints (11) express whether the manufacturer’s profit being greater than or equal to a given level  $\zeta$  in scenario  $\omega \in \Omega$ . Constraints (12) and (13) calculate the completion time of job  $j \in J$  in scenario  $\omega \in \Omega$ . Constraints (14) indicate whether the completion time of each job is no later than its due date. Constraints (15) represent the number of scenarios in which at least one job’s completion time exceeds its due date is no greater than the give level (*i.e.*,  $|\Omega| \cdot \alpha$ ). Constraints (16) denote the domain of decision variables.

The SAA method transforms the original DRCC programming model **P1** into a scenario-based mixed-integer linear programming (MILP) model **P2**, which can be directly solved by commercial solvers such as Gurobi. However, the computation time of the method will grow exponentially with the size of the problem. Therefore, to solve the large-scale problem efficiently, we propose a model-based constructive heuristic.

### 4.2. The model-based constructive heuristic

In this subsection, we propose a new model-based constructive heuristic to solve the large-scale problem efficiently. Before presenting the heuristic, we first employ some common methods to transform the DRCC programming model **P1** into a deterministic mathematical programming model.

4.2.1. An equivalent transformation

In the first step, we introduce an auxiliary continuous decision variable  $\beta \in [0, 1]$  to equivalently transform the original objective function (1) into formulas (17)–(19) as follows.

$$\min \beta \tag{17}$$

$$\text{s.t. } \inf_{\mathbb{P} \in \mathcal{P}} \mathbb{P} \left\{ \sum_{j \in J} f_j \cdot l_j \cdot h_j - \sum_{j \in J} e_j \cdot (1 - h_j) - \sum_{k \in K} c_k^L \cdot z_k - \sum_{n \in N} \sum_{k \in K} c_{nk}^M \cdot u_{nk} \geq \zeta \right\} \geq 1 - \beta \tag{18}$$

$$\beta \in [0, 1]. \tag{19}$$

4.2.2. Bonferroni inequality

In the second step, we apply Bonferroni inequality to break the joint chance constraint (8) into a set of individual chance constraints. Specifically, joint chance constraint (8) can be conservatively approximated as individual chance constraints (20) and (21), where  $\alpha'_j$  and  $\alpha''_{ijnk}$  must satisfy  $\sum_{j \in J} \alpha'_j + \sum_{i \in J} \sum_{j \in J, j \neq i} \sum_{n \in N} \sum_{k \in K} \alpha''_{ijnk} = \alpha$  [4]. For convenience, we divide the risk tolerance  $\alpha$  evenly in line with Do Chung *et al.* [6], *i.e.*,  $\alpha'_j = \alpha''_{ijnk} = \frac{\alpha}{|J| + |J| \cdot (|J| - 1) \cdot |N| \cdot |K|}$ .

$$\inf_{\mathbb{P} \in \mathcal{P}} \mathbb{P} \left\{ \sum_{n \in N} \sum_{k \in K} r_{jk} \cdot x_{jnk} + p_j \leq d_j \right\} \geq 1 - \alpha'_j, \quad \forall j \in J \tag{20}$$

$$\inf_{\mathbb{P} \in \mathcal{P}} \mathbb{P} \left\{ C_i + p_j - \mathcal{H} \cdot (1 - y_{ijnk}) \leq d_j \right\} \geq 1 - \alpha''_{ijnk}, \quad \forall i, j \in J, i \neq j, n \in N, k \in K. \tag{21}$$

4.2.3. A deterministic mathematical programming model

In the third step, we refer to Theorem 2.2 in Wagner [23] to equivalently transform the DR individual chance constraints into deterministic constraints (*i.e.*, without probability). Before applying this theorem, we first introduce some new parameters.

New input parameters

- $\mu_j$  The expectation of  $l_j$ , where  $j \in J$ ;
- $\sigma_j^2$  The variance of  $l_j$ , where  $j \in J$ ;
- $\xi_j$  The expectation of  $p_j$ , where  $j \in J$ ;
- $\gamma_j^2$  The variance of  $p_j$ , where  $j \in J$ .

Referring to Theorem 2.2 in Wagner [23], DR individual chance constraints (18), (20) and (21) can be equivalently transformed into the following constraints (22), (23) and (24), respectively.

$$\sqrt{\sum_{j \in J} f_j^2 \cdot \sigma_j^2 \cdot h_j^2} \leq \sqrt{\frac{\beta}{1 - \beta} \left( - \sum_{j \in J} e_j \cdot (1 - h_j) - \sum_{k \in K} c_k^L \cdot z_k - \sum_{n \in N} \sum_{k \in K} c_{nk}^M \cdot u_{nk} - \zeta + \sum_{j \in J} f_j \cdot \mu_j \cdot h_j \right)}, \tag{22}$$

$$\gamma_j \leq \sqrt{\frac{\alpha'_j}{1 - \alpha'_j} \left( d_j - \sum_{n \in N} \sum_{k \in K} r_{jk} \cdot x_{jnk} - \xi_j \right)}, \quad \forall j \in J \tag{23}$$

$$\gamma_j \leq \sqrt{\frac{\alpha''_{ijnk}}{1 - \alpha''_{ijnk}} \left( d_j + \mathcal{H} \cdot (1 - y_{ijnk}) - C_i - \xi_j \right)}, \quad \forall i, j \in J, i \neq j, n \in N, k \in K. \tag{24}$$

After that, a new deterministic mathematical programming model **P3** is presented as follows.

$$[\mathbf{P3}]: \quad \min \quad \beta \quad (25)$$

$$\text{s.t.} \quad (2) - (7), (9), (19), (22) - (24). \quad (26)$$

Model **P3** still cannot be directly solved *via* commercial solvers due to the nonconvex constraint (22). However, note that this nonconvex constraint become a convex (*i.e.*, second-order conic) one if the decision variable  $\beta$  is fixed. Motivated by this fact, we design a constructive heuristic based on model **P3**.

#### 4.2.4. The constructive heuristic

The basic idea of the heuristic is trial-and-error. That is to say, given a certain  $\beta$ , we can check whether model **P3** is feasible by calling commercial solvers such as Gurobi. If the condition is true, the value of  $\beta$  is reduced *via* the golden-section search method, and *vice versa*. Besides, to offset the impact of conservative approximation caused by Bonferroni inequality, we enlarge the solution search space *via* modifying the preset profit level and job due date, *i.e.*,  $\zeta' = \zeta/\theta_1$  and  $d'_j = \theta_2 \cdot d_j$ , with tuning parameters  $\theta_1, \theta_2 > 1$ . The time complexity of this constructive heuristic is  $\mathbf{O}(\mathcal{M}\mathcal{N}\mathcal{T})$ , where  $\mathcal{M}$  denotes the maximum number of updates for  $\zeta$  and  $d_j$ ,  $\mathcal{N}$  represents the number of iterations for  $\beta$ , and we use  $\mathbf{O}(\mathcal{T})$  to denote the time complexity of checking the feasibility of **P3** with fixed  $\beta$  (Note that the feasibility version of model **P3** with fixed  $\beta$  is a decision problem in terms of complexity theory. Such a decision problem is NP-complete, because Constraints (24) are Knapsack-like constraints. Therefore, for simplicity, we use  $\mathbf{O}(\mathcal{T})$  to denote the time complexity of solving this decision problem). Algorithm 1 presents the details of the constructive heuristic.

---

**Algorithm 1:** The model-based constructive heuristic.

---

**Input:** Relevant problem parameters, model **P3**,

$\mathcal{M}$  (% The maximum number of updates for  $\zeta$  and  $d_j$ ),

$\mathcal{N}$  (% The number of iterations for  $\beta$ ).

```

1  $\beta^{(1)} = 0.618$  (% Initialize  $\beta$ );
2  $S = \{0, 1\}$  (% The set of trial  $\beta$ );
3  $S^* = \emptyset$  (% The set of  $\beta$  such that the feasibility version of model P3 is feasible);
4 for  $i = 1 : \mathcal{M}$  do
5    $\zeta' = \zeta/\theta_1$  and  $d'_j = \theta_2 \cdot d_j$ ;
6   for  $t = 1 : \mathcal{N}$  do
7      $S = S \cup \beta^{(t)}$ ;
8     Call Gurobi to solve model P3;
9     if model P3 is feasible then
10       $S^* = S^* \cup \beta^{(t)}$ ;
11       $\beta^{(t+1)} = \beta^{(t)} - 0.618 \cdot \left( \beta^{(t)} - \max_{s \in S} \{S(s) | S(s) < \beta^{(t)}\} \right)$ ;
12    else
13       $\beta^{(t+1)} = \beta^{(t)} + 0.618 \cdot \left( \min_{s \in S} \{S(s) | S(s) > \beta^{(t)}\} - \beta^{(t)} \right)$ ;
14    end
15  end
16  if  $S^*$  is not empty then
17    break;
18  end
19 end

```

**Output:** The minimum  $\beta$  in  $S^*$  and its corresponding solution.

---

## 5. NUMERICAL EXPERIMENTS

In this section, the illustrated example in Section 3 is first solved to demonstrate the validity of the SAA method and the constructive heuristic. Then, the performance of the two developed methods, and one adapted version of the best heuristic in stochastic ScheLoc area from the literature (*i.e.*, the scenario-based heuristic proposed by Liu *et al.* [19]), are further compared by 300 instances adapted from Heßler and Deghdak [8]. All numerical experiments are performed on a personal computer with Core I5, 2.11 GHz processor, 16 GB RAM, and Windows 10 Operating System, and the two solution methods are both coded in Python 3.6 combined with Gurobi 9.0. The computation time of each method is limited to 3600 s.

### 5.1. Evaluation standard

Solutions obtained by different methods are compared by the out-of-sample test in line with Liu *et al.* [20]. Specifically, 10 000 scenarios are first randomly generated, which represents various realizations of uncertain parameters. Then, the profit likelihood (denoted as  $\Upsilon_1$ ) and service likelihood (denoted as  $\Upsilon_2$ ) are measured as follows:

- (1) The profit likelihood  $\Upsilon_1$  is measured by  $(\eta_1/10,000)$ , where  $\eta_1$  denotes the number of scenarios in which manufacturer's profit no less than the given level;
- (2) The service likelihood  $\Upsilon_2$  is measured by  $(n_2/10,000)$ , where  $\eta_2$  denotes the number of scenarios in which no job's completion time exceeds its due date.

Accordingly, the performance of different methods is compared by the values of  $\Upsilon_1$  and  $\Upsilon_2$ .

### 5.2. The solutions obtained by the two methods for the example

In this part, the example in Section 3 is solved to demonstrate the effectiveness of the two methods in Section 4. The number of scenarios (*i.e.*,  $|\Omega|$ ) for the SAA method is set to be 30, and the values of parameters  $\mathcal{M}$  and  $\mathcal{N}$  for the constructive heuristic are set to be 5 and 10, respectively. The given profit level is set to be 6, and the risk tolerance  $\alpha$  is set to be 0.2.

The schedules obtained by the two methods are presented in Figure 3. From Figure 3, we can see that the selected locations of two methods are identical, *i.e.*, both locations 1 and 3 are selected. Recall that the capacity of each location is set to be 1. Therefore, two machines are assigned to two locations, respectively. For the SAA method, it can be observed that jobs 2 and 4 are rejected. Jobs 1, 5 and 3 are assigned to machine 1 at location 1, while jobs 6, 8 and 7 are distributed to machine 2 at location 3. In contrast, for the constructive heuristic, we can see that jobs 3 and 4 are rejected. Jobs 2, 1 and 5 are distributed to machine 1 at location 1, while jobs 6, 8 and 7 are assigned to machine 2 at location 3.

The expected profits of solutions obtained by the two methods are calculated as 6.7 and 10.6, respectively. Moreover, out-of-sample test shows the profit likelihood proposed by the constructive heuristic is  $(0.9259 - 0.5905)/0.5905 \times 100\% = 56.8\%$  higher than that provided by the SAA, while the service likelihood of the constructive heuristic is only  $(0.8537 - 0.8014)/0.8537 \times 100\% = 6.1\%$  lower than that of the SAA. Note that the service likelihood of the two methods are both higher than the given level (*i.e.*,  $1 - \alpha = 0.8$ ).

### 5.3. Numerical experiments on 300 instances

In this part, the performance of the two developed methods and the scenario-based heuristic from the literature are compared by 300 adapted instances. In the following, the instance generation rule is first illustrated. Then, computation results of different methods for these instances are analyzed.

#### 5.3.1. Instance generation rule

The instances are adapted from the large-scale data set for the ScheLoc problem provided by Heßler and Deghdak [8]. This data set includes the following information (i) the number of candidate locations; (ii) the number of machines; (iii) the number of jobs; (iv) job processing time; (v) job release time.

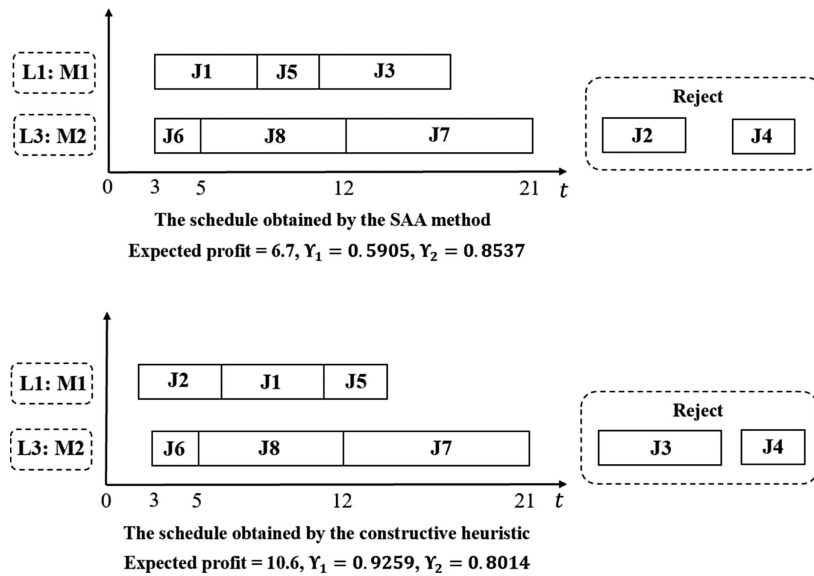


FIGURE 3. The schedules obtained by the two methods for the example in Section 3.1.

Since our work is an extension of the classic parallel machine ScheLoc problem, some necessary parameters are not included in the data set. Therefore, referring to the relevant literature and considering the characteristic of our addressed problem, we carefully generate the rest parameters. Specifically, customer payment probability is randomly generated following  $U[0, 1]$  [20]. Each job's due date is randomly generated following  $U\left[(1 - \psi_1 - \psi_2/2) \cdot \sum_{j \in J} \mathbb{E}(p_j), (1 - \psi_1 + \psi_2/2) \cdot \sum_{j \in J} \mathbb{E}(p_j)\right]$ , in which both  $\psi_1$  and  $\psi_2$  are adjustable parameters [17]. In this paper,  $\psi_1$  is set to be 0.2, and  $\psi_2$  is set to be 0.6. The given profit level  $\zeta$  is randomly generated following  $U\left[0.2 \cdot \sum_{j \in J} (f_j - e_j) \cdot l_j - \min_{k \in K} c_k^L, 0.4 \cdot \sum_{j \in J} (f_j - e_j) \cdot l_j - \min_{k \in K} c_k^L\right]$ , in which the job processing revenue without credit risk and the job rejection penalty are randomly generated following  $U[10, 15]$  and  $U[1, 5]$ , respectively [20], and the location selection cost is randomly generated following  $U[10, 50]$ . The capacity of each location is uniformly set to be 1, and the machine-to-location assignment cost is uniformly set to be 0. The standard deviation of each uncertain parameter is set to be 0.3 times its expectation [20].

Note that we adopt the first sixty combinations in Heßler and Deghdak [8]'s data set, and for each problem setting, we randomly generate five instances. Thus a total of 300 instances are tested.

### 5.3.2. Computation results

The computation results obtained by different methods for the 300 instances are shown in Table 2. Specifically, the first column denotes the sequence number of the tested problem setting, and the second column reports its size which includes the number of candidate locations, machines and jobs. The third to fifth columns give the profit likelihood, the service likelihood and the computation time of the SAA method, respectively. The sixth to eighth and the last three columns show the corresponding results of the scenario-based heuristic and the constructive heuristic, respectively. The average value of each indicator is reported in the last row of the table, which is calculated based on the instances for which all the methods can find feasible solutions within 3600 s.

From Table 2, we can see that the scenario-based heuristic and our proposed constructive heuristic can obtain solutions for all the combinations, while the SAA method can only obtain solutions for 53 of them. Moreover, the average computation time of the constructive heuristic is only  $51.4/1212.7 = 4.24\%$  and  $51.4/231.9 = 22.2\%$  of those needed by the SAA and the scenario-based heuristic, respectively. These results demonstrate the efficiency of the constructive heuristic for solving the studied problem.

TABLE 2. Computation results obtained by different methods.

NO.	( K ,  N ,  J )	The SAA			The scenario-based heuristic			The constructive heuristic		
		$\Upsilon_1$	$\Upsilon_2$	$T$ (s)	$\Upsilon_1$	$\Upsilon_2$	$T$ (s)	$\Upsilon_1$	$\Upsilon_2$	$T$ (s)
1	(13,4,59)	0.9640	0.7411	415.4	0.9768	0.4919	200.8	0.9973	0.9680	42.5
2	(10,6,55)	0.9262	0.7025	2253.9	0.9292	0.2771	8.7	0.9891	0.9534	40.7
3	(14,5,8)	0.9687	0.4084	94.9	0.9492	0.3886	226.9	0.9390	0.5284	3.7
4	(13,6,97)	0.7802	0.7820	3600.0	0.8304	0.3808	428.0	0.9028	0.9612	193.8
5	(15,10,46)	0.8082	0.8840	3600.0	0.9725	0.5755	305.2	0.9633	0.9814	74.6
6	(14,2,10)	0.9840	0.2305	148.8	0.8883	0.1681	150.2	0.9706	0.2351	4.1
7	(7,6,35)	0.9015	0.6987	501.3	0.9347	0.1846	198.4	0.9929	0.9861	11.7
8	(12,8,29)	0.9117	0.2304	2523.7	0.9050	0.2285	303.4	0.9911	0.2510	129.5
9	(7,4,77)	0.8880	0.9923	1527.0	0.9495	0.6497	97.3	0.9837	0.9970	42.4
10	(6,2,68)	0.8366	0.8672	815.2	0.9047	0.4066	293.9	0.9730	0.9738	14.9
11	(6,2,39)	0.9992	0.0525	321.0	0.9866	0.0925	172.0	0.9807	0.1027	5.4
12	(15,10,31)	0.9473	0.7335	100.9	0.8839	0.5442	382.9	0.9015	0.9558	38.9
13	(12,10,49)	0.7934	0.3892	77.2	0.9184	0.1053	484.9	0.9115	0.6344	68.1
14	(9,7,95)	0.6403	0.9450	3600.0	0.9335	0.2432	154.2	0.9226	0.9592	157.1
15	(12,3,95)	0.9337	0.9150	1470.4	0.8856	0.4568	525.6	0.9229	0.9745	83.1
16	(8,5,79)	0.8635	0.6877	1897.4	0.9108	0.1985	321.8	0.9003	0.9563	64.0
17	(12,10,51)	0.9802	0.7826	576.2	0.9211	0.0903	56.7	0.9615	0.9807	75.7
18	(9,4,4)	0.8017	0.8262	93.6	0.9343	0.1793	330.5	0.9194	0.9780	3.5
19	(12,9,50)	0.8816	0.4933	3600.0	0.9626	0.3111	108.0	0.9906	0.4257	67.9
20	(6,4,15)	0.8511	0.4221	144.1	0.8893	0.4964	185.8	0.9238	0.5824	2.6
21	(15,3,61)	0.9014	0.2627	720.7	0.9791	0.0884	348.1	0.9947	0.3961	42.5
22	(5,2,94)	0.8450	0.8549	1920.6	0.9977	0.5744	310.0	0.9643	0.9976	37.3
23	(7,3,58)	0.8007	0.1764	198.7	0.8953	0.0607	201.1	0.9185	0.2013	15.7
24	(14,6,51)	0.8963	0.4732	355.3	0.9910	0.5384	63.3	0.9866	0.5093	49.6
25	(13,6,29)	0.9313	0.6231	1816.8	0.8159	0.9347	62.1	0.8999	0.8796	93.3
26	(6,2,62)	0.8176	0.7648	536.5	0.9581	0.1947	414.7	0.9652	0.9389	10.7
27	(14,7,46)	0.8822	0.5128	2046.8	0.8915	0.3475	239.7	0.9099	0.5066	45.8
28	(6,4,32)	0.9314	0.8700	17.2	0.8895	0.5343	103.2	0.9606	0.9998	4.1
29	(15,3,21)	0.9024	0.6337	488.2	0.9265	0.7588	504.2	0.9919	0.8348	16.2
30	(15,10,31)	0.9253	0.8642	521.3	0.8650	0.3016	262.9	0.9456	0.9996	69.7
31	(12,3,45)	0.8612	0.8864	506.8	0.9425	0.9328	191.6	0.9522	0.9932	15.3
32	(15,2,87)	0.8553	0.7551	1275.7	0.8434	0.3662	274.6	0.9038	0.9987	68.7
33	(13,2,20)	0.9146	0.1102	4.5	0.9340	0.1745	83.3	0.9581	0.1024	5.7
34	(8,3,74)	0.8972	0.6933	1169.2	0.9081	0.1833	118.0	0.9120	0.9675	40.3
35	(15,6,44)	0.9816	0.8927	119.5	0.9888	0.8055	406.6	0.9608	0.9355	38.3
36	(14,8,56)	0.9262	0.7691	3309.9	0.8922	0.8353	96.3	0.9310	0.9998	90.4
37	(10,6,46)	0.9236	0.7657	412.7	0.9464	0.1282	198.8	0.9524	0.9391	29.8
38	(15,8,71)	—	—	3600.0	0.8864	0.5959	433.9	0.8975	0.8361	178.2
39	(8,5,100)	0.7370	0.7519	3600.0	0.9153	0.4491	516.8	0.9567	0.9193	102.9
40	(6,4,16)	0.8831	0.1963	134.2	0.9502	0.1022	144.7	0.9794	0.2375	2.7
41	(5,2,38)	0.7993	0.8533	42.0	0.9005	0.4753	86.4	0.9710	0.9556	2.7
42	(7,6,46)	0.9544	0.9148	989.4	0.8546	0.9582	311.1	0.9249	0.9196	19.0
43	(8,5,69)	0.7169	0.9921	2371.0	0.8816	0.0944	111.0	0.9317	0.9924	45.6
44	(13,3,98)	0.8501	0.1481	1693.5	0.9277	0.1978	593.5	0.9909	0.1203	123.8
45	(9,8,68)	0.8012	0.7421	1650.9	0.9559	0.1782	340.6	0.9870	0.9961	83.9
46	(10,7,40)	0.7945	0.8116	718.9	0.9367	0.6528	85.5	0.9843	0.9997	24.5
47	(13,7,59)	0.8072	0.8962	2621.4	0.9818	0.2582	122.1	0.9913	0.9090	75.6
48	(9,8,47)	0.8145	0.3364	1087.2	0.9302	0.3042	92.7	0.8970	0.4219	35.4
49	(9,6,50)	0.8521	0.5621	1905.5	0.9387	0.1475	111.1	0.9296	0.6377	33.8

**Notes.** “—” represents no feasible solution can be found within 3600 s;  $\Upsilon_1$  represents the profit likelihood;  $\Upsilon_2$  represents the service likelihood;  $T$  represents the computation time.

TABLE 2. continued.

NO.	( K ,  N ,  J )	The SAA			The scenario-based heuristic			The constructive heuristic		
		$\Upsilon_1$	$\Upsilon_2$	$T$ (s)	$\Upsilon_1$	$\Upsilon_2$	$T$ (s)	$\Upsilon_1$	$\Upsilon_2$	$T$ (s)
50	(10,3,51)	0.8466	0.8082	181.7	0.8344	0.9899	309.0	0.8843	0.9919	17.1
51	(16,14,21)	–	–	3600.0	0.8796	0.7499	299.2	0.9129	0.8328	258.3
52	(22,18,66)	–	–	3600.0	0.8675	0.7031	270.8	0.8855	0.9876	472.8
53	(22,16,26)	0.8516	0.7401	2776.3	0.9905	0.5042	89.3	0.9881	0.9430	54.3
54	(24,12,45)	–	–	3600.0	0.9487	0.3612	542.8	0.9736	0.9310	146.9
55	(17,14,90)	–	–	3600.0	0.9519	0.6547	778.4	0.9950	0.9609	574.6
56	(17,14,30)	0.7451	0.7562	729.0	0.8237	0.8923	24.6	0.8411	0.9975	50.1
57	(20,18,58)	–	–	3600.0	0.9040	0.2969	725.2	0.9369	0.3685	330.0
58	(19,10,57)	0.8955	0.8821	859.6	0.9842	0.3984	378.9	0.9948	0.9808	166.4
59	(19,6,57)	0.8161	0.9564	130.6	0.9279	0.5976	161.9	0.9616	0.9910	87.3
60	(22,19,28)	–	–	3600.0	0.8414	0.7732	924.6	0.8237	0.9904	529.4
	Average	0.8683	0.6611	1212.7	0.9220	0.4043	231.9	0.9502	0.7849	51.4

Focusing on the out-of-sample performance of the three methods, it can be observed that the average profit likelihood (*i.e.*,  $\Upsilon_1$ ) and the service likelihood (*i.e.*,  $\Upsilon_2$ ) of the constructive heuristic are  $(0.9502 - 0.8683)/0.8683 = 9.43\%$  and  $(0.7849 - 0.6611)/0.6611 = 18.73\%$  higher than those of the SAA method, respectively, and  $(0.9502 - 0.9220)/0.9220 = 3.06\%$  and  $(0.7849 - 0.4043)/0.4043 = 94.14\%$  superior to those provided by the scenario-based heuristic, respectively. Based on these results, we conclude that the model-based constructive heuristic performs better than the SAA method and the scenario-based heuristic.

## 6. CONCLUSION

In this work, we study a new and general stochastic parallel machine ScheLoc problem with limited location capacity and customer credit risk. The problem consists of determining machine-to-location assignment, job acceptance, job-to-machine assignment, and scheduling of accepted jobs on each machine. The objective is to maximize the profit likelihood. For the problem, a DRCC programming model is established. Then, a SAA method and a model-based constructive heuristic are proposed. An illustrated example demonstrates the effectiveness of the model, and computation results of 300 adapted instances show the efficiency of the constructive heuristic.

Several research directions can be explored in the future. Firstly, the ScheLoc problem can be investigated in a more complex environment, by considering uncertain job release time and job due date, for example. Secondly, in this study, the DR joint chance-constraint is conservatively approximated as a set of individual chance constraints *via* Bonferroni inequality. It is a worthy topic of developing new techniques that can equivalently rewrite the DR joint chance-constraint. Thirdly, the characteristics of stochastic ScheLoc problems can be further explored such that some valid inequalities can be derived to speed up the solution procedure. Finally, it is an interesting topic to adapt the existing algorithms developed for the classical parallel machine ScheLoc problem (such as LBB and ILS) to solve our studied problem.

*Acknowledgements.* The authors thank the editor and the three anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 72021002, 71771048, 71832001, 72071144 and 71871159.

## REFERENCES

- [1] B. Basciftci, S. Ahmed and S. Shen, Distributionally robust facility location problem under decisiondependent stochastic demand. *Eur. J. Oper. Res.* **292** (2021) 548–561.

- [2] V. Belvedere, A. Grando and P. Bielli, A quantitative investigation of the role of information and communication technologies in the implementation of a product-service system. *Int. J. Prod. Res.* **51** (2013) 410–426.
- [3] M.L. Bentaha, A. Dolgui, O. Battaï, R.J. Riggs and J. Hu, Profit-oriented partial disassembly line design: dealing with hazardous parts and task processing times uncertainty. *Int. J. Prod. Res.* **56** (2018) 7220–7242.
- [4] G.C. Calafiore and L. El Ghaoui, On distributionally robust chance-constrained linear programs. *J. Optim. Theory Appl.* **130** (2006) 1–22.
- [5] E. Delage and Y. Ye, Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Oper. Res.* **58** (2010) 595–612.
- [6] B. Do Chung, T. Yao and B. Zhang, Dynamic traffic assignment under uncertainty: a distributional robust chance-constrained approach. *Netw. Spat. Econ.* **12** (2012) 167–181.
- [7] L. Han, P. Wu and C. Chu, Service-oriented distributionally robust lane reservation. *J. Ind. Inf. Integr.* **25** (2022) 100302.
- [8] C. Heßler and K. Deghdak, Discrete parallel machine makespan ScheLoc problem. *J. Comb. Optimiz.* **34** (2017) 1159–1186.
- [9] H. Hennes and H.W. Hamacher, *Integrated scheduling and location models: single machine makespan problems*. Technical Report, University of Kaiserslautern, Shaker Verlag, Aachen (2002) 82.
- [10] M.T. Kalsch, *Scheduling-Location (ScheLoc) Models, Theory and Algorithms*. Verlag Dr. Hut (2009).
- [11] M.T. Kalsch and Z. Drezner, Solving scheduling and location problems in the plane simultaneously. *Comput. Oper. Res.* **37** (2010) 256–264.
- [12] R. Kramer and A. Kramer, An exact framework for the discrete parallel machine scheduling location problem. *Comput. Oper. Res.* **132** (2021) 105318.
- [13] Y. Li, J.F. Ct, L. Callegari-Coelho and P. Wu, Novel formulations and logic-based benders decomposition for the integrated parallel machine scheduling and location problem. *INFORMS J. Comput.* **34** (2021) 1048–1069.
- [14] Y. Li, X. Li, J. Shu, M. Song and K. Zhang, A general model and efficient algorithms for reliable facility location problem under uncertain disruptions. *INFORMS J. Comput.* **34** (2022) 407–426.
- [15] Y. Liao, F. Deschamps, E.d.F.R. Loures and L.F.P. Ramos, Past, present and future of industry 4.0 – a systematic literature review and research agenda proposal. *Int. J. Prod. Res.* **55** (2017) 3609–3629.
- [16] M. Liu and X. Liu, Distributionally robust parallel machine ScheLoc problem under service level constraints. *IFAC-PapersOnLine* **52** (2019) 875–880.
- [17] M. Liu, S. Wang, C. Chu and F. Chu, An improved exact algorithm for single-machine scheduling to minimise the number of tardy jobs with periodic maintenance. *Int. J. Prod. Res.* **54** (2016) 3591–3602.
- [18] M. Liu, X. Liu, F. Chu, F. Zheng and C. Chu, Service-oriented robust parallel machine scheduling. *Int. J. Prod. Res.* **57** (2019) 3814–3830.
- [19] M. Liu, X. Liu, E. Zhang, F. Chu and C. Chu, Scenario-based heuristic to two-stage stochastic program for the parallel machine ScheLoc problem. *Int. J. Prod. Res.* **57** (2019) 1706–1723.
- [20] M. Liu, X. Liu, F. Chu, F. Zheng and C. Chu, Profit-oriented distributionally robust chance constrained flowshop scheduling considering credit risk. *Int. J. Prod. Res.* **58** (2020) 2527–2549.
- [21] H. Scarf, A min-max solution of an inventory problem, in *Stud. Mathematical Theory of Inventory and Production*. Stanford University Press, Redwood City, CA (1958) 201–209.
- [22] C. Tsosie and S. Nicastro. <https://www.nerdwallet.com/blog/small-business/businesscredit-score-basics/> (2017).
- [23] M.R. Wagner, Stochastic 01 linear programming under limited distributional information. *Oper. Res. Lett.* **36** (2008) 150–156.
- [24] S. Wang, J. Wan, D. Li and C. Zhang, Implementing smart factory of industrie 4.0: an outlook. *Int. J. Distrib. Sens. Netw.* **12** (2016) 3159805.
- [25] S. Wang, R. Wu, F. Chu, J. Yu and X. Liu, An improved formulation and efficient heuristics for the discrete parallel-machine makespan ScheLoc problem. *Comput. Ind. Eng.* **140** (2020) 106238.
- [26] J.C. Yepes-Borrero, F. Perea, R. Ruiz and F. Villa, Bi-objective parallel machine scheduling with additional resources during setups. *Eur. J. Oper. Res.* **292** (2021) 443–455.
- [27] P. Yunusoglu and S. Topaloglu Yildiz, Constraint programming approach for multi-resourceconstrained unrelated parallel machine scheduling problem with sequence-dependent setup times. *Int. J. Prod. Res.* **60** (2022) 2212–2229.
- [28] C. Zhang, Y. Li, J. Cao, Z. Yang and L.C. Coelho, Exact and matheuristic methods for the parallel machine scheduling and location problem with delivery time and due date. *Comput. Oper. Res.* **147** (2022) 105936.

**Please help to maintain this journal in open access!**



This journal is currently published in open access under the Subscribe to Open model (S2O). We are thankful to our subscribers and supporters for making it possible to publish this journal in open access in the current year, free of charge for authors and readers.

Check with your library that it subscribes to the journal, or consider making a personal donation to the S2O programme by contacting [subscribers@edpsciences.org](mailto:subscribers@edpsciences.org).

More information, including a list of supporters and financial transparency reports, is available at <https://edpsciences.org/en/subscribe-to-open-s2o>.