Taylor & Francis
Taylor & Francis Group

International Journal of Production Research

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

Scenario-based heuristic to two-stage stochastic program for the parallel machine ScheLoc problem

Ming Liu, Xin Liu, E. Zhang, Feng Chu & Chengbin Chu

To cite this article: Ming Liu, Xin Liu, E. Zhang, Feng Chu & Chengbin Chu (2019) Scenario-based heuristic to two-stage stochastic program for the parallel machine ScheLoc problem, International Journal of Production Research, 57:6, 1706-1723, DOI: 10.1080/00207543.2018.1504247

To link to this article: https://doi.org/10.1080/00207543.2018.1504247

Published online: 01 Nov 2018.

International Journal of Production Research, 2019
Vol. 57, No. 6, 1706–1723, https://doi.org/10.1080/00207543.2018.1504247

Scenario-based heuristic to two-stage stochastic program for the parallel machine ScheLoc problem

Ming Liu[a], Xin Liu[a], E. Zhang[b*], Feng Chu[c,d] and Chengbin Chu[a,e]

[a]*School of Economics & Management, Tongji University, Shanghai, People's Republic of China;* [b]*School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, People's Republic of China;* [c]*IBISC, Univ Évry, University of Paris-Saclay, France;* [d]*Management Engineering Research Center, Xihua University, Chengdu, People's Republic of China;* [e]*Systems Engineering Department, Université Paris-Est, ESIEEParis, Noisy-le-Grand Cedex, France*

(*Received 13 December 2017; accepted 30 June 2018*)

Scheduling-Location (ScheLoc) problem is a new and interesting topic in manufacturing, considering location and scheduling decisions simultaneously. Most existing works focus on the deterministic problems. In practice, however, job-processing times are usually uncertain due to some factors. This paper investigates the stochastic parallel machine ScheLoc problem to minimise the weighted sum of the location cost and the expectation of the total completion time. A two-stage stochastic programming formulation is proposed, then the sample average approximation (SAA) method is adapted to solve the small-size problems. To efficiently address the large-scale problems, a genetic algorithm (GA) and a scenario-based heuristic are designed. Numerical experiments on 450 instances are conducted. Computational results show that the scenario-based heuristic outperforms SAA method and GA in terms of solution quality and computational time.

Keywords: ScheLoc problem; uncertain processing times; two-stage stochastic integer program; scenario-based heuristic

1. Introduction

Scheduling-Location (ScheLoc) problem is a relatively new and important issue in manufacturing industry, which can offer great potential to reduce system cost consisting of facility or factory location cost and job-processing cost. The problem mainly deals with the selection of machine locations and the schedule of jobs, in which the release time of each job depends on the distance between the job storage location and its processing machine, in order to optimise the objective under some certain constraints (Heßler and Deghdak 2015).

ScheLoc problem has a wide range of applications, e.g. in production planning and logistics. One of the applications comes from a container harbour, where the containers should be loaded on ships (Kalsch and Drezner 2010). In this application, we should determine the best positions for ships on the berth (location problem) and decide the sequence for loading containers (scheduling problem). Another application is from the mining industry, where the minerals should be moved to the crushing machines. The decisions on the positions of crushing machines and schedule of minerals should be made. Moreover, the usage of movable machines in production system can also be considered as an application of ScheLoc problem (Kalsch 2009).

In ScheLoc problem, the location problem is of tactical level and the scheduling problem is of the operational level. There is a common assumption in various existing researches that these two problems are separate. However, the optimal solution to the integrated problem may not include the optimal solutions of the two component problems. Due to the interrelationship between the location problem and the scheduling problem, it is necessary and meaningful to study integrated methods to solve the problems simultaneously. The location problem and the scheduling problem are NP-hard, respectively, thus the ScheLoc problem is even more difficult.

Most of the existing works on the parallel machine scheduling problem assume that the machine locations are fixed (e.g. Lei 2010; Yin et al. 2014a, 2014b, 2015, 2017, etc.). Besides, the related researches studying ScheLoc problem mainly focus on the deterministic single machine and the parallel machine ScheLoc problem. However, in practice, the job-processing times are usually uncertain due to that (i) uncertain machine maintenance may affect job-processing times (Wang and Liu 2013, 2014; Liu et al. 2016), (ii) the damage to jobs during transportation from the storage location to the corresponding machines is unpredictable, (iii) the effectiveness of operators may vary according to the work rate, skill level and motivation

*Corresponding author. Email: eezhang@163.com

(Battaïa and Dolgui 2013). Motivated by the above observation, this paper considers a parallel machine ScheLoc problem with uncertain job-processing times. For the studied problem, a two-stage stochastic programming formulation is proposed, and a genetic algorithm (GA) and a scenario-based heuristic are developed. The contribution of this work mainly includes:

(1) We consider the stochastic optimisation for the parallel machine ScheLoc problem with uncertain job-processing times, and a new two-stage stochastic programming formulation is proposed.
(2) Based on the given historical data on job-processing times under different scenarios, the sample average approximation (SAA) method is adapted to solve the small-scale instances.
(3) In order to efficiently solve the large-scale instances, based on the characteristics of the problem, a GA and a scenario-based heuristic are then developed.

The remainder of this paper is organised as follows. Section 2 gives a brief literature review. In Section 3, problem description is given and a new two-stage stochastic programming formulation is proposed. In Section 4, an adapted SAA method, a GA and a scenario-based heuristic are developed. Computational results on benchmark instances are reported in Section 5. Section 6 summarises this work and suggests future research directions.

2. Literature review

The scheduling problem has been well studied in the literature (e.g. Yin et al. 2012; Wang et al. 2013; Gafarov, Dolgui, and Werner 2014; Yin et al. 2014b; Jia, Jiang, and Li 2015; Zeppetella et al. 2016; Zhu, Zheng, and Chu 2017, etc.). Since our study falls within the scope of stochastic parallel machine ScheLoc problem, in the following, we mainly review the existing literature on ScheLoc problem. Then we review the two-stage stochastic program for parallel scheduling problems.

Hamacher and Hennes (2002) first considers scheduling and location decisions simultaneously, in which the candidate machine locations are in the network and the location of a single machine defines release times of jobs. Integrated models are proposed, and based on earliest release date (ERD) rule, polynomial-time algorithms are developed for some special cases. Elvikis, Hamacher, and Kalsch (2007) study the single machine makespan ScheLoc problem in the planar space, where machines can be located anywhere in a given planar region. In the paper, release dates are represented by a distance function and some geometrical properties are proposed, an algorithm based on ERD rule is then proposed. Then based on that, Elvikis, Hamacher, and Kalsch (2009) investigate a single machine ScheLoc problem in the planar space to minimise the makespan. Based on the geometrical properties, i.e. bisectors and ordered regions, two global search algorithms and a local search algorithm are developed. Kalsch and Drezner (2010) investigate the single machine ScheLoc problem in the planar region and analyse two objectives, i.e. the makespan and the total completion time, respectively. Basic properties of models are described and lower bounds on the objectives are developed, and the ‘big triangle and small triangle’ branch-and-bound global optimisation algorithm is proposed.

Heßler and Deghdak (2015) first investigate the discrete parallel ScheLoc problem, where the candidate locations for machines are discrete. A mixed integer programming (MIP) formulation is proposed, and different versions of clustering heuristics, where jobs are split into clusters and each cluster is assigned to one of the candidate machine locations, are then developed for the problem. Rajabzadeh, Ziaee, and Bozorgi-Amiri (2016) study the parallel machine makespan ScheLoc problem in discrete and continuous spaces, and an MIP formulation and a linear programming (LP) formulation are proposed.

To better understand the current research status, the comparison between related works is reported in Table 1. We can obtain that most existing researches mainly address the ScheLoc problem in deterministic environments with perfect and complete information.

As stated above, job-processing times are usually uncertain (e.g. Choi and Wang 2012; Lu, Ying, and Lin 2014; Wang and Choi 2014; Ying 2015; Zheng et al. 2018, etc.). A popular approach addressing the stochastic optimisation problem is two-stage stochastic programming approach. Al-Khamis and M’Hallah (2011) propose a two-stage stochastic programming model for the parallel machine scheduling problem with uncertain job due dates, to determine the machines’ capacities to maximise the expected net profit of on-time jobs. An adapted SAA method is proposed.

Concluding, to the best of our knowledge, there is no result on parallel machine ScheLoc problem with uncertain job-processing times.

3. Problem description and formulation

In this section, the problem is firstly described in detail in the following, and then a two-stage stochastic programming formulation is proposed.

Table 1. Comparison of researches studying ScheLoc problems.

Literature	Problem	Environment setting: Deterministic	Environment setting: Stochastic	Objective	Approach
Hamacher and Hennes (2002)	Single machine ScheLoc on a network	✓	–	Minimising the makespan	Algorithm based on ERD rule
Elvikis et al. (2007)	Single machine ScheLoc on the planar space	✓	–	Minimising the makespan	Algorithm based on bisectors and ordered regions and ERD rule
Elvikis et al. (2009)	Single machine ScheLoc on the planar space	✓	–	Minimising the makespan	Global search algorithms, Local search algorithm based on bisectors and ordered regions and and ERD rule
Kalsch and Drezner (2010)	Single machine ScheLoc on the planar space	✓	–	Minimising (i) the makespan, (ii) the total completion time	the 'big triangle and small triangle' branch-and-bound algorithm
Heßler and Deghdak (2015)	Discrete parallel machine ScheLoc (candidate machine locations are discrete)	✓	–	Minimising the makespan	Clustering heuristics, where jobs are clustered first and clusters are assigned to the candidate machine locations
Rajabzadeh et al. (2016)	Parallel machine in discrete and continuous spaces	✓	–	Minimising the makespan	MIP formulation and LP formulation
This paper	Stochastic (discrete) parallel machine ScheLoc	–	✓	Minimising the weighted sum of the location cost and the expectation of the total completion time	SAA method, Scenario-based heuristic, GA

3.1. *Problem description*

For the addressed problem, there are a set of discrete candidate machine locations, i.e. $M = \{1, \ldots, m\}$, from which we can choose at most K locations for machines, where $K \leq m$. There is at most one machine placed at a candidate location, and one machine cannot be placed at two or more candidate locations. There is a fixed cost c_k for placing a machine at location $k \in M$. Besides, there is a set of jobs to be processed, i.e. $I = \{1, \ldots, n\}$. Jobs are stored at the so-called storages in the beginning and should be transited to the corresponding machines for processing. Therefore, the release date r_{ik} of job i on the machine at location k is related to the distance $d(i,k)$ between them and calculated as $r_{ik} = 1/v_{ik} \cdot d(i,k)$, where v_{ik} denotes the speed of transportation vehicles. Throughout this paper, it is assumed that $v_{ik} = 1$, in line with Heßler and Deghdak (2015). During processing, preemption is not allowed, and each machine can process at most one job at one time.

Given a set of samples (i.e. scenarios) including data of job-processing times, this problem is to decide machine locations before the realisations of uncertainties and schedule jobs with full stochastic information such that the weighted sum of the fixed cost and the expectation of total completion time is minimised.

3.2. *Two-stage stochastic program*

We formulate the addressed problem as a two-stage stochastic program. The decision on machine locations is regarded as the first-stage decision, which must be made before therealisation of uncertain job-processing times. After the first stage, full information of job-processing times is obtained, then the second-stage decision on the schedule of jobs should be made, which is also called the recourse decision.

In the following, we give basic notations, define decision variables, and present the two-stage stochastic programming formulation for the parallel machine ScheLoc problem with uncertain job-processing times.

Parameters:

i, j, l: index of jobs.
k: index of candidate machine locations.
ω: index of scenarios.
I: set of jobs, i.e. $I = \{1, \ldots, n\}$.
J: $J = I \cup \{0\}$, where job $\{0\}$ is the dummy job serving as the predecessor of the first job and the successor of the last job processed on a machine. The release time and the processing time of the dummy job equal 0.

M: set of candidate machine locations, i.e. $M = \{1, \ldots, m\}$.
Ω: set of scenarios, which consists of a finite number of scenarios.
K: the maximum number of locations allowed to be chosen.
$p_i(\omega)$: the processing time of job i under scenario $\omega \in \Omega$, we define vector $\boldsymbol{p(\omega)} = \{\boldsymbol{p_1(\omega)}, \boldsymbol{p_2(\omega)}, \ldots, \boldsymbol{p_n(\omega)}\}$ for simplicity.
r_{ik}: the release time of job i on a machine at location k, if any.
c_k: the fixed cost for locating a machine at location k, if any.
θ_1, θ_2: the weight coefficients of the fixed cost and the expectation of total completion time in the objective function.
M: a large enough number.

Decision variables:

z_k: a binary variable equal to 1 if a machine is placed at candidate location $k \in M$; 0 otherwise. We define $\boldsymbol{z} = \{\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_m}\}$.
$x_{ij}^k(\omega)$: a binary variable equal to 1 if job i directly precedes job j on the machine at location k under scenario $\omega \in \Omega$, if any; 0 otherwise.
$C_i(\omega)$: the completion time of job i under scenario $\omega \in \Omega$.

Two-stage stochastic programming formulation (P1):

$$\min \quad g_{\text{stoch}} = \left\{ \theta_1 \cdot \sum_{k \in M} c_k \cdot z_k + \theta_2 \cdot \mathbb{E}_{p(\omega)} \sum_{i \in I} C_i(\omega) \right\}$$

$$s.t. \quad \sum_{k \in M} z_k \le K \tag{1}$$

$$\sum_{k \in M} \sum_{j \in J, j \ne i} x_{ij}^k(\omega) = 1, \quad \forall i \in I, \omega \in \Omega \tag{2}$$

$$\sum_{j \in I} x_{0j}^k(\omega) \le z_k, \quad \forall k \in M, \omega \in \Omega \tag{3}$$

$$\sum_{l \in J, l \ne i} x_{il}^k(\omega) - \sum_{l \in J, l \ne i} x_{li}^k(\omega) = 0, \quad \forall i \in J, k \in M, \omega \in \Omega \tag{4}$$

$$C_i(\omega) \ge \sum_{k \in M} r_{ik} \cdot \sum_{j \in J, j \ne i} x_{ij}^k(\omega) + p_i(\omega), \quad \forall i \in I, \omega \in \Omega \tag{5}$$

$$C_i(\omega) \ge C_j(\omega) + p_i(\omega) - M \cdot (1 - x_{ji}^k(\omega)), \quad \forall i \in I, j \in J, j \ne i, \; k \in M, \omega \in \Omega \tag{6}$$

$$C_0(\omega) = 0, \quad \forall \omega \in \Omega \tag{7}$$

$$C_i(\omega) \ge 0, \quad \forall i \in I, \omega \in \Omega \tag{8}$$

$$x_{ij}^k(\omega), z_k \in \{0,1\}, \quad \forall i,j \in J, k \in M, \omega \in \Omega. \tag{9}$$

The objective function g_{stoch} is to minimise the weighted sum of (i) the location cost, i.e. $\sum_{k \in M} c_k \cdot z_k$ and (ii) the expectation of the total completion time, i.e. $\mathbb{E}_{\boldsymbol{p(\omega)}} \sum_{k \in M} C_i(\omega)$.

Constraint (1) restricts that the number of selected locations for machines is no more than K. Constraint (2) states that each job is assigned to one machine and has only one successor under scenario $\omega \in \Omega$. Constraint (3) ensures the opened stations. Constraint (4) ensures that if job $i \in I$ is assigned to machine at location $k \in M$ under scenario $\omega \in \Omega$, there is a predecessor and a successor of job i processed on the machine. Constraint (5) implies that no job is started before its release date under each scenario $\omega \in \Omega$. Constraint (6) restricts that no job can be started before the completion time of its predecessor under each scenario $\omega \in \Omega$, which is linearised by the big-M method. Constraints (7)–(9) are the restrictions on decision variables.

When the number of scenarios equals 1, this stochastic problem corresponds to a deterministic problem. For the problem in deterministic environment, we provide the tradeoff between the first-stage and second-stage objectives in Appendix.

4. Solution approaches

Since variables x_{ij}^k are required to be binary, the second-stage recourse decision is complex. In this section, in order to handle the difficulties in solving the two-stage stochastic optimisation program, different solution approaches are proposed. Firstly, an SAA method with CPLEX is adapted. However, we obtain that CPLEX loses its power to solve large-scale instances. Therefore, to efficiently deal with large-scale problems, we then develop a GA and a scenario-based heuristic. The solution approaches are detailed in the following.

4.1. *SAA method*

SAA method is a popular and easy to implement approach to address stochastic problems, and it performs well under sufficient number of scenarios (Wang and Ahmed 2008). SAA is an approach for solving stochastic optimisation problems by deterministic optimisation techniques using Monte Carlo simulation. In SAA, the expected objective of the original stochastic problem is approximated by a sample average estimate derived from a set of samples (Kleywegt, Shapiro, and Homem-De-Mello 2001; Verweij et al. 2003). This method has been widely applied to solve the two-stage stochastic optimisation problems (Pagnoncelli, Ahmed, and Shapiro 2009; Wang and Ahmed 2008).

In practice, the distribution of the uncertain job-processing times is unknown. We are given data of job-processing times $\{\boldsymbol{p(1)}, \boldsymbol{p(2)}, \ldots, \boldsymbol{p(N)}\}$ under scenarios $\Omega = \{1,2,\ldots,N\}$, where N denotes the number of given scenarios. The scenarios are independently and identically distributed. The SAA approach approximates problem **P1** by the following problem **P2**:

P2:

$$\begin{aligned} \min \quad & \hat{g}_{SAA} = \theta_1 \cdot \sum_{k \in M} c_k z_k + \theta_2 \cdot \frac{1}{N} \sum_{\omega=1}^{N} \sum_{i \in I} C_i(\omega) \\ s.t. \quad & (1) - (9), \quad \omega = 1,2,\ldots,N \end{aligned}$$

Formulation **P2** can be solved by calling commercial optimisation softwares, e.g. CPLEX. We can obtain the optimal solution $\hat{z} = \{\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_K\}$ and the optimal value $\hat{g}_{SAA}$ of **P2**, which are considered as approximate solution and objective value of original problem **P1**. As the number of scenarios $N \to \infty$, the optimal value $\hat{g}_{SAA}$ converge to the true optimal value of g_{stoch} almost surely (Bertsimas, Gupta, and Kallus 2016).

4.2. *Genetic algorithm*

From the computational results reported in Section 5, we can find that SAA cannot optimally solve the stochastic problem with more than 30 jobs within 3600 s. Therefore, it is necessary to develop methods to efficiently solve the large-scale problems. In this section, we develop a GA for the proposed problem.

GA is based on the rules of biological reproduction, which is first introduced by Holland (1975). It starts with an initial set of feasible solutions. During each iteration, offspring solutions are reproduced by current solutions via crossover and mutation operators. Then the population combining current solutions and offspring solutions is renewed. When a stopping criterion is reached, the algorithm stops. In the following, we present an example (Figure 1) to illustrate the studied two-stage ScheLoc problem. It shows that there are three candidate machine locations, i.e. $M = \{1,2,3\}$, and six jobs to be processed, i.e. $I = \{1,2,3,4,5,6\}$. Location 2 and location 3 are selected for machines. Jobs are ordered with the sequence 2,6,1,5,3,4. Jobs are scheduled to machines at opened locations via earliest completion time (ECT) first rule, i.e. the first job among those not yet processed should be assigned to a machine such that its completion time is the earliest. The job schedules are shown by Gantt chart at the bottom of Figure 1.

4.2.1. *Coding and initialisation*

In our GA for the two-stage stochastic optimisation problem, feasible solutions are represented by chromosomes (individuals) composed of the two parts corresponding to machine location and job sequence decisions: (i) the first part is the

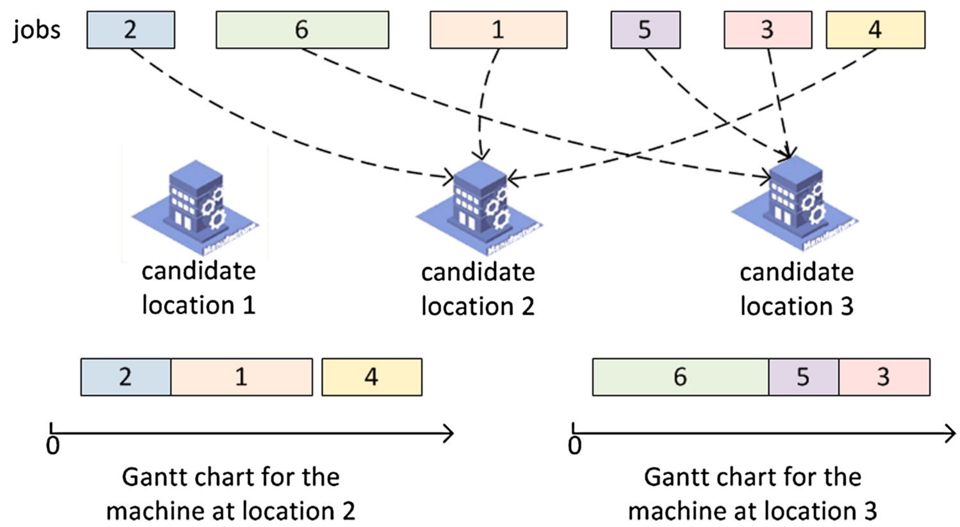

Figure 1. Illustration of the example.

selection of machine locations, where that the number at the kth position equals 1 denotes that candidate location k is selected for placing a machine, 0 otherwise, where $k \in M$. (ii) The second part is the sequence of jobs to be put on the machines under each scenario. Based on the machine locations and the sequence of jobs are determined, under each scenario, jobs are scheduled by ECT rule. The gene structure of this example is illustrated by the first gene part and the second gene part under scenario 1 in Figure 2, where opened locations are labelled by grey.

For the generation of an initial individual, firstly the number at each position in the first gene part is randomly selected from $\{0,1\}$, where 1 denotes that the location is selected and 0 otherwise. If the number of selected locations is more than K or less than 1, the first gene part will be regenerated. The second part of an initial solution under each scenario is a random permutation of job indices. Since jobs are scheduled by the ECT rule, the completion time of each job will be obtained accordingly.

4.2.2. *Crossover and mutation*

During each iteration, offspring individuals are generated by current solutions via genetic operators, including crossover and mutation procedures. For the crossover procedure, the identical parts of two parents will be reproduced to the two offsprings. For each location, if it is selected in both parent solutions, then it will still be selected in the offsprings. Otherwise, it will be reselected randomly for the offsprings. If the sequences of a job are the same in two parents individuals, then its sequence will be not changed in the offsprings. Figure 3 illustrates the procedure. The identical parts of two parents are labelled with a circle in dotted line, which are copied to the offspring individuals. For example, possible location 2 is selected in parent 1 and parent 2, then it is still selected for placing a machine in the offsprings. Similarly, the sequences of jobs 6 and 5 under scenario 1 is unchanged in the offsprings. Besides, the number on each of the rest gene position of possible machine locations, i.e. locations 1 and 2, are randomly selected from $\{0,1\}$. The rest jobs under each scenario are randomly ordered. Then jobs are scheduled via ECT rule. If the offspring individual is infeasible, i.e. the number of selected locations is more than K or less than 1, the number at each of the rest position in the first gene part will be regenerated.

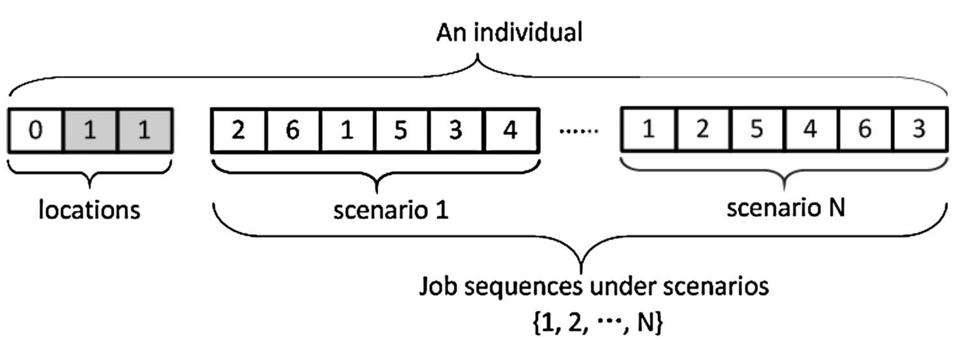

Figure 2. Coding (jobs are scheduled via ECT rule).

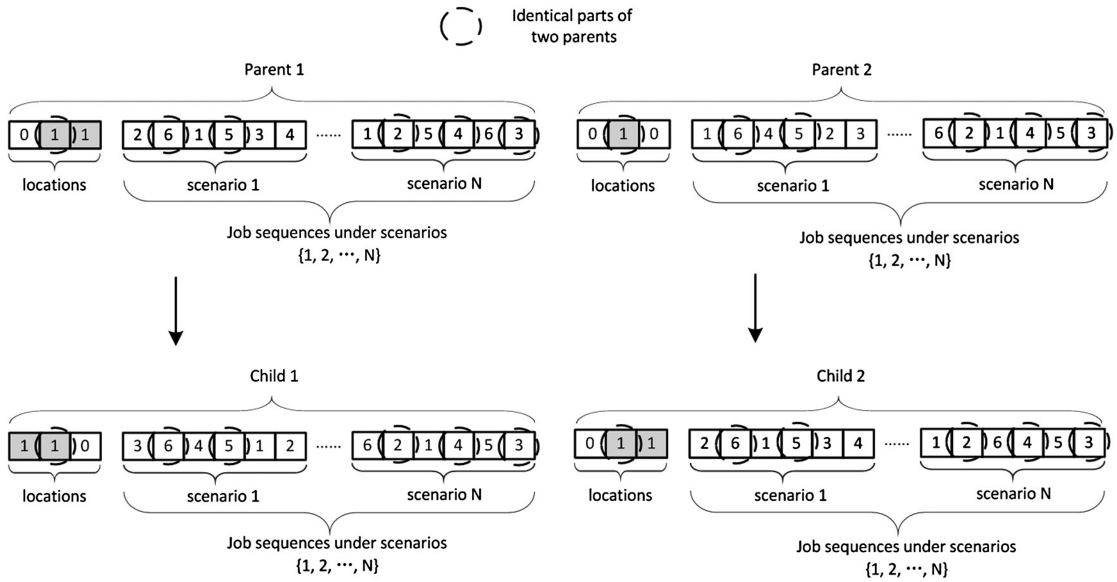

Figure 3. Crossover (reproducing the identical gene parts between parents to offsprings).

There are two ways for mutation: (1) random modification of machine locations and (2) random swap of the sequences of two jobs in the second gene part. For example in Figure 4, the first mutation way consists of (i) copying the job sequence part to offspring individual, (ii) randomly selecting a candidate machine location, i.e. location 1, (iii) modifying the number on its gene position, and (iv) if the number of selected locations is more than K or less than 1, the first gene part will be regenerated. The second mutation way is illustrated in Figure 5, which includes (i) copying the gene parts to offspring individual, (ii) randomly selecting two jobs under each scenario, and (iii) the sequences of selected jobs are swaped with each other under each scenario. Then jobs are scheduled via the ECT rule.

4.3. *Scenario-based heuristic*

From the computational results in Section 5, it can be obtained that the computational time of SAA is relatively large and the solutions obtained by GA are with worse quality than SAA. In order to solve the large-scale problems more efficiently, we

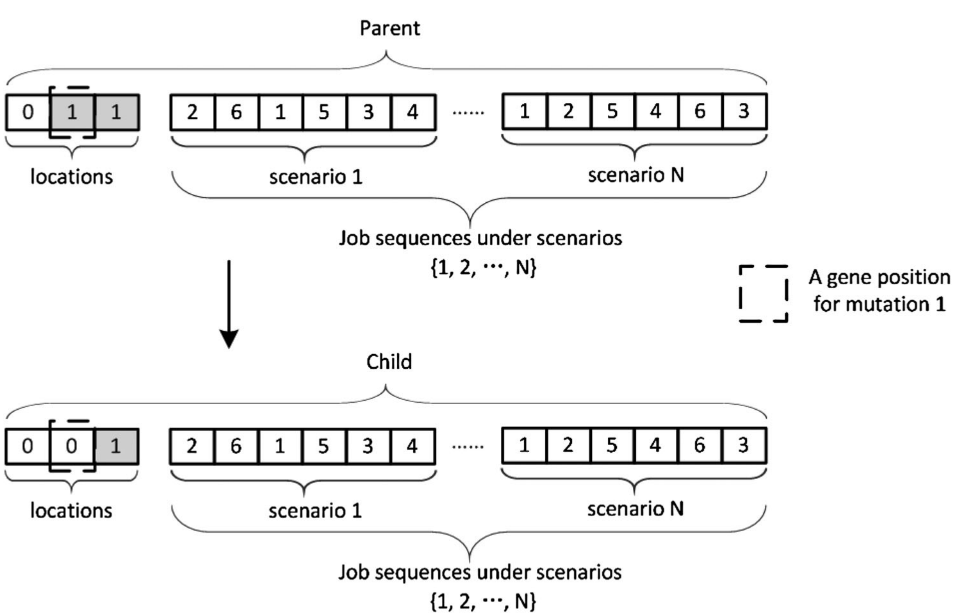

Figure 4. Mutation 1 (random modification of machine locations).

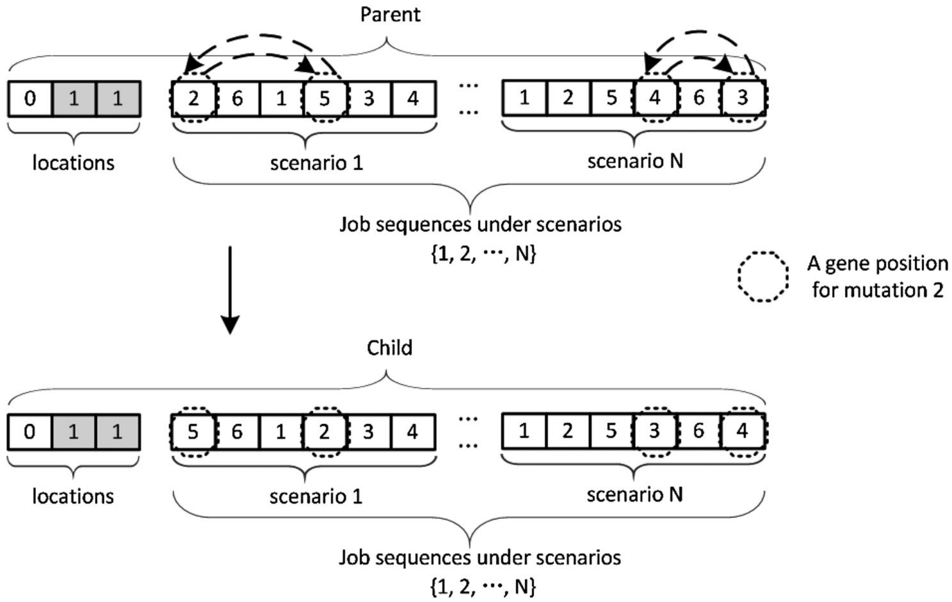

Figure 5. Mutation 2 (random swap of the sequences of two jobs).

design a scenario-based heuristic in this section. Computational results show that the scenario-based heuristic outperforms SAA and GA in terms of computational time and solution quality.

We can obtain the negative correlation between the two objectives in different stages (c.f., Appendix). Thus the basic idea of the scenario-based heuristic is to deal with the decisions of two stages separately, i.e. the location problem **P3_1** without the realisation of uncertainties and the scheduling problem **P3_2** under given information on job-processing times, as shown in the following. The output of problem **P3_1**, which is the decision on machine locations, i.e. $\boldsymbol{z} = [z_1, z_2, \ldots, z_m]$, is taken as the input of problem **P3_2** in each iteration. In problem **P3_2**, $C_i(\omega)$ denotes the completion time of job $i \in I$ with given machine locations $\boldsymbol{z}$ under scenario $\omega \in \Omega$.

$$\textbf{P3\_1}: \quad \min \quad f_1 = \theta_1 \cdot \sum_{k \in M} c_k z_k$$

$$s.t. \quad (1),(9).$$

$$\textbf{P3\_2}: \quad \min \quad f_2 = \theta_2 \cdot \frac{1}{N} \sum_{\omega=1}^{N} \sum_{i \in I} C_i(\omega)$$

$$s.t. \quad (2)-(9), \quad \forall\ \omega \in \Omega$$

Scenario-based heuristic is illustrated in Algorithm 1, where *num_iter* denotes the maximum number of iterations, which is calculated as $num\_iter = \sum_{k=1}^{k=K} C_m^k$. During each iteration, the first-stage optimal solution is restricted to be different from the solutions obtained in the previous iterations. Each time the decision on locations is made, then under each scenario, jobs are scheduled according to a modified shortest processing time (SPT) first rule: jobs are firstly sorted in ascending order of their processing times, the shortest job among those not yet processed should be put on a machine such that its completion time is the earliest. Scenario-based heuristic takes O(max$\{K, n \log n\}$)= O($n \log n$) time in solving the second-stage problem. A preliminary analysis has been conducted to adjust the stopping criteria. Based on the tradeoff between the solution quality and the computational time, we set that the proposed scenario-based heuristic stops when the objective value of the complete problem increases in five successive iterations.

Algorithm 1: Scenario-based heuristic

Input: $K,\ \Omega,\ \boldsymbol{p},\ c_k,\ r_{ik},\ \forall i \in I, k \in M.$

$t = 0$; (Iteration counter)
$\boldsymbol{z}^t = \emptyset$; (Machine locations)
$Obj^t = \infty$; (Objective value of the complete problem)
while $t < num\_iter$ **do**
  Solve **P3_1** with CPLEX and obtain the first-stage optimal solution, i.e., machine locations $\boldsymbol{z}^t$;
  Under $\boldsymbol{z}^t$, jobs are scheduled via the modified SPT rule;
  Obtain objective value of the complete problem $Obj^t = f_1 + f_2$;
  if *The objective value of the complete problem increases in 5 successive iterations* **then**
    Stop;
  else
    $t = t + 1$;
    Add new constraint $\boldsymbol{z} \neq \boldsymbol{z}^t$ to **P3_1**; (Avoid the same solutions in previous iterations)
  end
  end
end
Compare Obj^t in each iteration, and select the solution of machine locations with the best objective value;

Output: Machine locations and the best objective value.

5. Computational experiments

In this section, the performance of proposed approaches are evaluated by 450 instances, including small-scale and large-scale instances. All solution approaches were coded in MATLAB_2014b, SAA method were combined with CPLEX 12.6. All computational experiments were conducted on a PC with 3.60 GHz processor and 8.00 GB RAM under windows 7 operating system. The total computation time of formulation combined with CPLEX is limited to 3600 s.

For the GA, a preliminary analysis was conducted to fine-tune the parameters, which is presented in Table 1. Population size and generation number are set to be 80 and 20, respectively.

5.1. *Data generation*

For the instances, data set is derived from benchmark instances created by Heßler and Deghdak (2015), which can be found on the Internet at https://drive.google.com/open?id=0B_MYN0r5mjqOEdGSEhoLW9Ubk0. Their benchmark instances contain four different types of instance sets, namely the small-scale instances, large-scale instances, network instances, and planar instances. The set of small-scale instances contains 50 instances with up to 30 jobs, 10 locations, and 8 machines. The large-scale instances contains 400 instances with up to 300 jobs, 60 locations, and 50 machines. For the third set, 350 networks are randomly generated. For the last set of instances, the coordinates of job locations are randomly selected in a square with a minimum distance between two jobs equal to 0.1 units.

The fixed cost of each location is calculated as $c_k = \hat{c} + \bar{n}_k \cdot \bar{c}$, where $\hat{c}$ denotes the construction cost and $\bar{c}$ is the cost related to the number $\bar{n}_k$ of jobs with distance less than a predetermined threshold $\bar{d}$. In the computational experiments, we set $\hat{c} = 20$, $\bar{c} = 10$, and the unit of cost is \$10,000. For the value of $\bar{d}$, firstly we sort the distances between jobs and locations in ascending order and obtain vector d, then set the $\bar{d}$ to be $\lfloor \frac{1}{3} \cdot n \rfloor$th element in d. Assume there are three jobs and two possible locations, and the ascending order of distances is denoted by $d = [3,6,13,13,14,15]$, the value of $\bar{d}$ is set to be $\lfloor (\frac{1}{3} \cdot 6) \rfloor$th element in d, namely $\bar{d} = 6$.

For small-scale and large-scale instances in the computational experiments, the distance between each job and possible location is derived from the original instances. The processing times of jobs are randomly generated within [5,25]. During the computational experiments, we set θ_1 and θ_2 to be 1.

Table 2. Parameters for GA.

Parameter	Value (GA)
Population size (*pop*)	80
Generation number (*gen*)	20
Crossover probability	0.9
Mutation 1 probability	0.2
Mutation 2 probability	0.8

5.2. *Results and discussion*

Computational results on the small-size and large-size instances are reported in Tables 2–4. For each instance, we run each algorithm 30 times and obtain its average value.

Table 2 represents the computational results of the small-size instances in which the number of jobs ranges from 2 to 30. Since SAA method directly uses CPLEX, the number of scenarios it can deal with is limited and set to be 30. Due to the capacity of processing large-scale data, the number of scenarios for our scenario-based heuristic and the GA is set to be 300. One of the properties of SAA method is that as the number of scenarios $N \to \infty$, the optimal value $\hat{g}_{SAA}$ converge to the optimal value of g_{stoch} almost surely. Therefore, one instance with more random scenarios means the closer it is to the original stochastic problem. We can obtain from Table 2 that its solution quality of SAA method with 30 scenarios is little higher than that of GA with 300 scenarios. However, the computational time of SAA method is the largest on average, which is about 6 times longer than that of GA and about 23 times longer than that of scenario-based heuristic. Besides, given the same number of scenarios, the average objective yielded by scenario-based heuristic is 263.1, which is about 18.3% smaller than that yielded by GA. Thus, we can find that the scenario-based heuristic can yield solutions with highest quality. Moreover, the average computational time of scenario-based heuristic is 19.1, which is the least among the three solution approaches. For the proposed GA, its computational time is smaller than that of SAA, while its solution quality is the worst. Concluding, for small-size instances, scenario-based heuristic is the most efficient in terms of the solution quality and computational time.

The computational results of 400 large-scale instances with jobs ranging from 20 to 300 are reported in Tables 3 and 4. Since SAA combined with CPLEX cannot solve most of the large-scale instances to optimality within 3600 s, thus we compare the performance of scenario-based heuristic and genetic algorithm (GA). In Tables 3 and 4, S-heuristic denotes the proposed scenario-based heuristic. The number of scenarios for scenario-based heuristic is 300, and so is that for GA. From the first to the sixth columns in Table 3, we can find that the average objectives obtained by GA is 1456.9 and the average objectives obtained by scenario-based heuristic is 1151.8, namely the solution quality of GA is about 26.5% worse than that of scenario-based heuristic. Besides, the computational time of scenario-based heuristic is 61.0, which is about 75.0% smaller than the computational time of GA. From the seventh to twelfth columns in Table 3, we can obtain that the average objective and computational time of scenario-based heuristic are 33.5% and 64.4% smaller than that of GA respectively. From the13th to 18th columns in Table 3, it can be obtained that the average objective and computational time of Scenario-based heuristic are 52.0% and 69.5% smaller than that of GA, respectively. We can obtain from the 19th to 24th columns in Table 3 that the average objective yielded by scenario-based heuristic is 2086.5, which is about 23.0% smaller than the average objective obtained by GA. Moreover, the computational time of scenario-based heuristic is 76.9% smaller than that of GA on average. In Table 3, concluding, the average objective of scenario-based heuristic reported is about 33.7% smaller than that of GA, and the average computational time of scenario-based heuristic is about 71.4% smaller that that of GA.

The instances in Table 4 are with larger scales than that in Table 3 on average. We can find that the average computational times of scenario-based heuristic and GA reported in Table 4 are larger than that reported in Table 3. From the first to sixth columns in Table 4, we can find that scenario-based heuristic can obtain objectives about 31.3% smaller than GA within a shorter time. The7th to 12th columns in Table 4 reports that the average objective and computational time of scenario-based heuristic are 51.2% and 76.2% smaller than that of GA, respectively. The 13th to 18th columns in Table 4 represents that the average objective and computational time of scenario-based heuristic are 27.6% and 49% smaller than that of GA, respectively. Besides, from 19th to 24th in Table 4, it can be observed that the objective obtained by the scenario-based heuristic is about 47.7% smaller that of GA, and the computational time of the scenario-based heuristic is about 57.3% smaller than that of GA.

By computational results, we conclude that:

Table 3. Computational results on small-size instances.

Set	(m,p,n)	SAA with 30 scenarios		S-heuristic with 300 scenarios		GA with 300 scenarios	
		Obj	CT	Obj	CT	Obj	CT
1	(2,1,2)	266	69.2	225	1.2	267	49.5
2	(2,1,2)	73	16.8	74	0.1	74	13.4
3	(3,1,2)	261	37.9	239	0.1	259	45.3
4	(8,2,2)	55	20.9	56	0.0	56	13.5
5	(9,2,2)	202	26.0	200	0.1	211	41.8
6	(7,2,2)	89	22.6	72	0.6	88	18.9
7	(4,3,3)	121	22.6	118	0.1	118	27.6
8	(4,2,4)	123	23.5	98	0.6	120	23.6
9	(8,3,4)	227	38.0	167	1.0	223	46.7
10	(2,1,5)	70	21.2	66	0.1	71	14.1
11	(5,3,6)	265	52.9	200	4.1	263	52.2
12	(9,7,7)	59	23.0	50	2.0	59	14.5
13	(2,1,8)	69	23.4	52	5.5	71	14.2
14	(9,2,8)	198	50.6	134	79.1	197	38.1
15	(3,1,9)	66	22.6	51	3.1	68	14.3
16	(5,2,9)	230	76.6	173	89.2	251	48.1
17	(10,7,9)	192	54.1	163	4.9	189	42.6
18	(10,8,9)	104	26.8	83	8.3	109	23.2
19	(5,2,10)	153	27.4	128	1.6	155	32.0
20	(6,3,10)	243	110.4	170	62.4	254	45.7
21	(5,4,10)	290	47.6	258	0.9	276	55.5
22	(4,2,10)	362	129.2	278	3.2	370	68.1
23	(10,2,10)	369	152.4	278	3.2	359	67.2
24	(4,2,11)	256	46.8	206	2.9	254	50.2
25	(5,4,12)	420	120.4	371	0.9	419	72.6
26	(8,7,13)	416	99.2	371	0.6	426	77.1
27	(9,8,13)	370	164.4	295	3.3	373	73.8
28	(5,4,14)	464	298.2	353	2.9	461	87.7
29	(5,4,14)	263	35.6	229	0.7	262	49.8
30	(3,2,14)	356	66.0	313	0.5	361	69.2
31	(6,2,14)	429	968.2	344	89.9	442	91.6
32	(4,2,15)	321	172.6	281	2.2	326	68.3
33	(5,4,15)	448	794.2	374	42.8	488	97.5
34	(10,4,15)	303	215.6	233	37.0	318	64.5
35	(3,2,16)	388	694.6	287	92.2	412	83.2
36	(10,9,17)	318	264.2	247	68.3	321	64.3
37	(5,3,18)	419	828.6	349	6.3	424	88.2
38	(10,2,18)	346	506.6	262	70.0	368	74.4
39	(10,7,19)	287	93.2	242	3.0	284	59.2
40	(8,7,20)	226	143.6	191	4.7	236	52.2
41	(5,2,21)	650	1406.0	590	2.3	652	136.5
42	(4,3,22)	524	1669.8	410	65.8	553	110.5
43	(9,5,23)	666	2950.2	517	42.1	673	138.2
44	(10,7,23)	571	1661.2	463	43.3	613	123.8
45	(8,7,25)	488	403.6	426	1.9	487	106.1
46	(2,2,27)	679	1599.2	567	4.9	680	145.6
47	(8,2,28)	613	364.0	567	0.3	613	124.0
48	(5,2,29)	537	1895.6	402	85.7	545	115.1
49	(8,5,29)	459	1459.4	406	2.1	458	101.3
50	(5,4,30)	634	2630.6	527	5.9	637	132.4
	Average	318.7	452.9	263.1	19.1	322.0	65.3

(1) for small-size instances, SAA with 30 scenarios outperforms GA with 300 scenarios in terms of solution quality with 1.27% improvement on average; however, its computational time is about 23 times and 6 times larger than that of scenario-based heuristic and GA,

Table 4. Computational results on the first 200 large-size instances.

		GA		S-heuristic				GA		S-heuristic				GA		S-heuristic				GA		S-heuristic	
Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT
51	(13,4,59)	1636	279.9	1374	44.1	101	(16,14,21)	743	102.0	480	11.9	151	(29,22,48)	2159	241.0	1096	157.2	201	(19,10,120)	3808	592.8	2829	85.8
52	(10,6,55)	1507	260.9	1189	37.4	102	(22,18,66)	2478	324.2	1477	133.1	152	(39,24,55)	3341	293.5	1289	103.5	202	(11,7,24)	715	115.3	496	31.3
53	(14,5,8)	1116	179.1	935	6.5	103	(22,16,26)	2934	338.7	1640	123.0	153	(39,29,35)	3857	446.7	2162	25.7	203	(10,5,154)	2275	307.8	1476	104.5
54	(13,6,97)	859	151.5	709	22.3	104	(24,12,45)	1339	204.3	900	162.8	154	(30,18,61)	4919	491.4	2370	118.1	204	(10,6,116)	1567	253.2	1263	22.3
55	(15,10,46)	1455	240.5	1072	100.9	105	(17,14,90)	1753	296.2	1342	120.3	155	(25,20,64)	1890	226.8	937	123.9	205	(10,5,65)	4244	734.2	3440	122.9
56	(14,2,10)	2599	447.6	1972	168.1	106	(17,14,30)	2236	311.6	1475	20.6	156	(34,22,95)	6639	513.7	2652	17.5	206	(17,6,81)	2268	378.4	1870	44.4
57	(7,6,35)	2573	450.1	2025	207.3	107	(20,18,58)	3195	465.9	2174	147.8	157	(38,17,79)	1309	115.4	513	14.4	207	(15,6,100)	781	135.5	554	59.8
58	(12,8,29)	2155	362.8	1890	11.2	108	(19,10,57)	744	112.5	578	1.2	158	(32,24,94)	2902	381.1	1841	38.6	208	(16,13,101)	362	59.7	234	28.4
59	(7,4,77)	1448	240.0	1079	165.1	109	(19,6,57)	1705	225.0	1089	46.6	159	(31,23,51)	2550	309.2	1481	63.7	209	(16,11,15)	2414	400.5	1979	46.0
60	(6,2,68)	118	22.3	85	14.0	110	(22,19,28)	1237	191.7	898	57.2	160	(32,23,71)	3123	411.0	1807	220.9	210	(19,8,185)	3881	632.1	3220	35.7
61	(6,2,39)	1425	234.0	1170	20.3	111	(22,15,70)	1734	233.4	1087	113.7	161	(32,23,87)	4193	445.4	2131	131.6	211	(16,13,65)	5224	819.5	4278	38.8
62	(15,10,31)	413	72.7	331	3.8	112	(24,9,42)	663	90.1	477	1.4	162	(35,16,98)	1806	193.6	855	248.2	212	(11,7,54)	3470	596.4	2679	289.4
63	(12,10,49)	246	42.4	173	23.1	113	(16,9,61)	868	132.9	583	38.2	163	(36,17,43)	2027	185.5	819	29.5	213	(12,5,155)	4334	689.5	3537	72.9
64	(9,7,95)	1602	293.3	1337	77.5	114	(23,16,63)	955	165.6	818	17.8	164	(39,29,98)	4081	464.4	2164	248.5	214	(13,6,78)	314	46.3	184	21.9
65	(12,3,95)	2550	409.2	2256	9.9	115	(19,16,96)	1961	263.9	1270	44.4	165	(35,21,22)	2703	305.9	1446	92.7	215	(11,7,28)	3502	571.3	2639	309.3
66	(8,5,79)	1617	265.4	1290	31.2	116	(25,7,22)	1285	206.2	971	63.6	166	(26,15,77)	3132	359.8	1634	161.7	216	(11,8,12)	646	94.9	437	14.2
67	(12,10,51)	1421	234.6	1064	143.2	117	(20,17,46)	1870	305.9	1516	73.9	167	(28,22,63)	3389	363.4	1687	85.6	217	(14,6,84)	2648	374.3	1928	11.4
68	(9,4,4)	839	138.8	609	77.4	118	(16,8,40)	2389	319.7	1671	16.9	168	(27,20,81)	3106	344.9	1487	436.1	218	(12,7,137)	3291	530.5	2673	37.4
69	(12,9,50)	1656	275.9	1406	10.0	119	(20,14,48)	1575	222.3	1015	130.4	169	(25,24,91)	3529	394.0	1844	112.7	219	(13,11,175)	2871	472.1	2215	136.9
70	(6,4,15)	1280	216.8	986	79.1	120	(17,10,18)	2172	362.5	1632	312.2	170	(30,23,38)	4648	453.8	2090	98.7	220	(10,9,129)	5570	765.2	3814	136.5
71	(15,3,61)	861	147.0	648	16.4	121	(15,10,34)	1756	309.7	1312	189.9	171	(26,17,95)	3336	474.8	2239	74.4	221	(12,7,9)	2969	408.0	2084	21.3
72	(5,2,94)	597	104.0	460	51.2	122	(23,15,55)	1845	284.5	1313	104.2	172	(27,22,62)	4604	458.1	2417	18.1	222	(10,6,125)	3861	607.1	3025	140.8
73	(7,3,58)	926	147.1	680	66.9	123	(15,8,44)	2820	368.7	1675	114.8	173	(29,21,72)	2700	268.9	1153	146.2	223	(19,8,19)	2484	395.8	2022	31.6
74	(14,6,51)	3017	463.7	2203	113.2	124	(15,11,65)	1546	218.6	1046	96.8	174	(27,20,72)	3058	302.0	1400	214.7	224	(15,8,78)	3171	524.7	2249	798.0
75	(13,6,29)	1269	214.3	990	58.6	125	(24,12,66)	717	103.5	468	81.0	175	(29,16,68)	5085	448.2	1952	207.7	225	(16,7,111)	3022	505.2	2209	288.5
76	(6,2,62)	2348	409.2	2028	50.0	126	(21,9,45)	861	118.1	525	59.0	176	(31,22,79)	2443	209.5	899	115.4	226	(13,6,99)	5543	953.4	4684	111.2
77	(14,7,46)	560	97.9	408	20.7	127	(24,6,74)	1583	228.2	1092	72.3	177	(37,20,89)	4503	439.8	1965	242.4	227	(20,14,159)	3067	428.9	2355	10.1
78	(6,4,32)	2066	339.4	1610	58.3	128	(22,6,62)	2402	284.1	1311	77.1	178	(27,18,95)	3709	451.1	2222	41.6	228	(20,9,84)	2761	453.9	2074	266.4
79	(15,3,21)	1359	211.6	976	107.2	129	(20,6,59)	1040	181.6	882	11.1	179	(32,27,91)	3833	396.6	1800	72.6	229	(10,8,137)	6759	904.0	4750	36.7
80	(15,10,31)	1686	268.4	1268	25.0	130	(25,20,73)	724	107.1	561	2.8	180	(33,21,52)	5441	469.8	2276	71.4	230	(17,6,84)	3335	529.2	2801	17.1
81	(12,3,45)	1282	213.7	1075	16.1	131	(19,8,21)	1518	243.9	1129	114.9	181	(39,25,84)	5678	503.7	2305	58.3	231	(10,8,109)	4669	764.1	3446	376.1
82	(15,2,87)	1943	339.1	1532	100.7	132	(22,9,24)	1852	223.1	1222	6.1	182	(37,30,40)	1875	201.7	977	18.7	232	(20,5,199)	1050	169.6	798	33.9
83	(13,2,20)	2837	453.8	2098	177.2	133	(16,9,48)	1343	185.7	872	22.1	183	(36,24,86)	4640	478.3	2196	318.0	233	(20,13,88)	3189	445.0	2225	54.6
84	(8,3,74)	435	77.6	333	6.4	134	(25,14,57)	2482	306.9	1460	72.8	184	(25,16,93)	2901	337.8	1627	49.8	234	(12,6,97)	1535	245.6	1211	29.2
85	(15,6,44)	1368	215.9	1060	48.5	135	(18,11,37)	2705	434.4	2060	116.4	185	(40,23,77)	2755	318.6	1525	24.7	235	(20,12,195)	1811	316.2	1396	61.9
86	(14,8,56)	1074	172.9	904	4.4	136	(16,11,22)	634	94.1	444	17.2	186	(34,23,94)	4427	459.7	2240	42.8	236	(13,8,114)	910	128.6	607	22.7
87	(10,6,46)	1264	212.2	890	59.4	137	(23,18,50)	1498	187.0	849	103.6	187	(38,24,97)	3897	394.5	1889	37.1	237	(11,10,165)	1450	246.2	1403	2.1
88	(15,8,71)	1967	324.6	1578	47.9	138	(22,15,46)	419	70.5	306	8.6	188	(35,20,40)	1832	176.0	774	36.9	238	(15,8,35)	1360	216.5	996	74.7
89	(8,5,100)	2668	460.1	2091	322.5	139	(22,11,37)	1633	228.6	1155	22.4	189	(34,24,94)	5332	505.7	2280	97.5	239	(20,9,93)	436	61.9	249	40.1
90	(6,4,16)	1872	317.0	1498	41.9	140	(23,10,62)	1599	214.1	1033	23.9	190	(27,26,67)	2189	218.2	1099	6.1	240	(19,9,51)	2057	345.9	1582	53.2
91	(5,2,38)	1094	188.6	862	24.8	141	(17,11,19)	1708	262.9	1124	357.5	191	(30,27,94)	2335	185.8	786	275.6	241	(18,9,27)	1864	274.9	1179	133.0
92	(7,6,46)	1696	282.7	1391	23.4	142	(21,12,38)	695	110.0	517	20.8	192	(31,16,79)	4930	448.6	2081	38.4	242	(11,9,52)	2035	348.7	1547	105.7

(*Continued*)

Table 4. Continued.

		GA		S-heuristic				GA		S-heuristic				GA		S-heuristic				GA		S-heuristic	
Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT
93	(8,5,69)	1319	221.4	1032	46.8	143	(16,9,14)	2504	287.5	1411	32.1	193	(32,16,35)	3298	298.0	1343	87.5	243	(14,9,46)	4038	692.3	3585	25.6
94	(13,3,98)	1387	226.3	1027	124.5	144	(17,12,48)	2214	279.5	1319	123.1	194	(39,28,97)	1899	191.3	894	27.3	244	(15,12,12)	1522	256.4	1293	13.9
95	(9,8,68)	1395	234.1	1081	75.0	145	(21,14,44)	1823	290.9	1287	165.0	195	(36,16,42)	3881	455.7	2104	90.2	245	(11,10,73)	2023	334.4	1439	166.3
96	(10,7,40)	282	52.2	212	16.9	146	(20,11,53)	1060	146.2	662	58.0	196	(39,23,36)	5384	472.8	2344	35.1	246	(16,14,57)	2435	394.7	1798	151.8
97	(13,7,59)	1011	161.9	728	32.2	147	(21,5,22)	2134	290.0	1238	290.5	197	(38,20,87)	3828	408.3	1892	44.3	247	(10,7,75)	2906	471.2	2386	43.7
98	(9,8,47)	818	136.2	602	36.3	148	(24,12,59)	1843	279.3	1311	24.4	198	(40,19,57)	4183	459.7	2171	223.4	248	(11,5,149)	3519	481.7	2485	18.6
99	(9,6,50)	2091	351.6	1802	15.6	149	(23,18,57)	1747	276.3	1330	67.9	199	(37,29,37)	2625	259.1	1142	122.9	249	(12,11,53)	499	75.2	348	8.9
100	(10,3,51)	1834	310.9	1572	11.1	150	(17,11,61)	1042	138.6	615	47.6	200	(28,23,90)	3126	363.1	1631	105.9	250	(12,11,70)	5078	866.8	4355	155.6
Average		**1456.9**	**243.5**	**1151.8**	**61.0**	**Average**		**1631.7**	**232.6**	**1092.4**	**82.8**	**Average**		**3500.6**	**359.9**	**1678.5**	**109.5**	**Average**		**2710.9**	**428.3**	**2086.5**	**98.9**

Table 5. Computational results on the remaining 200 large-scale instances.

		GA		S-heuristic				GA		S-heuristic				GA		S-heuristic				GA		S-heuristic	
Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT
251	(29,19,175)	7185	862.1	4229	63.1	301	(30,22,100)	4406	506.8	2305	130.6	351	(17,10,86)	2868	413.9	1912	369.9	401	(29,24,182)	5672	256.8	3356	310.4
252	(26,12,149)	6141	757.6	3579	169.0	302	(27,21,169)	7438	837.2	4035	361.4	352	(22,14,31)	1054	154.9	742	348.3	402	(30,24,142)	10242	339.6	5654	306.4
253	(20,12,200)	1806	238.8	1117	47.8	303	(39,29,119)	9101	771.3	3543	343.0	353	(18,17,184)	5248	889.6	4143	590.5	403	(25,22,299)	7148	795.4	3169	296.8
254	(22,17,130)	1192	134.8	603	37.5	304	(40,24,164)	1437	174.7	844	11.4	354	(19,17,43)	6258	963.4	4966	128.7	404	(40,28,270)	6052	851.2	3448	52.4
255	(21,16,165)	5729	790.5	3954	64.5	305	(34,26,195)	4744	440.1	2061	35.1	355	(22,17,150)	5203	886.0	4461	37.3	405	(29,23,250)	7575	186.9	4492	69.2
256	(21,17,137)	5259	597.7	2795	256.6	306	(39,20,86)	6839	644.3	3013	136.6	356	(17,11,195)	9587	1358.3	5851	4227.5	406	(38,16,106)	9134	253.4	5298	246.7
257	(22,11,178)	5154	731.7	3659	29.0	307	(36,30,175)	3269	345.6	1582	65.6	357	(19,16,203)	1409	162.1	777	20.3	407	(27,19,89)	13378	468.5	6562	647.1
258	(22,20,79)	1322	215.5	1035	21.3	308	(30,25,154)	8940	736.7	3884	17.2	358	(22,14,162)	3130	450.4	2185	68.3	408	(25,17,275)	1025	669.8	449	67.1
259	(27,11,70)	6036	917.6	4362	212.4	309	(31,24,169)	5548	673	3291	199.4	359	(21,12,177)	7905	1187.4	5578	1126.1	409	(40,17,35)	12108	950.2	6450	126.6
260	(29,12,200)	1802	215.9	970	145.3	310	(25,24,158)	9147	963.5	4317	452.7	360	(23,19,24)	7475	1039.4	5097	116.4	410	(39,25,205)	11309	1661.2	6273	202.1
261	(26,22,47)	7156	919.4	4619	164.8	311	(39,30,150)	5327	607.1	2944	58.6	361	(16,11,184)	4644	782.4	3778	114.9	411	(30,17,252)	2921	403.6	1514	371.0
262	(27,14,26)	5067	652.1	3171	125.7	312	(27,25,34)	6423	692	3587	16.1	362	(17,16,208)	3820	611.4	2841	168.4	412	(40,17,137)	11403	799.2	7034	706.3
263	(20,19,162)	6511	970.9	4744	118.1	313	(36,26,86)	6679	619.6	2795	180.6	363	(16,14,184)	5348	873.3	4092	520.8	413	(28,24,143)	13616	364.0	6034	345.9
264	(29,24,120)	5304	804.7	3698	277.1	314	(38,24,125)	5343	576	2564	316.2	364	(24,17,270)	8009	1257.7	5803	933.4	414	(35,15,166)	5162	1295.6	2758	149.2
265	(23,18,151)	5751	696.8	3411	162.4	315	(37,20,67)	4088	399.2	1925	30.2	365	(25,15,32)	1100	142.2	630	389.0	415	(29,18,228)	3784	459.4	1534	63.6
266	(21,10,43)	5234	603.9	2783	229.3	316	(40,26,142)	6481	763.6	3596	77.3	366	(23,10,92)	1075	150.2	677	339.2	416	(32,18,275)	9001	1130.6	5666	427.5
267	(20,14,186)	2871	410	1856	173.8	317	(26,20,139)	4067	444.8	2053	106.8	367	(23,13,245)	9413	1294.3	6267	515.5	417	(29,15,19)	6983	685.1	5273	182.0
268	(23,21,44)	5744	782.4	3913	195.5	318	(32,28,187)	3202	359.7	1825	15.9	368	(19,14,213)	6976	1179.4	5967	179.6	418	(30,24,268)	6537	792.5	3414	316.4
269	(20,19,190)	5880	849.7	4103	189.0	319	(26,20,124)	7490	796.9	3720	103.2	369	(15,12,162)	3617	487.5	2461	243.4	419	(36,15,262)	5404	461.8	2525	703.5
270	(26,23,134)	1358	155.1	692	75.3	320	(40,27,133)	7236	627.7	2915	136.9	370	(18,15,124)	10699	1438.5	6604	1002.1	420	(28,21,67)	7320	269.7	5039	26.5
271	(28,12,160)	5019	495.8	2225	145.3	321	(33,29,111)	4705	573.5	2878	33.5	371	(16,11,263)	4570	723.9	3674	63.7	421	(37,28,256)	14097	794.5	6806	355.8
272	(23,12,144)	2100	227.6	1098	42.3	322	(38,20,79)	5765	650.3	2926	317.2	372	(23,11,28)	6810	1109.0	5191	1028.4	422	(38,22,116)	5960	469.8	4262	354.6
273	(27,19,121)	6096	903.5	4428	119.9	323	(35,21,151)	2907	402	1816	232.8	373	(22,13,30)	6057	938.1	4768	56.2	423	(40,25,59)	15598	761.3	6492	197.4
274	(23,12,82)	3859	638.3	3067	130.7	324	(30,29,86)	9721	863.2	3841	242.0	374	(24,18,265)	1438	209.7	1042	92.3	424	(27,20,235)	5674	625.4	3019	769.5
275	(23,14,163)	1913	215.6	969	150.2	325	(35,25,71)	1831	228.6	1004	152.5	375	(17,13,251)	3498	545.8	2822	22.1	425	(26,18,216)	9568	614.9	4268	682.6
276	(28,10,172)	1422	158.4	720	106.4	326	(37,23,153)	7906	916.2	4819	36.0	376	(18,16,104)	8283	1346.6	6694	193.3	426	(30,25,145)	6388	598.1	3044	190.1
277	(26,20,31)	3823	490.3	2251	286.5	327	(35,26,121)	2838	322.4	1525	77.1	377	(23,16,293)	7131	1142.8	5263	812.8	427	(40,27,108)	7449	726.4	3762	65.8
278	(30,24,97)	3426	505.7	2361	70.9	328	(25,21,120)	6399	820.2	3859	230.9	378	(15,13,154)	10447	1473.2	7067	411.2	428	(34,26,188)	3209	499.8	1219	126.5
279	(25,24,46)	5130	599.9	2834	74.1	329	(29,23,130)	5476	502.5	2211	240.8	379	(21,11,232)	1579	206.9	987	476.7	429	(37,20,279)	7383	591.6	3617	196.4
280	(21,12,189)	7502	958.5	4638	182.4	330	(26,25,80)	1628	162.3	683	168.3	380	(16,15,200)	5001	738.8	3591	59.0	430	(25,17,181)	11420	469.2	7183	133.3
281	(24,14,44)	4731	500.1	2389	70.8	331	(27,23,45)	2002	244.2	1081	74.4	381	(15,13,115)	802	101.9	496	48.6	431	(30,16,129)	4409	562.6	2386	454.4
282	(27,14,32)	7070	797.2	3736	114.5	332	(35,24,180)	3549	379	2117	3.2	382	(17,15,277)	9300	1360.3	7581	58.1	432	(39,17,178)	2591	1002.3	946	721.8
283	(21,15,102)	8120	927.9	4660	50.7	333	(26,21,66)	7905	673.2	3162	190.2	383	(19,11,235)	5721	850.4	3885	332.5	433	(32,28,129)	5578	491.6	2146	330.5
284	(25,18,100)	6674	791.6	3907	327.1	334	(27,26,166)	10157	955.8	4404	589.8	384	(23,18,299)	2326	381.1	1678	294.7	434	(35,28,139)	10869	743.5	6056	587.1
285	(30,25,120)	5086	791.6	3737	197.7	335	(40,27,96)	10606	981	4717	329.6	385	(24,17,42)	4906	722.6	3487	90.0	435	(39,27,53)	13634	675.3	7309	117.5
286	(22,17,196)	2285	277.6	1287	38.6	336	(36,23,31)	7555	751	3502	90.2	386	(20,18,150)	9384	1203.3	6723	37.8	436	(35,30,148)	8284	523.6	4304	213.0
287	(26,15,100)	6297	969.1	4484	594.9	337	(29,21,48)	5229	629.2	2995	72.1	387	(16,10,21)	4746	644.2	3211	142.0	437	(25,22,298)	6577	446.1	3285	118.7
288	(29,17,156)	5054	730.6	3327	44.2	338	(34,20,74)	5720	563.6	2520	243.8	388	(23,15,280)	7401	1084.7	5273	68.0	438	(36,20,102)	3281	375.3	1425	569.7
289	(28,21,190)	5313	734.5	3399	177.6	339	(36,21,131)	5765	535.2	2466	170.6	389	(15,14,177)	5591	904.2	4227	428.2	439	(36,27,43)	5172	769.4	2858	221.0
290	(29,12,164)	2965	409.8	1951	35.8	340	(35,28,188)	11191	903.1	4258	67.7	390	(19,15,78)	3701	622.3	3042	142.0	440	(38,27,90)	14176	1137.8	8443	20.2
291	(24,20,56)	8314	993.2	4656	261.4	341	(35,29,144)	7154	858.2	4057	234.7	391	(22,15,247)	2126	324.6	1460	286.6	441	(29,23,297)	9147	369.3	5426	407.0
292	(30,11,193)	4873	644.2	3186	82.9	342	(25,21,127)	3681	336.9	1577	52.1	392	(22,10,132)	4284	643.8	3231	180.5	442	(31,21,178)	1481	723.0	648	39.0

(*Continued*)

Table 5. Continued.

		GA		S-heuristic				GA		S-heuristic				GA		S-heuristic				GA		S-heuristic	
Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT	Set	(m,p,n)	Obj	CT	Obj	CT
293	(30,17,139)	6435	868.4	4104	153.1	343	(38,26,108)	5663	540.1	2446	165.6	393	(20,16,221)	9183	1190.3	5526	923.4	443	(35,28,134)	5255	315.3	3659	360.8
294	(25,18,148)	5953	629.3	3122	52.2	344	(38,24,103)	6191	765.9	3640	46.1	394	(15,12,192)	1916	279.2	1246	878.6	444	(32,19,66)	3568	336.4	1475	824.6
295	(26,12,82)	4369	592.9	3346	10.4	345	(40,30,174)	1571	135.9	658	4.8	395	(18,10,130)	4330	649.3	3237	112.5	445	(25,24,124)	4030	703.0	2496	229.2
296	(29,23,199)	4489	648.7	3067	437.4	346	(29,24,173)	4751	441.6	2011	71.8	396	(18,10,68)	6350	917.3	4626	148.3	446	(38,16,299)	4241	495.9	2137	196.1
297	(24,14,132)	6887	869.3	4267	60.7	347	(32,21,66)	9394	903.9	4794	27.2	397	(19,16,137)	5653	796.8	4986	64.7	447	(26,24,227)	10451	715.4	6597	66.0
298	(30,10,172)	2908	387.6	1772	156.7	348	(39,27,104)	6813	773	3758	132.8	398	(21,13,246)	5791	851.9	3868	121.7	448	(40,19,26)	1799	820.0	811	342.0
299	(26,20,130)	2639	347.8	1924	3.1	349	(31,20,151)	8406	850.7	4028	80.4	399	(15,14,60)	814	121.1	546	369.2	449	(26,15,153)	13213	763.4	5637	32.0
300	(23,13,124)	7574	996.1	4819	167.9	350	(39,23,25)	6191	771.6	3759	56.4	400	(19,11,137)	9859	875.1	6665	341.4	450	(35,28,63)	20552	1507.2	6665	341.4
Average		**4816.6**	**628.2**	**3020.4**	**142.1**	**Average**		**5917.6**	**608.3**	**2886.1**	**144.6**	**Average**		**5276.6**	**773.6**	**3818.5**	**394.5**	**Average**		**7816.7**	**654.4**	**4086.4**	**297.6**

(2) for small-size instances, scenario-based heuristic with 300 scenarios outperforms GA with 300 scenarios and SAA with 30 scenarios in terms of solution quality with 18.32% and 17.45% improvement on average,
(3) for large-size instances, given 300 scenarios, the computational time of scenario-based heuristic is about 69.5% smaller than that of GA on average, and scenario-based heuristic outperforms GA in terms of solution quality with about 36.5% improvement on average.

For the practical problems, especially for the large-scale problems, our proposed scenario-based heuristic can save a lot of computational time and it is very easy to implement (Table 5).

6. Conclusion

This paper investigates the two-stage stochastic parallel machine ScheLoc problem with uncertain job-processing times. A two-stage stochastic programming formulation is proposed, then SAA combined with CPLEX is adopted to solve the small-size problems. For large-scale problems, a scenario-based heuristic and a GA are designed. Computational experiments on benchmark instances of different scales are conducted to evaluate the performances of the solution approaches. Computational results show that our proposed scenario-based heuristic is the most efficient in terms of solution quality and computational time.

In future research, several issues may be studied. Firstly, different objectives, such as makespan and total workload of machines, should be analysed. Besides, more practical conditions should be considered, such as the situation that the probability distribution is hard to estimate. Moreover, robust stochastic programming is worth to investigate.

Funding

This work was supported by the Natural Science Foundation of China (NSFC) [grant numbers 71531011, 71428002 and 71571134]. This work was also supported by the Fundamental Research Funds for the Central Universities, Shanghai Pujiang program [grant number 17PJC046], and MOE (Ministry of Education in China) Project of Humanities and Social Sciences [grant number 12YJC630284].

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Al-Khamis, T., and R. M'Hallah. 2011. "A Two-stage Stochastic Programming Model for the Parallel Machine Scheduling Problem with Machine Capacity." *Computers and Operations Research* 38 (12): 1747–1759.
Battaïa, O., and A. Dolgui. 2013. "A Taxonomy of Line Balancing Problems and their Solution Approaches." *International Journal of Production Economics* 142 (2): 259–277.
Bertsimas, D., V. Gupta, and N. Kallus. 2016. "Robust Sample Average Approximation." *Mathematical Programming* 00 (3): 1–66.
Bérubé, J. F., M. Gendreau, and J. Y. Potvin. 2009. "An Exact ε-Constraint Method for Bi-objective Combinatorial Optimization Problems: Application to the Traveling Salesman Problem with Profits." *European Journal of Operational Research* 194 (1): 39–50.
Choi, S. H., and K. Wang. 2012. "Flexible Flow Shop Scheduling with Stochastic Processing Times: A Decomposition-based Approach." *Computers and Industrial Engineering* 63 (2): 362–373.
Elvikis, D., H. W. Hamacher, and M. T. Kalsch. 2007. "Scheduling and Location (ScheLoc): Makespan Problem with Variable Release Dates." Technical Report, University of Kaiserslautern, Department of Mathematics, Report in Wirtschaftsmathematik Nr. 106, Kaiserslautern.
Elvikis, D., H. W. Hamacher, and M. T. Kalsch. 2009. "Simultaneous Scheduling and Location (ScheLoc): The Planar ScheLoc Makespan Problem." *Journal of Scheduling* 12: 361–374.
Gafarov, E. R., A. Dolgui, and F. Werner. 2014. "A New Graphical Approach for Solving Single-machine Scheduling Problems Approximately." *International Journal of Production Research* 52 (13): 3762–3777.
Hamacher, H. W., and H. Hennes. 2002. "Integrated Scheduling and Location Models: Single Machine Makespan Problems." *Studies in Locational Analysis* 16 (0): 77–90.
Heßler, C., and K. Deghdak. 2015. "Discrete Parallel Machine Makespan ScheLoc Problem." *Journal of Combinatorial Optimization* 34: 1–28.
Holland, J. H. 1975. "Adaptation in Natural and Artificial Systems." *Quarterly Review of Biology* 6 (2): 126–137.
Jia, W., Z. Jiang, and Y. Li. 2015. "Combined Scheduling Algorithm for Re-entrant Batch-processing Machines in Semiconductor Wafer Manufacturing." *International Journal of Production Research* 53 (6): 1866–1879.
Kalsch, M. T. 2009. "Scheduling-Location (ScheLoc) Models, Theory and Algorithms." PhD thesis, University of Kaiserslautern.

Kalsch, M. T., and Z. Drezner. 2010. "Solving Scheduling and Location Problems in the Plane Simultaneously." *Computers and Operations Research* 37 (2): 256–264.
Kleywegt, A. J., A. Shapiro, and T. Homem-De-Mello. 2001. "The Sample Average Approximation Method for Stochastic Discrete Optimization." *SIAM Journal on Optimization* 12 (2): 479–502.
Lei, D. 2010. "A Genetic Algorithm for Flexible Job Shop Scheduling with Fuzzy Processing Time." *International Journal of Production Research* 48 (10): 2995–3013.
Liu, M., S. Wang, C. Chu, and F. Chu. 2016. "An Improved Exact Algorithm for Single-Machine Scheduling to Minimise the Number of Tardy Jobs with Periodic Maintenance." *International Journal of Production Research* 54 (12): 1–12.
Lu, C. C., K. C. Ying, and S. W. Lin. 2014. "Robust Single Machine Scheduling for Minimizing Total Flow Time in the Presence Of Uncertain Processing Times." *Computers and Industrial Engineering* 74 (1): 102–110.
Pagnoncelli, B. K., S. Ahmed, and A. Shapiro. 2009. "Sample Average Approximation Method for Chance Constrained Programming: Theory and Applications." *Journal of Optimization Theory and Applications* 142 (2): 399–416.
Rajabzadeh, M., M. Ziaee, and A. Bozorgi-Amiri. 2016. "Integrated Approach in Solving Parallel Machine Scheduling and Location (ScheLoc) Problem." *International Journal of Industrial Engineering Computations* 7 (4): 573–584.
T'kindt, V., and J. C. Billaut. 2001. "Multicriteria Scheduling Problems: A Survey." *RAIRO-Operations Research* 35 (2): 143–163.
Verweij, B., S. Ahmed, A. J. Kleywegt, G. Nemhauser, and A. Shapiro. 2003. "The Sample Average Approximation Method Applied to Stochastic Routing Problems: A Computational Study." *Computational Optimization and Applications* 24 (2-3): 289–333.
Wang, W., and S. Ahmed. 2008. "Sample Average Approximation of Expected Value Constrained Stochastic Programs." *Operations Research Letters* 36 (5): 515–519.
Wang, K., and S. H. Choi. 2014. "A Holonic Approach to Flexible Flow Shop Scheduling under Stochastic Processing Times." *Computers and Operations Research* 43 (4): 157–168.
Wang, S., and M. Liu. 2013. "A Branch and Bound Algorithm for Single-Machine Production Scheduling Integrated with Preventive Maintenance Planning." *International Journal of Production Research* 51 (3): 847–868.
Wang, S., and M. Liu. 2014. "Two-stage Hybrid Flow Shop Scheduling with Preventive Maintenance Using Multi-objective Tabu Search Method." *International Journal of Production Research* 52 (5): 1495–1508.
Wang, S., L. Wang, M. Liu, and Y. Xu. 2013. "An Estimation of Distribution Algorithm for the Multi-objective Flexible Job-shop Scheduling Problem." *International Journal of Production Research* 51 (51): 1–8.
Yin, Y., T. C. E. Cheng, C. C. Wu, and S. R. Cheng. 2014a. "Single-machine Batch Delivery Scheduling and Common Due-date Assignment with a Rate-modifying Activity." *International Journal of Production Research* 52 (19): 5583–5596.
Yin, Y., T. C. E. Cheng, D. Xu, and C. C. Wu. 2012. "Common Due Date Assignment and Scheduling with A Rate-modifying Activity to Minimize the Due Date, Earliness, Tardiness, Holding, and Batch Delivery Cost." *Computers & Industrial Engineering* 63 (1): 223–234.
Yin, Y., Y. Wang, T. C. E. Cheng, W. Liu, and J. Li. 2017. "Parallel-machine Scheduling of Deteriorating Jobs with Potential Machine Disruptions." *Omega* 69: 17–28.
Yin, Y., W. H. Wu, T. C. E. Cheng, and C. C. Wu. 2014b. "Due-date Assignment and Single-machine Scheduling with Generalised Position-dependent Deteriorating Jobs and Deteriorating Multi-maintenance Activities." *International Journal of Production Research* 52 (8): 2311–2326.
Yin, Y., W. H. Wu, T. C. E. Cheng, and C. C. Wu. 2015. "Single-machine Scheduling with Time-dependent and Position-dependent Deteriorating Jobs." *International Journal of Computer Integrated Manufacturing* 28 (7): 781–790.
Ying, K. C. 2015. "Scheduling the Two-machine Flowshop to Hedge Against Processing Time Uncertainty." *Journal of the Operational Research Society* 66 (9): 1413–1425.
Zeppetella, L., E. Gebennini, A. Grassi, and B. Rimini. 2016. "Optimal Production Scheduling with Customer-driven Demand Substitution." *International Journal of Production Research* 55 (6): 1692–1706.
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* 00 (0): 00–00. doi: 10.1080/00207543.2018.1430909.
Zhu, Z., F. Zheng, and C. Chu. 2017. "Multitasking Scheduling Problems with a Rate-modifying Activity." *International Journal of Production Research* 55 (1): 296–312.

Appendix

In this part, based on the ε-constraint method, we briefly provide the tradeoff between objectives of two stages, namely the fixed cost and the total completion time. In deterministic environment, the original problem is transformed into a bi-objective optimisation problem, which is represented as follows:

$$\begin{array}{ll} \min & \{f_1(\mathrm{x}), f_2(\mathrm{x})\} \\ s.t. & \mathrm{x} \in X \end{array} \% \qquad (P\prime)$$

where $f_1(\mathrm{x})$ and $f_2(\mathrm{x})$ denote the fixed cost and total completion time respectively. Item x represents a decision variable vector, which belongs to the feasible region X defined by constraints (1)–(9). A solution x^* is called Pareto optimal if no $x' \in X$ exists such that $f_1(\mathrm{x}') \leq f_1(\mathrm{x}^*), f_2(\mathrm{x}') \leq f_2(\mathrm{x}^*)$ with at least one inequality being strict (T'kindt and Billaut 2001). The image of corresponding objective values of all Pareto optimal solutions is called the Pareto front.

The ε-constraint method is to transform the bi-objective problem into a series of problems, which optimises one objective with restricting another by a bound ε. For our problem, the second objective is considered as a constraint and restricted by ε, the value of which decreases by the minimum unit value of $f_2(\mathrm{x})$ during each iteration. The range of $\boldsymbol{\varepsilon}$ is $[f_2^I, f_2^N]$, which is obtained by following ideal point and nadir point (Bérubé, Gendreau, and Potvin 2009).

- Ideal point: $\mathbf{f}^I = (f_1^I, f_2^I)$, where $f_1^I = \min\{f_1(\mathrm{x})\}$ and $f_2^I = \min\{f_2(\mathrm{x})\}$, $x \in X$;
- Nadir point: $\mathbf{f}^N = (f_1^N, f_2^N)$, where $f_1^N = \min\{f_1(\mathrm{x}) : f_2(\mathrm{x}) = f_2^I\}$ and $f_2^N = \min\{f_2(\mathrm{x}) : f_1(\mathrm{x}) = f_1^I\}$, $x \in X$;

We consider the instance with 8 possible locations and 28 jobs and the maximum number of locations we can select is 2, which is the Example 4_9 in Heßler and Deghdak (2015)'s small-size instances. Figure 1 shows the Pareto front of this instance. We can obtain that the two objectives are negatively correlated with each other.

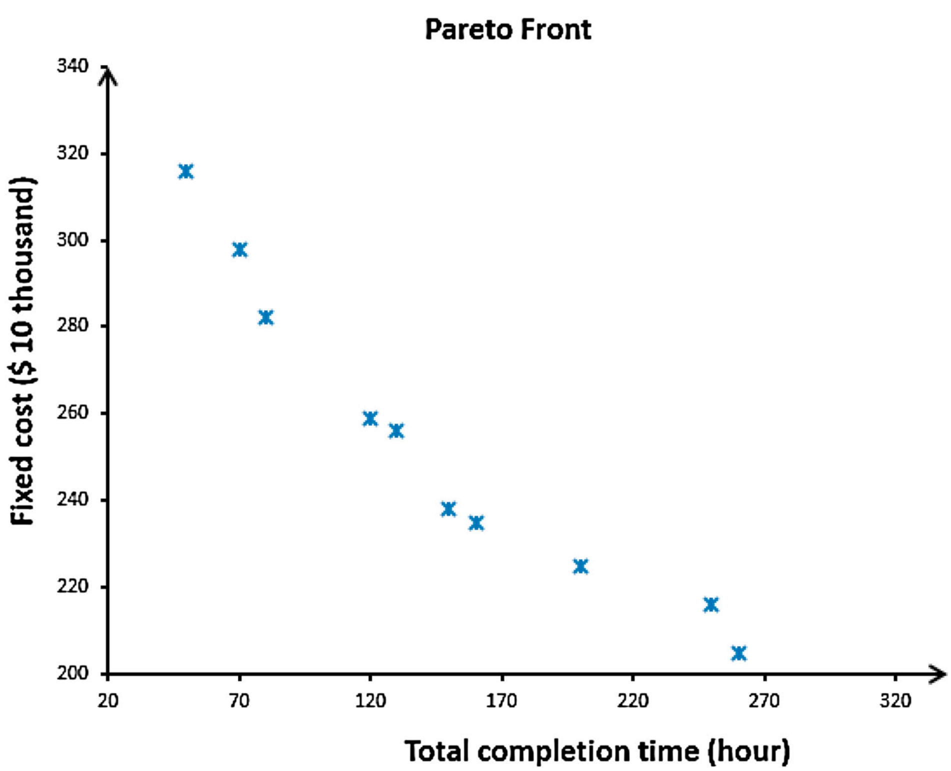

Figure A.1. Perato Front.