Distributed Appointment Assignment and Scheduling under Uncertainty

Li Xue
School of Management, Northwestern Polytechnical University, Xi'an, China xueli@nwpu.edu.cn

Yantong Li*, Zheng Wang
School of Maritime Economics and Management, Dalian Maritime University, Dalian, China yantongli@163.com

Sai-Ho Chung, Xin Wen
Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

We investigate a stochastic distributed appointment assignment and scheduling problem, which consists of assigning appointments to distributed service units and determining service sequences at each service unit. In particular, the service time duration and release time uncertainties are well-considered. The solution to this generic problem finds interesting applications in distributed production systems, healthcare systems, and post-disaster operations. We formulate the problem as a two-stage stochastic program to minimize the total transportation cost and expected makespan, and apply the sample average approximation method to make the problem tractable. We then develop a stochastic logic-based Benders decomposition method, decomposing the problem into a master problem and a subproblem. The master problem determines the appointment assignment variables, and the subproblem handles the sequence variables and the service start time variables. Benders optimality cuts are generated from the solution of the subproblem and are added to the master problem. The developed stochastic logic-based method is advantageous since it can manage many scenarios in parallel. We further consider the due date of each appointment and minimize the total weighted earliness and tardiness and adjust the developed method to solve this variant. Experiments on random instances demonstrate the excellent performance of the proposed model and methods.

Key words: Appointment scheduling; uncertainty; sample average approximation; stochastic programming; logic-based Benders decomposition.

1. Introduction

With the popularization of information technology and smartphones, online reservation systems are adopted by more service agencies. The application of reservation systems can significantly improve service efficiency and customer satisfaction. To implement the service model with reservation systems, one must solve a class of appointment scheduling problems to provide optimal service solutions.

* Corresponding author

We investigate the appointment (job, task) scheduling problem arising from the operational practice of reservation systems, in which multiple appointments must be assigned to distributed service units (hospitals, service agencies). In particular, the service units and customers that make appointments are dispersed on a network and the customer must travel to the allocated service unit to get service. We consider a practical situation where the release (ready) time and service duration of appointments are uncertain. An optimal solution to this interesting problem finds various applications in healthcare, production, transportation, and post-disaster operations.

In the healthcare system, patients make reservations online and get service offline at the hospital. In a traditional single-hospital service model, the patient makes a reservation indicating the preferable service date and time, and decision-makers then arrange medical resources (healthcare workers, doctors) to serve the appointments. A recent challenge is the hospital strain during the carnivorous pandemic. Medical demand surges have stressed hospital systems and result in the crisis-level shortage of beds and staff to provide adequate care for patients. As a result, medical consortium became a new tendency in public healthcare systems, where multiple clinics and hospitals are integrated to provide healthcare services. One of the biggest advantages of the medical consortium compared with the traditional single-hospital one is that appointments can be scheduled among different hospitals. The implementation of this new model impels the decision-maker to solve a distributed appointment scheduling problem (DASP), which makes full use of medical resources to release the hospital strain and avoid the potential risk of cross infections during epidemics.

Another interesting application of the DASP is the large-scale vaccination campaign undergoing during the outbreak of the COVID-19 pandemic. Recipients make appointments online and get vaccinated offline. Decision-makers must assign appointments made by recipients to vaccination sites and sequence these appointments (Zhang et al. 2022). In the production system, one application of the DASP comes from the medical resources production scheduling during epidemics (Li et al. 2021), where orders (appointments) must be assigned to production plants, and the production schedule at each plant must be determined. We can easily find applications of the DASP in various systems. A notable feature of the DASP is that the release time and the service duration frequently correspond to uncertainty. For example, the release time of an appointment may be impacted by the traffic condition and other factors, and the service duration of an appointment may be uncertain due to the material preparation, fatigue level of the service provider, and other factors. Therefore, it is essential to include uncertainty when solving the DASP problem.

The literature seems to focus on solving the appointment assignment and scheduling problem applied to the outpatient and other healthcare services (Marynissen and Demeulemeester 2019, Yu et al. 2020). Most studies aim to schedule the patients' service sequences to optimize a specific quality standard in healthcare systems like to minimize the patient waiting times and hospital work time. However, most appointment scheduling studies in the literature focus on a single hospital or outpatient clinic (Li et al. 2022). There is a lack of flexibility for assigning patients to different hospitals to reduce potential operational costs and guarantee service quality. Moreover, most address the deterministic appointment scheduling problem and the variability and unpredictability in medical service duration and patient arrival time, which may be incurred by various practical situations such as health conditions and medical equipment faults, have received less attention.

To fill the research gap, this paper addresses a stochastic distributed appointment assignment and scheduling problem (SDASP) to minimize the total travel (transportation) cost and expected makespan. Decision-makers must make a plan that determines the appointments' allocation to service centers and the service sequences of appointments at each service center. The total cost component represents the distance-related cost when the service recipients travel from their homes to the corresponding service center. The makespan represents the key performance indicator of the whole service system, which can be considered as an index to evaluate the efficiency or cost of the system. In particular, the appointment release time and service duration uncertainty are integrated into the model. To solve the SDASP, we first develop a two-stage stochastic programming model. The model is intractable due to the continuous feature of uncertain parameters, leading to many scenarios. In this case, we apply the popular sample average approximation (SAA) method to approximate the original model. The approximated SAA model can then be solved using off-the-shelf commercial solvers. However, given the NP-hardness of the studied scheduling problem, commercial solvers lose efficiency when practical-sized instances are involved. We propose a stochastic logic-based Benders Decomposition (SLBBD) method to solve the approximate SAA model. The SLBBD method decomposes the SAA model into a master problem (MP) and subproblem (SP). The MP determines the assignments of appointments to testing sites. The SP can be further decomposed by each testing site and scenario, resulting in a series of single-machine scheduling problems. An optimality Benders cut derived from the SP is introduced to strengthen the MP solution. The MP is then iteratively solved with cuts added during each iteration. We extend the model to solve an interesting variant of the SDASP where the due date of appointments is considered and we optimize a different objective, i.e., the expected total eraliness and tardiness of all appointments. Numerical experiments on randomly generated instances are conducted to demonstrate the effectiveness and efficiency of the proposed approaches.

The remainder of the paper is structured as follows. Section 2 reviews related literature on scheduling applications in healthcare and scheduling problems under uncertainty, respectively. Section 3 presents the description of the considered problem and proposes a two-stage stochastic programming formulation. In Section 4, the classical SAA method and the newly proposed SLBBD method are depicted. Computational experiments on randomly generated instances are conducted and reported in Section 5. In Section 6, we summarise this work and suggest some future research directions.

2. Literature review

Appointment scheduling is a common challenge in many industries, such as production systems, machine scheduling and healthcare. We next review the related literature, including the scheduling applications in healthcare and scheduling problems under uncertainty.

2.1. Scheduling applications in healthcare

Scheduling studies involved in healthcare system are receiving increasing attention in the literature (Abdalkareem et al. 2021). Recent studies focus on scheduling medical resources like surgery, beds, operating rooms, and human resources. such as patient admission scheduling (Demeester et al. 2010, Turhan and Bilgen 2017, Bastos et al. 2019), nurse restoring (Santos et al. 2016, Mischek and Musliu 2019, Ceschia et al. 2020) and operating room scheduling (May et al. 2011, Turhan and Bilgen 2020, Sigurpalsson et al. 2020). Abdalkareem et al. (2021) gives a comprehensive review of healthcare scheduling research in the optimization context. We next briefly review appointment scheduling problems closely related to the current study.

Most appointment scheduling studies in the healthcare field focus on one single service provider. De Vuyst et al. (2014) studied a healthcare appointment scheduling problem with a healthcare facility and a fixed-length session. An algorithm based on the discrete-time setting and Lindley's recursion was designed to minimize the patient waiting time and physician idle time. Feldman et al. (2014) investigated an appointment scheduling problem considering patients' time preferences. Patients with appointments may fail to appear due to the unpredictability of patient behavior, which is called no-shows. Static and dynamic programs are presented to maximize the expected daily profit. Alizadeh et al. (2020) studied a scheduling problem concerning the booking of non-emergency outpatient appointments with limited medical staff. They proposed a MILP model considering the different duration of the appointments and priorities of patients. Besides, A genetic algorithm was designed, which outperforms the model through computational experiments. Some researchers consider how to schedule patients for multiple and serial assessments in a healthcare program. In this scenario, patients take a series of assessments once at their appointment time, and the hospital must consider the capacity and utilization of each assessment. Diamant et al. (2018) investigated appointment scheduling in a multidisciplinary, multistage health care program. In this problem, the clinic assigns patients to an appointment day and schedules assessments order pending the patients' arrival situation. The authors presented an MDP model of this problem and an approximate dynamic programming was proposed to solve it. Yu et al. (2020) studied the appointment scheduling problem with series patients, where patients must take serial treatments after their first appointment day. An MDP model and the Index Policy based on a one-step policy improvement algorithm are proposed to solve this problem.

The studies on the appointment scheduling problem with multiple service providers are limited. Zhou et al. (2021) studied patient-and-physician matching and appointment scheduling in specialty care to minimize the matching and operational costs. A two-stage formulation was proposed, where the first stage was the patient–physician assignments, and each service provider decided the appointment scheduling in the second stage. The sample average approximation (SAA) method and an improved Benders decomposition method were designed for the problem. Soltani et al. (2019) considered an appointment scheduling problem with multiple identical providers, stochastic service times, and customer no-shows. The authors used a time-in homogeneous discrete-time Markov chain process to minimize the weighted sum of customers' waiting time, providers' idle time, and overtime. A load-based appointment scheduling heuristic based on some optimal conditions was proposed to find near-optimal solutions. Shnits et al. (2020) investigated appointment scheduling with parallel servers and pre-sequenced patients. The randomly distributed service duration and no-shows are also considered. The objective is to minimize the end of day and increase resource utilization while a minimal probability of each appointment starting on time is required. A deterministic MILP model was formulated, and a sequential multi-server numerical-based algorithm was developed to overcome the model's limitations. The studies above only consider the random service time and omit the impact of patients' uncertain arrival time, i.e., the delay of arrivals. Besides, with the outbreak of the COVID-19 pandemic, decision-makers frequently face the problem of organizing a large scale of the population to perform rapid testing or vaccination activities, which also involve multiple service providers(Li et al. 2022, Zhang et al. 2022).

2.2. Scheduling problems under uncertainty

The scheduling theory and its applications have been widely studied in the literature. Most studies assume that the processing (service) time of jobs is deterministic (Pinedo and Hadavi 2012, Cao et al. 2005, Tran et al. 2016, Dolgui et al. 2018, Wu and Che 2020). In recent years, an increasing amount of literature generalizes the deterministic scheduling problem to consider uncertain factors in industrial practice (Aydilek et al. 2015, Feng et al. 2016, Liu et al. 2019b).

Current literature mostly focuses on situations where job processing times are uncertain (Liu et al. 2021). Peng and Liu (2004) develop a methodology for modeling parallel machine scheduling problems with fuzzy processing times. Aydilek et al. (2015) propose a fuzzy mathematical programming model of parallel machines scheduling problems to minimize the setup cost and develop an approximated method. The scheduling problem with processing time uncertainty is more complicated than its deterministic counterpart. Most methods developed for it are heuristics (Torabi et al. 2013, Mir and Rezaeian 2016).

Many works assume that the probability distributions of job processing times are known. Tang et al. (2010) consider a stochastic scheduling problem of minimizing the total weighted completion time on preemptive identical parallel machines and develop an algorithm based on a multi-machine list scheduling policy. Ranjbar et al. (2012) develop two branch-and-bound algorithms to solve a robust scheduling problem to maximize the customer service level, which is represented by the probability of the makespan not exceeding the due date. Two exact algorithms are developed by using a general iterative relaxation procedure. Xu et al. (2013) study an identical parallel machine scheduling problem to minimize the makespan, in which the job processing times are stochastic within known closed intervals. Skutella et al. (2016) propose a novel time-indexed linear programming relaxation of the stochastic unrelated parallel machine scheduling problem with minimizing the weighted sum of completion times. Liu et al. (2019a) investigates a stochastic parallel machine scheduling problem and assumes that only the mean and covariance matrix of the processing times are known. SAA method and hierarchical approach based on mixed-integer second-order cone programming formulation are designed. The above literature generally deals with the uncertainty of job processing times. In the rapid-testing scheduling problem, we must include the release time uncertainty in the model.

The literature on scheduling problems considering uncertain job release times is generally scarce. Shen et al. (2016) study a multi-objective flexible job-shop scheduling problem with release time uncertainties and develop an improved multi-objective evolutionary algorithm based on decomposition. Yue et al. (2018) investigate a single machine scheduling problem under stochastic release time within intervals. An efficient two-stage heuristic is proposed to minimize the maximum waiting time. Zheng et al. (2019) study a single yard crane scheduling to minimize the expected total tardiness of tasks with uncertain release times of retrieval tasks. A two-stage stochastic programming model is proposed, and the SAA approach and a genetic algorithm are developed.

Few works study parallel machine scheduling problems simultaneously considering stochastic job release times and processing times. Liu and Liu (2019) first address the stochastic parallel machine scheduling problem to minimize the expected total weighted earliness and tardiness in a Just-in-Time mode. A two-stage stochastic programming formulation is proposed, and the SAA method is applied. Liu et al. (2021) then develop a scenario-reduction-based decomposition approach to solve the problem presented in Liu and Liu (2019).

To sum up, we conclude from the above literature review that although the OR/MS methods have received increasing attention in healthcare systems, the scheduling problem considering multiple service providers and the uncertainties of service duration and release times have not been addressed yet. In this paper, we tackle the difficulties in solving the SDASP by providing a two-stage stochastic program, an SAA approximation model, and a logic-based Benders decomposition algorithm.

3. Problem description and formulation

In this section, we describe the studied SDASP in detail and present a two-stage stochastic programming formulation.

3.1. Problem description

The SDASP generalizes the parallel machine scheduling problem with release dates by considering processing (service) time and release (ready) time uncertainties. Consider a a set $\mathcal{N} = \{1, \ldots, n\}$ of appointments. A set $\mathcal{M} = \{1, \ldots, m\}$ of service centers is launched to provide services. There is a trip cost c_{jk} when an appointment $j \in \mathcal{N}$ travels from its location to the location of the service center $k \in \mathcal{M}$. The travel time between appointment $j \in \mathcal{N}$ and service center $k \in \mathcal{M}$ is denoted by t_{jk}. Each appointment $j \in \mathcal{N}$ has a specific release time r_j indicating the time when the recipients in an appointment are ready to set out, and $r_j + t_{jk}$ is the arrival time of the recipients in the appointment arriving at the corresponding service center. The service time of appointment j is denoted as p_j . Due to various reasons, the release time and service time of an appointment are uncertain. Since no historical data is available, the probability distribution of these uncertain parameters is unknown. These uncertainties are represented by a set Ω of scenarios. Each scenario $\omega \in \Omega$ represents a specific realization of the release time and service time denoted by $r_j(\omega)$ and $p_j(\omega)$, respectively. The following assumptions are made according to classic scheduling problems: 1) Each service center can process at most one appointment at a time. This assumption can be justified because appointments are served sequentially. 2) Each appointment is assigned to exactly one service center. 3) The service of an appointment cannot be disrupted once being started.

The decisions to be made include allocating appointments to service centers and determining the service sequences of appointments at each service center. The objective is to minimize the total trip costs of all appointments and the expected makespan, i.e., the expected time when all appointments finish services.

We formulate the SDASP as a two-stage stochastic program. In the first stage, we determine the assignments of appointments to the service centers before the realization of random information leading to appointments' release and service times. The total travel costs of all appointments and the expected makespan are minimized. In the second stage, we determine the serving sequences of appointments at each service center to minimize the makespan with the realization of appointments' release and service times. We next depict the proposed model.

3.2. Problem formulation

To present the model, we define the following notation.

Sets:

$\mathcal{N}$: set of appointments indexed by i, j and $\mathcal{N} = \{1, ..., n\}$;

$\mathcal{M}$: set of service centers indexed by k and $\mathcal{M} = \{1, 2, ..., m\}$;

Ω: set of independent scenarios indexed by ω.

Parameters:

$r_j(\omega)$: release time of appointment j under scenario ω;

$r_j(\omega) + t_{jk}$: arrival time of appointment j under scenario ω;

$p_j(\omega)$: service time of appointment j under scenario ω;

c_{jk}: travel cost from appointment j to service center k;

t_{jk}: travel time of appointment j from its location to the location of service center k;

h: a large enough number.

Decision variables:

v_{jk}: equal to 1 if appointment $j \in \mathcal{N}$ is assigned to service center $k \in \mathcal{M}$, and 0 otherwise;

$x_{ij}(\omega)$: equal to 1 if appointment $i \in \mathcal{N}$ is tested before appointment $j \in \mathcal{N}$;

(not necessarily immediately) at the same testing site, and 0 otherwise;

$C_{max}(\omega)$: makespan under scenario $\omega \in \Omega$;

$C_j(\omega)$: completion time of appointment $j \in \mathcal{N}$ under scenario $\omega \in \Omega$.

The two-stage stochastic programming model (P) can be formulated as follows:

$$\min \sum_{j \in N} \sum_{k \in M} c_{jk} v_{jk} + E_{\omega \in \Omega} C_{max}(\omega) \tag{1}$$

$$\text{s.t.} \sum_{k \in K} v_{jk} = 1, \quad \forall j \in \mathcal{N} \tag{2}$$

$$x_{ij}(\omega) + x_{ji}(\omega) \leq 1, \quad \forall i, j \in \mathcal{N}, i \neq \mathcal{N}, \omega \in \Omega \tag{3}$$

$$x_{ij}(\omega) + x_{ji}(\omega) \geq 1 - h\left(2 - v_{jk} - v_{ik}\right), \quad \forall i, j \in \mathcal{N}, i \neq j, k \in \mathcal{M}, \omega \in \Omega \tag{4}$$

$$C_j(\omega) \geq p_j(\omega) + \sum_{k \in K} (r_j(\omega) + t_{jk}) v_{jk}, \quad \forall j \in \mathcal{N}, \omega \in \Omega \tag{5}$$

$$C_j(\omega) \geq C_i(\omega) + p_j(\omega) - h\left(3 - x_{ij}(\omega) - v_{jk} - v_{ik}\right), \quad \forall i, j \in \mathcal{N}, i \neq j, k \in \mathcal{M}, \omega \in \Omega \tag{6}$$

$$C_{max}(\omega) \geq C_j(\omega), \quad \forall j \in \mathcal{N}, \omega \in \Omega \tag{7}$$

$$v_{jk} \in \{0, 1\}, \quad \forall j \in \mathcal{N}, k \in \mathcal{M} \tag{8}$$

$$x_{ij}(\omega) \in \{0, 1\}, \quad \forall i, j \in \mathcal{N}, i \neq j, \omega \in \Omega. \tag{9}$$

The objective function (1) minimizes the sum of travel cost from recipients' home in an appointments to service centers, i.e. $\sum_{j \in N} \sum_{k \in M} c_{jk} v_{jk}$, and the expected makespan, i.e. $E_{\omega \in \Omega} C_{max}(\omega)$.

Constraints (2) ensure that each appointment $j \in \mathcal{N}$ must be assigned to exactly one service center. Constraints (3) ensure that there are three optional test orders of two appointments i and j under scenario $\omega \in \Omega$: appointment i is processed before appointment j at the same service center ($x_{ij}(\omega)$=1), appointment j is processed before appointment i at the same service center ($x_{ji}(\omega)$=1) or appointment i and j are assigned to different service centers ($x_{ij}(\omega) = x_{ji}(\omega) = 0$). Constraints (4) denotes that if appointment i and j are processed at the same service center $k \in \mathcal{M}$, there must be one successor and one predecessor under each scenario $\omega \in \Omega$. Constraints (5) imply that each appointment j should be processed after he is ready and transported to the assigned service center under scenario ω. Constraints (6) denote that if appointment i is processed before appointment j at a service center $k \in \mathcal{M}$, appointment j's completion time must be no less than appointment i's completion time plus appointment j's service time under scenario $\omega \in \Omega$. Constraints (7) ensure the makespan must be no less than each appointment j's completion time under each scenario $\omega \in \Omega$. Constraints (8)- (9) define the ranges of decision variables.

The studied SRSP is NP-hard in the strong sense since its deterministic counterpart can be reduced to a parallel machine scheduling problem to minimize the makespan, which is well known to be strongly NP-hard (Pinedo and Hadavi 1992, Hino et al. 2005). The above model is intractable due to the continuous nature of stochastic parameters. We next introduce a well-known SAA method to approximate the model.

4. Solution Method

Since the set Ω of all possible scenarios can be very large, it is difficult to calculate the second stage expected value function for a given assignment scheme. To overcome this difficulty, we first apply the popular SAA method to obtain an approximate model that is tractable for off-the-shelf MIP solvers. We then develop an SLBBD method for solving the SAA model, given that the solver has difficulties in solving problem instances with many appointments and scenarios.

4.1. Sample Average Approximation

The SSA method is a well-known Monte Carlo-based simulation approach that has been successfully applied to solve a wide range of stochastic scheduling problems. For our problem, based on Monte Carlo simulation, a set $\mathcal{S} = \{1, 2, ..., |S|\}$ of scenarios is used to approximate the set Ω, where $S \subset \Omega$. The second stage expected value $E_{\omega \in \Omega} C_{max}(\omega)$ is replaced by the sample average function $\frac{1}{|\mathcal{S}|}\sum_{s \in \mathcal{S}} C(s)$. With the above definitions, model P can be reformulated into an SAA-based model, denoted as model P_1:

$$\min \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{M}} c_{jk} v_{jk} + \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} C_{max}(s) \tag{10}$$

$$\text{s.t.} (2), (8) \text{ and to} \tag{11}$$

$$x_{ij}(s)+x_{ji}(s)\leq 1, \quad \forall i,j\in\mathcal{N}, i\neq\mathcal{N}, s\in\mathcal{S} \tag{12}$$

$$x_{ij}(s)+x_{ji}(s)\geq 1-h\left(2-v_{jk}-v_{ik}\right), \quad \forall i,j\in\mathcal{N}, i\neq j, k\in\mathcal{M}, s\in\mathcal{S} \tag{13}$$

$$C_j(s)\geq p_j(s)+\sum_{k\in K}(r_j(s)+t_{jk})v_{jk}, \quad \forall j\in\mathcal{N}, s\in\mathcal{S} \tag{14}$$

$$C_j(s)\geq C_i(s)+p_j(s)-h\left(3-x_{ij}(s)-v_{jk}-v_{ik}\right), \quad \forall i,j\in\mathcal{N}, i\neq j, k\in\mathcal{M}, s\in\mathcal{S} \tag{15}$$

$$C_{max}(s)\geq C_j(s), \quad \forall j\in\mathcal{N}, s\in\mathcal{S} \tag{16}$$

$$x_{ij}(s)\in\{0,1\}, \quad \forall i,j\in\mathcal{N}, i\neq j, s\in\mathcal{S}. \tag{17}$$

Model P_1 can be solved using off-the-shelf commercial solvers, such as Cplex and Gurobi. The performance of P_1 negatively correlates to the number of scenarios, i.e. $|\mathcal{S}|$, and the optimal objective value of the SAA model P_1 converges to the true optimal value of model P almost surely when $|\mathcal{S}|\to+\infty$ (Bertsimas et al. 2018). However, since the deterministic counterpart of model P_1 is strongly NP-hard, solving the SAA model P_1 with many scenarios is time-consuming. Therefore, an SLBBD method is developed in the next section to tackle practical-sized problem instances in a reasonable computation time.

4.2. The stochastic Logic-based Benders Decomposition

In this section, we develop a stochastic logic-based Benders decomposition (SLBBD) method for solving the SAA model P_1. The logic-based Benders decomposition (LBBD) is widely applied to solve large-scale deterministic MILPs. The basic idea of the LBBD method is to decompose a complex problem into a master problem (MP) and subproblem (SP). The MP and SP are then iteratively solved until they converge to optimality (Hooker 2007). The SP must be linear programs in the classic Benders decomposition method, and Benders cuts are generated based on the dual information. The LBBD method relaxes this restriction and allows the SP to take any form. Thus, the LBBD method has a wider range of applications in solving large-scale optimization problems as long as the problem fits in the decomposition structure, and Benders cuts can be generated by the inference dual (Hooker 2019). The SLBBD method generalizes the LBBD method to solve two-stage stochastic optimization problems. The method shows good performance in solving a stochastic planning and scheduling problem (Elci and Hooker 2020) and stochastic distributed operating room scheduling problem (Guo et al. 2021).

The studied SDASP is a variant of the parallel machine scheduling problem with release dates, which further considers appointments' and service centers' locations, stochastic release time, and service time. The problem implies a good decomposition structure, which motivates us to apply the SLBBD method. In particular, we formulate an MP that determines the appointments' assignments to service centers to minimize the total travel costs and relaxed expected makespan. Given a feasible solution to the MP, we get the assignment results of appointments to service centers. Each service center and scenario can decompose the resulting SP. Thus it is equivalent to solving many single machine scheduling problems with release time to minimize the makespan, i.e., $(1|r_j|C_{max})$. The $1|r_j|C_{max}$ problem can be optimally solved in polynomial time using the earliest release date (ERD) rule. Next, we introduce the MP, SP, and Benders cuts in detail.

4.2.1. The master problem Following the idea of the LBBD method, let a continuous variable $\xi_{max}(s)$ denote the lower bound of the real makespan $C_{max}(s)$ under each scenario $s \in S$ corresponding to a specific assignment of appointments. The MP can be formulated as follows:

$$\min \sum_{k \in \mathcal{M}} \sum_{j \in \mathcal{N}} c_{jk} v_{jk} + \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \xi_{max}(s) \tag{18}$$

$$\text{s.t.}(2),(8) \text{ and to} \tag{19}$$

$$\xi_{max}(s) \geq 0, \quad \forall s \in \mathcal{S} \tag{20}$$

$$\xi_{max}(s) \geq r_j(s) + t_{jv_{jk}} + p_j(s), \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, s \in \mathcal{S} \tag{21}$$

$$\xi_{max}(s) \geq \tau_k(s) + \sum_{j \in J} p_j(s) v_{jk}, \forall k \in M, s \in \mathcal{S}. \tag{22}$$

The MP aims to minimize the total travel cost and the lower bound of the expected makespan for all scenarios. Constraints (21) ensure that $\xi_{max}(s)$ must be no less than the earliest completion time of every appointment j under scenario s. Let $\tau_k(s) = \min_{j \in J}\{t_{jk} + r_j(s)\}, \forall k \in K, s \in \mathcal{S}$ be the earliest beginning time at site k under scenario s. Constraints (22) ensure that the makespan $\xi_{max}(s)$ must be no less than the sum of all appointments' service time assigned to site k under scenario s.

4.2.2. The subproblem The solution of the MP gives the value of appointment assignment variables $\hat{v}_{jk}$. Let S_k be the set of service centers assigned to service center k in the solution, i.e., $S_k = \{j | \hat{v}_{jk} = 1, j \in \mathcal{N}\}$. The SP can be further decomposed into independent subproblems for each service center k under each scenario s. Each subproblem corresponds to a $1|r_j|C_{max}$ problem, which can be optimally solved in polynomial time using the ERD rule. Let $\hat{\xi}_k(s)$ be the optimal makespan for the set S_k of appointments assigned to service center k under scenario s. We next introduce Benders cuts generated from the solution of the SP.

4.2.3. Benders cuts There are generally two kinds of Benders cut, i.e., feasibility cuts and optimality cuts. As the MP solution is always feasible to the SP, feasibility Benders cuts are irrelevant here. We generate optimality cuts to improve the lower bound of the MP objective. The solution of the SP gives the optimal makespan $\hat{\xi}_k(s)$ for each set S_k assigned to service center k under scenario s, we can generate the following optimality cuts.

$$\xi_{max}(s) \geq \hat{\xi}_k(s)\left(\sum_{j\in S_k} v_{jk} - |S_k| + 1\right), \forall k \in \mathcal{M}, s \in \mathcal{S}. \tag{23}$$

Cuts(23) ensure that if all service centers in a set S_k are assigned to service center k in subsequent iterations, the makespan under scenario s should be greater than or equal to $\hat{\xi}_k(s)$ obtained in the current iteration. Once the Benders optimality cut (23) is added to the MP, the augmented MP continues to be solved. The algorithm stops until the MP, and the SP converge to optimality or a given time limit is reached.

5. Computational study

In this section, we conduct numerical experiments on random instances to evaluate the performance of the proposed model and the SLBBD method. The tested instances and detailed results are available at `https://www.dmu-yantongli.com/instances`. All solution approaches are coded in Java linked with IBM ILOG CPLEX 12.10. The MP is solved using CPLEX, and the Benders cuts are added using the embedded lazy callback procedure. All runs are performed on a PC with a Core i7 CPU at 3.60GHz and 16GB RAM under Windows 10 operating system.

5.1. Instance generation and parameter setting

We randomly generate a set of instances with discrete uniform distribution based on the rules introduced in Liu et al. (2021). The transportation time t_{jk} between appointments and service centers is randomly generated from the interval [1, 3]. The ready time r_j of appointment j is randomly generated from the interval [1, 5]. Following Liu et al. (2016), the mean service time is randomly generated from a discrete uniform distribution from the interval [1, 10]. To reflect the stochastic variation, the standard deviations of each appointment's service time and ready time are set to $0.1\mathbb{E}[p_j]$ and $0.1\mathbb{E}[r_j]$, respectively. For each instance, a reference set of 1000 scenarios (Xie and Ahmed 2018), where the release times and service times are randomly generated with normal distribution under the given mean and standard deviation values.

The computation time for each run is limited to 1800 seconds. The sample size $|\mathcal{S}|$ is set to 100 (Xie and Ahmed 2018). The instance size is characterized by the number of appointments n and service centers m. For each instance with specific n and m, we generate five random instances.

5.2. Results for small-sized instances

First, we test some small-sized instances with up to $n=20$ appointments and two service centers. Table 1 reports the average results of each set with a specific n and m. To evaluate the performance of the proposed methods, we compare the results obtained by directly solving the SAA model with CPLEX (SAA-C) and the SLBBD method. We introduce five performance indicators, i.e., the mean objective value, variance, 85th percentiles of the objective, 99th percentiles of the objective, and computation time. Mean objective values over the tested 1000 scenarios are denoted by Obj_1 and Obj_2 obtained by the SAA-C method and the SLBBD method, respectively. The variances of the objective values (over 1000 scenarios) are denoted by V_1 and V_2, respectively, for the SAA-C and the SLBBD methods. Values PT_1^r and PT_2^r denote the rth percentiles of the objective values over 1000 scenarios obtained by the SAA-C and the SLBBD methods, respectively. Columns T_1 and T_2 denote the computation times for the SAA-C and the SLBBD methods, respectively.

Table 1 Numerical results for small-sized instances

		The SSA-C method					The SLBBD method				
n	m	Obj_1	V_1	PT_1^{85}	PT_1^{99}	T_1 (s)	Obj_2	V_2	PT_2^{85}	PT_2^{99}	T_2 (s)
4	2	22.90	1.03	23.97	25.41	77.76	22.90	1.03	23.97	25.41	0.35
5	2	26.03	1.13	27.22	28.79	1800	26.03	1.13	27.22	28.79	0.37
6	2	28.56	1.11	29.72	31.28	1800	28.43	1.17	29.65	31.31	0.56
7	2	31.87	1.21	33.12	34.81	1800	31.36	1.16	32.55	34.30	0.59
8	2	40.35	1.52	41.93	44.05	1800	38.72	1.31	40.10	41.96	0.54
9	2	43.11	1.42	44.58	46.64	1800	41.79	1.32	43.13	45.05	0.35
10	2	52.98	1.77	54.81	57.19	1800	48.97	1.50	50.53	52.61	0.36
11	2	55.53	1.77	57.37	59.63	1800	52.86	1.47	54.38	56.55	0.57
12	2	58.29	1.75	60.12	62.54	1800	56.64	1.57	58.32	60.53	0.79
13	2	67.58	2.02	69.74	72.44	1800	62.73	1.65	64.46	66.80	0.57
14	2	69.52	1.97	71.58	74.39	1800	67.48	1.79	69.35	72.05	0.61
15	2	69.72	1.94	71.77	74.37	1800	65.17	1.69	66.89	69.36	2.11
16	2	77.97	2.17	80.26	83.18	1800	69.98	1.71	71.73	74.20	1.06
17	2	91.32	2.47	93.89	97.13	1800	75.44	1.83	77.34	80.02	0.83
18	2	79.70	2.21	82.00	84.96	1800	74.87	1.80	76.73	79.35	1.02
19	2	87.29	2.22	89.62	92.44	1800	82.41	1.88	84.41	87.04	0.88
20	2	105.56	2.65	108.34	111.72	1800	90.92	2.00	93.02	95.88	2.31
Average		59.31	1.79	61.18	63.59	1698.69	55.10	1.53	56.69	58.89	0.82

From Table 1, we observe that the SAA-C method only solves instances with four appointments to optimality within the 1800s, and the average computation time is 1698.69 seconds. However, the proposed SLBBD method can solve all instances to optimality and the average computation time is only 0.82 seconds. The average objective value obtained by the SAA-C method is 59.31, about 7.64% larger than that of the SLBBD method. The average variance of the objective value obtained by the SAA-C method is 1.79, about 16.99% larger than that of the SLBBD method. In addition, the average 85th and 99th percentiles of the objective values obtained by the SAA-C method are 111.56 and 114.83, which are about 26.49% and 26.17% larger than those of the SLBBD method, respectively. In summary, the SAA-C method is far more time-consuming than the SLBBD method. The SLBBD method outperforms the SAA regarding the five performance indicators.

5.3. Results for practical-sized instances

In practice, the studied SDASP involves dozens of appointments and several service centers. We further generate some practical-sized instances. We consider the number of es $n = \{20, 40, \ldots, 100\}$ and the number of testing sites $m = \{2, 4, \ldots, 10\}$. Our preliminary results indicate that the SAA-C method cannot obtain any feasible solution within the given time limit of 3600 seconds. Therefore, these instances are only solved using the proposed SLBBD method with a time limit of 3600 seconds for each run. The results are presented in Table 2. We can observe from the table that the objective value, variance, 85th percentile, and 99th percentile increase with the number of appointments and the number of service centers. This is reasonable since the SDASP involves binary variables v_{jk} whose number is positively related to n and m. The SLBBD method can optimally solve instances with 70 appointments and two service centers within 3600 seconds. However, when the number of service centers increases to 4, it can only solve instances with 30 appointments to optimality within 3600s. Moreover, the SLBBD method can not find feasible solutions to instances with six service centers and more than 60 appointments. A similar conclusion can be drawn by observing the results for instances with eight testing sites and more than 50 appointments. The results indicate that the number of service centers has a more substantial impact on the computation efficiency than the number of appointments.

5.4. Sensitivity analysis

We next conduct sensitivity analysis based on the results of an instance with eight appointments and two service centers. The detailed results are shown in Table 3 to illustrate the differences in the results obtained by the SAA-C and SLBBD methods. In each row $j \in \{1, 2, ..., 8\}$, a symbol $'\checkmark'$ denotes the appointment j is assigned to service center k, i.e. $v_{jk} = 1$. We can see from Table 3 that the assignments of appointments obtained by the SAA-C and SLBBD methods are different.

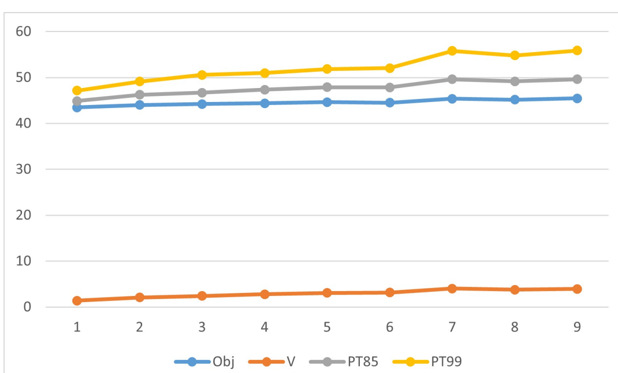

Figure 1 Analysis on the standard deviation $E[p_j]$

We further examine the impact of the standard deviation of service times by varying its value in the range of $0.1E[p_j], \ldots, 0.9E[p_j]$. We then solve the instance using the SLBBD method and

Table 2 Numerical results for practical-sized instances

Instance	Obj	V	PT85	PT95	T (s)
10-2	46.48	1.46	48.03	50.32	0.35
20-2	92.80	1.92	94.77	97.64	0.39
30-2	137.46	2.39	140.02	143.45	1.83
40-2	189.74	2.90	192.77	196.74	115.43
50-2	223.11	3.17	226.38	231.07	546.40
60-2	277.41	3.31	280.91	285.75	868.26
70-2	299.16	3.47	302.56	307.91	3544.32
80-2	356.65	3.85	360.75	365.64	3600
90-2	380.84	4.05	384.90	391.41	3600
100-2	439.42	4.31	443.69	449.73	3600
10-4	33.13	1.23	34.40	36.09	1.50
20-4	64.00	1.36	65.41	67.31	5.63
30-4	89.46	1.62	91.16	93.74	80.55
40-4	105.83	1.75	107.78	109.98	3600
50-4	136.89	1.92	138.87	141.66	3600
60-4	163.73	2.08	165.94	168.99	3600
70-4	188.17	2.13	190.38	193.30	3600
80-4	207.99	2.30	210.29	214.16	3600
90-4	229.89	2.36	232.27	235.94	3600
100-4	264.56	2.63	267.33	271.33	3600
10-6	27.73	0.94	28.73	30.15	0.48
20-6	42.13	0.92	43.02	44.54	603.39
30-6	64.45	1.18	65.70	67.65	3600
40-6	84.67	1.53	86.27	88.75	3600
50-6	109.81	1.60	111.47	114.19	3600
60-6	133.60	1.63	135.27	137.91	3600
10-8	28.00	1.16	29.23	30.67	0.44
20-8	42.44	0.84	43.34	44.64	2723.57
30-8	57.63	0.96	58.66	60.21	3600
40-8	74.50	1.13	75.74	77.22	3600
50-8	92.62	1.23	93.95	96.00	3600
10-10	25.62	0.91	26.59	27.80	2.30
20-10	37.99	0.82	38.86	40.30	13.90
30-10	54.67	0.86	55.49	57.29	3600

Table 3 Results for the illustrative example

	The SSA-C method		The SLBBD method	
Appointment\Service center	1	2	1	2
1	✓			✓
2	✓		✓	
3	✓		✓	
4		✓		✓
5		✓		✓
6		✓	✓	
7		✓		✓
8		✓	✓	

reports the results of the four performance indicators in Figure 1. It can be seen that there has been a gradual increase in the number of objective values, variances, and percentiles, along with the increase of the standard deviation of service time. When the standard deviation of service time increases, long service times are more likely generated, leading to a longer makespan. Besides, the service time range becomes wider with the standard deviation increase, resulting in larger variances and larger objective values in extreme cases.

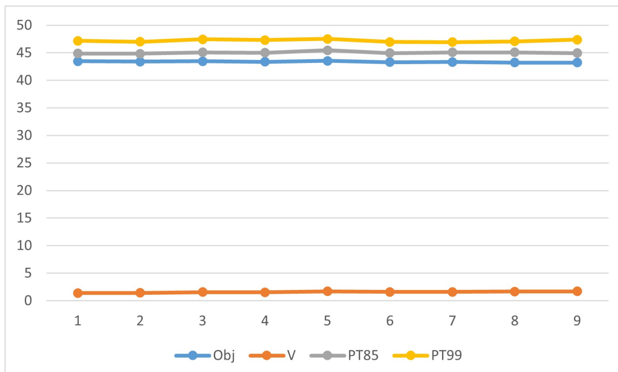

Figure 2 Analysis on the standard deviation $E\left[r_j\right]$

We also analyze the impact of the standard deviation of release times by varying its value in the range of $0.1E\left[r_j\right], \ldots, 0.9E\left[r_j\right]$. The obtained results are shown in Figure 2, from which we see that no significant difference can be observed regarding the mentioned performance indicators. A reasonable explanation is that the range of release times is smaller than the service time (Liu et al. 2021). Hence, the above results suggest that the standard deviation of service time has a larger effect on the performance, i.e., larger objective values and weak robustness in extreme cases.

5.5. Extend the model to consider due dates of appointments

We extend the studied SDASP to include due time of appointments' service. The due time of a appointment specifies the time that the appointment expects to finish service. Then the SDASP with due time consideration is closely related to the stochastic parallel machine scheduling problem (denoted as S-PMSP) with uncertain processing and release times presented in Liu et al. (2021). The S-PMSP minimizes the setup cost of assignments and the expected penalty cost of jobs' tardiness and earliness. While our SDASP aims to minimize the travel costs (equivalent to the setup costs in the S-PMSP) and the expected makespan. We refer to Liu et al. (2021) for a formulation and the SAA model for the problem. We further devise the proposed SLBBD method to solve the SAA model of S-PMSP.

Following the same framework presented in Section 4.2, we decompose the S-PMSP into a MP and a SP, which are iteratively solved to obtain the optimal solution. Let $w_k(s)$ be a lower bound of penalty cost for earliness and tardiness of all jobs (appointments) assigned on machine (service center) k under scenario s. The MP of the S-PMSP can be formulated as follows:

$$\min \quad \sum_{j\in N}\sum_{k\in M} c_{jk}v_{jk} + \frac{1}{|S|}\sum_{s\in S}\sum_{k\in K} w_k(s) \tag{24}$$

$$\text{s.t.} \sum_{k\in K} v_{jk} = 1, \quad \forall j \in N \tag{25}$$

$$w_k(s) \geq 0 \quad \forall k \in K, \forall s \in S. \tag{26}$$

The MP of S-PMSP can be solved using the branch-and-cut method. Once the MP is solved, we get the assignments of jobs to machines. The SP corresponds to sequencing jobs on each machine and each scenario, generating many single machine scheduling problems with release time to minimize the total weighted earliness and tardiness, i.e., $1\,|r_j|\,\theta_j^E E_j + \theta_j^T T_j$. The notations E_j and E_j denote the earliness and tardiness of job j, and θ_j^E and θ_j^T denote the corresponding weights, respectively. The $1\,|r_j|\,\theta_j^E E_j + \theta_j^T T_j$ problem can be optimally solved using the dynamic programming method proposed in Tanaka and Fujikuma (2012). We use the implementation of SiPS/SiPSi C libraries devised by Shunji Tanaka, which is publicly available at `https://sites.google.com/site/shunjitanaka/sips`. $\hat{E}_j(s)$ and $\hat{T}_j(s)$ are the value of earliness and tardiness of job j in the current assignment under scenario s, which can be calculated by applying SiPSi C libraries. Let S_k be the set of jobs assigned to machine k in the solution of the MP. Let $\hat{w}_k(s)$ represent the weighted penalty cost of jobs' tardiness and earliness of jobs assigned to machine k under scenario s. $\hat{w}_k(s)$ can be formulated as (27). The optimality Benders cut can be formulated as cut (28).

$$\hat{w}_k(s) = \sum_{j\in S_k}\left(\theta_j^E \cdot \hat{E}_j(s) + \theta_j^T \cdot \hat{T}_j(s)\right) \tag{27}$$

$$w_k(s) \geq \hat{w}_k(s)\left(\sum_{j\in S_k} x_{jk} - |S_k| + 1\right), \forall k \in M, s \in S. \tag{28}$$

The MP is then continued to be solved upon adding the Benders cuts. The above procedure is repeated until the method converges to optimality or a given time limit is reached.

We generate instances using the same scheme proposed in Liu et al. (2021). We then solve these instances using SAA-C and the SLBBD methods. The computation time for each run is limited to 3600 seconds. The results are presented in Table 4, from which we can observe that the average computation time of SAA-C is 3180 seconds, about 30.25% larger than the SLBBD method. The average objective value obtained by the SAA-C method is 94.28, about 53.35% larger than that obtained by the SLBBD method. Moreover, the average variance (V), average 85th percentiles (PT^{85}), and average 99th percentiles (PT^{99}) of the objective values obtained by the SAA-C method are all significantly larger than those obtained by the SLBBD method. Therefore, the results demonstrate that the proposed SLBBD method outperforms the SAA-C method.

Table 4 Numerical results for the S-PMSP instances

		The SAA-C Method					The SLBBD Method				
set	n	obj_1	V_1	PT_1^{85}	PT_1^{99}	T_1 (s)	obj_2	V_2	PT_2^{85}	PT_2^{99}	T_2 (s)
1	4	8.97	2.74	11.98	17.68	9	8.97	2.74	11.98	17.68	2
2	5	14.58	1.57	16.23	18.51	51	14.58	1.57	16.23	18.51	4
3	6	15.72	2.94	19.06	24.72	3600	15.72	2.94	19.06	24.72	9
4	7	16.18	0.89	16.81	19.97	3600	14.58	1.57	16.23	18.51	10
5	8	68.91	5.37	74.06	79.13	3600	47.63	6.80	54.82	64.02	283
6	9	96.33	8.56	105.04	115.61	3600	68.93	8.84	77.94	91.09	854
7	10	63.58	9.51	73.99	86.43	3600	56.07	6.20	62.21	71.68	3600
8	11	85.23	4.41	89.35	95.63	3600	37.61	3.61	40.84	49.57	3600
9	12	84.82	10.65	96.14	113.75	3600	56.00	8.67	65.05	77.57	3600
10	13	188.91	11.18	200.37	218.80	3600	98.58	15.26	115.05	133.77	3600
11	14	110.21	6.94	117.59	129.65	3600	84.16	7.47	91.86	105.79	3600
12	15	104.13	8.45	113.49	124.72	3600	65.77	5.45	70.9	79.35	3600
13	16	125.51	14.75	141.59	162.49	3600	81.25	9.39	91.96	104.58	3600
14	17	132.97	11.13	144.75	163.94	3600	103.63	6.91	111.08	120.3	3600
15	18	159.58	16.1	176.99	196.8	3600	57.11	5.51	63.12	70.03	3600
16	19	200.24	10.97	211.44	226.93	3600	135.00	12.22	148.61	163.33	3600
17	20	126.92	7.70	134.35	148.18	3600	99.66	6.84	106.46	118.28	3600
Average		94.28	7.87	102.54	114.29	3180	61.48	6.59	68.44	78.16	2398

In summary, we conclude that 1) the proposed model and SLBBD method can provide practical optimal solutions for the studied SDASP; 2) the SLBBD method outperforms the SAA-C method in solving both the SDASP and the S-PMSP; and 3) the SDASP becomes more challenging to solve when the uncertain level of processing time increases.

6. Conclusion

We have investigated the stochastic distributed appointment assignment and scheduling problem. The aim is to provide an optimal plan that minimizes the total travel costs and the expected makespan. We have considered two critical uncertain factors, i.e., the uncertain appointment service time and release time. By analyzing the problem, we have abstracted the problem as a parallel machine scheduling problem with release dates to minimize the total cost and expected makespan. The studied problem is complex due to its NP-hard and stochastic nature. We first propose a two-stage stochastic programming formulation for this problem. The first stage makes assignments decisions without realizing the service and release times of appointments, and the second stage optimizes the makespan with the assignments variables given and fixed. We apply the widely-used

SAA method to reformulate the problem with a set of sample scenarios to make the problem tractable. The reformulated SAA model can be directly solved using off-the-shelf solvers, denoted as the SAA-C method. However, the SAA-C method loses efficiency when the instance size becomes large. We develop a novel SLBBD method that decomposes the approximate SAA model into an MP and SP to handle practical-sized instances. Benders cuts are added to the MP upon finding each feasible solution of the SP. We perform extensive numerical experiments to validate the proposed model and solution methods. Results on the SDAASP instances show that our proposed model can provide optimal solutions. In particular, our SLBBD manages to solve practical-sized instances with up to 100 appointments within an acceptable time limit. We further extend the model to consider the due date of appointments to minimize the expected earliness and tardiness. We devise our SLBBD method to solve this variant. Numerical results consistently demonstrate the excellent performance of the SLBBD method. These experiments confirm that the SLBBD method is efficient in solving such kinds of scheduling problems under uncertainty.

The studied distributed appointment assignment and scheduling problem is a timely and essential topic and finds a wide range of applications in healthcare, production, and transportation systems. The current study focus on a single period planning. Future research may involve multiple periods. In this case, decision-makers must assign appointments to different dates and service centers, and decide the operational time and work schedule of each service center. Efficient algorithms should be developed to solve this variant. In addition, We find that the SLBBD method degenerates quickly as the number of testing sites increases. Therefore, a future study may focus on developing efficient heuristics to address large-sized instances.

Data availability statement

The data that support the findings of this study are openly available at `https://www.dmu-yantongli.com/instances`.

Acknowledgment

This work was partly supported by the National Natural Science Foundation of China (NSFC) under grants 72201044 and 71971036, the Humanities and Social Science Foundation of the Chinese Ministry of Education under Grant 22YJC630071, the China Postdoctoral Science Fund under grant 2022M710018, and the Canadian Natural Sciences and Engineering Research Council (NSERC) under grant 2019-00094. These supports are greatly acknowledged.

References

Z. A. Abdalkareem, A. Amir, M. A. Al-Betar, P. Ekhan, and A. I. Hammouri. Healthcare scheduling in optimization context: a review. *Health and Technology*, pages 1–25, 2021.

R. Alizadeh, J. Rezaeian, M. Abedi, and R. Chiong. A modified genetic algorithm for non-emergency outpatient appointment scheduling with highly demanded medical services considering patient priorities. *Computers & Industrial Engineering*, 139:106106, 2020.

A. Aydilek, H. Aydilek, and A. Allahverdi. Production in a two-machine flowshop scheduling environment with uncertain processing and setup times to minimize makespan. *International Journal of Production Research*, 53(9):2803–2819, 2015.

L. S. Bastos, J. F. Marchesi, S. Hamacher, and J. L. Fleck. A mixed integer programming approach to the patient admission scheduling problem. *European Journal of Operational Research*, 273(3):831–840, 2019.

D. Bertsimas, V. Gupta, and N. Kallus. Robust sample average approximation. *Mathematical Programming*, 171(1):217–282, 2018.

D. Cao, M. Chen, and G. Wan. Parallel machine selection and job scheduling to minimize machine cost and job tardiness. *Computers & Operations Research*, 32(8):1995–2012, 2005.

S. Ceschia, R. Guido, and A. Schaerf. Solving the static inrc-ii nurse rostering problem by simulated annealing based on large neighborhoods. *Annals of Operations Research*, 288(1):95–113, 2020.

S. De Vuyst, H. Bruneel, and D. Fiems. Computationally efficient evaluation of appointment schedules in health care. *European Journal of Operational Research*, 237(3):1142–1154, 2014.

P. Demeester, W. Souffriau, P. De Causmaecker, and G. V. Berghe. A hybrid tabu search algorithm for automatically assigning patients to beds. *Artificial Intelligence in Medicine*, 48(1):61–70, 2010.

A. Diamant, J. Milner, and F. Quereshy. Dynamic patient scheduling for multi-appointment health care programs. *Production and Operations Management*, 27(1):58–79, 2018.

A. Dolgui, D. Ivanov, S. P. Sethi, and B. Sokolov. Scheduling in production, supply chain and industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications. *International Journal of Production Research*, 57(1-2):411–432, 2018.

O. Elci and J. Hooker. Stochastic planning and scheduling with logic-based benders decomposition. *arXiv preprint arXiv:2012.14074*, 2020.

J. Feldman, N. Liu, H. Topaloglu, and S. Ziya. Appointment scheduling under patient preference and no-show behavior. *Operations Research*, 62(4):794–811, 2014.

X. Feng, F. Zheng, and Y. Xu. Robust scheduling of a two-stage hybrid flow shop with uncertain interval processing times. *International Journal of Production Research*, 54(12):3706–3717, 2016.

C. Guo, M. Bodur, D. M. Aleman, and D. R. Urbach. Logic-based benders decomposition and binary decision diagram based approaches for stochastic distributed operating room scheduling. *INFORMS Journal on Computing*, page available online, 2021.

C. M. Hino, D. P. Ronconi, and A. B. Mendes. Minimizing earliness and tardiness penalties in a single-machine problem with a common due date. *European Journal of Operational Research*, 160(1):190–201, 2005.

J. N. Hooker. Planning and scheduling by logic-based benders decomposition. *Operations Research*, 55(3): 588–602, 2007.

J. N. Hooker. Logic-based benders decomposition for large-scale optimization. In *Large Scale Optimization in Supply Chains and Smart Manufacturing*, pages 1–26. Springer, Heidelberg, 2019.

Y. Li, Y. Li, J. Cheng, and P. Wu. Order assignment and scheduling for personal protective equipment production during the outbreak of epidemics. *IEEE Transactions on Automation Science and Engineering*, 19(2):692–708, 2021.

Y. Li, X. Wen, T.-M. Choi, and S.-H. Chung. Optimal establishments of massive testing programs to combat covid-19: A perspective of parallel-machine scheduling-location (scheloc) problem. *IEEE Transactions on Engineering Management*, 2022.

M. Liu and X. Liu. Machine scheduling with stochastic release and processing times. *IFAC-PapersOnLine*, 52(13):2116–2121, 2019.

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. *International Journal of Production Research*, 54(11-12):1–12, 2016.

M. Liu, X. Liu, F. Chu, F. Zheng, and C. Chu. Service-oriented robust parallel machine scheduling. *International Journal of Production Research*, 57(12):3814–3830, 2019a.

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. *International Journal of Production Research*, 57(6):1706–1723, Mar. 2019b.

X. Liu, F. Chu, F. Zheng, C. Chu, and M. Liu. Parallel machine scheduling with stochastic release times and processing times. *International Journal of Production Research*, 59(20):6327–6346, 2021.

J. Marynissen and E. Demeulemeester. Literature review on multi-appointment scheduling problems in hospitals. *European Journal of Operational Research*, 272(2):407–419, 2019.

J. H. May, W. E. Spangler, D. P. Strum, and L. G. Vargas. The surgical scheduling problem: Current research and future opportunities. *Production and Operations Management*, 20(3):392–405, 2011.

M. S. S. Mir and J. Rezaeian. A robust hybrid approach based on particle swarm optimization and genetic algorithm to minimize the total machine load on unrelated parallel machines. *Applied Soft Computing*, 41:488–504, 2016.

F. Mischek and N. Musliu. Integer programming model extensions for a multi-stage nurse rostering problem. *Annals of Operations Research*, 275(1):123–143, 2019.

J. Peng and B. Liu. Parallel machine scheduling models with fuzzy processing times. *Information Sciences*, 166(1-4):49–66, 2004.

M. Pinedo and K. Hadavi. Scheduling: theory, algorithms and systems development. In *Operations Research Proceedings 1991*, pages 35–42. Springer, 1992.

M. Pinedo and K. Hadavi. *Scheduling : Theory, Algorithms, and Systems*, volume 29. Springer, Heidelberg, 2012.

M. Ranjbar, M. Davari, and R. Leus. Two branch-and-bound algorithms for the robust parallel machine scheduling problem. *Computers & Operations Research*, 39(7):1652–1660, 2012.

H. G. Santos, T. A. Toffolo, R. A. Gomes, and S. Ribas. Integer programming techniques for the nurse rostering problem. *Annals of Operations Research*, 239(1):225–251, 2016.

X. Shen, Y. Sun, and M. Zhang. An improved moea/d for multi-objective flexible job shop scheduling with release time uncertainties. In *2016 IEEE Congress on evolutionary computation (CEC)*, pages 2950–2957. IEEE, 2016.

B. Shnits, I. Bendavid, and Y. N. Marmor. An appointment scheduling policy for healthcare systems with parallel servers and pre-determined quality of service. *Omega*, 97:102095, 2020.

A. O. Sigurpalsson, T. P. Runarsson, and R. J. Saemundsson. Stochastic master surgical scheduling under ward uncertainty. In *International Conference on Human-Centred Software Engineering*, pages 163–176, 2020.

M. Skutella, M. Sviridenko, and M. Uetz. Unrelated machine scheduling with stochastic processing times. *Mathematics of operations research*, 41(3):851–864, 2016.

M. Soltani, M. Samorani, and B. Kolfal. Appointment scheduling with multiple providers and stochastic service times. *European Journal of Operational Research*, 277(2):667–683, 2019.

S. Tanaka and S. Fujikuma. A dynamic-programming-based exact algorithm for general single-machine scheduling with machine idle time. *Journal of Scheduling*, 15(3):347–361, 2012. ISSN 1094-6136; 1099-1425/e.

X. Tang, K. Li, and F. Wu. Approximation for preemptive scheduling stochastic jobs on identical parallel machines. In *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, pages 1287–1290. IEEE, 2010.

S. A. Torabi, N. Sahebjamnia, S. A. Mansouri, and M. A. Bajestani. A particle swarm optimization for a fuzzy multi-objective unrelated parallel machines scheduling problem. *Applied Soft Computing*, 13(12): 4750–4762, 2013.

T. T. Tran, A. Araujo, and J. C. Beck. Decomposition methods for the parallel machine scheduling problem with setups. *INFORMS Journal on Computing*, 28(1):83–95, 2016.

A. M. Turhan and B. Bilgen. Mixed integer programming based heuristics for the patient admission scheduling problem. *Computers & Operations Research*, 80:38–49, 2017.

A. M. Turhan and B. Bilgen. A hybrid fix-and-optimize and simulated annealing approaches for nurse rostering problem. *Computers & Industrial Engineering*, 145:106531, 2020.

X. Wu and A. Che. Energy-efficient no-wait permutation flow shop scheduling by adaptive multi-objective variable neighborhood search. *Omega*, 94:102117, 2020.

W. Xie and S. Ahmed. Distributionally robust simple integer recourse. *Computational Management Science*, 15(3):351–367, Oct. 2018. ISSN 1619-6988.

X. Xu, W. Cui, J. Lin, and Y. Qian. Robust makespan minimisation in identical parallel machine scheduling problem with interval data. *International Journal of Production Research*, 51(12):3532–3548, 2013.

S. Yu, V. G. Kulkarni, and V. Deshpande. Appointment scheduling for a health care facility with series patients. *Production and Operations Management*, 29(2):388–409, 2020.

F. Yue, S. Song, Y. Zhang, J. N. Gupta, and R. Chiong. Robust single machine scheduling with uncertain release times for minimising the maximum waiting time. *International Journal of Production Research*, 56(16):5576–5592, 2018.

C. Zhang, Y. Li, J. Cao, and X. Wen. On the mass covid-19 vaccination scheduling problem. *Computers & Operations Research*, 141:105704, 2022.

F. Zheng, X. Man, F. Chu, M. Liu, and C. Chu. A two-stage stochastic programming for single yard crane scheduling with uncertain release times of retrieval tasks. *International Journal of Production Research*, 57(13):4132–4147, 2019.

S. Zhou, D. Li, and Y. Yin. Coordinated appointment scheduling with multiple providers and patient-and-physician matching cost in specialty care. *Omega*, 101:102285, 2021.