European Journal of Operational Research 310 (2023) 100–116

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Dynamic scheduling of patients in emergency departments

Thiago Alves de Queiroz [a,*], Manuel Iori [b], Arthur Kramer [c,d], Yong-Hong Kuo [e]

[a] Institute of Mathematics and Technology, Federal University of Catalão, Catalão, 75704-020, GO, Brazil
[b] Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Reggio Emilia, 42122, Italy
[c] Departament of Production Engineering, Federal University of Rio Grande do Norte, Natal, 59077-080, RN, Brazil
[d] Mines Saint-Étienne, Univ. Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, Saint-Étienne F-42023, France
[e] Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, China

ARTICLE INFO

Article history:
Received 22 September 2021
Accepted 3 March 2023
Available online 9 March 2023

Keywords:
Scheduling
Healthcare
Emergency department
Variable neighborhood search
Weighted tardiness

ABSTRACT

Emergency department overcrowding is a global issue that poses a great threat to patient health and safety. The timeliness of medical services provided to patients is crucial to emergency departments as it directly impacts the mortality and morbidity of urgent patients. However, critical resources (e.g., doctors and nurses) are typically constrained due to the limited financial budget. Thus, hospital administrators may need to investigate solutions to improve the efficiency of the emergency department. In this work, we study the dynamic problem of scheduling patients to doctors, aiming at minimizing the total weighted tardiness. We propose a simple reoptimization heuristic based on multiple queues of patients in accordance with their urgency levels, and then combine it with an effective variable neighborhood search. We also propose a scenario-based planning approach that uses sampled scenarios to anticipate future events and the variable neighborhood search to schedule patients. The methods are adapted to handle a problem variant where information on arrival time and urgency level of some patients can be received in advance by the emergency department. With a comprehensive computational study on two sets of realistic instances from Hong Kong SAR of China and Italy, we validate the performance of the proposed methods, evaluating the benefits of having more doctors and receiving early information.

1. Introduction

Emergency Department (ED) overcrowding has been continuously reported for decades in various regions in the world (see, e.g., Di Somma et al. 2015; Richardson 2006). Since the ED is the 24/7 gateway to the hospital for patients who require immediate emergency medical services, congestion within the facility may prevent those patients from accessing the required treatments in a timely manner. Such delays in the necessary medical treatments can lead to life-altering (or even life-ending) cases. Other adverse consequences of ED overcrowding include public safety at risk, prolonged suffering, patient dissatisfaction, violence in the ED waiting room, and increased chances of decision errors (Derlet & Richards, 2000). While one of the most effective ways to alleviate the ED overcrowding situation is to expand the ED capacity, this may not be feasible for most hospitals due to financial constraints. Therefore, hospital managers have kept investigating possible ways to improve patient flow by optimizing the ED processes and operations.

The flow of patients and the prolonged patient waiting times within EDs have been the subject of intense study and discussion. This topic is relevant to a variety of optimization problems, ranging from the scheduling of work shifts to the minimization of service costs. Our research is motivated by the recent advance in information technologies adopted by EDs. With modern information systems at the hospital, most of the activities within the ED are now tracked in real time for more effective communications and responsive actions. Traditionally, as information might not be updated and comprehensive, protocols were set up to guide daily operations (e.g., patient prioritization, medical professional assignment, and process flow). A conventional and commonly adopted practice was (and still is) to prioritize patients according to their urgency level. While this is a sensible way of giving priority to patients, the utilization of additional information about patients and resource availability may further enhance the efficiency of the ED and the effectiveness of emergency medical services provided to patients, thereby leading to better outcomes (see, e.g., Saghafian, Hopp, Van Oyen, Desmond, & Kronick 2014).

This paper addresses the ED patient scheduling problem, where information about the patient characteristics (arrival times, ser-

* Corresponding author.
E-mail addresses: taq@ufcat.edu.br (T. Alves de Queiroz), manuel.iori@unimore.it (M. Iori), arthur.kramer@ufrn.br, arthur.kramer@emse.fr (A. Kramer), yhkuo@hku.hk (Y.-H. Kuo).

https://doi.org/10.1016/j.ejor.2023.03.004
0377-2217/

vice times, and urgency levels) and the availability of doctors is revealed dynamically on a time horizon. We focus on the minimization of the weighted tardiness of patients, by optimizing their schedules. This is a challenging optimization problem, because of the large number and wide heterogeneity of the patients that arrive at the ED. We propose different methods to solve the problem, ranging from simple reoptimization heuristics to more complex scenario-based approaches, also making use of inner metaheuristic components. We also study the influence of the number of doctors and the possibility of receiving early information on patients as ways to improve the efficiency of the ED.

1.1. Our contributions

The dynamic patient scheduling problem we present involves a number of questions whose answers would help a decision-maker to better manage an ED, for instance:

- How many doctors are needed to efficiently provide essential medical care to the patients?
- How can we develop effective schedules for patients?
- Can the utilization of shared (early) information be helpful to devise better schedules?

We propose solution methods that can guide the decision-maker to adequate answers to these questions. In particular, our contributions rely on:

- a strong research motivation that originates from two large EDs, one located in the Hong Kong Special Administrative Region (SAR) of China and the other in the northern part of Italy. Our problem definition derives indeed from the real-world activities performed in these hospitals, and our computational tests are based on realistic instances derived from real-world data;
- the proposal of predictive-reactive and event-driven solution methods:
  - a simple reoptimization heuristic that schedules patients to doctors according to the patients' urgency levels. The most urgent patients are scheduled as soon as possible. We also investigate the case where patients receive care based on their release time, less urgent patients have their urgency level increased as time goes on to avoid excessive waiting times, and doctors can take a break between two visits in the presence of only low-priority patients;
  - an improved heuristic that optimizes the schedule with a Variable Neighborhood Search (VNS) whenever an event happens (e.g., the arrival of a patient in the ED). Past decisions and decisions in progress are fixed, while the VNS (re)schedules patients already assigned to doctors with the newly revealed patients;
  - an online anticipatory algorithm, which makes decisions online using samples of the future, i.e., a Scenario-Based Planning Approach (SBPA) that uses sampled scenarios with fictive patients generated according to the probability distributions inferred from historical data. The VNS obtains a schedule for each scenario, and a consensus function calculates the score of each schedule, providing high scores to schedules containing the most common decisions in terms of the assignment of patients to doctors. The idea is to anticipate future decisions concerning the possible arrival of urgent patients;
- the study of the problem variant in which (part of) the triage is anticipated during the transportation of patients. The ED receives a phone call with the patient's information, and so the patient can be inserted at an earlier stage in the scheduling process. This may be advantageous, especially when urgent patients are expected to arrive;
- the formal mathematical modeling of the deterministic (offline) problem, where all information is supposed to be known in advance, by means of an arc-flow formulation, used to assess the quality of the solution produced by the heuristic methods.

We test all solution methods on realistic instances from the two hospitals, carrying out more than 30,000 computational runs. The simple reoptimization heuristic works well only when the number of doctors is large. Its combination with the VNS brings relevant improvements in the reduction of weighted tardiness, without requiring much computing time. The SBPA leads to further improvements, but at the expense of a non-negligible computational effort. Additional numerical experiments are conducted to analyze the main parameters of the VNS and its neighborhood structures. We also study the influence of the number of scenarios, the size of the time horizon for fictive patients, and the benefits of using early information about patients. Overall, we believe these computational results represent an essential contribution to the important problem of dynamic scheduling of patients in EDs.

The remainder of the paper is organized as follows. Section 2 reviews the research studies in ED operations and dynamic scheduling. Section 3 presents the formal problem description. Sections 4 and 5 outline the solution methods that we developed for the deterministic and dynamic problems, respectively. The performance of the proposed methods is demonstrated through computational experiments in Section 6. Conclusions and future research directions are provided in Section 7.

2. Literature review

There is a vast literature that investigates how to improve operations in EDs. In this section, we limit our discussion to the contributions that are most relevant to our paper, and we refer to Saghafian, Austin, & Traub (2015) for a more comprehensive review.

The intervention strategies suggested by existing research studies and adopted in practice for patient prioritization are mainly based on the triage category (Fernandes et al., 2005) – typically associated with the level of urgency or patient criticality – assessed and assigned by a triage nurse. The more urgent a patient, the higher her priority is. While this practice is reasonable and popularly adopted, there has been clinical research showing that factoring patient complexity (e.g., in terms of the number of consultations and procedures needed) into the triage assessment can help improve ED key performance indicators (KPIs) (Ieraci, Digiusto, Sonntag, Dann, & Fox, 2008). By using both an analytical framework and a simulation model, Saghafian et al. (2014) also found that a complexity-augmented triage scheme, which takes the patient complexity into account for patient prioritization, can improve both patient safety and increase operational efficiency.

For example, fast tracks and patient streaming are relevant mechanisms that share similar concepts. Fast tracks are typically dedicated to patients of lower acuity, who are expected to require a shorter service time. Kuo, Leung, Graham, Tsoi, & Meng (2018) adopted a simulation approach to examine the impacts of the adoption of a fast track. They found that the fast track is more effective for EDs that have a higher proportion of urgent patients. Patient streaming is an idea that identifies patients who are likely to be discharged from the ED or to be admitted to the hospital (see, e.g., King, Ben-Tovim, & Bassham 2006). Saghafian, Hopp, Van Oyen, Desmond, & Kronick (2012) proposed a "virtual-streaming" scheme, where resources are also shared across the streams. They provided conditions for which this streaming scheme is particularly effective. However, recent research found that patient complexity is still not a major factor considered in

patient prioritization decisions (Ding, Park, Nagarajan, & Grafstein, 2019). In this paper, we aim to embed patient complexity, represented by urgency levels and stochastic service times, in our patient scheduling decisions for improving ED efficiency.

Due to the multi-dimensional sources of uncertainty (e.g., patient arrival processes, required medical treatments, and durations of the activities), the high variations in the realized outcomes, and the complexity of the system, simulation modeling is the most popular approach to providing practical solutions for managing ED operations. For a review of the recent developments of ED simulation modeling, we refer the reader to Vanbrabant, Braekers, Ramaekers, & Van Nieuwenhuyse (2019). There have been various applications of simulation modeling for improving ED operations. Hoot et al. (2008) developed a discrete event simulation model with the use of patient data to forecast near-term ED KPIs (e.g., the waiting count, waiting time, and length of stay). Kuo, Rado, Lupia, Leung, & Graham (2016) developed a simulation model of an ED in Hong Kong SAR of China (one of the two EDs that motivate this research) and tackled the challenges of data incompleteness with a simulation-optimization approach. By using the simulation model, they examined the effect of a number of intervention strategies (e.g., expanding capacity, staggered shifts, and the recruitment of nurse practitioners) and studied the trade-offs between different KPIs. Oh et al. (2016) applied a simulation model to assess the outcomes of the redesign of an ED in Florida, USA. They studied potential improvement areas such as resource allocation, ED process flow, and other operational policies. Dosi, Iori, Kramer, & Vignoli (2019, 2020, 2021) studied the overcrowding problem faced by an ED in the northern part of Italy (the other ED that motivates this research). They developed a simulation model, and proposed and tested a set of what-if scenarios evaluating them in terms of reduction of patients' waiting times and lengths of stay. Queuing models have also been used in analyzing ED operations (e.g., Huang, Carmeli, & Mandelbaum 2015; Kamali, Tezcan, & Yildiz 2018). The main goals of these studies are to derive theoretical properties of the models and managerial insights into optimizing ED processes.

Scheduling has been an important topic in the optimization of ED systems. Most of the research has been performed on staff scheduling, e.g., physicians (Camiat, Restrepo, Chauny, Lahrichi, & Rousseau, 2021) and nurses (Legrain, Omer, & Rosat, 2020), material resources allocation, such as beds (Bastos, Marchesi, Hamacher, & Fleck, 2019) and operating rooms (Bovim, Christiansen, Gullhav, Range, & Hellemo, 2020), and activities planning, including surgeries (Tsai, Yeh, & Kuo, 2021) and appointments (Jiang, Abouee-Mehrizi, & Diao, 2020; Pan, Geng, & Xie, 2021). Because of the stochastic environment of EDs, queuing and simulation models have been incorporated into optimization frameworks. For example, Green, Soares, Giglio, & Green (2006) used a lag stationary independent period-by-period queuing model to determine the number of providers in each hour. Their approach could reduce the effects of the time lags in peak hours caused by the time-varying arrival rates. Ahmed & Alkhamis (2009) proposed a simulation-optimization framework to determine the right resource levels in the ED (e.g., numbers of physicians, lab technicians, and nurses) with the objective to optimize patient throughput and total time in the system. Kuo (2014) integrated an ED simulation model into a simulated annealing to determine the number of physicians required. He, Sim, & Zhang (2019) proposed a hybrid robust-stochastic framework, with a family of uncertainty sets, to optimize ED patient scheduling decisions, considering the randomness of patients' arrival, consultation times, and heterogeneous physicians. Lee & Lee (2020) used a deep reinforcement learning algorithm to schedule patients of different weights to appropriate medical resources, aiming at minimizing the weighted waiting time of patients.

Despite the relevance of dynamic ED patient scheduling problems, the number of studies addressing such areas is relatively limited. In the survey by Ouelhadj & Petrovic (2009), the authors emphasized that dynamic scheduling approaches can be classified into three main categories: reactive, predictive-reactive, and robust proactive. The authors also highlighted predictive-reactive as the most common category in the literature. According to this approach, an initial schedule is built and then adjusted (rescheduled) once new information is available. The rescheduling activity seeks to maintain the solutions aligned to the chosen optimization criterion even after the occurrence of unexpected events. Moreover, two main decisions are associated with the rescheduling problem. The first is the time to reschedule, and the second is how to reschedule. Regarding time, three main policies are possible: periodic, event-driven, and a combination of both. In the periodic policy (also known as the rolling horizon), the solution is adjusted at fixed intervals. In the event-driven policy (which is the policy we adopt in this work), rescheduling is required each time a new event (e.g., the arrival of a patient at the ED) changes the system status.

For what concerns dynamic scheduling, a general overview has been proposed by Vieira, Herrmann, & Lin (2003). Gupta, Maravelias, & Wassick (2016) reviewed the literature on rescheduling activities covering motivation, main issues to be addressed, and the most popular solution approaches. The authors also emphasized the importance of reviewing the system every time a new event takes place, which is the key characteristic of the event-driven policy. Another branch of research that is highly relevant for our study is that of (semi-)online scheduling problems (Albers & Hellwig, 2012; Epstein, 2018), where makespan (or another objective function) is to be minimized given that only partial information is available to the decision maker. These problems have been considered in patient scheduling and in other healthcare settings, e.g., in pathology laboratories (Azadeh, Baghersad, Farahani, & Zarrin, 2015). Besides that, simple sequencing rules that require full information about jobs (i.e., offline) may no longer be applied (e.g., Arkin & Roundy, 1991; Shim & Kim, 2007; Souayah, Kacem, Haouari, & Chu, 2009). Research efforts have been devoted to tackling scheduling problems in both offline and online settings to compare their solution qualities (e.g., Albers & Hellwig, 2012; Angelelli, Speranza, & Tuza, 2008; Chen, Sterna, Han, & Blazewicz, 2016; Ng, Tan, He, & Cheng, 2009). Furthermore, the problem we are facing is a combination of stochastic programming and online optimization (i.e., an online stochastic optimization problem) as denoted by Bakker, Dunke, & Nickel (2020).

Recently, Larsen & Pranzo (2019) proposed a general framework that models dynamic rescheduling problems. Basically, the framework consists of (a) executing the solver to obtain a solution for a deterministic problem based on the available data at a given time, (b) monitoring the behavior of the system under the implementation of the solution, and taking into account the disruptions that may occur (based on a simulator), (c) deciding when rescheduling is necessary, (d) creating a new deterministic problem, and (e) re-invoking the solver to obtain a new solution. The authors applied the framework to a job shop scheduling problem. A similar approach has been discussed in Rossit, Tohmé, & Frutos (2019), who highlighted the importance of dynamic scheduling in the context of Industry 4.0.

While there is a rich literature on both ED operations and dynamic scheduling, to the best of our knowledge, our research is the first study that proposes dynamic algorithms for scheduling patients in EDs with the aim of minimizing total weighted tardiness (apart from our preliminary research in Queiroz, Iori, Kramer, & Kuo (2021), that presented preliminary results by early versions of the simple and VNS-based reoptimization heuristics). The problem we face differs from the existing dynamic scheduling problems noticed in other applications and is particularly challenging in the

sense that: (i) patient arrivals are uncertain, (ii) patients are heterogeneous because of their triage categories (i.e., urgency levels), and consultation times, (iii) the triage category of a patient is not revealed before her arrival, and (iv) there are service targets (i.e., target response times for different triage categories) that guide the optimization process. Our research aims to address this significant and practical problem and foster further research in this relevant area.

3. Formal problem description

We address the problem of assigning patients to doctors in an ED. We consider a time horizon that starts at zero and ends in T. The patients arrive dynamically at the ED during this time horizon. More in detail, we are given a set $J = \{1, \ldots, n\}$ of patients to be visited by a set $M = \{1, \ldots, m\}$ of doctors. Doctors work in parallel and each of them can visit at most one patient at a time. Each patient has to be uniquely assigned to a doctor. The visit, once started, cannot be interrupted (so, in scheduling terms, no preemption of the activities is allowed).[1] We suppose doctors are equally efficient, so the time required for a visit (denoted as processing time in the following) is independent of the doctor.

Each patient j arrives at the ED at time r_j, which is known as the release date or release time in scheduling terms and can be immediately visited by a doctor. The patient is assigned an urgency level (i.e., a triage category) denoted by u_j. We are given a list of ℓ possible triage categories, so $u_j \in \{1, \ldots, \ell\}$. On the basis of her urgency level, the patient is also assigned a priority weight w_j, a target due time d_j for the beginning of her visit, and an expected processing time p_j^a for her visit. The weight w_j gives an estimation of the importance of the patient (the more critical the urgency level of the patient is, the higher the weight is) and is used in the objective function to be minimized. The due time d_j (or due date in scheduling terms) follows some internal/governmental protocols that impose maximum waiting times for the different urgency levels. This implies that a patient arriving in r_j has due time $d_j = r_j + \delta_j$, where δ_j is the maximum waiting time associated with the urgency level of j. When the visit begins after the due time, then a tardiness T_j is created. The value of T_j is computed as $T_j = \max\{0; s_j - d_j\}$, where s_j is the beginning of the visit to patient j. The expected processing time p_j^a is computed once the patient arrives on the basis of her urgency level. The effective (realized) processing time is known only after the doctor finishes providing the medical services, and is denoted by p_j^e. Thus, the completion time C_j of the visit of j is known only at the end of the visit and results in $C_j = s_j + p_j^e$. As soon as a doctor completes a visit, she can start the next one. The Dynamic Scheduling of Patients with identical parallel Doctors, release dates, and non-deterministic service times (DSPD) requires assigning all patients to doctors by minimizing the total weighted tardiness, computed as $\sum_{j \in J} w_j T_j$.

The choice of a weighted tardiness minimization objective function is made based on the main characteristics of the problem under study. It is straightforward to translate the internal/governmental protocols concerning maximum waiting times into due dates for patients that depend on their urgency levels. Thus, the definition of a due date-based objective function seemed appropriate. On one hand, since serving patients earlier than their due dates is not discouraged, penalizing earliness (in addition to tardiness) in the objective function does not represent the reality of EDs. Another option would be to consider a lateness-based objective function, since in this case serving patients earlier than their due dates would have a positive impact on the total cost of a given solution. In this case, a situation where a solution with a late high-urgency patient and an early low-urgency patient could be equivalent (in terms of cost) to a solution where both are visited on time. However, in practice, late patients mean a higher risk of death, which is not acceptable at all. In this way, a lateness-based objective function is not considered. Another option would be to consider a late-work-based objective function. In the late work situation, only the amount of work performed after the due dates are penalized, in other words, the waiting time of patients would not be penalized if this criterion would be selected. Since waiting times play an important role in the ED context, this possibility is also not considered in our problem. Therefore, the total weighted tardiness is a very appropriate objective function for the problem under study.

The DSPD has two types of dynamic events that affect the status of the system: (i) a patient arrives and there is a doctor available to visit her; and (ii) a doctor finishes visiting a patient and the list of waiting patients is not empty. Whenever any of these events occur, we need to take a decision on the next assignments. In our approach, this is obtained by the optimization algorithms of Section 5.

In this paper, we also solve the static deterministic problem (also known as offline or utopic problem in the literature), where all information is precisely known in advance. Solving this problem is important because it allows us to acquire relevant information to assess the quality of the solutions obtained for the dynamic (online) stochastic problem. We attempt the solution to the deterministic problem both exactly and heuristically, as outlined in the next section. If we consider the deterministic problem and adopt the scheduling classification of Graham, Lawler, Lenstra, & Kan (1979), then the DSPD can be defined as $P|r_j|\sum w_j T_j$. This problem requires to schedule jobs with release dates on identical parallel machines by minimizing the total weighted tardiness. The $P|r_j|\sum w_j T_j$ is $\mathcal{NP}$-hard because it is a generalization of the $1|r_j|\sum w_j C_j$, which was proven $\mathcal{NP}$-hard by Lenstra, Kan, & Brucker (1977).

4. Solution of the static deterministic problem

Despite the fact that the deterministic $P|r_j|\sum w_j T_j$ is a classical scheduling problem, the literature concerning heuristics and/or exact methods to solve this specific problem is quite scarce. We can highlight the contributions of Baptiste, Jouglet, & Savourey (2008) that proposed several lower bounds for identical parallel machine scheduling problems with release dates and min-sum objective functions, including the total weighted tardiness minimization; Jouglet & Savourey (2011) that proposed dominance rules; and Kacem, Souayah, & Haouari (2012) that developed a branch-and-bound able to solve instances with up to 50 jobs and 4 machines.

However, many works in the literature address problems that are closely related to the $P|r_j|\sum w_j T_j$. For example, the $P|r_j|\sum w_j C_j$ can be obtained by considering the due dates of all jobs as zero ($d_j = 0,\ \forall j \in J$), the $P||\sum w_j T_j$ by assuming that there are no release dates for the jobs; and the $R|r_j|\sum w'_j E_j + w_j T_j$ by considering heterogeneous machines and adding weighted earliness penalization. Concerning the proposition of lower bound

[1] We remark that, in reality, preemption is sometimes allowed in the EDs we collaborate with. Depending on the levels of the patients' urgency, the doctor may finish the consultation with the current patient first, or choose to temporarily interrupt it and provide service to an urgent patient. In the literature, both preemptive-priority (Kwasnick, 2017) and non-preemptive-priority (Granja, Almada-Lobo, Janela, Seabra, & Mendes, 2014) queues have been applied for ED operations. In particular, Luo, Bayati, Plambeck, & Aratow (2017) and Wen, Geng, & Xie (2020) study the performance of both types of priority queues in EDs. We made the assumption of non-preemptive-priority queues in our environment because (i) preemption is rare to take place (e.g., critical patients are less than 3% of the overall patient volume) and (ii) service times for less urgent patients (around 10 minutes) are much shorter than those for critical patients (over 30 minutes) and so the impact of preemption is not very significant.

schemes, Kedad-Sidhoum, Solis, & Sourd (2008) and Şen & Bülbül (2015) tackled the $R|r_j|\sum w'_j E_j + w_j T_j$. Kedad-Sidhoum et al. (2008) proposed two lower bounds schemes. The first one is based on an assignment formulation, and the second one is based on a time-indexed formulation. Şen & Bülbül (2015), in turn, proposed a lower-bound scheme based on a preemptive relaxation of the problem.

With regard to exact methods, we highlight the works by Pessoa, Uchoa, de Aragão, & Rodrigues (2010), Kramer, Dell'Amico, Feillet, & Iori (2020), and Bulhões, Sadykov, Subramanian, & Uchoa (2020). Pessoa et al. (2010) dealt with the $P||\sum w_j T_j$ by proposing an arc-time-indexed formulation and a tailored branch-and-price algorithm that can solve instances with up to 100 jobs and 4 machines. Kramer et al. (2020) proposed an arc-flow model and a branch-and-price algorithm to solve the problem with release dates, aiming to minimize the total weighted completion times ($\sum w_j C_j$). The proposed methods are able to solve instances with up to 200 jobs and 10 machines. Bulhões et al. (2020) developed a generic branch-cut-and-price algorithm to solve a class of scheduling problems, including the $R|r_j|\sum w'_j E_j + w_j T_j$ (and the $P|r_j|\sum w_j T_j$ by consequence).

Concerning heuristics methods, Kramer & Subramanian (2019) also addressed a class of earliness and tardiness scheduling problems, including the $P|r_j|\sum w_j T_j$ as a special case. The authors proposed a metaheuristic based on the iterated local search that, depending on the characteristics of the problem, uses tailored data structures to speed up the local search phase, obtaining practical results for various problems. In addition, the recent survey by Durasević & Jakobović (2022) gives an overview of the use of heuristics and metaheuristics methods for unrelated parallel machine problems (including the $P|r_j|\sum w_j T_j$).

Based on the results from the literature, in this paper, we attempt solving the static deterministic problem both exactly, with an arc-flow formulation based on the ideas of Kramer et al. (2020), and heuristically, with a VNS that is relatively simple to design, has been widely employed to parallel machine scheduling problems (as shown in Durasević & Jakobović 2022), and is precisely known by proving good results. As in this case, all information is known in advance, and the expected processing time p_j^q is always equal to the realized time p_j^e, for all $j \in J$. Hence, for simplicity, in this section, we simply use p_j to denote the processing time of patient j.

4.1. Arc-flow model

Arc-Flow (AF) formulations (see de Lima, Alves, Clautiaux, Iori, & de Carvalho 2022 for a recent overview) aim to schedule jobs over a sufficiently long time horizon by making use of the concept of flows in a capacitated network. To this aim, we consider an acyclic-directed multigraph $G = (N, A)$, where the set N of vertices contains the time instants, represented by the normal patterns obtained by the use of the p_j values (see, e.g., Almeida Cunha, de Lima, & Queiroz 2020; Côté & Iori 2018 for recent research on normal patterns). Let $A = A_0 \cup_{j\in J} A_j$ be the set of arcs. Let $A_j, j \in J$ be the set of arcs (p, q, j) representing the fact that patient j is visited by a doctor in the interval $[p, q)$. In addition, let A_0 be the set of dummy arcs $(p, q, 0)$ used to represent idle times between time instants p and q. Algorithm 1 shows the way in which G can be built.

In Algorithm 1, T should be sufficiently large to ensure that there exists at least an optimal solution for the problem but should also be as short as possible in order to reduce the problem size. In order to estimate the time horizon T, we use:

$$T = \left\lfloor \frac{1}{m}\sum_{j\in J} p_j + \frac{(m-1)}{m} p_{\max} \right\rfloor + r_{\max}, \tag{1}$$

Algorithm 1 Construction of $G = (N, A)$.

```
Let T be the end of the time horizon
A_j ← ∅, j ∈ J_+                          ▷ J_+: set of patients plus dummy 0
N ← ∅; P[0, ..., T − 1] ← false; P[T] ← true
for all j ∈ J do P[r_j] ← true
for t ← 0 to T − 1 do
    if P[t] = true then
        for all j ∈ J | r_j ≤ t and t + p_j ≤ T do
            P[t + p_j] ← true; A_j ← A_j ∪ {(t, t + p_j, j)}
for t ← 0 to T | P[t] = true do N ← N ∪ {t}
for all i ← 1 to |N| − 1 do A_0 ← A_0 ∪ {(N_i, N_{i+1}, 0)}
A ← A ∪_{j∈J_+} A_j
return N, A
```

which is an extension of the estimation given by van den Akker, Hoogeveen, & Van de Velde (1999) and already adopted by Kramer, Dell'Amico, & Iori (2019), Kramer et al. (2020). The values $p_{\max} = \max_{j\in J} p_j$ and $r_{\max} = \max_{j\in J} r_j$ in (1) represent the maximum processing time and the latest release time, respectively.

In the AF formulation, we associate a binary variable x_{pqj} to each arc $(p, q, j) \in A \setminus A_0$, stating that patient j is visited (or not) by a doctor from time p to q, and a continuous variable $0 \le x_{pq0} \le m$ with each arc $(p, q, 0) \in A_0$ representing the number of doctors idling between times p to q. The selection of an arc $(p, q, j) \in A \setminus A_0$ incurs a cost $c_{pqj} = w_j \max\{0, p - d_j\}$ that represents the tardiness of patient j if she receives the doctor's visit at time p. The formulation is as follows:

$$\text{(AF)} \quad \min \sum_{(p,q,j)\in A\setminus A_0} c_{pqj} x_{pqj} \tag{2}$$

$$\text{s.t.} \sum_{(p,q,j)\in A_j} x_{pqj} \ge 1 \qquad j \in J \tag{3}$$

$$\sum_{(q,r,j)\in A} x_{qrj} - \sum_{(p,q,j)\in A} x_{pqj} = \begin{cases} m, & \text{if } q = r_{\min} \\ -m, & \text{if } q = T \\ 0, & \text{otherwise} \end{cases} \qquad q \in N \tag{4}$$

$$x_{pqj} \in \{0, 1\} \qquad (p, q, j) \in A \setminus A_0 \tag{5}$$

$$0 \le x_{pq0} \le m \qquad (p, q, 0) \in A_0 \tag{6}$$

The objective function (2) aims to minimize the total weighted tardiness. Constraints (3) ensure that all patients are served. Constraints (4) are the flow conservation constraints, ensuring that no more than one patient is seen at the same time by the same doctor. Constraints (5)-(6) define the domain of the variables. The formulation is characterized by a pseudo-polynomial number of variables $\mathcal{O}(nT)$ and constraints $\mathcal{O}(n + T)$. A variant of this AF was presented by Kramer et al. (2020) to solve the $P|r_j|\sum w_j C_j$.

4.2. Variable neighborhood search heuristic

The VNS, first proposed by Mladenović & Hansen (1997), employs a systematical change of neighborhood structures to obtain a solution that is globally optimal for all neighborhoods. The VNS carries the optimization with a single solution and has been successfully applied to many optimization problems. A recent survey of the extensive use of VNS in healthcare-related problems can be found in Lan, Fan, Yang, Pardalos, & Mladenović (2021).

The main steps of VNS are to: create an initial solution; obtain a neighbor solution of the current one with a given neighborhood

structure; (optionally) perform a local search on the neighbor solution; accept the neighbor solution and restart the search to the first neighborhood structure if the new solution is better than the current, otherwise continue with the search on the current solution by the next neighborhood structure. We refer to Hansen, Mladenović, Todosijević, & Hanafi (2017) for a survey of VNS variants. We consider the general variant, described in Algorithm 2, in which the local search consists of the Variable Neighborhood Descent heuristic (VND).

Algorithm 2 General VNS for the problem.

1: Let x be an initial solution
2: **while** number of consecutive iterations without improvement $\leq NO_{imp}$ **do**
3: $k \leftarrow 1$
4: **while** $k \leq K$ **do**
5: $x' \leftarrow$ neighbor solution from $N_k(x)$
6: $x'' \leftarrow \text{VND}(x')$
7: **if** $wT(x'') < wT(x)$ **then** $x \leftarrow x''$; $k \leftarrow 1$
8: **else** $k \leftarrow k+1$
9: **return** x

In our VNS implementation, a solution x is coded as a vector of lists of integers. Each position of the vector represents a doctor; the vector has size m and contains an ordered list of patients identified by their index. The initial solution is created as follows: sort the patients by non-decreasing order of r_j, breaking ties by the highest w_j first; following this order, assign each patient (as the last one) to the doctor whose completion time is the minimum. The start time s_j of each patient j is defined as the maximum between r_j and the completion time of the last patient assigned to the same doctor. The cost of x is given by $wT(x)$ and corresponds to $\sum_{j \in J} w_j T_j$, where $T_j = \max\{0, s_j - d_j\}$. The search is iterated until NO_{imp} consecutive iterations without improvements are performed.

The performance of the VNS depends on the number, size, and order of its neighborhood structures $N_k()$. On the basis of preliminary experiments, we built $K = 5$ neighborhoods associated with swap and insertion operations:

- N_1: select two patients, j_1 and j_2, served by the same doctor and swap them;
- N_2: as N_1, but now insert j_1 before j_2;
- N_3: select two patients, j_1 and j_2, served by different doctors and swap them;
- N_4: as N_3, but now, remove j_1 from her doctor and insert her before j_2;
- N_5: select a patient j_1 served by a doctor, and a sequence $\{j_2, j_3, \ldots\}$ of patients served by a different doctor, then remove the sequence from its doctor and insert it before j_1. The size of the sequence is limited to MAX_s% of the number of patients served by the corresponding doctor.

We only perform a move if the new start time of a visited patient is no later than her target due time plus a maximum violation constant MAX_d. The only exception to this rule is in N_5, where, for simplicity, we check MAX_d only for j_1 and not for the other patients in the sequence. We study the impact of these neighborhoods and show how they contribute to the VNS in achieving high-quality solutions in Section 6.2.

Line 5 of Algorithm 2 represents the shaking phase, which, according to Hansen et al. (2017), aims at generating a random neighbor solution from the neighborhood $N_k()$ of the current solution. In this way, we consider that doctors and patients in the neighborhood structures are selected at random. On the other hand, line 6 represents the local search phase, for which we use the VND. This heuristic changes the neighborhoods in a sequential deterministic way, in such a way that the final solution is optimal with respect to all neighborhoods $N_k()$. In the local search phase, we use the same neighborhood structures from the shaking phase; however, instead of random selections, all possibilities for doctors and patients are explored, and the one with the best improvement is chosen. For more details about VNS and VND schemes, we refer to Hansen et al. (2017) and Hansen & Mladenović (2018).

5. Solution of the dynamic stochastic problem

As we are facing an online stochastic optimization problem, we propose different methods to solve the it. According to Ulmer, Goodson, Mattfeld, & Thomas (2020), in the reoptimization methodology, the scheduling is determined by observing the information currently available. Each time an event occurs, a (smaller) deterministic scheduling problem is solved giving no importance to future uncertainties. On the other hand, as we can infer the probability distributions of the problem's inputs from historical data of the EDs, we would like to anticipate future decisions and thus minimize the effect of uncertainties. In this sense, van Hentenryck & Bent (2006) pointed out the proposal of online anticipatory algorithms, also called *lookahead algorithms* by Ulmer et al. (2020), which do not ignore the future and use sampled scenarios from the known/inferred distributions. Considering the occurrence of an event, these scenario-based algorithms also solve a deterministic scheduling problem but with possible realizations of the future.

Some effective contributions in the literature regarding the proposal of scenario-based methods are, e.g., Bent & van Hentenryck (2004), who tackled the multiple vehicle routing problem with time windows; Hvattum, Løkketangen, & Laporte (2007), who solved a stochastic and dynamic vehicle routing problem; Tirado, Hvattum, Fagerholt, & Cordeau (2013) who solved an integrated ship routing and scheduling problem; Schilde, Doerner, & Hartl (2014), who tackled a dynamic dial-a-ride problem; Voccia, Campbell, & Thomas (2019), who tackled the same day delivery problem; and Song, Ulmer, Thomas, & Wallace (2020), who solved the team-orienteering problem with time windows and mandatory visits.

We propose from simple reoptimization heuristics to an enhanced scenario-based method that makes decisions using samples of the future. The rationale behind these heuristics is to update/determine new decisions each time an event occurs and new information is revealed. Our methods can thus be classified as event-driven algorithms. We consider only events of type (i) a patient arrives at the ED and there is a doctor available, and (ii) a doctor can start a new visit. When an event occurs, all information about the schedule is collected (i.e., patients waiting to be visited or being visited at the moment, doctors in service or idle/waiting for the next patient, etc.). Past decisions (e.g., patients already visited) and decisions in progress (e.g., patients being visited) are immutable (i.e., fixed). In other words, only future decisions can be updated.

5.1. Reoptimization heuristics

Reoptimization heuristics (sometimes also called greedy rules or myopic heuristics in the literature about dynamic problems) are algorithms that take decisions, whenever an event occurs, only based on the information currently available. Hence, they do not explore the stochastic aspect of the problem (see, e.g., Archetti, Feillet, Mor, & Speranza 2020; Ulmer et al. 2020). We develop a first reoptimization heuristic that schedules patients to doctors according to the patients' triage category (i.e., urgency level). Recall we are given a list of ℓ possible urgency levels. For each level $q \in \{1, \ldots, \ell\}$, we create a queue Q_q and insert there all patients having $u_j = q$. The first queue, Q_1, holds the most urgent patients,

and so on, until the last queue, Q_ℓ, which holds the less urgent patients. Patients in each queue are sorted by non-decreasing release time (i.e., the first-in, first-out policy). The reoptimization heuristic based on priority queues is presented in Algorithm 3. The algorithm iterates until all events have been considered. The idea is to assign the highest-priority patients as soon as possible to the free doctors.

Algorithm 3 REO-QUEUE - reoptimization heuristic based on priority queues.

1: **while** there is an event **do**
2: $t \leftarrow$ time at which the event happens
3: $Q \leftarrow$ add the newly revealed patients j that are ready ($r_j \leq t$)
4: **for all** doctors i that are free at time t **do**
5: **for** $q \leftarrow 1$ to ℓ **do**
6: **if** $Q_q \neq \emptyset$ **then** assign the first patient $j \in Q_q$ to doctor i, setting $s_j \leftarrow t$, and go to line 5

In line 3 of Algorithm 3, the queues are updated with the new patients ready to receive medical services. The queues are updated to have patients organized in non-decreasing order of their release times. Lines 4 to 6 are related to the assignment of high-priority patients, following the queues, to the doctors who are available at time t. These patients have their service start time set to t and their completion time set to $t + p^e_j$. Notice that at time $t = 0$, we have all doctors available and (possibly) some patients ready to be seen by the doctors.

We also compare some other variants of Algorithm 3 in order to infer useful policies to the EDs. One variant is related to the scheduling according to the patients' release time by considering a single queue and the first-in, first-out policy. The other variant considers the different queues in the number of existing urgency levels but patients whose due times are violated are moved to the next higher-priority queue as a way to avoid excessive waiting times for low-priority patients. Besides considering that patients can move from their queue to the next ones in the case of violating their target due times, the last variant also assumes that free doctors can take a break between two visits if there are only low-urgency patients in the ED. It is important to emphasize that, as the patient's effective (realized) processing time is not known in advance, patients with the same priority weight have the same expected processing time (and the same maximum waiting time). Thus, variants that schedule patients by the expected processing time (and due time) will result in one of the above variants. More details about the variants and the obtained results are given in Section 6.3.

In order to take advantage of the VNS, we developed a second reoptimization heuristic in which the VNS is used to assign patients to doctors. We call this heuristic REO-VNS and provide its pseudocode in Algorithm 4. Let x be the current solution, where past decisions and decisions in progress at time t are fixed (line 3). We optimize x with the VNS by (re)scheduling the new set of patients who are ready and patients whose start times are set to be after t ($s_j > t$) (lines 4 and 5). In the optimized schedule, patients whose new start times are equal to t start receiving service (decisions in progress) (line 6).

5.2. Scenario-based planning approach

Different from reoptimization heuristics, the SBPA considers sampled scenarios with *fictive* patients that might appear in the future. Scenarios are generated by incorporating fictive patients through samplings from the given probability distributions. When an event occurs, considering the current solution, we optimize each scenario by using the VNS; then, with a consensus function, we evaluate the solutions of each scenario to decide which decisions to adopt for the instance being solved. All sampled scenarios are generated in a preprocessing step, following the same procedure used to create the instances. Scenarios do not contain the real patients of the instance, but just fictive ones, so the SBPA does not know in advance any information about the future. More details on the instances and scenario generation are given in Section 6.1. We refer the reader interested in this form of dynamic heuristics to Bent & van Hentenryck (2004) for a seminal paper, van Hentenryck & Bent (2006) for a classic book, and Ulmer et al. (2020) and Bakker et al. (2020) for recent surveys. The SBPA we propose for the DSPD is presented in Algorithm 5.

Algorithm 4 REO-VNS - reoptimization heuristic with VNS.

1: **while** there is an event **do**
2: $t \leftarrow$ time at which the event happens
3: $x \leftarrow$ current solution where past and in progress decisions at t are fixed ($s_j < t$)
4: Add the newly revealed patients j that are ready ($r_j \leq t$) to x
5: Optimize x with the VNS
6: Start servicing the patients j whose start time in x is equal to t ($s_j = t$)

Algorithm 5 SBPA-VNS - scenario-based planning approach with VNS.

1: Let Ω be the set of scenarios with fictive patients
2: **while** there is an event **do**
3: $t \leftarrow$ time at which the event happens
4: $x \leftarrow$ add the newly revealed patients that are ready
5: **for all** scenarios $\omega \in \Omega$ **do**
6: $x_\omega \leftarrow x \cup$ {fictive patients j of scenario ω having $r_j \in [t, t + t_{SH}]$}
7: Optimize x_ω with the VNS
8: Update x_ω by substituting each fictive patient by an idle time t_{wait}
9: $x_{best(\omega)} \leftarrow$ the solution x_ω with the highest consensus function score
10: Update x with the decisions in $x_{best(\omega)}$

In Algorithm 5, x contains the current schedule of patients, where past decisions and decisions in progress at time t are fixed. Besides, it considers the new patients that are ready at t. These are the real patients of the input instance being solved. In line 5, considering all information available at time t, we use the scenarios in Ω to guide the next decisions and update solution x. For each scenario ω, we use the VNS to solve a deterministic scheduling problem on x_ω considering all the real patients currently available and the fictive patients $i \in \omega$ whose release time r_i is between t and $t + t_{SH}$, where t_{SH} is the length of the sampling horizon (i.e., the time interval in which we extract the fictive patients). The number of scenarios and the value of t_{SH} may be critical parameters for SBPAs and are computationally investigated in Section 6.2.

In lines 5 to 8 of Algorithm 5, we consider x_ω as the current solution x (with real patients) plus the fictive patients of scenario ω having $r_i \in [t,\ t + t_{SH}]$. We first optimize x_ω with the VNS, and next substitute all fictive patients from it. A fictive patient might represent a future patient in the ED. Therefore, we substitute each fictive patient with an idle time t_{wait}. This means that a doctor that would service a fictive patient is, in fact, not in service during t_{wait} minutes. Then, after this waiting time, the doctor will be free, triggering an event of type (ii). A proper value for parameter

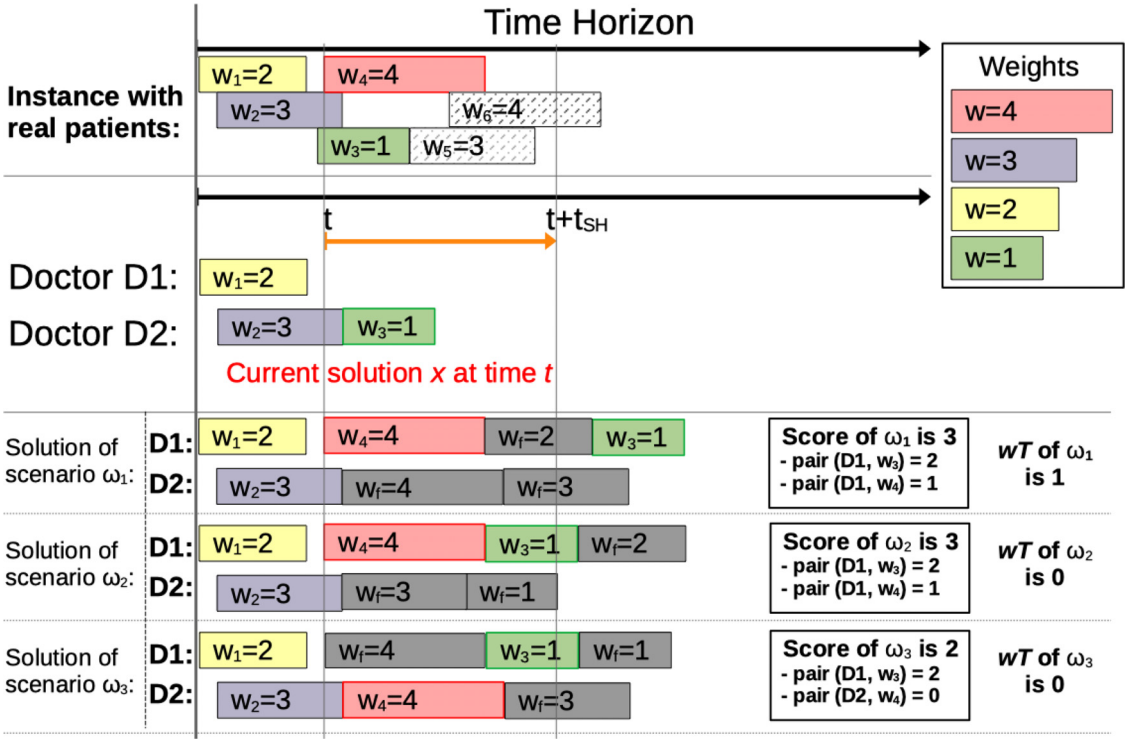


Fig. 1. Illustrative example of how the score of the solution of scenarios is calculated.

t_{wait} is investigated in Section 6.2. As t_{wait} may be different from the service time required by each fictive patient, it is necessary to correct/update the start time of the non-fictive patients in x_ω as well as the solution value (line 8). For these, we simply shift left (or right) the start time of the patients and then recompute the tardiness.

The last two lines of Algorithm 5 are related to the consensus function, which calculates the score of each solution x_ω. We take the most common decisions in all scenarios regarding the assignment of patients to doctors. In other words, for each x_ω, this function calculates the number of times the pair (doctor o, real patient j), for each doctor and the real patient scheduled to this doctor, appears in all other solutions $x_{\bar{\omega}}$, for each $\bar{\omega} \neq \omega$. Every time a pair (o, j) is found in the solution of another scenario, the score of x_ω is increased by one. The current solution x is updated to perform the decisions obtained in the solution x_ω having the highest score. In the case of ties, the largest score is given by the solution with the smallest weighted tardiness.

Figure 1 illustrates how our consensus function calculates the score for an instance with 2 doctors and 6 patients, assuming that there are 3 scenarios with fictive patients. The current solution x, at time t, has patients 1 (weight $w_1 = 2$) and 2 (weight $w_2 = 3$) fixed since their start times are smaller than t. On the other hand, patient 4 (weight $w_4 = 4$, the most urgent patient) is just revealed on the time horizon, while patient 3 (weight $w_3 = 1$, the less urgent patient) was revealed before t but has her start time after t. This means that patients 3 and 4 can have their start times updated/defined by the VNS, while patients 5 (with weight $w_5 = 3$) and 6 (with weight $w_6 = 4$) are not known at time t. Now, each scenario ω is optimized with the VNS. Besides having the real patients 1, 2, 3, and 4, given the interval $[t,\ t + t_{SH}]$, scenarios ω_1, ω_2, and ω_3 have, each one, 3 fictive patients (ω_1 with weights $w_f = 2, 3, 4$; ω_2 with weights $w_f = 1, 2, 3$; and ω_3 with weights $w_f = 1, 3, 4$). Observing the solution of each scenario, in ω_1 we have the pairs (doctor, real patient) $= \{(D1, w_3), (D1, w_4)\}$. Notice the pair $(D1, w_3)$ is also in the solutions of ω_2 and ω_3 (score of ω_1 is increased by 2), and the pair $(D1, w_4)$ is also in the solution of ω_2 (score of ω_1 is increased by 1). After all, the score of the solution of ω_1 is 3. Similarly, we calculate the score of ω_2, which is 3, and ω_3, which is 2. As the solutions of ω_1 and ω_2 have the same score, we select the solution of ω_2 since it has the smallest total weighted tardiness (wT). This means that the current solution x will be updated to have doctor D1 visiting patients 4 (at time t) and 3 (after finishing patient 4), while doctor D2 will be idle after finishing serving patient 2.

5.3. Use of early information on patients

In this section, we investigate the option of anticipating some information on the arrival time and triage category of the patients, before they actually reach the ED. This can be achieved, for example, during transportation by calling the ED from the ambulance. This may be an important piece of information for the ED because a patient can be inserted at an earlier stage in the scheduling process, and this can hopefully lead to a reduction of the total weighted tardiness, especially when there is high demand for higher priority patients.

We call this problem variant *DSPD with early information*. The use of early information might be possible only for a small fraction of the patients arriving in the ED. If a patient j is treated with early information, then the ED receives a call t_{call} minutes before r_j. This implies that the patient is revealed on the time horizon at time $r_j - t_{call}$, but she still will receive medical services at or after time r_j. To handle this problem variant, we consider the following modifications in the proposed heuristics:

- REO-QUEUE: (Algorithm 3) we also insert in queues the patients with early information ($r_j - t_{call} \leq t$) that have the two highest urgency levels. The others with early information but lower urgency levels are only considered at time r_j. The idea is

to schedule first higher priority patients, even those ready only in the near future;
- REO-VNS: (Algorithm 4) we use the VNS to optimize the schedule, which also includes the patients j with early information (again having $r_j - t_{call} \leq t$ at time t), independently of their weights;
- SBPA-VNS: (Algorithm 5) as done in REO-VNS, we also include the patients with early information, independently of their weights. The scenarios also consider fictive patients with early information.

6. Computational experiments

The proposed methods have been coded in the C++ programming language. We use Gurobi Optimizer (version 10.0) to solve the mixed integer programming model, with all parameters set to their default values, except the time limit set to 3600 seconds. The experiments were carried out on a single core of a computer with Intel Xeon E3-1245v5 3.50 GHz, with 32 GB of RAM, running under Linux Ubuntu 16.04.7 LTS.

We first introduce the realistic instances that we generated by considering datasets from two large hospitals in Italy and Hong Kong SAR of China. We present how the scenarios with the fictive patients are generated. Next, we analyze the influence of some parameters on the performance of the methods. Then, we provide a full computational analysis of the methods, and, finally, we evaluate the impact of the early information on the EDs.

6.1. Instance generation

The instances used in the computational experiments were generated based on two major EDs, one located in Italy and the other in Hong Kong. Real-world data about these EDs and their events and activities (e.g., patients' arrival time) have been collected during previous research (Dosi et al., 2019; 2020; Kuo et al., 2016) and new on-site visits. All these data have been used to simulate the problem parameters in the instances and infer the probability distributions. We organize the instances of both hospitals in 6 groups, each representing a shift of 4 hours in a day. Group 1 contains the working hours in the interval $[00AM;\ 04AM)$, while Group 2 represents the interval $[04AM;\ 08AM)$, and so on until Group 6, which is associated with the interval $[08PM;\ 12AM)$.

The hospital in Italy covers a region with approximately one million inhabitants. This hospital admits around 80,000 patients per year and has almost 1500 beds, around 5500 employees, and 30 pavilions. A detailed discussion on the probability distributions and the simulation to infer some instances values for this hospital was previously performed by Dosi et al. (2019, 2020). The data have been retrieved from the hospital's database and an on-site time study to collect service times. The arrival distribution of patients and their triage categories have been obtained from a full analysis of this database. After the on-site time study and inferring the probability distribution, the service time of patients has been estimated by simulation. We simulate 31 days of service, for a total of 186 instances. The arrival times of patients (r_j) are characterized by an exponentially distributed inter-arrival time with a mean equal to $1/irr_t$, where irr_t is the inter-arrival time rate for the one-hour time period starting at hour $t \in \{0, 1, \ldots, 23\}$. This means that the number of patients varies in accordance with the group.

The hospital in Italy considers four urgency levels, $u \in \{1, 2, 3, 4\}$, having $\delta \in \{0, 15, 60, 120\}$ as the maximum waiting time, following internal/governmental protocols. We set the weights associated with the urgency levels to $w = \{4, 3, 2, 1\}$, respectively. Each patient is assigned to one of the four urgency levels by considering the inferred probability distribution of the triage categories and the patient arrival time. According to the inferred probability distributions, the realized service times (p_j^e) are obtained from a truncated normal distribution with the minimum, maximum, mean, and standard deviation equal to 1, 60, $9 + 4f_j$ and $3 + 2f_j$, respectively, for $f_j \in \{0, 1, 2, 3, 4\}$ representing the family of patients with similar symptoms. More urgent patients are likely to receive higher service times.

We also collected data from the ED of the Prince of Wales Hospital (PWH) in Hong Kong. PWH is one of the largest hospitals in the city. It is equipped with more than 1700 beds and has around 5500 staff members. PWH provides 24-hour accident and emergency services. In previous work, Kuo et al. (2016) performed a detailed study to infer the probability distributions for PWH and then to obtain some instance values. The patient arrival data and patient triage categories have been retrieved from the hospital's computer information system. On-site visits have occurred to conduct a time study to obtain information on patients. The interarrival time of each patient follows an exponential distribution. On the other hand, the duration of some activities, e.g., realized service times, could not be recorded directly. In this way, to include a variety of distributions, the authors assumed a Weibull distribution, integrating simulation and meta-heuristics to tune the distribution parameters in accordance with what was observed in practice. Therefore, we have been allowed to use data for five days, resulting in 30 instances. Such instances consider four urgency levels $\delta \in \{0, 10, 30, 180\}$ and the weights $w \in \{4, 3, 2, 1\}$, respectively. The number of patients varies in accordance with the time of the day, having more patients during the daytime. The average service time (p_j^a) of each patient j is calculated as the average of the realized service time (p_i^e) for all patients i with the same urgency level and group.

All the proposed methods are applied to solve the realistic instances above-mentioned of the EDs in Italy and Hong Kong. Besides that, SBPA-VNS, when solving an instance, also considers the inclusion of fictive patients from sampled scenarios. We associate with each of the six groups of instances a set of scenarios created in a preprocessing step. Each scenario contains fictive patients that might appear later on. As already mentioned, fictive patients are not the real patients of the instances and the methods do not have access to information about future real patients. The fictive patients are sampled from the inferred probability distributions from the EDs historical data and thus the generation of scenarios uses the same procedure adopted to generate the instances. This is in accordance with the previous literature on online stochastic optimization problems (see, e.g., Bakker et al. 2020; van Hentenryck & Bent 2006; Voccia et al. 2019).

The number of scenarios may impact the solution and the computing time of the SBPA-VNS. Some authors have pointed to the use of as many scenarios as possible, e.g., Hvattum, Løkketangen, & Laporte (2006), who did experiments with 30 up to 600 scenarios, improving solution quality with the use of 600 scenarios. On the other hand, Hvattum et al. (2007) and Voccia et al. (2019) achieved different conclusions on the number of scenarios to consider. Hvattum et al. (2007) performed experiments with 1 up to 30 scenarios and concluded that considering 30 scenarios lead to better results. Voccia et al. (2019), in turn, considered using from 10 up to 60 scenarios and concluded that using 10 scenarios allowed the algorithm to achieve better results. Clearly, these conclusions may depend on the proposed methods, problem complexity, and instances being solved. van Hentenryck & Bent (2006) observed that, in general, sophisticated methods do not need many scenarios as in the case of weaker methods that depends on many more scenarios. Therefore, we created 60 scenarios for each group of the ED in Italy, resulting in 360 scenarios. In this way, when solving instances, e.g., of Group 1, the SBPA-VNS considers only the 60 scenarios created for this group.

Table 1
Distribution of the patients by urgency level (percentages) in the two EDs according to the realistic instances.

Group	ED in Italy					ED in Hong Kong						
	#patients*	$u=1$	$u=2$	$u=3$	$u=4$	#patients*	$u=1$	$u=2$	$u=3$	$u=4$	$\%EI$	$\bar{t}_{call}$
1	[5,17]	2.96	34.02	51.78	11.24	[35,69]	0.36	2.88	33.45	63.31	17.99	7.56
2	[5,17]	3.13	23.76	60.05	13.05	[47,74]	0.96	2.25	27.33	69.45	16.08	6.78
3	[41,62]	2.16	24.84	57.37	15.63	[140,169]	0.51	2.19	22.37	74.94	21.21	7.41
4	[34,57]	3.04	29.60	54.90	12.46	[112,139]	0.31	3.40	32.92	63.37	19.47	7.29
5	[25,49]	2.76	28.62	59.73	8.89	[102,138]	0.70	2.97	34.44	61.89	18.53	7.42
6	[7,37]	2.58	34.53	51.86	11.03	[68,98]	0.48	4.78	36.36	58.37	18.42	7.47
Average	-	2.77	29.23	55.95	12.05	-	0.55	3.08	31.15	65.22	18.62	7.32
	$\%EI$ per group and urgency level					$\%EI$ per group and urgency level						
1	-	3.91	39.06	41.80	15.23	-	0.00	0.00	24.00	76.00	-	-
2	-	2.09	24.39	55.75	17.77	-	2.00	0.00	32.00	66.00	-	-
3	-	1.78	22.84	55.41	19.97	-	0.00	3.03	23.64	73.33	-	-
4	-	2.97	29.83	53.32	13.88	-	0.79	3.17	26.19	69.84	-	-
5	-	1.78	26.52	62.31	9.39	-	0.94	3.77	35.85	59.43	-	-
6	-	1.54	30.31	58.11	10.04	-	0.00	6.49	45.45	48.05	-	-
Average	-	2.35	28.83	54.45	14.38	-	0.62	2.75	31.20	65.44	-	-

* $[a, b]$ Denotes the range of the number of patients.

Concerning the presence of early information, this is a common practice in the ED of Hong Kong. On the other hand, the ED in Italy has not mentioned the possibility of patients having this. We decide to study its impact on such instances by attempting different options. The objective is to assess whether the possibility of having early information could bring some benefits to patients and hospitals in general, knowing that now these EDs cannot implement it for all patients. Clearly, this study could corroborate future decisions and investments they intend to make, as well as extend such good practices to other EDs and hospitals. The instances of the ED in Hong Kong have the presence of early information for 18.62% of the patients. The value of t_{call} varies from 1 to 10 minutes, with an average of 7.32 minutes. On the other hand, to the ED in Italy, we assume a standard normal distribution to select either 10%, 15%, 20%, or 30% of the patients of each group to have information revealed t_{call} minutes before their release times. To define t_{call}, we use a truncated normal distribution with the minimum, maximum, mean, and standard deviation equal to 0, 60, 10, and 3 minutes, respectively.

We summarize in Table 1 information about the percentage of patients and the percentage of patients having early information per each priority and group, considering the 186 and 30 instances of the EDs in Italy and Hong Kong, respectively. We present the ranges of the numbers of patients in all groups of the two EDs. For the ED in Hong Kong, we also show for each group the percentage of patients having early information (column $\%EI$) and the average time $\bar{t}_{call}$ the ED receives early information on such patients. Observing the table, the ED in Hong Kong has a predominance of urgency levels with $u=4$ for 65.22% of the patients, and $u=3$ for 31.15% of the patients. Besides that, 65.44% and 31.20% of these patients have early information, respectively. Patients with the two largest urgency levels ($u=1$ and $u=2$) are in fewer numbers. Differently, the ED in Italy is characterized by patients with urgency levels $u=3$ for 55.95%, and $u=2$ for 29.23% of the patients. Patients with urgency levels $u=4$ (smallest one) and $u=1$ (largest one) are in fewer numbers. This is also reflected in the percentage of patients having early information, with the predominance of urgency levels $u=3$ for 54.45%, and $u=2$ for 28.83% of the patients.

All results and discussions presented next consider the instances of the ED in Italy because of their large number and generality. The instances of the ED in Hong Kong are used in Section 6.5 to assess the impact of early information. All instances are available from the authors upon request.

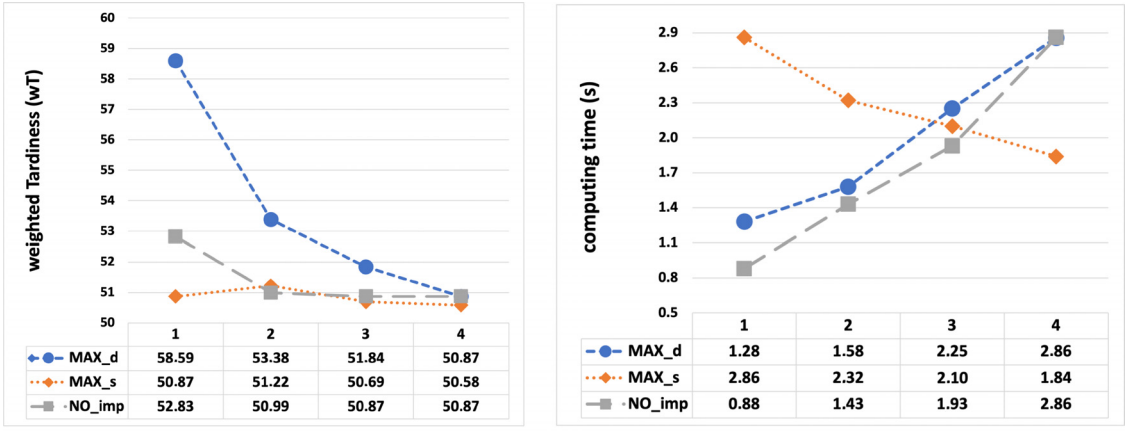
6.2. Sensitivity analysis

In this section, we investigate how the main parameters of the methods affect the solution and computing time. All these experiments consider the instances of the ED in Italy and assume there are 3 doctors available in the ED. We present the average values of the total weighted tardiness (wT) and computing time ($time$, in seconds) for all 186 instances of this ED. We first study the impact of the VNS neighborhood structures on the solution quality and execution time. We consider the following VNS configurations, with $K=1$ (N_1), $K=2$ (N_1+N_2), $K=3$ ($N_1+N_2+N_3$), $K=4$ ($N_1+N_2+N_3+N_4$), and $K=5$ ($N_1+N_2+N_3+N_4+N_5$). This means that with $K=2$ we consider the VNS to have only the first (N_1) and second (N_2) neighborhood structures, and so on for the other values of K. Notice that the last configuration, with $K=5$, corresponds to the VNS proposed in Section 4.2. For these experiments, we use the following basic configuration for the parameters: $MAX_d=120$, $MAX_s=1$, and $NO_{imp}=50$. Table 2 summarizes the obtained results. In the last four lines of Table 2, we consider the values of wT and show the percentage reduction for the VNS with $K=1$, $K=2$, $K=3$, and $K=4$. For example, in the line "Red. $N_1+N_2+N_3$ (%)", for $K=3$, we have the VNS with the neighborhoods N_1, N_2, and N_3 and report the percentage decrease from the results obtained by the configuration with $K=4$ instead of this with $K=3$, and with $K=5$ instead of this with $K=3$. Notice that a negative value would mean that having $K=3$ is better for the VNS.

We observe in Table 2 that only swap movements (as in the case of $K=1$ and $K=2$) are not sufficient to explore the space of solutions adequately. The percentage of decrease of wT from $K=1$ to $K=2$ is 8.79%. The consideration of insert movements (as in the case of $K=3$ and $K=4$) brings a substantial improvement in the solutions. The percentage of decrease of wT from $K=2$ to $K=3$ (respectively, $K=4$) is 42.44% (respectively, 43.24%). Finally, we have neighborhood N_5 that, besides considering insertion movements, works with sequences of patients as a way to escape from local optima solutions. This is achieved, although the percentage of decrease of wT from $K=3$ (respectively, $K=4$) to $K=5$ is 1.96% (respectively, 0.58%). The best solutions are then found with the VNS that considers all neighborhood structures (i.e., with $K=5$). All these neighborhoods take advantage of the way the solution has been coded and, consequently, it is possible to better explore the space of solutions since a very large number of possibilities are tried. Comparing such a configuration with the one with $K=1$, we

Table 2
Impact of the VNS neighborhood structures in the deterministic problem (Italian ED instances, 3 doctors).

	N_1	N_1+N_2	$N_1+N_2+N_3$	$N_1+N_2+N_3+N_4$	$N_1+N_2+N_3+N_4+N_5$
computing time (s)	0.05	0.11	0.45	0.80	2.88
wT	99.90	91.12	52.45	51.72	51.42
Red. N_1 (%)	-	8.79	47.50	48.23	48.53
Red. N_1+N_2 (%)	-	-	42.44	43.24	43.57
Red. $N_1+N_2+N_3$ (%)	-	-	-	1.39	1.96
Red. $N_1+N_2+N_3+N_4$ (%)	-	-	-	-	0.58



(a) Results in terms of wT (b) Results in terms of runtime

Fig. 2. Individual impact of each VNS parameter (Italian ED instances, 3 doctors).

can notice that the percentage of decrease of wT is 48.53% and the computing time still remains acceptable. In this way, we decided to use the VNS with $K = 5$ in the next experiments.

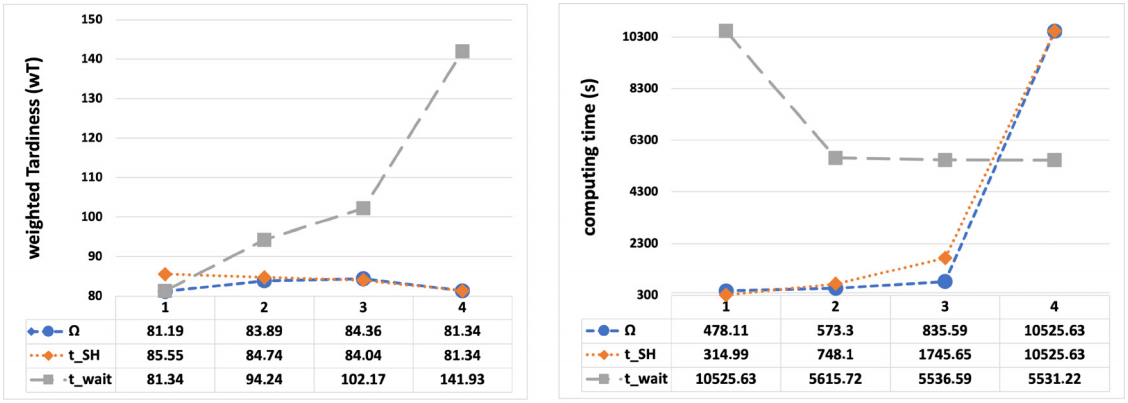
Once we have defined the VNS configuration to use, the next study is related to the impact of its parameters: maximum violation constant MAX_d in the neighborhoods, the maximum size MAX_s of the sequence in neighborhood N_5, and the number of consecutive iterations without improvement, NO_{imp}, used as stopping criterion (line 2 of Algorithm 2). For these experiments, we evaluate each parameter individually, considering the basic configuration with $MAX_d = 120$, $MAX_s = 1$, and $NO_{imp} = 50$. The parameters we test are $MAX_d = \{0, 15, 60, 120\}$, $MAX_s = \{1, 0.5, 0.33, 0.25\}$, and $NO_{imp} = \{10, 20, 30, 50\}$. It means that when testing MAX_d, we set $MAX_s = 1$, and $NO_{imp} = 50$, and similarly for the other parameters.

We illustrate in Fig. 2 the individual impact of each parameter in terms of average total weighted tardiness (wT) and computing time. On the x-axis we report the four values each parameter can assume, where the value 1 means the first value in the set of each parameter, and so on. For example, in Fig. 2(a), for MAX_d, the value 1 on the x-axis indicates the wT (of 58.59) returned by the VNS when solving all instances with $MAX_d = 0$, for $MAX_s = 1$ and $NO_{imp} = 50$. Larger values of MAX_d (e.g., having neighborhood solutions where the start time of patients may violate their due time by $MAX_d = 120$) lead to solutions with smaller wT because the search space can be better explored. On the other hand, allowing sequences whose size MAX_s is too large (e.g., considering all patients of a doctor, by setting $MAX_s = 100\%$) is time-consuming, especially due to the local search. Better solutions are achieved when $MAX_s = 50\%$ and $MAX_s = 25\%$, which also leads to a decrease in the computing time with respect to $MAX_s = 100\%$. Concerning NO_{imp}, better values of wT are obtained with larger values of such a parameter. In our case, it is sufficient to set $NO_{imp} = 30$ since it returns the same solution as with $NO_{imp} = 50$ but having a much lower computing time.

Based on these results, we set $MAX_d = 120$, $MAX_s = 25\%$, and $NO_{imp} = 30$ in the VNS. With this combination of parameters, the VNS returns the wT equal to 50.93, for the average runtime equal to 1.13 seconds. On the other hand, the AF model returns the wT equals to 49.60 weighted minutes and an average computing time of 21.18 seconds, not optimally solving one instance within the imposed time limit. The percentage of increase from the AF model solution compared with the VNS one is 2.68%, in terms of wT, and -94.66%, in terms of computing time. Therefore, the VNS is a proper choice considering that the problem is dynamic stochastic and, thus, needs accurate and fast decisions.

The next analysis concerns the influence of some parameters on the performance of SBPA-VNS. We evaluate the impact of the number $|\Omega|$ of scenarios, the size t_{SH} of the sampling horizon, and the time t_{wait} a doctor should wait to anticipate possible urgent future patients. We evaluate each parameter individually, assuming the basic configuration with $|\Omega| = 60$, $t_{SH} = 120$, and $t_{wait} = 1$. The parameters we test are $|\Omega| = \{10, 20, 30, 60\}$, $t_{SH} = \{30, 45, 60, 120\}$, and $t_{wait} = \{1, 5, 10, 20\}$. It means that when evaluating $|\Omega|$, we set $t_{SH} = 120$, and $t_{wait} = 1$, and so on. We present the impact of each parameter in terms of average wT and computing time in Fig. 3. On the x-axis we have the four values each parameter can assume.

Observing Fig. 3, SBPA-VNS achieves solutions with the smallest wT by using the smallest number of scenarios ($|\Omega| = 10$). The more scenarios it uses, the higher the computing effort is, without any real improvement in terms of wT. For example, having $|\Omega| = 60$ highly impacts on the computing time, changing to 10525.63 seconds instead of 478.11 seconds for $|\Omega| = 10$, which in turn is a decrease of 95.46% over $|\Omega| = 60$. This is a big difference in the computing time that adds no improvement in terms of wT. Considering the dynamic nature of the problem, it is much more appropriate to consider $|\Omega| = 10$. On the other hand, better values of wT are obtained when allowing a large sampling horizon t_{SH} (i.e., maintaining many fictive patients in the sampled scenarios) to anticipate future decisions, but this could be very time-consuming if using a large number of scenarios. Concerning the waiting time (t_{wait}) for doctors, the smaller the value, the better wT is. This could come with a substantial increase in the computing time if



(a) Results in terms of wT (b) Results in terms of runtime

Fig. 3. Individual impact of each SBPA-VNS parameter (Italian ED instances, 3 doctors).

using all 60 scenarios. With these results, we set SBPA-VNS to have $|\Omega| = 10$, $t_{SH} = 120$, and $t_{wait} = 1$. This combination of parameters gives SBPA-VNS the value of wT equal to 81.19, with an average runtime of 478.11 seconds.

6.3. The impact of different policies

Aiming to identify policies that could guide decision-making in the reoptimization heuristic, we present results for different REO-QUEUE variants. In the basic REO-QUEUE heuristic (see Algorithm 3), when a patient's due time is violated her priority level is not updated, i.e., a patient does not migrate from her queue to another one. She remains on her initial queue until receiving the doctor's visit. As discussed in Section 5.1, the following variants of the REO-QUEUE are tested:

- REO-NO-QUEUE: patients are assigned to a single queue and scheduled according to their release time by the first-in, first-out policy. Ties are broken by the highest urgency level first;
- REO-QUEUE-UPDATE: as low-priority patients could wait for a long period and their health conditions could be worsening with time, these patients are moved from their current queues to the respective next ones (of higher priority) if their target due times have been violated. Then, a new due time is set for the patient, depending on the queue this patient is moved to. For example, a patient j with $u_j = 3$ has due time set to $d_j = r_j + \delta_3$ minutes and is initially inserted in queue Q_3. If her visit does not start by d_j, then she is moved to queue Q_2 and her due time is updated to $d_j + \delta_2$. The total weighted tardiness of a solution continues being calculated by using the original input data, as the update in the due times only affects the way the patients are assigned, not the computation of their tardiness;
- REO-QUEUE-WAIT: similar to the REO-QUEUE-UPDATE variant, in addition to considering that a free doctor waits for t_{wait} minutes if there are only low-priority patients (i.e., with $u = 4$ in the lowest urgency level queue). A doctor is not allowed to wait for t_{wait} minutes one right after the other, i.e., after finishing her waiting time, the doctor must necessarily visit the next patient (if one exists) according to the priority queues. We evaluate the following values: $t_{wait} = \{1, 5, 10, 20\}$.

Table 3 contains the results obtained by each REO-QUEUE variant. We solve the instances of the ED in Italy by assuming that there are 3 doctors. In the last six lines of this table, we consider the values of wT and show the percentage reduction on each variant. In terms of computing time, all the variants are very fast, reporting execution times below 0.01 seconds on average. In terms of total weighted tardiness, we notice that scheduling the patients according to the first-in, first-out policy is not satisfactory for the ED. In fact, the REO-NO-QUEUE variant has the worst results overall, with a value of wT equal to 281.74. Results are better when other patients' characteristics are taken into consideration, as in the case of the urgency level. As already mentioned, in our problem, patients with the same urgency level have the same expected processing and waiting times. Thus, to consider such characteristics refers to the use of release time or urgency level.

According to the results shown in Table 3, we can observe that the value of wT of the REO-QUEUE-WAIT variant is at least 35.60% lower than the one of REO-NO-QUEUE. For the other variants (REO-QUEUE-UPDATE and REO-QUEUE), the percentage of decrease of wT is even larger. This justifies the importance of considering the patients' urgency level when taking decisions. On the other hand, we observe that allowing free doctors to become idle could be advantageous if the idle time is as short as possible. The decrease of wT is 37.30% when moving from $t_{wait} = 20$ minutes to $t_{wait} = 1$ minute. This decrease is also impactful (i.e., 6.84%) even moving from $t_{wait} = 5$ to $t_{wait} = 1$. In fact, slightly better results are obtained when no waiting time for doctors is allowed. Moreover, it seems better to keep patients waiting in their queues (REO-QUEUE) instead of moving them to more urgent queues as time goes on (REO-QUEUE-UPDATE). The decrease of wT is 1.05% when going from REO-QUEUE-UPDATE to REO-QUEUE.

6.4. The impact of different number of doctors

We present in Table 4 the results obtained by varying the number of doctors in the set $\{2, 3, 4, 5, 6\}$. For each method, we show the average total weighted tardiness (wT) and computing time (in seconds) for all 186 instances of the ED in Italy. The first two lines of the table contain the results of the AF model and the VNS for the deterministic problem, while the other lines refer to the dynamic problem (DSPD). In the last four lines, we consider the SBPA-VNS results and show the percentage reduction for the ED with 2, 3, 4, and 5 doctors. For example, in the line "Red. 3 doctors (%)" we assume the ED has 3 doctors and report the percentage decrease from having 4 doctors instead of 3 ones, 5 doctors instead of 3, and 6 doctors instead of 3 (a negative value would mean that having 3 doctors could be better). In general, the more doctors the ED has, the smaller the wT and computing times are. In the deterministic problem, the VNS is quite competitive with the AF model, obtaining the same wT for the ED with 5 and 6 doctors, while presenting a much shorter computing time for the ED with 2 and 3 doctors.

Table 3
Impact of different policies in the heuristic REO-QUEUE (Italian ED instances, 3 doctors).

	REO-NO-QUEUE	REO-QUEUE-WAIT				REO-QUEUE-UPDATE	REO-QUEUE
		$t_{wait} = 20$	$t_{wait} = 10$	$t_{wait} = 5$	$t_{wait} = 1$		
execution time (s)	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
wT	281.74	181.44	143.41	122.12	113.77	112.39	111.21
Red. REO-NO-QUEUE (%)	-	35.60	49.10	56.66	59.62	60.11	60.53
Red. REO-QUEUE-WAIT $t_{wait} = 20$ (%)	-	-	20.96	32.69	37.30	38.06	38.71
Red. REO-QUEUE-WAIT $t_{wait} = 10$ (%)	-	-	-	14.85	20.67	21.63	22.45
Red. REO-QUEUE-WAIT $t_{wait} = 5$ (%)	-	-	-	-	6.84	7.97	8.93
Red. REO-QUEUE-WAIT $t_{wait} = 1$ (%)	-	-	-	-	-	1.21	2.25
Red. REO-QUEUE-UPDATE (%)	-	-	-	-	-	-	1.05

Table 4
Impact of the number of doctors in the ED (Italian ED instances).

Method	2 doctors		3 doctors		4 doctors		5 doctors		6 doctors	
	time	*wT*	time	*wT*	time	*wT*	time	*wT*	time	*wT*
ARC-FLOW	34.66	858.87	23.20	49.60	0.09	0.11	0.04	0.00	0.01	0.00
VNS	2.95	871.88	1.13	50.93	0.42	0.21	0.29	0.00	0.23	0.00
REO-QUEUE	< 0.01	1417.91	< 0.01	111.21	< 0.01	4.94	< 0.01	1.36	< 0.01	0.65
REO-VNS	2.33	1191.42	0.76	87.84	0.10	3.17	0.07	1.31	0.07	0.56
SBPA-VNS	592.12	1175.55	478.11	81.19	238.41	3.17	235.32	1.23	231.36	0.56
Red. SBPA-VNS 2 doctors (%)	-	-	19.25	93.09	59.74	99.73	60.26	99.90	60.93	99.95
Red. SBPA-VNS 3 doctors (%)	-	-	-	-	50.13	96.10	50.78	98.49	51.61	99.31
Red. SBPA-VNS 4 doctors (%)	-	-	-	-	-	-	1.30	61.20	2.96	82.33
Red. SBPA-VNS 5 doctors (%)	-	-	-	-	-	-	-	-	1.68	54.47

The results in Table 4 for the DSPD show that the REO-QUEUE heuristic is the fastest heuristic but is not competitive in terms of weighted tardiness when the number of available doctors is small (e.g., for 2 and 3 doctors in the ED). It has overall the largest values of *wT*, which are an increase of 19.01% and 26.61% if compared to REO-VNS for 2 and 3 doctors, respectively. On the other hand, having VNS in the reoptimization heuristic (REO-VNS) results in much better values of *wT* compared to REO-QUEUE. The REO-VNS is still competitive with SBPA-VNS, returning the same value of *wT* in the presence of 4 and 6 doctors but requiring much shorter computing time. In the other configurations, with fewer doctors in the ED, SBPA-VNS is the most effective approach in terms of weighted tardiness. It is indeed able to decrease the value of *wT* obtained with REO-VNS by 1.33%, 7.57%, and 6.11% for the ED with 2, 3, and 5 doctors, respectively, leading to a satisfactory overall average reduction of 5.01%.

Concerning the savings the ED may achieve when having more doctors, we observe in Table 4 that a small change from 2 to 3 doctors in the ED is able to reduce the value of *wT* by 93.09% with SBPA-VNS. When we assume the possibility of having 6 doctors instead of 2, the value of *wT* is reduced by 99.95%. The reductions are still large when we have more doctors, e.g., going from 3 to 4 doctors (i.e., with one more doctor), with *wT* decreasing by 96.10%. These are really important improvements in the services if we think about the well-being of patients and the improvement of hospital KPIs. As the value of *wT* decreases in the presence of more doctors, when the number of doctors increases from 4 to 5 and from 5 to 6, the *wT* is reduced less significantly but still with a notable benefit.

We summarize in Table 5 some additional information from the results obtained by our proposed methods on the Italian ED instances with 3 doctors. They are related to the average values of:

- Number of changes in the solution: modifications in the schedule, counted over the number of changes in the positions of already scheduled patients. The insertion of newly arrived patients does not count as a change. In this way, we present results only for the REO-VNS and SBPA-VNS heuristics;
- Number of tardy patients: patients whose start time is after their due time;
- Number of patients waiting: patients currently in the ED after the arrival (release time) of the last patient;
- Violation of due time / #tardy patients: average due time violation per tardy patients;
- Total weighted tardiness (*wT*) / #doctors: the value of total weighted tardiness divided by the number of available doctors in the ED;
- *wT* of each doctor: the value of *wT* obtained from patients served by each doctor. The sum of these values, for each doctor, results in the total weighted tardiness;
- Total weighted tardiness (*wT*): solution cost, computed as $\sum_{j \in J} w_j T_j$;
- *wT* of patients by each urgency level: the value of *wT* obtained from patients having the given urgency level. The sum of these values, for each urgency level, results in the total weighted tardiness.

Observing Table 5, we can notice that the number of changes in the solution reaches a maximum average value of 34 with the scenario-based planning approach (SBPA-VNS). This value represents an increase of 41.67% compared to the number of changes obtained when the REO-VNS heuristic is applied. These additional changes allow SBPA-VNS to obtain solutions with reduced *wT* when compared to the solutions obtained by REO-VNS. Considering that patients are usually called by a monitor (modern information systems) or by a person in charge of that in the ED and that patients commonly do not have access to the calling order (e.g., her position in the queue), the person/system could be easily updated in real-time with the new schedule (from the proposed methods). The average numbers of tardy patients and patients waiting are quite small on average, being below 3 and 4 patients, respectively, for all methods. On the other hand, in the REO-QUEUE heuristic, the average violation of due time per tardy patient is superior to 7 minutes on average, while it remains below 4 minutes for all other methods. These results reinforce the importance of reoptimizing the schedules each time a new patient arrives in the ED.

Table 5
Additional information from the results for 3 doctors (Italian ED instances).

	VNS	REO-QUEUE	REO-VNS	SBPA-VNS
Changes in the solution	-	-	24	34
Number of tardy patients	1.15	2.54	2.52	2.44
Number of patients waiting	3.31	3.46	3.31	3.31
Violation of due time / #tardy patients	2.81	7.58	3.48	3.29
Average weighted tardiness (wT) / #doctors	16.98	37.07	29.28	27.06
wT of doctor #1	19.65	38.45	31.66	28.01
wT of doctor #2	17.05	38.35	27.87	24.41
wT of doctor #3	14.23	34.41	28.31	28.77
Average weighted tardiness (wT)	50.93	111.21	87.84	81.19
wT of patients with $u = 4$ (less urgent)	6.62	41.20	8.65	11.34
wT of patients with $u = 3$	34.94	57.96	59.32	52.41
wT of patients with $u = 2$	8.85	5.58	12.77	10.71
wT of patients with $u = 1$ (most urgent)	0.52	6.47	7.10	6.73

Table 6
Impact of the early information in the EDs (3 doctors available).

Method	ED in Hong Kong SAR				ED in Italy									
	no EI		with EI		no EI		10% of EI		15% of EI		20% of EI		30% of EI	
	Time	wT	Time	wT	Time	wT	Time	wT	Time	wT	Time	wT	Time	wT
REO-QUEUE	< 0.01	43.60	< 0.01	41.67	< 0.01	111.21	< 0.01	114.05	< 0.01	116.77	< 0.01	120.29	< 0.01	124.67
REO-VNS	79.75	20.80	71.24	15.97	0.76	87.84	0.87	87.69	0.88	86.19	0.98	86.13	0.94	85.53
SBPA-VNS	-	-	-	-	478.11	81.19	468.73	81.19	464.63	79.50	461.18	79.50	454.39	77.88
Red. SBPA-VNS no EI (%)	-	-	7.57	23.22	-	-	1.96	0.00	2.82	2.08	3.54	2.08	4.96	4.08
Red. SBPA-VNS 10% (%)	-	-	-	-	-	-	-	-	0.87	2.08	1.61	2.08	3.06	4.08
Red. SBPA-VNS 15% (%)	-	-	-	-	-	-	-	-	-	-	0.74	0.00	2.20	2.04
Red. SBPA-VNS 20% (%)	-	-	-	-	-	-	-	-	-	-	-	-	1.47	2.04

Concerning the value of wT of each doctor in Table 5, we notice that the methods could satisfactorily balance this value among the available doctors. They are generally close to the value of "average weighted tardiness (wT) / #doctors". On the other hand, the values of wT according to the patients' urgency level are very different from each other. In general, the least urgent patients, with $u = 4$ and $u = 3$, have the highest values of wT, especially patients with $u = 3$. We notice the methods can reach small values of wT for the most urgent patients, around 7 minutes on average for patients with $u = 1$, which is highly desirable.

6.5. The impact of early information

The results of having early information (EI) about patients' arrival are presented in Table 6. The methods are applied to solve the 30 instances of PWH and the 186 instances of the ED in Italy. We also present the results without considering EI (columns "no EI") to investigate how this practice could benefit the EDs, supporting future decisions and investments to improve their services and KPIs. In addition to showing the results of the individual methods for the dynamic problem, the last lines of the table contain the percentage reductions of the values of wT (obtained with SBPA-VNS for the ED in Italy and REO-VNS for the ED in Hong Kong) when considering a certain percentage of EI and the previous percentage. We notice that, overall, having EI helps EDs in reducing the total weighted tardiness. The reduction in terms of wT is 23.22% for the ED in Hong Kong and at most 4.08% for the ED in Italy. For the latter, the more the percentage of patients having EI, the higher the reduction is.

The results of REO-QUEUE in Table 6 show a relevant improvement of 4.43% in the total weighted tardiness for the ED in Hong Kong when changing from "no EI" to "with EI". However, for the ED in Italy, results worsen as the percentage of patients with early information increases. In fact, for the latter, the best results are achieved when EI is not present, which is justified by the determinism of this heuristic to prioritize and schedule more urgent patients as soon as they are revealed. On the other hand, we notice the benefits when REO-VNS and SBPA-VNS are used to solve the instances with patients having EI. For the ED in Hong Kong, the reduction in terms of wT obtained with REO-VNS with respect to REO-QUEUE is 61.68%, which is a huge improvement. REO-VNS also returns a very satisfactory improvement of wT when changing from "no EI" to "with EI" in this ED.

For the ED in Italy in Table 6, the reductions obtained with REO-VNS with respect to those obtained with REO-QUEUE are 23.11%, 26.19%, 28.40%, and 31.39%, for the percentage of patients receiving EI being 10%, 15%, 20%, and 30%, respectively. Comparing SBPA-VNS and REO-QUEUE, these reductions are even larger, i.e. 28.81%, 31.92%, 33.91%, and 37.53% respectively. Indeed, SBPA-VNS is the most effective approach in terms of wT. It decreases the values obtained with REO-VNS by 7.41%, 7.76%, 7.70%, and 8.94%, respectively, leading to a satisfactory overall average reduction of 7.95%.

Concerning the savings that having early information could bring to emergency departments, we observe for the ED in Italy with the results of SBPA-VNS in Table 6 that changing from "no EI" to 15% of patients with EI reduces the value of wT by 2.08%. If, instead, we assume the possibility of having 30% of patients with EI, the value of wT decreases by 4.08%. As the wT decreases in the presence of more patients with EI, when going from 15% to 30% of patients with EI, the value of wT is reduced by 2.04%. No reduction is observed when the percentage of patients with EI increases from "no EI" to 10% and also from 15% to 20%. This can be justified by the EDs instances according to Table 1. The ED in Italy is characterized by patients with urgency levels $u = 3$ and $u = 2$, which in turn reflect the high number of patients of these levels with early information. Differently, the ED of Hong Kong has a predominance of less urgent patients ($u = 4$ and $u = 3$) with early information. In this way, the proposed methods has been showing that the presence of EI in EDs is indeed a good practice and greater savings can be achieved when EI is associated with more patients.

7. Concluding remarks

The flow of patients within emergency departments (EDs) has been the subject of intense study and discussion. This topic involves a large list of possible optimization problems, starting from the scheduling of work shifts to the minimization of service costs. In this paper, we focus on the minimization of the weighted tardiness of patients arriving dynamically at the ED over a given time horizon by optimizing their assignment to the available doctors. We propose a set of solution methods, ranging from simple reoptimization heuristics to a sophisticated scenario-based planning heuristic that considers sampled scenarios to anticipate future decisions. We also propose an arc-flow (AF) formulation and a variable neighborhood search (VNS) heuristic to handle the static problem and provide a quality assessment for the dynamic stochastic problem.

The methods have been validated on realistic instances obtained from hospitals in Italy and Hong Kong SAR, China. Results of the reoptimization heuristics show the need for more elaborate policies to guide decisions, as in the case of using the patients' urgency level. Another important finding is related to the waiting times of free doctors, which can be advantageous if this time is as short as possible. Besides that, if we desire to have high-quality solutions, instead of convenient policies, it is much better to use an optimization method to obtain improved schedules. Obviously, the use of policies has the advantage of being easily implemented by decision-makers. On the other hand, nowadays due to the computerization of EDs and hospitals, it would not be a problem to introduce optimized decision-supporting systems in such environments. Therefore, we use a general VNS to achieve this purpose. We notice that better results can indeed be obtained if stochastic information is used to anticipate future decisions. In this case, the scenario-based planning approach (combined with the VNS) using a consensus function based on the most common decisions has provided the best overall results for the dynamic problem. With such a method, we obtain really important savings in the total weighted tardiness with respect to the reoptimization heuristics.

Concerning the influence of the number of doctors in the EDs efficiency, results have suggested smaller weighted tardiness when more doctors are available. While EDs operating with two doctors have solutions in which the average weighted tardiness is superior to 1000, this number decreases below 4 weighted minutes in the presence of four doctors, achieving values near zero when six doctors are considered. Similarly, receiving early information about patients is a beneficial practice for EDs, especially if more patients with high urgency levels have information revealed as soon as possible. It would be worthwhile to have such a practice in EDs and hospitals around the globe.

Finally, we would like to point out some future research directions. Concerning methodological aspects, methods based on sampled scenarios can be further improved by new consensus functions and strategies that make better use of information from scenarios, e.g., sophisticated branch-and-regret heuristics to evaluate different possible alternatives for each available doctor (Hvattum et al., 2007). It could be also interesting to propose VNS neighborhood structures based on patients' characteristics (e.g., urgency level). Another direction is to consider a scenario tree structure where scenarios are updated in an on-the-fly manner taking into account past information. This could lead to very large scenario trees, impacting the computational effort. However, they could be adjusted in the process of solution while maintaining as much information as possible (Birge & Louveaux, 2011). For what concerns similar optimization problems from the dynamic scheduling domain, it would be interesting to verify the performance of the proposed methods, after the necessary modifications, on problems considering other objective functions (e.g., total flow time) or different characteristics of machines (e.g., uniform or unrelated machines). This research could also involve the case of deteriorating jobs, implying that the performance of a machine (in the ED, the efficiency of a doctor) decreases after having processed one or more activities. It would also be interesting to study the case of (stochastic) preemptions of jobs (in the ED, less urgent patients are laid aside for a while in case very urgent patients enter the ED and need immediate care).

Acknowledgments

This research was funded by Research Grants Council of Hong Kong (grant no. ECS 27200419), the National Council of Technological and Scientific Development (CNPq - grants no. 405369/2021-2 and 311185/2020-7), and the State of Goiás Research Foundation (FAPEG).

References

Ahmed, M. A., & Alkhamis, T. M. (2009). Simulation optimization for an emergency department healthcare unit in Kuwait. *European Journal of Operational Research, 198*(3), 936–942.

Albers, S., & Hellwig, M. (2012). Semi-online scheduling revisited. *Theoretical Computer Science, 443*, 1–9.

Almeida Cunha, J. G., de Lima, V. L., & Queiroz, T. A. (2020). Grids for cutting and packing problems: A study in the 2D knapsack problem. *4OR-A Quarterly Journal of Operations Research, 18*, 293–339.

Angelelli, E., Speranza, M. G., & Tuza, Z. (2008). Semi-online scheduling on two uniform processors. *Theoretical Computer Science, 393*(1–3), 211–219.

Archetti, C., Feillet, D., Mor, A., & Speranza, M. (2020). Dynamic traveling salesman problem with stochastic release dates. *European Journal of Operational Research, 280*(3), 832–844.

Arkin, E. M., & Roundy, R. O. (1991). Weighted-tardiness scheduling on parallel machines with proportional weights. *Operations Research, 39*(1), 64–81.

Azadeh, A., Baghersad, M., Farahani, M. H., & Zarrin, M. (2015). Semi-online patient scheduling in pathology laboratories. *Artificial intelligence in medicine, 64*(3), 217–226.

Bakker, H., Dunke, F., & Nickel, S. (2020). A structuring review on multi-stage optimization under uncertainty: Aligning concepts from theory and practice. *Omega, 96*, 102080.

Baptiste, P., Jouglet, A., & Savourey, D. (2008). Lower bounds for parallel machine scheduling problems. *International Journal of Operational Research, 3*(6), 643–664. https://doi.org/10.1504/IJOR.2008.019731. https://www.inderscienceonline.com/doi/abs/10.1504/IJOR.2008.019731.

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

Bent, R. W., & van Hentenryck, P. (2004). Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research, 52*(2), 977–987.

Birge, J. R., & Louveaux, F. (2011). *Introduction to stochastic programming*. New York: Springer.

Bovim, T. R., Christiansen, M., Gullhav, A. N., Range, T. M., & Hellemo, L. (2020). Stochastic master surgery scheduling. *European Journal of Operational Research, 285*(2), 695–711.

Bulhões, T., Sadykov, R., Subramanian, A., & Uchoa, E. (2020). On the exact solution of a large class of parallel machine scheduling problems. *Journal of Scheduling, 23*, 411–429.

Camiat, F., Restrepo, M. I., Chauny, J.-M., Lahrichi, N., & Rousseau, L.-M. (2021). Productivity-driven physician scheduling in emergency departments. *Health Systems, 10*(2), 104–117.

Chen, X., Sterna, M., Han, X., & Blazewicz, J. (2016). Scheduling on parallel identical machines with late work criterion: Offline and online cases. *Journal of Scheduling, 19*(6), 729–736.

Côté, J.-F., & Iori, M. (2018). The meet-in-the-middle principle for cutting and packing problems. *INFORMS Journal on Computing, 30*(4), 646–661.

de Lima, V. L., Alves, C., Clautiaux, F., Iori, M., & de Carvalho, J. M. V. (2022). Arc flow formulations based on dynamic programming: Theoretical foundations and applications. *European Journal of Operational Research, 296*(1), 3–21.

Derlet, R. W., & Richards, J. R. (2000). Overcrowding in the nation's emergency departments: Complex causes and disturbing effects. *Annals of Emergency Medicine, 35*(1), 63–68.

Di Somma, S., Paladino, L., Vaughan, L., Lalle, I., Magrini, L., & Magnanti, M. (2015). Overcrowding in emergency department: An international issue. *Internal and Emergency Medicine, 10*(2), 171–175.

Ding, Y., Park, E., Nagarajan, M., & Grafstein, E. (2019). Patient prioritization in emergency department triage systems: An empirical study of the canadian triage and acuity scale (CTAS). *Manufacturing & Service Operations Management, 21*(4), 723–741.

Dosi, C., Iori, M., Kramer, A., & Vignoli, M. (2019). Computational simulation as an organizational prototyping tool. *Proceedings of the Design Society: International Conference on Engineering Design, 1*(1), 1105–1114. https://doi.org/10.1017/dsi.2019.116.

Dosi, C., Iori, M., Kramer, A., & Vignoli, M. (2020). Facing implementation barriers to healthcare simulation studies. In V. Bélanger, N. Lahrichi, E. Lanzarone, & S. Yalçındağ (Eds.), *Health care systems engineering* (pp. 117–129). Springer International Publishing.

Dosi, C., Iori, M., Kramer, A., & Vignoli, M. (2021). Successful implementation of discrete event simulation: Integrating design thinking and simulation approach in an emergency department. *Production Planning & Control*, 1–15. Forthcoming

Durasević, M., & Jakobović, D. (2022). Heuristic and metaheuristic methods for the parallel unrelated machines scheduling problem: A survey. *Artifcial Intelligence Review*. Forthcoming

Epstein, L. (2018). A survey on makespan minimization in semi-online environments. *Journal of Scheduling, 21*(3), 269–284.

Fernandes, C. M., Tanabe, P., Gilboy, N., Johnson, L. A., McNair, R. S., Rosenau, A. M., … Bonalumi, N., et al., (2005). Five-level triage: A report from the ACEP/ENA five-level triage task force. *Journal of Emergency Nursing, 31*(1), 39–50.

Graham, R., Lawler, E., Lenstra, J., & Kan, A. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. In E. J. P. L. Hammer, & B. Korte (Eds.), *Discrete optimization ii proceedings of the advanced research institute on discrete optimization and systems applications*. In *Annals of Discrete Mathematics: vol. 5* (pp. 287–326). Elsevier.

Granja, C., Almada-Lobo, B., Janela, F., Seabra, J., & Mendes, A. (2014). An optimization based on simulation approach to the patient admission scheduling problem using a linear programing algorithm. *Journal of biomedical informatics, 52*, 427–437.

Green, L. V., Soares, J., Giglio, J. F., & Green, R. A. (2006). Using queueing theory to increase the effectiveness of emergency department provider staffing. *Academic Emergency Medicine, 13*(1), 61–68.

Gupta, D., Maravelias, C. T., & Wassick, J. M. (2016). From rescheduling to online scheduling. *Chemical Engineering Research and Design, 116*, 83–97. https://doi.org/10.1016/j.cherd.2016.10.035.

Hansen, P., & Mladenović, N. (2018). *Variable neighborhood search*. In R. Martí, P. M. Pardalos, & M. G. C. Resende (Eds.) (pp. 759–787)). Springer International Publishing.

Hansen, P., Mladenović, N., Todosijević, R., & Hanafi, S. (2017). Variable neighborhood search: Basics and variants. *EURO Journal on Computational Optimization, 5*, 423–454.

He, S., Sim, M., & Zhang, M. (2019). Data-driven patient scheduling in emergency departments: A hybrid robust-stochastic approach. *Management Science, 65*(9), 4123–4140.

Hoot, N. R., LeBlanc, L. J., Jones, I., Levin, S. R., Zhou, C., Gadd, C. S., & Aronsky, D. (2008). Forecasting emergency department crowding: A discrete event simulation. *Annals of Emergency Medicine, 52*(2), 116–125.

Huang, J., Carmeli, B., & Mandelbaum, A. (2015). Control of patient flow in emergency departments, or multiclass queues with deadlines and feedback. *Operations Research, 63*(4), 892–908.

Hvattum, L. M., Løkketangen, A., & Laporte, G. (2006). Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science, 40*(4), 421–438.

Hvattum, L. M., Løkketangen, A., & Laporte, G. (2007). A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems. *Networks, 49*(4), 330–340.

Ieraci, S., Digiusto, E., Sonntag, P., Dann, L., & Fox, D. (2008). Streaming by case complexity: Evaluation of a model for emergency department fast track. *Emergency Medicine Australasia, 20*(3), 241–249.

Jiang, Y., Abouee-Mehrizi, H., & Diao, Y. (2020). Data-driven analytics to support scheduling of multi-priority multi-class patients with wait time targets. *European Journal of Operational Research, 281*(3), 597–611.

Jouglet, A., & Savourey, D. (2011). Dominance rules for the parallel machine total weighted tardiness scheduling problem with release dates. *Computers & Operations Research, 38*(9), 1259–1266. https://doi.org/10.1016/j.cor.2010.12.006. https://www.sciencedirect.com/science/article/pii/S0305054810002960.

Kacem, I., Souayah, N., & Haouari, M. (2012). Branch-and-bound algorithm for total weighted tardiness minimization on parallel machines under release dates assumptions. *RAIRO-Operations Research, 46*(2), 125–147. https://doi.org/10.1051/ro/2012010.

Kamali, M. F., Tezcan, T., & Yildiz, O. (2018). When to use provider triage in emergency departments. *Management Science, 65*(3), 1003–1019.

Kedad-Sidhoum, S., Solis, Y. R., & Sourd, F. (2008). Lower bounds for the earliness–tardiness scheduling problem on parallel machines with distinct due dates. *European Journal of Operational Research, 189*(3), 1305–1316. https://doi.org/10.1016/j.ejor.2006.05.052. https://www.sciencedirect.com/science/article/pii/S0377221707006029.

King, D. L., Ben-Tovim, D. I., & Bassham, J. (2006). Redesigning emergency department patient flows: Application of lean thinking to health care. *Emergency Medicine Australasia, 18*(4), 391–397.

Kramer, A., Dell'Amico, M., Feillet, D., & Iori, M. (2020). Scheduling jobs with release dates on identical parallel machines by minimizing the total weighted completion time. *Computers & Operations Research, 123*, 105018.

Kramer, A., Dell'Amico, M., & Iori, M. (2019). Enhanced arc-flow formulations to minimize weighted completion time on identical parallel machines. *European Journal of Operational Research, 275*(1), 67–79.

Kramer, A., & Subramanian, A. (2019). A unified heuristic and an annotated bibliography for a large class of earliness–tardiness scheduling problems. *Journal of Scheduling, 22*, 21–57.

Kuo, Y. H. (2014). Integrating simulation with simulated annealing for scheduling physicians in an understaffed emergency department. *HKIE Transactions, 21*(4), 253–261.

Kuo, Y.-H., Leung, J. M., Graham, C. A., Tsoi, K. K., & Meng, H. M. (2018). Using simulation to assess the impacts of the adoption of a fast-track system for hospital emergency services. *Journal of Advanced Mechanical Design, Systems, and Manufacturing, 12*(3), JAMDSM0073.

Kuo, Y.-H., Rado, O., Lupia, B., Leung, J. M., & Graham, C. A. (2016). Improving the efficiency of a hospital emergency department: A simulation study with indirectly imputed service-time distributions. *Flexible Services and Manufacturing Journal, 28*(1-2), 120–147.

Kwasnick, S. (2017). *"Wait for It": Toward optimal emergency department wait time prediction*. Stanford University Ph.D. thesis.

Lan, S., Fan, W., Yang, S., Pardalos, P. M., & Mladenović, N. (2021). A survey on the applications of variable neighborhood search algorithm in healthcare management. *Annals of Mathematics and Artificial Intelligence*. https://doi.org/10.1007/s10472-021-09727-5. in press

Larsen, R., & Pranzo, M. (2019). A framework for dynamic rescheduling problems. *International Journal of Production Research, 57*(1), 16–33. https://doi.org/10.1080/00207543.2018.1456700.

Lee, S., & Lee, Y. H. (2020). Improving emergency department efficiency by patient scheduling using deep reinforcement learning. *Healthcare, 8*(2), 1–17.

Legrain, A., Omer, J., & Rosat, S. (2020). An online stochastic algorithm for a dynamic nurse scheduling problem. *European Journal of Operational Research, 285*(1), 196–210.

Lenstra, J., Kan, A. R., & Brucker, P. (1977). Complexity of machine scheduling problems. In P. Hammer, E. Johnson, B. Korte, & G. Nemhauser (Eds.), *Studies in integer programming*. In *Annals of Discrete Mathematics: vol. 1* (pp. 343–362). Elsevier.

Luo, D., Bayati, M., Plambeck, E. L., & Aratow, M. (2017). Low-acuity patients delay high-acuity patients in the emergency department. Available at SSRN 3095039.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research, 24*(11), 1097–1100.

Ng, C. T., Tan, Z., He, Y., & Cheng, T. E. (2009). Two semi-online scheduling problems on two uniform machines. *Theoretical Computer Science, 410*(8–10), 776–792.

Oh, C., Novotny, A. M., Carter, P. L., Ready, R. K., Campbell, D. D., & Leckie, M. C. (2016). Use of a simulation-based decision support tool to improve emergency department throughput. *Operations Research for Health Care, 9*, 29–39.

Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling, 12*(4), 417–431.

Pan, X., Geng, N., & Xie, X. (2021). Appointment scheduling and real-time sequencing strategies for patient unpunctuality. *European Journal of Operational Research, 295*(1), 246–260.

Pessoa, A., Uchoa, E., de Aragão, M. P., & Rodrigues, R. (2010). Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. *Mathematical Programming Computation, 2*(3), 259–290.

Queiroz, T. A., Iori, M., Kramer, A., & Kuo, Y.-H. (2021). Scheduling of patients in emergency departments with a variable neighborhood search. In N. Mladenović, A. Sleptchenko, A. Sifaleras, & M. Omar (Eds.), *Variable neighborhood search* (pp. 138–151). Cham: Springer International Publishing.

Richardson, D. B. (2006). Increase in patient mortality at 10 days associated with emergency department overcrowding. *Medical Journal of Australia, 184*(5), 213–216.

Rossit, D. A., Tohmé, F., & Frutos, M. (2019). Industry 4.0: Smart scheduling. *International Journal of Production Research, 57*(12), 3802–3813. https://doi.org/10.1080/00207543.2018.1504248.

Saghafian, S., Austin, G., & Traub, S. J. (2015). Operations research/management contributions to emergency department patient flow optimization: Review and research prospects. *IIE Transactions on Healthcare Systems Engineering, 5*(2), 101–123.

Saghafian, S., Hopp, W. J., Van Oyen, M. P., Desmond, J. S., & Kronick, S. L. (2012). Patient streaming as a mechanism for improving responsiveness in emergency departments. *Operations Research, 60*(5), 1080–1097.

Saghafian, S., Hopp, W. J., Van Oyen, M. P., Desmond, J. S., & Kronick, S. L. (2014). Complexity-augmented triage: A tool for improving patient safety and operational efficiency. *Manufacturing & Service Operations Management, 16*(3), 329–345.

Schilde, M., Doerner, K. F., & Hartl, R. F. (2014). Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. *European Journal of Operational Research, 238*(1), 18–30. https://doi.org/10.1016/j.ejor.2014.03.005.

Şen, H., & Bülbül, K. (2015). A strong preemptive relaxation for weighted tardiness and earliness/tardiness problems on unrelated parallel machines. *INFORMS Journal on Computing, 27*(1), 135–150. https://doi.org/10.1287/ijoc.2014.0615.

Shim, S.-O., & Kim, Y.-D. (2007). Scheduling on parallel identical machines to minimize total tardiness. *European Journal of Operational Research, 177*(1), 135–146.

Song, Y., Ulmer, M. W., Thomas, B. W., & Wallace, S. W. (2020). Building trust in home services—stochastic team-orienteering with consistency constraints. *Transportation Science, 54*(3), 823–838. https://doi.org/10.1287/trsc.2019.0927.

Souayah, N., Kacem, I., Haouari, M., & Chu, C. (2009). Scheduling on parallel identical machines to minimise the total weighted tardiness. *International Journal of Advanced Operations Management, 1*(1), 30–69.

Tirado, G., Hvattum, L. M., Fagerholt, K., & Cordeau, J.-F. (2013). Heuristics for dy-

namic and stochastic routing in industrial shipping. *Computers & Operations Research, 40*(1), 253–263. https://doi.org/10.1016/j.cor.2012.06.011.
Tsai, S. C., Yeh, Y., & Kuo, C. Y. (2021). Efficient optimization algorithms for surgical scheduling under uncertainty. *European Journal of Operational Research, 293*(2), 579–593.
Ulmer, M. W., Goodson, J. C., Mattfeld, D. C., & Thomas, B. W. (2020). On modeling stochastic dynamic vehicle routing problems. *EURO Journal on Transportation and Logistics, 9*(2), 100008.
van den Akker, J. M., Hoogeveen, J. A., & Van de Velde, S. L. (1999). Parallel machine scheduling by column generation. *Operations Research, 47*(6), 862–872.
van Hentenryck, P., & Bent, R. (2006). *Online stochastic combinatorial optimization*. Cambridge: MIT Press.
Vanbrabant, L., Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2019). Simulation of emergency department operations: A comprehensive review of KPIs and operational improvements. *Computers & Industrial Engineering, 131*, 356–381.
Vieira, G. E., Herrmann, J. W., & Lin, E. (2003). Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *Journal of Scheduling, 6*(1), 39–62.
Voccia, S. A., Campbell, A. M., & Thomas, B. W. (2019). The same-day delivery problem for online purchases. *Transportation Science, 53*(1), 167–184.
Wen, J., Geng, N., & Xie, X. (2020). Real-time scheduling of semi-urgent patients under waiting time targets. *International Journal of Production Research, 58*(4), 1127–1143.