CIRRELT

CIRRELT-2023-10

Real-Time Management of Intra-Hospital Patient Transport Requests: An Empirical Study

Vinicius M. Ton
Nathália C. O. da Silva
Angel Ruiz
José E. Pécora Jr.
Cassius T. Scarpin
Valérie Bélanger

February 2023

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2023-002

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Real-Time Management of Intra-Hospital Patient Transport Requests: An Empirical Study

Vinicius M. Ton[1,2,3,*], Nathália C. O. da Silva[3], Angel Ruiz[1,2], José E. Pécora Jr.[1,2], Cassius T. Scarpin[3], Valérie Bélanger[1,4]

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
2. Department of Operations and Decision Systems, Université Laval, Québec, Canada
3. Group of Technology Applied to Optimization (GTAO), Brazil
4. Department of Logistics and Operations Management, HEC Montréal, Québec, Canada

Abstract. This paper addresses the management of patients transportation requests within a hospital, a very challenging problem where waiting requests must be scheduled among the available porters in such a way that patients arrive at their destination timely and the resources invested in patient transport are kept as low as possible. Moreover, since transportation requests arrive unpredictably, the problem must be solved in real-time. To deal with such a dynamic context, all pending requests are rescheduled periodically. We propose several strategies to trigger the rescheduling of waiting requests and three approaches (a mathematical formulation, a constructive heuristic, and a local search heuristic) to solve each rescheduling problem. A simulation tool is proposed to evaluate the potential of the rescheduling strategies and the proposed scheduling methods to tackle instances inspired by a real mid-size hospital. The local search heuristic, which produces the best results, is fast enough to be used in a real context and achieves significant reductions in the response time, total distance walked by porters, and percentage of late requests compared to the results produced by a constructive heuristic which mimics the manner in which our partner hospital presently manages requests.

Keywords: patient transportation, real-time scheduling, intra-hospital transportation, healthcare logistics

---

* Corresponding author: vinicius.martins-ton.1@ulaval.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2023

1. Introduction

Clinical diagnostics processes increasingly use a growing collection of tests, analyses, and consultations. Patients interact with a more extensive set of specialists and practitioners, leading to more complex pathways involving patient's moves between the hospital's services or departments. Patient flow, either within a given hospital or healthcare facility or involving transportation from one hospital to another, constitutes one of the most important logistics flows in the healthcare context.

Patient transportation distinguishes between *inter-hospital* transportation (defined as the patient's transport between their home and the hospital or even between two hospitals) and *intra-hospital* transportation, which concerns the transportation of a patient between two any services within the same hospital. Although *inter-* and *intra-hospital* transportation problems are very close, they also show, as it will be pointed out later, differences justifying the development of specific approaches to deal with each problem effectively and efficiently.

As the one we observed and inspired this research, a mid-size hospital handles around a thousand patients' transportation requests every weekday. These requests concern inpatients (i.e., patients hospitalized), but in some cases, also outpatients receiving ambulatory services at the hospital. To ensure the safety, comfort, and quality of the service offered to patients, hospitals have a transportation office. In this office, transportation requests are received, handled, and assigned among a crew of assistants referred to as porters. The performance of the transportation service directly impacts the hospital's operations, as patients arriving late to their appointments will cause delays in scheduled services. It can also impact the patients' experience related to waiting times before and after transport.

Finally, from a managerial standpoint, the number of resources invested in patient transport, including personnel and rolling equipment, is far from negligible. Therefore, hospitals seek to simultaneously minimize the waiting times for both patients and services and the cost of the transportation system, which is basically related to the porters' salaries.

In this context, this paper seeks to propose several methods to manage patients' transportation requests in real-time and assess their efficiency. We model the assignment of transport requests to porters as a parallel machine scheduling problem with sequence-dependent setup times (PMSP-SDST), and we propose a mathematical formulation, a constructive heuristic, and a local search heuristic for addressing the static situation whereby a set of transport requests are given.

To deal with the dynamic arrival of requests, we periodically reconsider the schedule for all waiting requests. This process will be referred to as rescheduling. Several approaches to trigger the rescheduling process are proposed, and a simulation tool empirically assesses their performance to handle instances inspired by the real case of a mid-sized hospital in the province of Quebec, Canada.

The remainder of the paper is organized as follows. The next section describes the problem of *intra-hospital* patient transportation, followed by a brief review of related papers. The mathematical formulation and the heuristics designed to deal with the problem of dispatching static requests are then proposed, followed by a description of the approaches for triggering rescheduling. Computational results are reported and analyzed. Finally, conclusions and suggestions for future research are presented.

2. Handling requests for patient transport

Intra-hospital patient transport involves the movement of patients with limited mobility or requiring supervised transport from one location to another for diagnosis or therapeutic reasons within the same building (Painchaud, Bélanger, & Ruiz 2017). In *intra-hospital* transport, porters move patients using stretchers, beds, or wheelchairs. The following paragraphs describe how *intra-building* transportation requests are handled.

The transportation office (TO) receives and handles requests from all services and departments of the hospital, including the admission and discharge offices. We assume that porters use some communication equipment that allows them to be in contact with the TO. Some requests are submitted in advance, but the majority require immediate transport. A transportation request i is characterized by an origin o_i (the location of the patient) and a destination d_i. It also contains the time the request was received by the TO, t_i^a, and its due time t_i^d, which means the latest time the patient is expected at the destination. Note that for requests placed in advance, t_i^a corresponds to the time at which the patient will be ready for transportation. In some hospitals, for example, in the one inspiring this research, the request's due time t_i^d is replaced by a level of priority λ_i that defines the maximum allotted time to complete the request. Finally, t_i, the traveling time from o_i to d_i, is known.

Let us assume that a request i having t_i^a smaller or equal to the current time is assigned to a porter p. Porter p leaves the TO (or their current location) and moves to location o_i. Then, the porter transports the patient to their destination d_i, where the patient is transferred to the service personnel. The porter contacts the TO to confirm the completion of the current task and to inquire about the subsequent request to perform. In the span between the completion of a request and his assignment to a new one, the porter is said to be idle. Finally, the delay of a request i is 0 if the patient arrives at the destination before its due time t_i^d. Otherwise, the delay for the request is computed as $C_i - t_i^d$, where C_i denotes the completion time (the arrival at the destination) of the request i.

During the day, porters become busy, and incoming requests are queued. The TO's dispatcher must decide which request (if any) to assign a porter whenever one becomes available; this decision is not straightforward. The TO can assign the queued requests according to their arrival time. However, this policy does not guarantee a shorter patient wait or an earlier arrival at the destination. Indeed, assigning requests in a FIFO (first come, first served) manner may force the available porter to travel a long distance from their current location to the origin of the request i, as illustrated by Figure 1. The left part (a) of the figure shows four requests waiting, i, $i+1$, $i+2$, and $i+3$ to be served. Two porters p_1 and p_2 are traveling to the destinations of their current requests d_{i-1} and d_{i-2}, respectively. Assuming that p_1 arrives at the destination before p_2, the dispatcher applies the FIFO rule and assigns p_1 to the earliest request i. Then, when p_2 arrives at d_{i-2}, the dispatcher assigns them to $i+1$. The central part (b) of Figure 1 illustrates these assignments. However, if the dispatcher had taken into account the arrival of porter p_2 at destination d_{i-2}, the distances traveled by the porters would have been reduced by the assignment of porter p_1 to request $i+1$, and porter p_2 to request r_i, as shown by (c), the right part of Figure 1.

Assignment decisions must therefore consider the wait times and expected duration of queued requests, but also how they will be scheduled. Indeed, one of the most important features of this problem is related to the porters' movement between two consecutive requests, which depends on the sequence of jobs assigned to each porter.

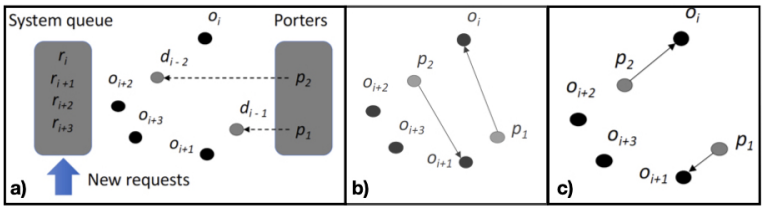

Figure 1. Example of dispatch decisions and their potential outcomes.

Moreover, since new requests arrive at the system from unpredictable locations at random times, what seems to be the right choice now, might not be ideal a few minutes later, rendering this a real-time decision-making problem.

3. Literature review

As mentioned previously, *inter-* and *intra-hospital* transportation problems are strongly related, although they also show significant differences. For that reason, this section reviews relevant works on the two problems and concludes with a discussion on the specific features of *intra-hospital* transportation to position and justify the contributions of this research.

To the best of our knowledge, Dershin and Schaik (1993) presented one of the first works devoted to improving *intra-hospital*, also referred to as in-house transport. Their work is three-part because it proposes improvements to the communication and control system, the development of a rational staffing model, and a database to monitor long-term performance. They used a queuing model to estimate waiting time based on the number of porters, assuming a fixed service rate (i.e., the number of services completed by a porter in an hour). Their recommendations considerably improved the system performance and, furthermore, demonstrated to managers that dramatic waiting times increase when call loading exceeds certain threshold levels.

The interest in *intra-hospital* patient transportation strongly increased at the end of the 2010s. Naesens and Gelders (2009) presented a case study of a hospital in Belgium, where the goal was to reduce the long waiting times for patient transportation between services. To this end, the authors performed a detailed study of transport operations. They analyzed patient flows and recommended replacing the existing centralized approach with a new decentralized service.

Fiegl and Pontow (2009) extended the transportation of patients to include the transport of medical items (e.g., records, forms, medicine, and laboratory samples) between services. Contrary to the transport of patients, where a request must be completed before starting the next, the problem becomes a variant of the pickup-and-delivery routing problem since several items can be transported together. They proposed an online optimization approach based on the highest density first rule proposed by Prughs, Sgall, and Torng (2004) to minimize the flow time and thereby ensure the highest possible task throughput.

Segev, Levi, Dunn, and Sandberg (2012) studied a particular case of patient transportation where patients waiting for surgery needed to be moved from the admission and preparation area to the operating theater. The work aimed to determine the appropriate number of dedicated elevators and porters to prevent delays in patient arrivals to operating theaters. To this end, a data-driven simulation tool was developed.

Although this work differs from the more general case where a large set of patients' origins and destinations are considered, it contributes accurate stochastic models for elevators and traveling times. It also presents a sensitivity analysis of the system performance concerning the number of dedicated elevators and available porters, which are among the most important considerations when designing a patient transportation system.

Hanne, Melo, and Nickel (2009) designed a computer-based planning system to coordinate *inter-building* transportation. This system is based on a dynamic DARP, which they solve with different heuristics. Indeed, several papers have modeled the *inter-building* transport problem faced by large hospitals owning several buildings, or even the transport of patients from their homes to hospitals and from hospitals back to their homes, as a DARP, a variant of a vehicle routing problem providing multi-occupancy, door-to-door transport service for people that aims to minimize simultaneously total traveled distance by the ambulances and patients inconvenience. This "inconvenience" measures the increase of the ride for any patient with respect to the shortest (direct) ride caused by sharing the route). We refer interested readers to the review on DARP by Cordeau and Laporte (2007) and the review on people transportation by Doerner and Salazar-González (2014).

Cordeau and Laporte (2003) proposed a two-neighborhood Tabu search metaheuristic to solve the static version of DARP (i.e., the transportation requests are known a priori). The first neighborhood removes a request from its current route and re-inserts it in a different route; the second neighborhood consists of rearranging the sequence of not yet serviced requests in their assigned route. Later, their metaheuristic was adapted by Beaudry, Laporte, Melo, and Nickel (2010) to be used in a *inter-building* transport context where requests arrive dynamically. To this end, the authors developed a two-phase procedure that is executed each time a new request arrives. In the first phase, a simple heuristic is used to insert the new request into an existing route to generate a feasible solution, which is improved in the second phase (the Tabu search). Kergosien, Lenté, Piton, and Billaut (2011) addressed a similar context, but they considered the possibility of outsourcing transport requests to a private company. They also proposed a Tabu search metaheuristic to minimize the system's total cost, encompassing both the hospital and external company costs.

Schmid and Doerner (2014) studied a joint patient scheduling and transportation problem. In their case, patients need to follow a given set of treatments at different services in the same hospital. Once treatment is completed, the patient needs to be transported to the next service by a porter. The difficulty lies in identifying, for each patient, the start time for each treatment and assigning a porter to each transportation request in such a way that the total patient inconvenience, idle times in rooms, and empty movements by porters are minimized. The problem is modeled as a variant of the multi-depot vehicle-routing problem with time windows (MDVRPTW). However, the problem is static, i.e., all the patients and their required treatments are known in advance.

Some other papers addressed the transport of patients from their homes to hospitals (outbound requests) and from hospitals back to their homes (inbound requests). These works aimed to elaborate routes to pick up several patients in a vehicle, thus sharing the same transport to the hospital or from the hospital to the patient's homes. By doing so, vehicles are used more efficiently, although they may become longer for some patients. Bowers, Lyons, and Mould (2012) proposed a decision support tool for strategic resource allocation decisions on this version of *inter-hospital* patient transportation. A constraint ensures that the patients' inconvenience (the increase in the

riding time concerning the shortest direct ride) will not exceed an acceptable distance, thus limiting potential patient discomfort. Such constraints help maintain the equity of the service and patient comfort. Schilde, Doerner, and Hartl (2011) explored the fact that most of the patients who travel to the hospital return to their homes once the service is completed and how this can be used to 'anticipate' their return requests and used this probabilistic information to improve the planning of inbound routes. Four metaheuristic solution approaches were proposed to handle the resulting dynamic stochastic dial-a-ride problem to demonstrate that taking stochastic information about future return transport into account is beneficial under certain conditions discussed in the paper.

Finally, in recent years, *intra-hospital* transportation problems have focused on the porters' ergonomic stress induced by pushing and pulling patients' beds and wheelchairs. Since the physical effort depends on the conveyance vehicle, tour length, and patient weight, some works proposed scheduling formulations that aim to minimize the porters' ergonomic strain, which in turn minimizes the risk of musculoskeletal disorders (von Elmbach, Boysen, Briskorn, & Mothes 2015; von Elmbach, Scholl, & Walter 2019). Unfortunately, the proposed formulations are static, meaning that all requests need to be known a priori, which is clearly not realistic in a healthcare context, although von Elmbach et al. (2019) suggested that their approach might be adapted to tackle the dynamic arrival of requests.

We can conclude that the scientific literature contains rich contributions related to the transport of hospital patients. However, most of the literature focuses on *inter-building* problems concerning static variants where the transport requests are known in advance. As pointed out by von Elmbach et al. (2019), the development of efficient online algorithms seems challenging but a valuable task for future research, specifically for *inter-hospital* problems for which the contributions are scarce. In this vein, our work contributes to a comprehensive evaluation of several algorithms capable of handling real-time patient transport requests.

It is also worth pointing out the differences between *inter-* and *intra-hospital* problems that motivate the development of specific approaches for the latter case. Foremost, *inter-hospital* problems seek to transport efficiency by moving several patients together. Since *intra-hospital* transport concerns a patient at the time, efficiency lies in minimizing porters' "empty" travels between two requests. That is the reason why inter-hospital formulations minimize (or limit) patients' inconvenience or the increase in their travel time, while *intra-hospital* problems focus on lateness.

Even more, we have observed that patients' travel times in *intra-hospital* problems are quite short. For instance, the average travel time between services in the hospital we observed is around 5 min, which according to the TO's managers, fit well with times they saw in other hospitals in the province. Comparatively, Beaudry et al. (2010) reports ride times of 30 minutes in an ambulance and up to 21 minutes for urgent requests for transport between the buildings spread over a hospital campus. Finally, another difference concerns the problem's degree of dynamism Ψ (the ratio between requests not known in advance to the total number of requests). Kergosien et al. (2011) reports that the average dynamism in the real instances they obtained was $\Psi = 58\%$, which contrasts with our real case where none of the requests was known in advance ($\Psi = 100\%$).

The following section proposes exact and approximated approaches to handle the PMSP-SDST, the static version of the *intra-hospital* patient transportation problem. Then, Section 5 presents strategies (i.e., rescheduling triggering policies) to use the described approaches in a dynamic context.

4. Static approaches to assign and schedule requests

In this section, it is assumed that when the schedule is created, all the requests to be performed are known in advance, and the objective is to minimize the weighted lateness of all the requests. An exact and two approximated approaches are proposed to assign and schedule the requests. The exact approach consists of a mixed integer linear programming model. Similar formulations have already been proposed in the literature (e.g., Radhakrishnan & Ventura 2000). The two approximated approaches are a constructive heuristic and a local search heuristic. The need for approximated approaches is justified by the combinatorial nature of the scheduling problem, which is NP-Hard. Indeed, according to Pinedo (2008), the single machine scheduling problem with sequence-dependent setup times (SMSP-SDST) can be solved in polynomial time for cases where the setup times have a particular structure. However, if the setup times are arbitrary, as in the problem we consider, the SMSP-SDST remains strongly *NP-Hard*. Nonetheless, the proposed formulation will be used to produce bounds that allow the quality of the approximated solutions to be estimated in some cases.

4.1. A mathematical formulation for the PMSP-SDST

We model transportation requests as tasks to perform by the porters (the servers), and the sequence-dependent setup times correspond to the time or the distance required to move from the location at the end of a task to the beginning of the next one. This modeling approach leads to a very compact and easy-to-understand formulation. Furthermore, although a scheduling problem can be reformulated as a vehicle routing problem, doing so requires several sets of constraints, such as the ones concerning depots from which routes start and finish, pairing constraints to ensure that the same porter visits the origin and destination locations of each request, and even precedence constraints forcing to visit the request' origin before its destination. All these restrictions are implicit, or their formulation is straightforward in the PMSP-SDST.

Let R be the set of requests to be performed and P the set of available porters. Let $R' = R \cup \{0\}$ be an enlarged set of tasks that contains 'dummy' requests allowing each porter to initiate their sequence of tasks. Recall that preemption is not allowed; once the porter has started a task, they cannot be interrupted or redirected to another request. Let t_{ij} be the porter's traveling time from the destination location d_i of request i to the origin o_j of request j, assuming that request j is performed right after request i. Moreover, let t_i, t_i^a, and t_i^d be the time to transport patient of request i from their origin location to their destination (i.e., from o_i to d_i), the time at which the request i was inserted in the system, and the maximum time at which request i can be finished before it is considered a delay, respectively. Dummy requests start at the hospital transportation office and require null traveling time. As the porters can stay at the destination of the last performed request, the final location of the dummy request does not need to be restricted to the hospital transportation office. Finally, we associate each request i with a penalty α_i that increases with the request's priority or urgency.

Several sets of variables are used to formulate the model. Continuous variables C_{ip} compute the time at which request i is completed by porter p; variables L_i compute the lateness (if any) incurred in serving request i; finally, decision variables x_{ijp} have a value of 1 if porter p executes request j immediately after request i, and a value of 0 otherwise.

The porter scheduling problem can be formulated as follows:

$$\textbf{Min} \sum_{i \in R} \alpha_i L_i \tag{1}$$

s.a.

$$\sum_{p \in P} \sum_{\substack{i \in R' \\ i \neq j}} x_{ijp} = 1, \qquad \forall j \in R; \tag{2}$$

$$\sum_{j \in R} x_{0jp} \leq 1, \qquad \forall p \in P; \tag{3}$$

$$\sum_{\substack{h \in R' \\ h \neq i,j}} x_{hip} \geq x_{ijp}, \quad \forall i \in R'; \forall j \in R; i \neq j; \forall p \in P; \tag{4}$$

$$\sum_{p \in P} \sum_{\substack{j \in R \\ i \neq j}} x_{ijp} \leq 1, \qquad \forall i \in R; \tag{5}$$

$$C_{jp} + M(1 - x_{ijp}) \geq C_{ip} + t_{ij} + t_j, \quad \forall i \in R'; \forall j \in R; i \neq j; \forall p \in P; \tag{6}$$

$$C_{0p} = 0, \qquad \forall p \in P; \tag{7}$$

$$C_{jp} \geq \sum_{\substack{i \in R' \\ i \neq j}} x_{ijp} * (t_j^a + t_{ij} + t_j), \quad \forall i \in R; \forall p \in P; \tag{8}$$

$$L_i \geq C_{ip} - t_i^d, \qquad \forall i \in R; \forall p \in P; \tag{9}$$

$$x_{ijp} \in \{0, 1\}, \quad \forall i \in R'; \forall j \in R; i \neq j; \forall p \in P; \tag{10}$$

$$C_{ip} \geq 0, \qquad \forall i \in R'; \forall p \in P; \tag{11}$$

$$L_i \geq 0, \qquad \forall i \in R; \tag{12}$$

The objective function (1) aims to minimize the weighted sum of the lateness over all requests. Coefficient α_i, which takes higher values for more urgent requests, forces the formulation to comply first with the due date of urgent requests. Constraints (2) ensure that each request is performed by one and only one of the porters, and it has a unique request as an immediate predecessor. Constraints (3) say that each porter p starts their working sequence with dummy request 0, followed by one request at most. Constraints (4) guarantees that the same porter p executes the previous and successive requests. Constraints (5) ensure that all the requests made by the same porter p are correctly ordered (i.e., each request has a unique immediate successor). Constraints (6) ensure, for each request j performed by a given porter p, its completion time must be at least the completion time of its precedent request, i, plus the traveling time from the destination of request i to the origin of the request, j, plus the execution time of j. The completion time for all dummy requests is set as 0 by constraints (7). Constraints (8) prevent requests from being started before the associated patient's pickup time t_i^a. Constraints (9) compute the lateness for each request. Constraints (10)-(12) define the variables' domains.

4.2. Constructive heuristic

Our observations at a partner hospital inspire this constructive heuristic (CH). It sorts a set of requests to be performed, first in decreasing order of their priority, and within the same priority, according to their arrival times. Then, the heuristic assigns the first request to the porter, who will become available first. The request is deleted from the set and inserted in the last position of the selected porter's schedule. The heuristic loops until the set of requests to be performed are empty.

4.3. Local search heuristic

The local search heuristic (LS) starts from an initial feasible solution and explores three neighborhoods to minimize weighted lateness. The procedure to generate the initial solution is basically the same as in CH; it sorts the requests similarly. However, the selection of the porter is different; instead of selecting the first available porter, it selects based on the best value for the objective function.

The neighborhoods $N1$ to $N3$ explored by LS are formed by combining the *shift* and *swap* moves described by Moser, Musliu, Schaerf, and Winter (2021). The three neighborhoods are used sequentially according to their level of computational complexity: Shift, External Swap + Internal Shift, and External Swap + Internal Swap. Starting with $N1$, every possible move in the neighborhood is explored, the move producing the best improvement over the best solution found so far is implemented, and the best solution is updated. If no improvement is possible, the heuristic moves to the following neighborhood. The heuristic stops when the exploration of the last neighborhood does not yield any improvement compared with the best solution found so far. At any time, if a solution better than the best found so far is reached, the heuristic goes back to the first neighborhood in the sequence. Finally, note that moves that schedule a request i earlier than the patient's pickup time t_i^a are allowed, but the transport cannot start before this time. The neighborhoods are described as follows.

- **N1—Shift:** In this neighborhood, the heuristic explores all possible solutions produced when a request i is removed from its current sequence p and inserted into any possible position in all other sequences different from p. All requests are considered for evaluation.
- **N2—External Swap + Internal Shift:** This neighborhood considers every possible swap between two requests, the request i assigned to the porter p, and request j assigned to any porter q such that $p \neq q$ (External Swap). Then, it explores every possible shift of request i in the sequence q and those of request j in the sequence p and selects the ones minimizing the objective value (Internal Shift).
- **N3—External Swap + Internal Swap:** This neighborhood, as the previous one, considers every possible swap between two requests, the request i, assigned porter p, and the request j assigned to any porter q such that $p \neq q$ (External Swap). For each of those exchanges, it explores every swap (exchanges) of request i with all the requests in the sequence q and those of request j with all the requests in the sequence p and selects the ones minimizing the objective value (Internal Swap).

5. A dynamic approach to rescheduling requests

As we explained in the previous sections, the management of transportation requests is a dynamic and uncertain problem. In manufacturing systems, *rescheduling* is defined as the process of updating a production sequence to incorporate an interruption in production or the arrival of new information or tasks (Vieira, Herrmann, & Lin 2003).

The instants at which rescheduling is launched can be fixed a priori (at a given frequency) or be dependent on the system state. A natural manner of handling rescheduling in our context is to trigger the rescheduling of the existing execution sequences every time a new request arrives. Hence, upon its arrival, the new request is added to the set of requests waiting to be executed, and, considering the current state of the porters (location if idle, and time and location of their current missions if busy), the set of waiting requests is rescheduled. Intuitively, such a strategy might allow the manager to profit from opportunities that arise due to the new event. On the other hand, reconsidering the actual schedule may not be optimal (da Silva, Scarpin, Pécora, & Ruiz 2019) and be very expensive in terms of computational effort. To find the best possible balance between the rescheduling frequency, quality of the produced solutions, and computational effort, we propose the following rescheduling triggering policies:

- **Policy 1 ($\Phi 1$):** Rescheduling is triggered whenever a new request arrives.
- **Policy 2 ($\Phi 2$):** Arriving requests are queued, and the rescheduling is triggered when either β requests are waiting or upon arrival of an urgent request (see Section 6.1). This policy aims to guarantee that a schedule, once produced, will be at least partly implemented before it might be reconsidered. Delaying the rescheduling trigger also reduces the number of reschedules compared to $\Phi 1$. Rescheduling is also triggered on the arrival of an urgent request to avoid delays for such requests. After several preliminary tests, we set $\beta = |P|$ in the numerical experiments.
- **Policy 3 ($\Phi 3$):** This policy is similar to $\Phi 2$, but rescheduling is triggered periodically every κ time unit rather than at variable times. Rescheduling is also triggered upon the arrival of an urgent request. In our numerical experiments, we set $\kappa = 5$ minutes.
- **Policy 4 ($\Phi 4$):** A reschedule is triggered every time a request is completed. However, since the number of pending requests can be large, only the $q \leq |P|$ oldest requests are considered to be rescheduled.

Each of the described rescheduling triggering policies offers a different compromise between the frequency of rescheduling, which impacts the computational effort, and the performance of the solutions' quality.

6. Numerical Experiments

This section aims to assess the relative performance of the methods proposed over a set of random instances generated using data provided by a real hospital and to discuss their potential as effective tools for real-life applications. It first presents the instances and then the methods that will be evaluated. Finally, it reports the numerical results and discusses the proposed methods' benefits in service metrics.

All the tests were performed on a computer with an Intel Xeon E5-2683 v4 2.1GHz processor using one core and a maximum of 8Gb of RAM running on a CentOS Linux release 7 operational system. CPLEX 12.8 was used to solve the proposed mathe-

matical formulation, and all other solving procedures were programmed in C++ and compiled using gcc 9.1.

6.1. Generation of test instances

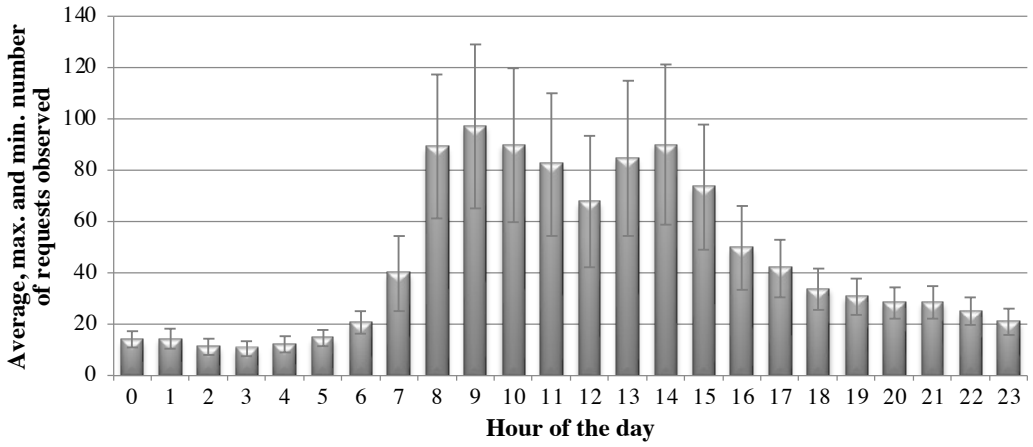

Figure 2. Historical distribution of requests' arrivals during the day (over 20 week-days).

We used the data provided by a hospital in the province of Quebec, Canada, with 800 beds, to generate random yet realistic instances. Historical data containing past requests were used to build a discrete distribution of the requests' arrival frequency at different moments of the day. Figure 2, which reports the average, largest, and the smallest number of requests received at each hour of the day over 20 week-days, illustrates the variability of the demand arrivals.

We limited our experiments to the day work-shift (8 a.m. to 4 p.m.). The 8-hours shift was divided into 96 intervals of 5 minutes, and the average number of requests received during each interval was computed. This empirical distribution was sampled to generate the interarrival time for each request i and then determine the actual arrival time t_i^a. The due date of each request was set to $t_i^d = t_i^a + \nu + \gamma_i$, the sum of the request's arrival time t_i^a, a fixed value ν corresponding to the longest distance between any two locations in the hospital in seconds, and a value γ_i which is related to the priority of request i. Four priority levels were considered: λ_1 to λ_4, with λ_4 corresponding to the highest priority. Parameter γ_i was set to 1800, 1000, 600, and 60 seconds for priorities 1 to 4, respectively, indicating that the higher priority requests are expected to be completed sooner. Finally, the penalty α_i in the objective function (1) was set, after running some preliminary tests, to 1, 10, 18, and 30 for priorities λ_1, λ_2, λ_3, and, λ_4, respectively.

Three sets of 12 instances were generated, with each set corresponding to a given profile H_1, H_2, and H_3. The instance profiles represent the prevalence of the priority levels in the generated requests. Profile H_1 contains a few requests of priority λ_4 (approximately 10%), with the remainder having any other level of service with a uniform probability of 30%. In the second profile H_2, the number of requests of each priority is homogeneous, so on average, there are 25% of each type. Finally, the third profile H_3, contains, on average, 34% of λ_4 requests, while the remaining requests have, on average, a probability of 22% for each of the other priorities.

The origin and destination of each request were randomly selected from the set of 28 locations provided by the hospital according to their average frequency of occurrence. The hospital's transportation office also provided travel times between any two locations. Traveling times vary between 1.5 and 8.5 minutes, with an average of 4.66 minutes. The experiments considered 16 porters.

6.2. Dispatching methods

Every time a rescheduling is launched, we can use any of the three scheduling approaches presented in Section 4 (a mathematical program referred to as *MP*, a construction heuristic *CH*, and a local search heuristic *LS*) to assign and sequence requests to porters. By combining a rescheduling policy with a scheduling approach, it is possible to form a variety of methods to handle dynamic patient transport requests. To improve the readability of the text, the resulting methods will be referred to as $XXYY$, where XX denotes the rescheduling policy ($\Phi 1$, $\Phi 2$, $\Phi 3$, or $\Phi 4$) and YY the scheduling approach (*MP*, *CH*, or *LS*).

6.3. Numerical results

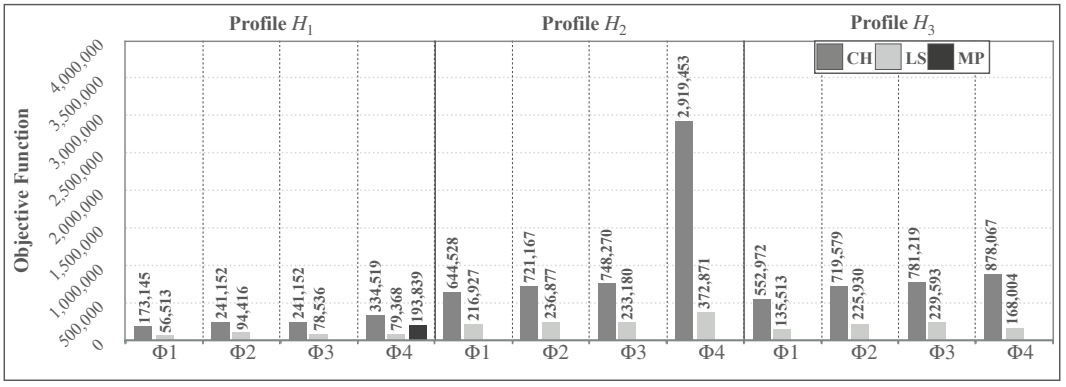

Figure 3. Average Objective Function, produced for each profile of instances and by each policy-solving method.

Figure 3 reports, for each of the 12 considered policy-solving method combinations, the average objective function value produced over the 12 instances of each profile H_1, H_2, and H_3. The best average value produced for each profile has been underlined for easier identification. It is worth mentioning that no computational times are reported because, in all the cases but for method $\Phi 4MP$, no instance required more than 2 seconds to be solved. Note that methods using *MP* as a scheduling approach require computational times that make them unsuitable for any real-time application. Indeed, to give the reader an idea of the computational requirements of methods using *MP*, when we limited the total execution time to 7 days and the maximum computational time to solve a schedule to 1 hour, only $\Phi 4MP$ was able to give solution to some instances (12 out of 36).

Figure 3 demonstrates the low performance of the constructive heuristic *CH* compared with *LS* and that for all the policies. The results reported in Figure 3 also confirm that the policy $\Phi 1$ clearly dominates the others and, when combined with the

solving approach LS, constitutes the most promising method to solve the problem in hand.

Finally, note that the results produced by $\Phi 4MP$, when available, compare poorly to the ones produced by $\Phi 1LS$. Two factors can explain that poor comparison; first, recall that the rescheduling computation time was limited to 1 hour. If this time limit is reached during partial rescheduling, the best solution found so far—which might be far from the optimal one—is kept. The other factor is the rescheduling triggered strategy. The combination of MP with rescheduling policy $\Phi 4$ does not seem very effective.

The previous numerical experiments focused on the value of the objective function, which corresponds to a sum of weighted lateness. In practice, managers are concerned with performance indicators related to service quality or efficiency that are aligned, yet different, from the objective, pursued by the proposed methods. For that reason, we evaluated the performance of the selected methods concerning other service-oriented performance metrics: the average response time (RT) in minutes, which for a given request i is defined as $RT_i = C_{ip} - t_i^a$, the percentage of late requests ($\%L$), and the average lateness of late requests (AvL) in minutes. By doing so, it should be possible to assess how well the proposed methods suit the managers' goals.

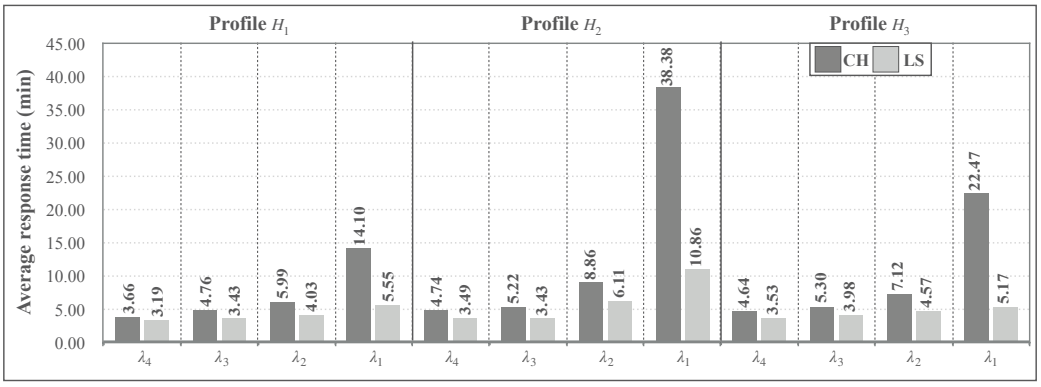

Figure 4. Average response time, in minutes, produced for each profile of instances by methods $\Phi 1CH$ and $\Phi 1LS$.

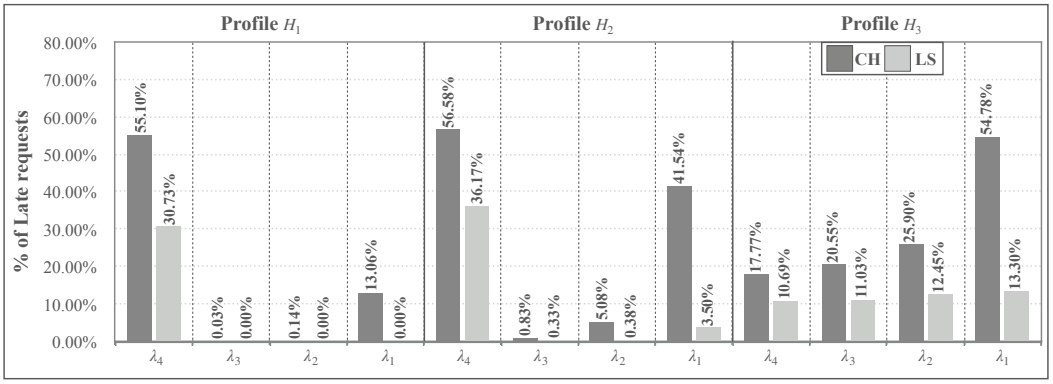

Figure 5. Average % of late requests, produced for each profile of instances by methods $\Phi 1CH$ and $\Phi 1LS$.

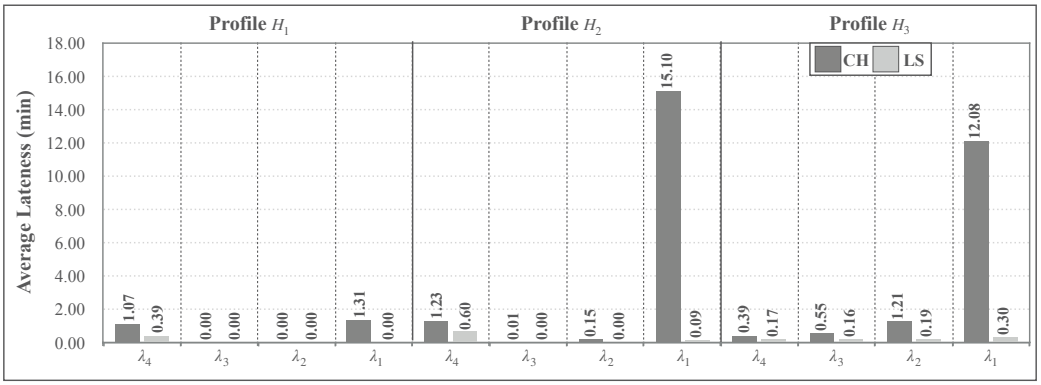

Figure 6. Average Lateness, in minutes, produced for each profile of instances by methods $\Phi 1CH$ and $\Phi 1LS$.

Figures 4, 5 and 6 report, for each type of profile of instances H_1, H_2, and H_3, the average results produced by policy $\Phi 1$ combined to methods CH and LS. Since the priority of requests is a key aspect of service, the results for the three performance metrics are reported separately for λ_1, λ_2, λ_3, and λ_4.

Let us begin the analysis by looking at the results produced for the average response time (Figure 4). Method $\Phi 1LS$ drastically reduces the average response times with respect to $\Phi 1CH$. Indeed, for the most urgent requests of type λ_4, the reduction in response time increases from 23.9% to 31.5% depending on the profile of the considered instances. Similar improvements are achieved for the requests belonging to other priorities. Moreover, although the response time produced by $\Phi 1LS$ tends to increase when the priority decreases, the increase is moderate, and the time consistently lower than those produced by $\Phi 1CH$. Regarding the percentage of late requests $\%L$, $\Phi 1LS$ produced the smallest percentage of late requests for all priorities and profiles (Figure 5). Finally, Figure 6 also shows that the resulting average delays whenever a request is late are reasonable and that this applies to both methods. In particular, $\Phi 1LS$ produces average delays of under one minute in all the cases.

To summarize, the numerical results confirmed the excellent performance of $\Phi 1LS$. Indeed, the noticeable improvements concerning the results produced by $\Phi 1CH$, the method that mirrors the one used at the hospital, suggest that the implementation of $\Phi 1LS$ might result in significant savings for the transportation office.

6.4. Managerial insights

The differences between the results produced for instances of profiles H_1, H_2, and H_3 raise certain managerial questions. Indeed, the response times produced by each method deteriorate as the proportion of urgent requests increases (i.e., from H_1 to H_2 and from H_2 to H_3). Although this deterioration is handled differently by the considered methods, the results produced for profile H_1 are preferred from a managerial standpoint. However, given that in the current system, most of the requests need immediate transport, it is possible that professionals and nurses overstate the priority of some requests to be certain that they will be performed in a timely manner. By doing so, the performance for all requests, including the highest priority requests, would decrease, which might provoke professionals' and nurses' dissatisfaction and, most likely, their tendency to increase the priority of the requests they place. Therefore,

it is essential to make stakeholders aware of both the benefits of placing requests in advance—which should relieve the urgency of requests— that might bring to the system and, even more importantly, the deterioration in system performance as a result of the increasing number of higher priority requests placed.

6.5. Sensitivity of the results concerning the number of porters

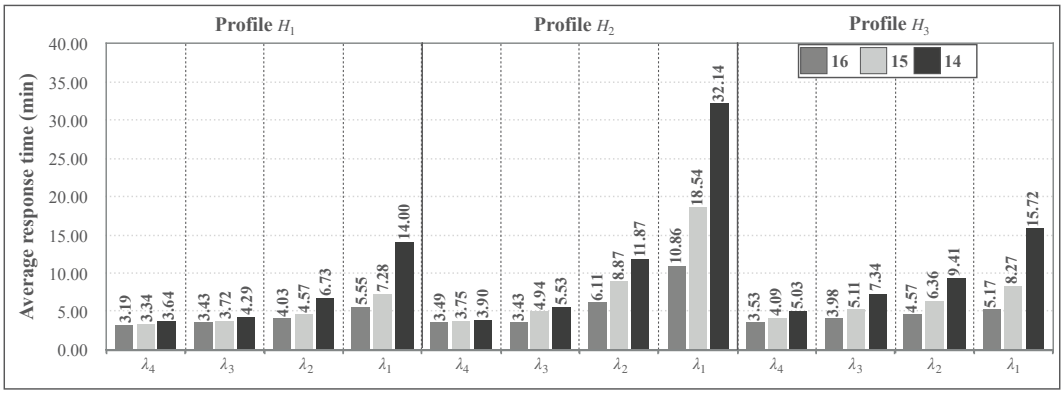

Figure 7. Average response time, in minutes, produced by $\Phi 1LS$ for 16, 15, and 14 porters.

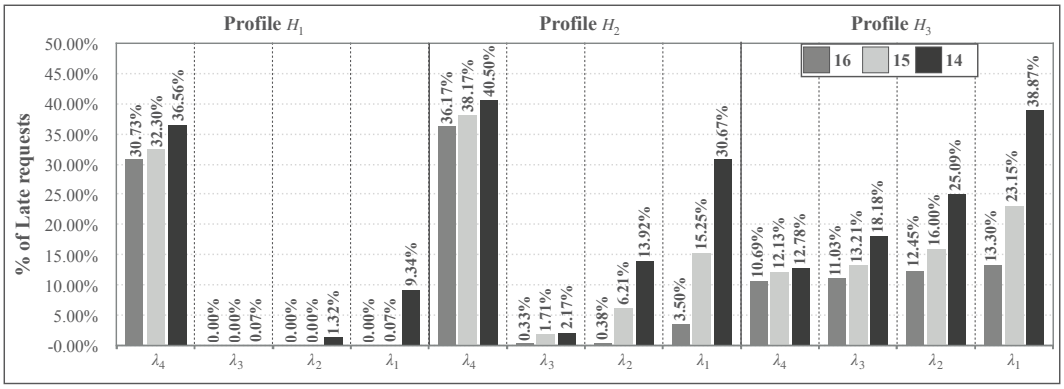

Figure 8. Average % of late requests, produced by $\Phi 1LS$ for 16, 15, and 14 porters.

To better characterize the improvements achieved by method $\Phi 1LS$ concerning $\Phi 1CH$, we computed, for both methods, the porters' daily empty moves that correspond to the setup time or, in other words, the daily time spent traveling from the end of requests to the origin of the following ones. We found that when $\Phi 1CH$ was used, each porter spent an average of 203 minutes per day on empty moves, while this time was reduced to only 138 minutes in the case of $\Phi 1LS$, an improvement of 65 minutes. This reduction in time positively affects the response time because it increases the availability of porters upon the arrival of requests. Moreover, considering that the 16 porters work 8-hour shifts in the instances, the efficiency achieved by $\Phi 1LS$ in terms of empty moves corresponds to the working time of two porters. In other words, implementing $\Phi, 1LS$ might help reduce the number of required porters. To explore the

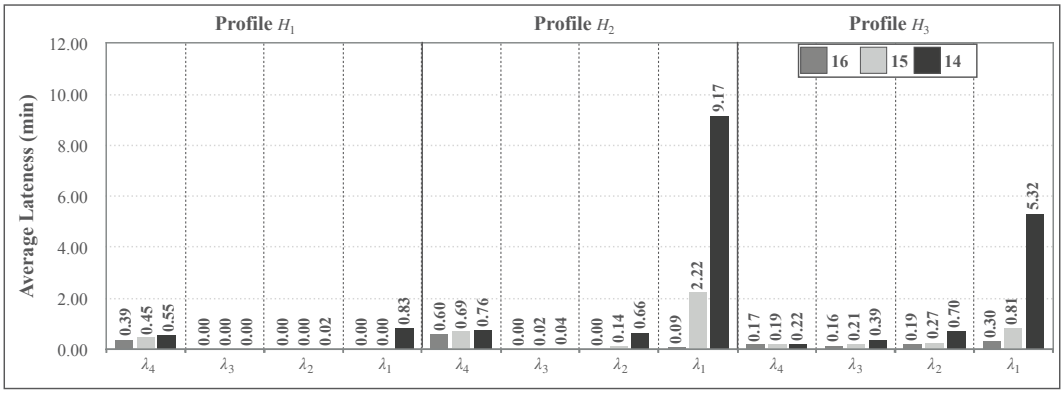

Figure 9. Average Lateness, in minutes, produced by $\Phi 1LS$ for 16, 15, and 14 porters.

extent to which a reduction in the number of porters would impact the system performance, we solved the previous instances again by $\Phi 1LS$ using $|P| = 15$ and $|P| = 14$ porters. The results are reported in Figures 7, 8 and 9 which detail the average results produced for performance metrics RT, $\%L$, and AvL for each type of request priority (λ_1 to λ_4) and profile of instances H_1, H_2, and H_3.

The numerical results confirm the expected performance deterioration as the number of porters is reduced for all the metrics. However, these deteriorations are not as large as one might expect. Indeed, the results produced by $\Phi 1LS$ with $|P| = 14$ porters are still better than the ones produced by $\Phi 1CH$ with 16 porters, representing an important potential reduction in the current resources used by the transportation office.

7. Conclusion

This paper describes the transport of patients within a healthcare facility and proposes efficient methods to manage patient transport requests in real-time. The problem of assignment and scheduling of transport requests to porters is modeled as a PMSP-SDST. Since the combinatorial nature of the resulting mathematical formulation makes it intractable even for very small instances, it proposes a constructive and a local search approach for addressing the static situation where the transport requests are known in advance.

However, in practice, new transport requests arrive in real-time, and the current schedule needs to be periodically revised in such a way that the requests not yet served and those that arrived after the last schedule was elaborated can be considered. This process is referred to as rescheduling. Several approaches to trigger the rescheduling process were proposed and combined with solving approaches to form specific methods that offer different performances.

Numerical experiments run on randomly generated instances inspired by a real hospital demonstrate that (i) the proposed heuristic methods are able to handle realistic situations in real-time, so they are suitable for real implementation, and (ii) the combination of a local search heuristic and a policy that triggers the rescheduling each time a new request arrives is particularly effective in solving this challenging problem. In particular, our experiments demonstrate that, compared to the management approach used in a real hospital, this heuristic drastically reduces the time that porters

travel empty (i.e., between two requests), translating into shorter response times and fewer late requests. In fact, our local search heuristic produces results that compare to the method currently used by the hospital but using only 14 porters rather than 16, representing an important potential reduction in the required resources.

This work also raises managerial and scientific questions for further research. From a managerial standpoint, our experiments demonstrate that the hospital's demand profile, in particular their priority, strongly impacts the potential performance of the transportation system. Hence, managers must understand the specific needs of their patients and set appropriate priorities for requests. Furthermore, in the current system, immediate transport is required for the majority of requests, putting a lot of pressure on the TO to serve them in a timely manner. It is of interest to ensure stakeholders are made aware of the benefits that placing requests in advance might bring to the TO and the patients. To this end, simulation tools such as the one used in this paper might be very helpful. From a scientific point of view, this work assumed the transport times to be deterministic and known in advance. It should be necessary, at least, to evaluate how the performance of the proposed methods is impacted by the variability of transport times, which will, most probably, require the development of specific models to tackle their uncertainty.

References

Beaudry, A., Laporte, G., Melo, T., & Nickel, S. (2010). Dynamic transportation of patients in hospitals. *OR Spectrum*, *32*(1), 77–107.

Bowers, J., Lyons, B., & Mould, G. (2012). Developing a resource allocation model for the Scottish patient transport service. *Operations Research for Health Care*, *1*(4), 84–94.

Cordeau, J.-F., & Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, *37*(6), 579–594.

Cordeau, J. F., & Laporte, G. (2007). The dial-a-ride problem: Models and algorithms. *Annals of Operations Research*, *153*(1), 29–46.

da Silva, N. C. O., Scarpin, C. T., Pécora, J. E., & Ruiz, A. (2019). Online single machine scheduling with setup times depending on the jobs sequence. *Computers & Industrial Engineering*, *129*, 251–258.

Dershin, H., & Schaik, M. S. (1993). Quality improvement for a hospital patient transportation system. *Hospital and Health Services Administration*, *38*(1), 111–119.

Doerner, K. F., & Salazar-González, J. J. (2014). Pickup-and-Delivery Problems for People Transportation. In *Vehicle routing: Problems, methods, and applications* (Second ed., pp. 193–212). Society for Industrial and Applied Mathematic.

Fiegl, C., & Pontow, C. (2009). Online scheduling of pick-up and delivery tasks in hospitals. *Journal of Biomedical Informatics*, *42*(4), 624–632.

Hanne, T., Melo, T., & Nickel, S. (2009). Bringing robustness to patient flow management through optimized patient transports in hospitals. *Interfaces*, *39*(3), 241–255.

Kergosien, Y., Lenté, C., Piton, D., & Billaut, J. C. (2011). A tabu search heuristic for the dynamic transportation of patients between care units. *European Journal of Operational Research*, *214*(2), 442–452.

Moser, M., Musliu, N., Schaerf, A., & Winter, F. (2021). Exact and metaheuristic approaches for unrelated parallel machine scheduling. *Journal of Scheduling*.

Naesens, K., & Gelders, L. (2009). Reorganising a service department: Central patient transportation. *Production Planning and Control*, *20*(6), 478–483.

Painchaud, M., Bélanger, V., & Ruiz, A. (2017). Discrete-event simulation of an intrahospital transportation service. *Springer Proceedings in Mathematics and Statistics*, *210*, 233–244.

Pinedo, M. L. (2008). *Scheduling: Theory, algorithms, and systems, fifth edition.* Springer

Cham.

Prughs, K., Sgall, J., & Torng, E. (2004). Online Scheduling. In J.-T. Leung (Ed.), *Handbook of scheduling* (pp. 329–373). London: Chapman and Hall.

Radhakrishnan, S., & Ventura, J. A. (2000, jul). Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent set-up times. *International Journal of Production Research*, *38*(10), 2233–2252.

Schilde, M., Doerner, K. F., & Hartl, R. F. (2011). Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers and Operations Research*, *38*(12), 1719–1730.

Schmid, V., & Doerner, K. F. (2014). Examination and operating room scheduling including optimization of intrahospital routing. *Transportation Science*, *48*(1), 59–77.

Segev, D., Levi, R., Dunn, P. F., & Sandberg, W. S. (2012). Modeling the impact of changing patient transportation systems on peri-operative process performance in a large hospital: Insights from a computer simulation study. *Health Care Management Science*, *15*(2), 155–169.

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.

von Elmbach, A. F., Boysen, N., Briskorn, D., & Mothes, S. (2015). Scheduling pick-up and delivery jobs in a hospital to level ergonomic stress. *IIE Transactions on Healthcare Systems Engineering*, *5*(1), 42–53.

von Elmbach, A. F., Scholl, A., & Walter, R. (2019). Minimizing the maximal ergonomic burden in intra-hospital patient transportation. *European Journal of Operational Research*, *276*(3), 840–854.