A dynamic scheduling approach to internal hospital logistics

Farzaneh Karami[a], Wim Vancroonenburg[a,b], Greet Vanden Berghe[a]
[a]KU Leuven, Department of Computer Science, CODeS & imec
[b]Research Foundation Flanders - FWO Vlaanderen
E-mail: {farzaneh.karami, wim.vancroonenburg, greet.vandenberghe}@cs.kuleuven.be

Keywords: Dynamic pickup and delivery problems, Internal hospital logistics, Heuristics

1 Introduction

This study is motivated by the problem of optimizing internal logistic flows in hospitals (Vancroonenburg et al., 2016). Internal hospital logistics (IHL) include assigning, routing and scheduling decisions. People who work in IHL are confronted with pickup and delivery problems with time windows (PDPTWs) on a daily basis. IHL continuously require patients, medicine or medical equipment to be transported between different locations in the hospital. These transportation requests must be handled while respecting their associated time constraints. From the modeling perspective, this may be interpreted as scheduling tasks to a fleet of porters while minimizing tasks' lateness, porters' overtime and travel time.
One important variant of the PDPTW is the *dynamic* PDPTW in which requests are unknown in advance and instead arrive as time progresses. In an IHL context, requests are inherently dynamic given that not all requests' locations and time windows are known at the beginning of the planning horizon. Gradual request arrivals demand revising the porters' schedule continuously.
Dynamic optimization has become a challenging research topic in the past two decades. The present paper proposes a *rolling-horizon-based* dynamic optimization approach which considers a buffering strategy for the dynamic PDPTW and seeks to iteratively group together sets of arrival requests and allocate them to available porters, thereby reducing operational expenses while maintaining high-quality service. Re-optimization is performed with respect to the input which consists of only those requests known at that particular moment in time and utilizes a two-step heuristic-based scheduling algorithm.

2 Problem formulation

To model IHL in a PDPTW context, its elements must first be introduced. Based on the work by Vancroonenburg et al. (2016), *locations* in the hospital and its layout are represented by an undirected graph $G(V, A)$. V is the set of vertices corresponding to all locations. A is the set of edges connecting locations and junctions inside hospital. d_{ij} is the time it takes a porter to travel from location $v_i \in V$ to $v_j \in V$.
A *request* represents a transport order and is denoted by $r = (i, j) \in R$, with R representing the set of all requests and $i, j \in T$ with T representing the set of all tasks. A request consists of a pickup task i at location $v_i \in V$ and a delivery task j at location $v_j \in V$ and is announced at time a_r, before which it is unknown to the system. In order to schedule and plan routes of tasks servicing, it is necessary to consider each task' time windows, which is a time slot in the planning horizon. A semi-soft time window $[st_i, dt_i)$ is specified for each task i signifying that the task may not be scheduled before the start of its corresponding time window, however it can be scheduled after the end of this time window. A task's service time is denoted by p_i. Finally, each request has a special type corresponding to its transportation facility, for example: 'bed', 'walking' or 'wheelchair'.
All *porters* are associated with an availability time window $[st_k, dt_k)$ during which they execute tasks. Porters

begin and conclude their workday at a certain depot $v_k \in V$. For this paper, during the dynamic planning, it is considered that the last-known location of each porter is the beginning of each route, this aspect increases the complexity of the model depending on the number of porters. Each porter has a certain set of skills and each request may require its associated porter to have that skill in order to perform in.
A *forbidden combination* is a set of requests $\{r, \ldots, r'\}$ which cannot be performed simultaneously by a single porter. The combination of 'wheelchair' and 'bed' is an example of a forbidden request combination.
If a request r has *precedence* over another request r', and both of them are assigned to the same porter, then the delivery of request r must be executed prior to the pickup of request r'. A very common example of this is that the transportation of a wheelchair to a patient's room must occur before the transportation of the patient themselves.
Some tasks may be combined together, for example when a nurse is walking with a patient, they may also carry certain medicine. Such task combinations may be permitted in some cases while prohibited in other cases.
The weighted objective function which must be minimized is the sum of three criteria: requests' tardiness, porters' overtime and travel time. For each task i, the difference between its completion time Ct_i and its time window end is called tardiness and is calculated as $l_i = Max(0, Ct_i - dt_i)$. A porter's overtime is defined as the completion time of their last request performed minus the end of their availability time window (or 0, if negative). For each porter, travel time corresponds to the total time it takes to travel between pickup/delivery locations in addition to their travel time to/from the depot.
In the context of IHL, routing corresponds to selecting appropriate paths for porters to travel along, while scheduling corresponds to assigning tasks to the porters. The total travel time largely depends on the hospitals size, whereas tardiness largely depends on the time windows of tasks. With the same objective weight, if travel times are larger than tardiness, then requests with urgent time windows may still be scheduled far in the future, thereby improving routing. By assigning proper weights in the objective function, it is possible to emphasize either the routing (minimizing travel time) or the scheduling (minimizing lateness). In practical applications of IHL, the lateness of tasks may be the primary concern.

2.1 MILP modelling

The IHL is formulated as a mixed integer linear programming (MILP) model for the PDPTW introduced by Savelsbergh and Sol (1995). First indices, sets, parameters, decision variables and the model formulation are defined.

Indices:

- i, j: Pickup/delivery task at a specific location.
- k: A specific porter.
- t: A request type.

Sets:

- R: The set of all requests.
- RT: The set of request types.
- T: The set of tasks.
- T^p, T^d: The set of pickup/delivery tasks and $T = T^p \cup T^d$.
- P: The set of porters.
- P_i: The set of porters with required skills for task $i \in T$.
- T_k: The set of tasks which may be assigned to porter k.
- $Prec$: The set of precedences between tasks.
- F: The set of forbidden request pairs (t, t'), where $t, t' \in RT$.

Parameters and constants:

- ω_i: The tardiness weight of pickup/delivery task i.
- M: A suitably large constant.
- C: Each porter capacity.
- N^0_{tk}: Number of items of type t carried by porter k.

Variables:

- $x_{ijk} \in \{0, 1\}$: Porter k advances to service j after performing service i.
- $x_{0ik} \in \{0, 1\}$: Porter k advances to service i from his depot location.

$x_{i0k} \in \{0,1\}$: Porter k advances to his/her depot location after servicing task i (end of route).
$y_{ijt} \in \mathbb{N}$: Number of items of type t travelling on edge (i,j).
$z_{ijt} \in \{0,1\}$: An item of type t travelling on edge (i,j).
$s_{ik} \geq 0$: The start time of servicing the task i (has no meaning if i is not assigned to k).
$l_i \geq 0$: Tardiness of task i.
$ot_k \geq 0$: Overtime of porter k.

$$\textbf{Minimize} \sum_{i \in T} \omega_i \cdot l_i + \sum_{k \in P} ot_k + \sum_{k \in P} \sum_{i,j \in V \cup \{0\}} d_{ij} \cdot x_{ijk} \tag{1}$$

Subject to

$$\sum_{i \in T_k} x_{0ik} = 1 \qquad \forall k \in P \tag{2}$$

$$\sum_{i \in T_k} x_{i0k} = 1 \qquad \forall k \in P \tag{3}$$

$$\sum_{j \in T_k} x_{ijk} = \sum_{j \in T_k} x_{jik} \qquad \forall k \in P, i \in T_k \tag{4}$$

$$\sum_{j \in T_k} \sum_{k \in P_i} x_{ijk} = 1 \qquad \forall i \in T \tag{5}$$

$$\sum_{j' \in T_k} x_{ij'k} \leq \sum_{j' \in T_k} x_{jj'k} \qquad j \neq j',\ (i,j) \in R, k \in P_i \cap P_j \tag{6}$$

$$st_i \cdot \sum_{j \in T_k} x_{ijk} \leq s_{ik} \qquad \forall i \in T, k \in P_i \tag{7}$$

$$s_{ik} + p_i - dt_i \leq l_i \qquad \forall i \in T, k \in P_i \tag{8}$$

$$s_{ik} + d_{i0} + p_i - dt_k \leq ot_k \qquad \forall k \in P, i \in T_k \tag{9}$$

$$s_{ik} + p_i + d_{ij} \leq s_{jk} + M \cdot (1 - x_{ijk}) \qquad \forall k \in P, i, j \in T_k \tag{10}$$

$$s_{ik} + p_i + d_{ij} \leq s_{jk} \qquad \forall (i,j) \in Prec, k \in P_i \cap P_j \tag{11}$$

$$\sum_{j \in T} y_{ijt} - \sum_{j \in T} y_{jit} = 1 \qquad \forall i \in T^p, t = type(i) \tag{12}$$

$$\sum_{j \in T} y_{ijt} - \sum_{j \in T} y_{jit} = -1 \qquad \forall i \in T^d, t = type(i) \tag{13}$$

$$N^0_{tk} \cdot x_{0ik} \leq y_{0it} \qquad \forall i \in T, k \in P_i, t \in RT^0_k \tag{14}$$

$$y_{ijt} \leq C \cdot z_{ijt} \qquad \forall i, j \in T, t \in RT \tag{15}$$

$$\sum_{t \in RT} y_{ijt} \leq C \cdot \sum_{k \in P_i \cap P_j} x_{ijk} \qquad \forall i, j \in T \tag{16}$$

$$z_{ijt} + z_{ijt'} \leq 1 \qquad \forall (t,t') \in F, i, j \in T \tag{17}$$

$$x_{ijk} \in \{0,1\} \qquad \forall k \in P; i, j \in T_k \tag{18}$$

$$u_i, l_i, ot_k \geq 0 \qquad \forall i \in T \tag{19}$$

Constraints (2), (3) and (4) are flow constraints. Constraints (5) guarantee the assignment of each task. Inequalities (6) guarantee that if a pickup task is assigned to porter k, then its corresponding delivery task is also assigned to the same porter. Constraints (7) enforce the commitment to the beginning of the time window. Constraints (8) and (9) derive tasks tardiness and porters overtime, respectively. Inequalities (10) relate the sequencing and scheduling of tasks, while inequalities (11) enforce the precedence constraint. Constraints (12) and (13) are flow constraints regarding 'request type flows' which ensure that each pickup of a request type increases its flow by one and each delivery reduces it by one. Constraints (14) define the initial flow. Constraints (15) are constraints linking y and z variables. Porter capacities are defined by (16), and inequalities (17) concern the forbidden combinations of tasks which cannot be simultaneously assigned to the same porter.

3 Previous research

In recent studies a few dynamic optimization methods have been developed within the health care application domain, with those generally limited to patient transports. Warren et al. (2004) develop guidelines for inter- and intra-hospital transportations. Beaudry et al. (2010) demonstrate an application of the dynamic dial-a-ride problem in terms of organizing intra-hospital patient transport in a German hospital. Hanne et al. (2009) focus on transportations between different buildings within a hospital site which contains more than 100 individual buildings. Their research ultimately resulted in the development of the Opti-TRANS system. Fiegl and Pontow (2009) presented the results of an algorithm which dynamically schedules pickup and delivery tasks within hospitals. They focus on minimizing the average weighted flow time. This, however, results in an objective function which possibly features several (potentially conflicting) objectives.
Most of the proposed solution approaches are based on an intelligent use of the time between event arrivals, however they ignore computation time which may prove troublesome when urgent requests arrive during computations. Thus in a large-scale IHL environment achieving high-quality solutions within that time is likely impossible. The proposed approach in this study helps remedy this deficiency.

4 Dynamic optimization approach

This study introduces a *rolling-horizon-based* optimization approach based on a buffering strategy. This method seeks to iteratively group together sets of dynamic arrival requests and allocate them to available porters, thereby reducing operational expenses while maintaining high quality levels of service. The positive effect of an inherent delay is that it helps compensate for request arrival information uncertainty. One negative effect, however, is that it may introduce tardiness regarding urgent requests. Consequently, the amount of delay must be set considering the urgency level of request arrivals. A visual representation of the proposed dynamic optimization approach and its components is illustrated in Figure 1.

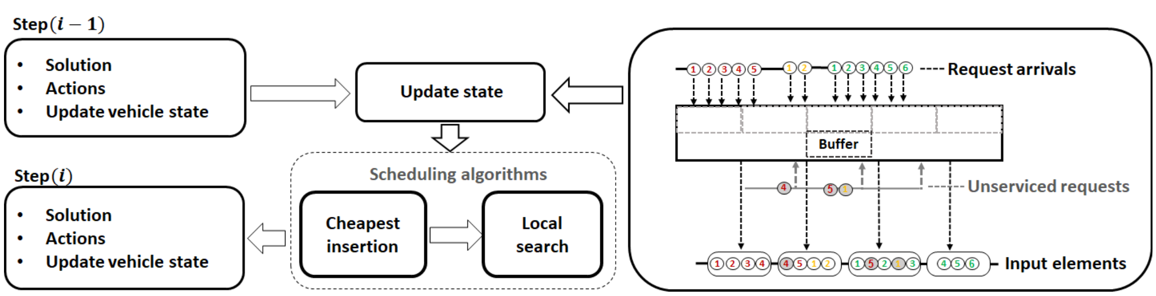

Figure 1: A visual representation of the buffering-strategy-based dynamic optimization approach

5 Results and future research

Real-word dynamic applications may benefit by applying more advanced scheduling algorithms. Due to the dynamic nature of these applications the scheduling algorithm must be implemented in a dynamic optimization framework. In this study, a MILP model was presented for scheduling IHL. Next, a dynamic optimization method was proposed which employs two-step heuristic scheduling algorithms. A computational study on different problem instances using the proposed approach will be presented at the workshop. Results on real-life and generated hospital instances which reveal how urgency levels and dynamism degrees of request arrivals impact cost will be presented. Solutions of the proposed approach will be evaluated both locally (at each re-optimization step) and globally (over the entire planning horizon) against the MILP solutions. In addition, the relative local and global gaps of the heuristic scheduling algorithm to the best found MILP

solution for which all information is known in advance are also evaluated.
In summary, using the maximum urgency of arrival requests as the re-optimization step length outperforms any other smaller step length. In addition, solution quality (relative gap) is only slightly affected by any change in instances' characteristics. Future work consists of the following steps: (1) generating instances for the dynamic PDPWT featuring different urgency levels and dynamism degrees as its two main characteristics to conduct a comprehensive study, (2) studying the proposed approach's sensitivity to the re-optimization frequency.

Acknowledgements

W. Vancroonenburg is a postdoctoral research fellow at Research Foundation Flanders - FWO Vlaanderen. Editorial consultation provided by Luke Connolly (KU Leuven).

References

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

Fiegl, C. and Pontow, C. (2009). Online scheduling of pick-up and delivery tasks in hospitals. *Journal of Biomedical Informatics*, 42(4):624–32.

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

Savelsbergh, M. W. and Sol, M. (1995). The general pickup and delivery problem. *Transportation science*, 29(1):17–29.

Vancroonenburg, W., Esprit, E., Smet, P., and Vanden Berghe, G. (2016). Optimizing internal logistic flows in hospitals by dynamic pick-up and delivery models. In *Proceedings of the 11th international conference on the practice and theory of automated timetabling, Udine, Italy, 23-26 August*, pages 371–383.

Warren, J., Fromm Jr, R. E., Orr, R. A., Rotello, L. C., Horst, H. M., of Critical Care Medicine, A. C., et al. (2004). Guidelines for the inter-and intra-hospital transport of critically ill patients. *Critical Care Medicine*, 32(1):256–262.