Combining Benders and Dantzig-Wolfe Decompositions for Online Stochastic Combinatorial Optimization

Antoine Legrain
Nadia Lahrichi
Louis-Martin Rousseau
Marino Widmer

October 2016

CIRRELT-2016-52

Combining Benders and Dantzig-Wolfe Decompositions for Online Stochastic Combinatorial Optimization

Antoine Legrain[1,*], Nadia Lahrichi[1], Louis-Martin Rousseau[1], Marino Widmer[2]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Mathematics and Industrial Engineering, Polytechnique Montréal, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7

[2] DIUF Decision Support & Operations Research, C325, Bd de Pérolles 90, 1700 Fribourg, University of Fribourg, Fribourg, Switzerland

Abstract. Online resource allocation problems are difficult because the operator must make irrevocable decisions rapidly and with limited (or nonexistent) information on future requests. We propose a mathematical-programming-based framework that takes into account all the available forecasts and the limited computational time. We combine Benders decomposition, which allows us to measure the expected future impact of each decision, and Dantzig-Wolfe decomposition, which can tackle a wide range of combinatorial problems. We illustrate the modeling process and demonstrate the efficiency of this framework on real data sets for two applications: appointment booking and scheduling in a radiotherapy center, and task assignment and routing in a warehouse.

Keywords: Stochastic optimization, online optimization, column generation, scheduling, routing

Acknowledgements. The authors sincerely thank the CICL and the head of its radio-oncology department, M.D. Marie-Andrée Fortin, JDA software and the director of operations, Catherine Gingras. They also gratefully acknowledge the supports of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Fonds de recherche du Québec - Nature et technologies (FRQNT), Mitacs, and the University of Fribourg.

* Corresponding author: antoine.legrain@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2016

1 Introduction

In resource allocation problems, the operator decides how to allocate requests to resources in order to maximize the profit or improve the service quality of an organization. These problems are challenging because the operator must quickly and continuously make irrevocable allocations without full knowledge of future requests. However, with the spread of information systems, there is now much historical data available to support online decisions. Many operators would benefit from decision support systems that use forecasts and optimization to guide allocation decisions.

Online resource allocation problems have been widely studied in the past 15 years and arise in domains such as:

- **Search-engine advertisements.** A search-engine operator allocates in an online fashion an advertisement to each new keyword search; it is displayed in the users' web navigator window. The goal is to maximize across all the keywords the expected advertisement revenue without exceeding the limited budget of each advertiser.
- **Revenue management.** Companies such as airlines that sell a limited quantity of goods match each new purchase to a selling price. They seek to maximize their expected revenue by choosing their selling prices dynamically without full knowledge of the demand.
- **Appointment booking.** Healthcare organizations such as clinics set up daily appointments with medical personnel and/or resources. The operator must find a compromise between efficiency and waiting-time targets that are based on patient priority.
- **Vehicle routing.** In a given time horizon, a fleet of vehicles must serve customers that are either known in advance or revealed dynamically. New customers must be added to the scheduled routes in real time.
- **Task assignment.** New tasks are generally assigned to workers according to their priority. The quality of the schedule must be continously reoptimized over a given horizon to ensure efficiency.

In an ideal (deterministic) world, the operator seeks a sequence of decisions leading to an optimal solution. However, in a realistic (stochastic) environment, not even feasibility can be ensured, because the operator makes irrevocable decisions without full knowledge of the future requests. The operator instead tries to minimize the expected deviation from the optimal solution based on the forecasts, which may be dynamically updated during the planning horizon.

The state-of-the-art methods for online resource allocation problems fail to evaluate the expected impact of allocation decisions. If the probability distribution of the requests is known at the beginning of the process and remains unchanged, approximate dynamic programming (Powell 2007) can be used to compute an offline policy. Online optimization (Buchbinder 2008) provides efficient algorithms with a proof of competitiveness, but it does not take advantage of available forecasts. Finally, online stochastic (OS) optimization (Van Hentenryck and Bent 2009) is a broad and practical paradigm, but it does not provide general mathematical tools to approximate the impact of allocation decisions.

In this paper, we extend the general framework of OS algorithms to provide such tools. We use Benders (Benders 1962) and Dantzig–Wolfe (Dantzig and Wolfe 1960) decompositions to solve OS resource allocation problems. The advantages of this new framework are threefold.

1. Benders subproblems are used to estimate the feasibility and optimality of the final solution. These estimations are handled through a given primal-ratio in the spirit of chance constraints, thus accommodating the operator's level of risk aversion.

2. Benders subproblems also allow us to infer the future load on the resources. A dual variable associated with each resource gives an approximation of the future cost of one resource unit.

3. Large-scale problems can be solved, because the combinatorial explosion and the complex constraints are managed through Dantzig–Wolfe decomposition.

We illustrate the use of this general framework on two specific applications that are representative of the complexity of online resource allocation problems. The first is an appointment booking and scheduling problem in a cancer treatment facility. Patients with varying priorities must be allocated consecutive treatment sessions on linear accelerators (linacs). The algorithm minimizes the patients' waiting time and schedules the necessary treatment-preparation steps. The second application concerns task assignment and routing decisions in a warehouse, where there is a queue of prioritized tasks that arrive continuously. When a worker finishes the current route, the operator must assign a new one. The algorithm must ensure the feasibility and efficiency of the assigned route while taking into account the priority ordering of the tasks.

The rest of the paper is organized as follows. Section 2 reviews existing techniques for online resource allocation problems. Section 3 formally defines the problem, and Section 4 details the algorithm. Section 5 discusses the appointment booking and scheduling application, while Section 6 presents the task assignment and routing application. Sections 5 and 6 both provide computational results for simulated and real data. Section 7 provides concluding remarks.

2 Literature review

Three main strategies have been proposed for online resource allocation problems: computing an offline policy, following a simple online policy, or reoptimizing the system for each request.

Markov decision processes (Puterman 2014) can be used to compute an offline policy. The problem is decomposed into two different sets (states and actions) and two functions (transition and reward). A state describes the value of the resources, and an action represents an available decision. The transition function indicates the probability of reaching one state from another with an action, and the reward function gives the reward for applying an action from a given state. An offline policy is then computed for each state. Approximate dynamic programming (Powell 2007) proposes ways to deal with the curse of dimensionality created by the exponential growth of the size of the state space. This technique has been successfully applied to financial optimization (Bäuerle and Rieder 2011), booking (Patrick et al. 2008), and routing (Novoa and Storer 2009) problems. However, the main advantage of this technique is also its main drawback: it concentrates all the computation at the beginning of the process, and then the operator has to follow the policy. This approach works only with a distribution known a priori; otherwise the transition function must be updated at each stage, thus negating the effort invested in the initial computation.

Online algorithms aim to solve dynamic problems rapidly without any knowledge of future requests. They ensure the quality of the final solution via a competitive ratio, which measures the gap between the optimal and worst-case solutions. Karp et al. (1990) solve the online matching problem. Mehta et al. (2007) introduce the Adwords problem and propose a $(1 - \frac{1}{e})$-competitive algorithm. Buchbinder (2008) proposes a primal-dual algorithm for a wide range of problems such as set covering, routing, and resource allocation problems. Feldman et al. (2009), Karande et al. (2011), Manshadi et al. (2012), and Jaillet and Lu (2014) introduce probabilistic knowledge. They compute offline strategies based on a prior distribution to help the online algorithm. Feldman et al. (2010) and Jaillet and Lu (2012) go further by reoptimizing their policy with the information available after one stage. However, these authors all consider the matching problem, which is a special polynomial case of the resource allocation problem. Legrain and Jaillet (2013) present a reoptimized primal-dual algorithm for the search-engine advertisement problem (i.e., the Adwords problem). Ciocan and Farias (2012) propose an OS algorithm for bipartite resource allocation problems. They compute an offline policy based on a prior distribution and reoptimize it at each time step. The policy estimates how to distribute each type of request among the different feasible allocations. There must be high volumes of each type of request in order for the algorithm to converge to the expected distribution. These authors do not provide a competitiveness proof for complex resource allocation problems, and they do not present generic techniques for using available forecasts of future requests.

OS algorithms reoptimize the problem for each new request using up-to-date forecasts. Classical techniques such as stochastic programming (Birge and Louveaux 2011) are not well suited to online optimization because they are too time-consuming; see Powell and Roy (2004). Van Hentenryck and Bent (2009) provide a general framework for OS problems and give three algorithms: expectation, consensus, and regret. These algorithms are all based on procedures that solve the offline version of the problem. The regret algorithm is the most advanced. In this algorithm, there are three steps to perform for each new request. First, build sample scenarios of future requests, then solve each scenario once with an offline procedure, and finally make a heuristic decision based on the solutions found. Our OS algorithm is based on this procedure. We propose a mathematical programming scheme for resource allocation problems to analyze the solution of each scenario and make the best decision for each request.

3 Problem formulation

We now give a formal description and formulation of the online resource allocation problem. A total of T requests arrive one at a time during a time horizon H. The operator must allocate the jth request so as to minimize the objective function. There is a set $\mathcal{R}$ of consumable resources, and resource r has b_r units available.

3.1 Offline formulation

We describe each allocation via a resource consumption pattern as in a Dantzig–Wolfe decomposition: each allocation pattern $i \in \mathcal{S}_j$ with cost c_{ij} for the jth request consumes an amount A_{ijr} of resource r. When the set $\mathcal{S}_j$ is large, the allocation patterns are generated during the solution process via column generation (Desaulniers et al. 2005). Complex and operational constraints (e.g., time constraints for the scheduling) are hidden in the patterns, giving a simple offline formulation for the resource allocation problem. Note that these constraints do not induce an integrality gap, thus strengthening the linear relaxation.

$$\min \sum_{j=1}^{T} \sum_{i \in \mathcal{S}_j} c_{ij} x_{ij} \tag{1}$$

subject to

$$\sum_{i \in \mathcal{S}_j} x_{ij} = 1, \qquad \forall j = 1 \dots T \tag{2}$$

$$\sum_{j=1}^{T} \sum_{i \in \mathcal{S}_j} A_{ijr} x_{ij} \leq b_r \qquad \forall r \in \mathcal{R} \tag{3}$$

$$x_{ij} \in \{0,1\} \qquad \forall j = 1 \dots T, \forall i \in \mathcal{S}_j \tag{4}$$

The variable x_{ij} is 1 if the jth request is matched to allocation pattern i, and 0 otherwise. The objective (1) minimizes the total cost of the allocations. Constraints (2) ensure that each request is matched to one allocation pattern. Constraints (3) manage the resource consumption. We now extend the offline formulation to the OS problem.

3.2 Online stochastic formulation

In a dynamic environment, the resource allocation problem becomes a multistage problem. One way to handle the allocation decision for the jth request is via a two-stage program with fixed recourse. Classical stochastic tools use a scenario-based optimization to solve this program. The number of requests T and the nature of each request are not known in advance and must be determined for each scenario. If the horizon H is known and sufficient historical data is available to build a probability distribution, which can be empirical, scenarios of future requests can be sampled through this probability distribution.

Let the sample set Ω_j be the set of possible scenarios of future requests. Each scenario ω has a probability p^ω and a total number of requests T^ω. The variable y_{il}^ω with cost c_{il}^ω is 1 if the lth request of scenario ω is matched to allocation pattern i, and 0 otherwise. The following stochastic formulation chooses the allocation of the jth request.

$$\min \sum_{i \in \mathcal{S}_j} c_{ij} x_{ij} + \sum_{\omega \in \Omega_j} p^\omega \sum_{l=1}^{T^\omega} \sum_{i \in \mathcal{S}_l^\omega} c_{il}^\omega y_{il}^\omega \tag{5}$$

subject to

$$\sum_{i \in \mathcal{S}_j} x_{ij} = 1 \tag{6}$$

$$\sum_{i \in \mathcal{S}_l} y_{il}^{\omega} = 1, \qquad \forall \omega \in \Omega_j, \forall l = 1 \ldots T^{\omega} \tag{7}$$

$$\sum_{i \in \mathcal{S}_j} A_{ijr} x_{ij} + \sum_{l=1}^{T^{\omega}} \sum_{i \in \mathcal{S}_l} A_{ilr} y_{il}^{\omega} \leq b_r, \qquad \forall \omega \in \Omega_j, \forall r \in \mathcal{R} \tag{8}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in \mathcal{S}_j \tag{9}$$

$$y_{il}^{\omega} \in \{0,1\}, \qquad \forall \omega \in \Omega_j, \forall l = 1 \ldots T^{\omega}, \forall i \in \mathcal{S}_l \tag{10}$$

The objective (5) minimizes the allocation cost of the jth request plus the expected cost of future allocations. Constraints (6) and (7) ensure that each (current or future) request is matched to one allocation pattern. Constraints (8) continue to manage the global resource consumption for each scenario. Constraints (9) and (10) define x_{ij} and y_{il}^{ω} as binary variables. This formulation leads to a huge model, which is difficult to solve in an online time-limited environment.

4 Methodology

In an online environment, an operator has generally at maximum few seconds to take the best possible decision: the computing speed of an online algorithm is thus a key-feature to get a realistic procedure. We propose a general, fast, and efficient L-shaped-based algorithm to rapidly solve the allocation problem for the jth request. Our procedure minimizes the total expected cost of future allocations in order to make the best decision for the current allocation. It first computes an expected descent direction using the value of the dual variables associated with the resource constraints (8). When this descent direction is imprecise for some resource constraints, we restrict the search space by generating probabilistic cuts for these constraints.

We first introduce the classical L-shaped procedure (Slyke and Wets 1969) for the online resource allocation problem. It applies Benders decomposition (Benders 1962) to transfer all the stochastic components (parts of the objective (5) and constraints (7), (8), and (10)) into an integer subproblem for each scenario ω.

$$Q(\overline{x}_{ij}, \omega) = \min \ \sum_{l=1}^{T^{\omega}} \sum_{i \in \mathcal{S}_l^{\omega}} c_{il}^{\omega} y_{il}^{\omega} \tag{11}$$

subject to

$$\sum_{i \in \mathcal{S}_l} y_{il}^{\omega} = 1, \qquad \forall l = 1 \ldots T^{\omega} \qquad (\alpha_l^{\omega}) \tag{12}$$

$$\sum_{l=1}^{T^{\omega}} \sum_{i \in \mathcal{S}_l} A_{ilr} y_{il}^{\omega} \leq b_r - \sum_{i \in \mathcal{S}_j} A_{ijr} \overline{x}_{ij}, \qquad \forall r \in \mathcal{R} \qquad (\beta_r^{\omega}) \tag{13}$$

$$y_{il}^{\omega} \in \{0,1\}, \qquad \forall l = 1 \ldots T^{\omega}, \forall i \in \mathcal{S}_l \tag{14}$$

The dual variables α_l^ω and β_r^ω (in parentheses) are associated with constraints (12) and (13). The integer subproblem calculates the recourse function $Q(\overline{x}_{ij}, \omega)$ for a solution $\overline{x}_{ij}$ and a scenario ω. We solve the linear relaxation of the subproblem because a two-stage recourse problem with integer subproblems is much more difficult and time-consuming (generally more than the few seconds allowed in an online environement). The solution of each relaxed subproblem gives the load on the resources: the dual variables β_r^ω give the expected future cost of one unit of resource r.

The relaxed subproblem feeds the following master problem for each scenario ω with an optimality cut (17), which approximates the recourse function $Q(x_{ij}, \omega)$:

$$\min \sum_{i \in \mathcal{S}_j} c_{ij} x_{ij} + \sum_{\omega \in \Omega_j} p^\omega \theta^\omega \tag{15}$$

subject to

$$\sum_{i \in \mathcal{S}_j} x_{ij} = 1 \tag{16}$$

$$\theta^\omega \geq \sum_{r \in \mathcal{R}} \beta_r^\omega (b_r - \sum_{i \in \mathcal{S}_j} A_{ijr} x_{ij}) + \sum_{l=1}^{T^\omega} \alpha_l^\omega, \qquad \forall \omega \in \Omega_j \tag{17}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in \mathcal{S}_j \tag{18}$$

The L-shaped procedure iteratively solves the master problem and the relaxed subproblems, which add new cuts to the master. The procedure stops when the relaxed subproblems have already been solved for the current solution of the master problem (i.e., the relaxed subproblems will not generate new optimality cuts).

In a dynamic environment, the L-shaped procedure remains too slow, as it needs several iterations before converging and slows down when combined with column generation. We therefore propose a much faster one-iteration L-shaped procedure.

4.1 One-iteration L-shaped procedure

If just one iteration is performed, the optimality cuts (17) can be transferred without the constant parts in the objective (19), which is now decomposed into two parts: the real cost c_{ij} of allocation pattern i and the average cost $\sum_{\omega \in \Omega_j} p^\omega (\sum_{r \in \mathcal{R}} \beta_r^\omega A_{ijr})$ implied by the resource utilization of this allocation. In this case, the master problem is equivalent to the following stochastic matching problem that minimizes the total expected cost of the allocation patterns:

$$\min \sum_{i \in \mathcal{S}_j} [c_{ij} - \sum_{\omega \in \Omega_j} p^\omega (\sum_{r \in \mathcal{R}} \beta_r^\omega A_{ijr})] \; x_{ij} \tag{19}$$

subject to

$$\sum_{i \in \mathcal{S}_j} x_{ij} = 1 \tag{20}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in \mathcal{S}_j \tag{21}$$

It is challenging to change an iterative method to a one-iteration procedure. The classical L-shaped procedure computes an initial solution of the master problem without considering any cuts and thus any stochastic information; it will iteratively refine this solution by adding new cuts. The one-iteration approach waits for optimality cuts to make a better decision based on insight into forthcoming requests. We consequently set an initial solution $\overline{x}_{ij} = 0$ that allows us to delay the allocation decision of the jth request, as in Van Hentenryck and Bent (2009).

Furthermore, the optimality cuts (17), which are generated by the relaxed subproblems for the solution $\overline{x}_{ij}$, are the best approximation of the recourse function $Q(x_{ij}, \omega)$ at the point $\overline{x}_{ij}$. However, they undervalue $Q(x_{ij}, \omega)$ for all the other points. Since they are computed only once, the solution $\overline{x}_{ij}$ is disadvantaged in the stochastic matching problem. Initializing $\overline{x}_{ij} = 0$ gives fair optimality cuts for all the nonzero solutions.

The relaxed subproblems must now deal with this initialization. This leads to two transformations: each scenario must include the jth request (index $l = 0$ in the scenario of future requests), and we add the constraint $\sum_{i \in \mathcal{S}_j} y^{\omega}_{ij} = 1 - \overline{x}_{ij} = 1$ (this constraint adds only a constant part to the objective (19)). In each scenario, the relaxed subproblem makes now the best allocation decision for the jth request.

To summarize, the one-iteration L-shaped procedure is a descent algorithm where the direction is equal to $c_{ij} - \sum_{\omega \in \Omega_j} p^{\omega} (\sum_{r \in \mathcal{R}} \beta^{\omega}_r A_{ijr})$. In a stochastic world, this direction indicates the region of the search space where we expect to find a better solution than the current one, $\overline{x}_{ij} = 0$. However, for some constraints we may not have enough dual information to guide the algorithm to an optimal solution. We use primal information on these constraints to remove decisions from the search space via feasibility cuts.

4.2 Probabilistic feasibility cuts

Usually, feasibility and optimality of the current decision $\overline{x}_{ij}$ are checked for each relaxed subproblem at each iteration of the classical L-shaped procedure. In our case, since each relaxed subproblem is solved once, feasibility and optimality are not fully checked. Instead, the algorithm retrieves primal information from the solution of the relaxed subproblems, as dual variables provide only a direction for each constraint. Indeed, different decisions may need to be taken by the algorithm on each resource constraint: a binary decision on whether the resources are consumed by a request or not, a positive quantity on how much of the resource should be consumed, and/or no information (dual variable equal to zero). While the first one is easy to deal with, the last two is challenging. The dual variable provides a direction but no information on the size of the step (i.e. consumption level) (see application 1 for illustration). The last case where there is no information about the direction arises when a resource includes precedence constraints on the resquests (see application 2). When the relaxed subproblems are solved with the decision $\overline{x}_{ij} = 0$, the precedence constraint is inactivated leading to a dual variable equal to zero. Additional information is needed by the algorithm in the later cases.

Let $\mathcal{D}$ be the set of resources for which the corresponding constraint does not provide enough dual information to the stochastic matching problem. When solving the relaxed subproblems, we store the optimal solution $\overline{y}^{\omega}_{il}$, which gives the optimal load on the resources. We assume that the optimal load of any resource in $\mathcal{D}$ remains close to optimal for any feasible decision $\overline{x}_{ij}$. We add the feasibility cuts $\sum_{i\in\mathcal{S}_j} A_{ijr}x_{ij} \le b_r - \sum_{l=1}^{T^\omega}\sum_{i\in\mathcal{S}_l} A_{ilr}\overline{y}_{il}^\omega$ to the stochastic matching problem for each scenario ω and for each resource r in $\mathcal{D}$. The jth request, which corresponds to the index $l=0$ in each scenario, is not taken into account in the optimal load because the variables x_{ij} determine the allocation of this request.

$$\min \sum_{i\in\mathcal{S}_j}[c_{ij} - \sum_{\omega\in\Omega_j} p^\omega(\sum_{r\in\mathcal{R}} \beta_r^\omega A_{ijr})]\ x_{ij} \tag{22}$$

subject to

$$\sum_{i\in\mathcal{S}_j} x_{ij} = 1 \tag{23}$$

$$\sum_{i\in\mathcal{S}_j} A_{ijr}x_{ij} \le b_r - \sum_{l=1}^{T^\omega}\sum_{i\in\mathcal{S}_l} A_{ilr}\overline{y}_{il}^\omega, \qquad \forall r\in\mathcal{D}, \forall\omega\in\Omega_j \tag{24}$$

$$x_{ij}\in\{0,1\}, \qquad \forall i\in\mathcal{S}_j \tag{25}$$

We transform the feasibility cuts (24) in the spirit of chance constraints to allow flexibility: since these cuts are approximations, they may lead to an allocation decision that is overly conservative. The goal is now to respect these cuts according to a certain confidence level η, which we call the primal-ratio. These cuts now become $\mathbb{P}_{\Omega_j}[\sum_{i\in\mathcal{S}_j} A_{ijr}x_{ij} \le b_r - \sum_{l=1}^{T^\omega}\sum_{i\in\mathcal{S}_l} A_{ilr}\overline{y}_{il}^\omega] \ge \eta$. Since the sample set Ω_j is finite and the relaxed subproblems are solved prior to the master problem, let Ω_j^r be a subset of scenarios for resource r such that $\sum_{\omega\in\Omega_j^r} p^\omega \ge \eta$. This subset is problem-related, easy to find, and computed in preprocessing of the master problem, as we will show when we discuss our applications.

$$\min \sum_{i\in\mathcal{S}_j}[c_{ij} - \sum_{\omega\in\Omega_j} p^\omega(\sum_{r\in\mathcal{R}} \beta_r^\omega A_{ijr})]x_{ij} \tag{26}$$

subject to

$$\sum_{i\in\mathcal{S}_j} x_{ij} = 1 \tag{27}$$

$$\sum_{i\in\mathcal{S}_j} A_{ijr}x_{ij} \le b_r - \sum_{l=1}^{T^\omega}\sum_{i\in\mathcal{S}_l} A_{ilr}\overline{y}_{il}^\omega, \qquad \forall r\in\mathcal{D}, \forall\omega\in\Omega_j^r \tag{28}$$

$$x_{ij}\in\{0,1\}, \qquad \forall i\in\mathcal{S}_j \tag{29}$$

This restricted stochastic matching problem takes into account the dynamic parts of the problem: the dual variables β^ω measure the expected cost of the resource utilization and, depending on the primal-ratio, constraints (28) forbid allocation decisions that might lead at the end of the horizon to infeasible or non-optimal final solutions. The primal-ratio should thus be as small as possible to avoid the removal of optimal decisions from the search space. To conclude, the relaxed subproblems feed the master problem with primal and dual information, then the master problem preprocesses this information, and can finally be solved as a restricted matching problem with column generation if needed (i.e. $\mathcal{S}_j$ cannot be explicitly enumerated).

4.3 Global online stochastic algorithm

Figure 1 presents the OS algorithm. This algorithm communicates with all the other parts of the system through an information system: it retrieves the state of the system before any decision and subsequently relays its decision to the rest of the system.

For each new request, the algorithm first finds feasible allocation patterns and then makes the best feasible decision. For some requests, the algorithm can save time by making decisions with an online policy instead of reoptimizing. Determining when this situation occurs is problem-dependent.

We simulate the other parts of the system in the two applications to evaluate our results in a realistic environment.

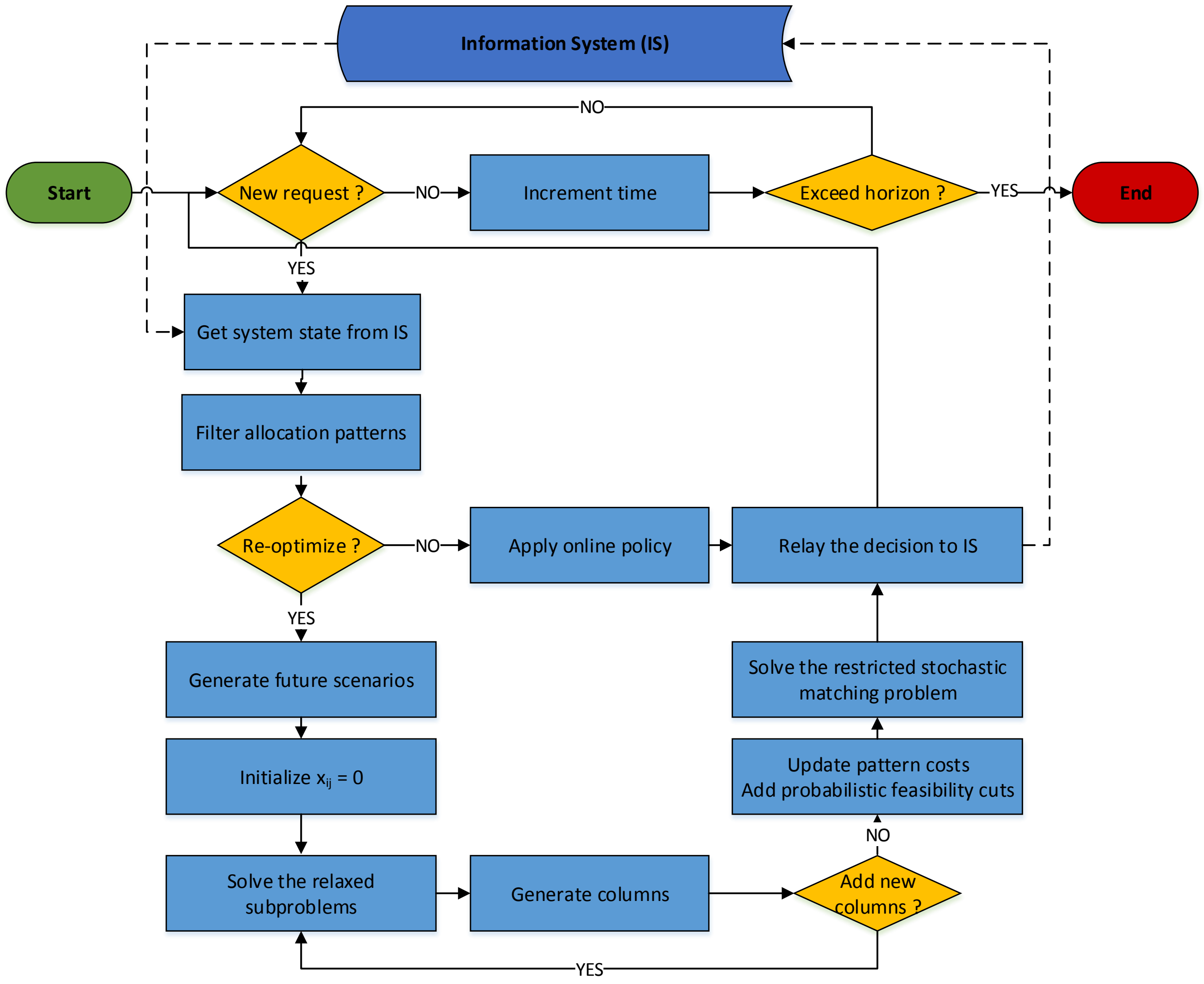

Figure 1: Flow chart of global online stochastic algorithm.

5 Application I: Appointment booking and scheduling problem

The online appointment booking problem (Gupta and Denton 2008) involves finding the best appointment for each new patient as he/she arrives. The main challenge is to maintain just enough free slots for high-priority patients that may arrive in the future. Online radiotherapy appointment booking is a relatively recent application. When patients arrive in a cancer treatment facility, they must undergo a series of examinations before receiving treatment on a linac, which irradiates the malignant tumor to kill the infected cells. After the first consultation, the patients undergo a scan to locate the tumor, and then the cancer treatment is prepared by the dosimetrists. The dosimetry primarily involves planning the shape, intensity, and direction of the beams of the linac. These steps form the pretreatment phase.

The total waiting time is measured as the number of days between the first consultation and the beginning of the treatment. The management must report detailed statistics to the state authorities on a regular basis. Furthermore, the center treats palliative (i.e., high priority) patients to relieve their pain and curative (i.e., low priority) patients to maximize their chances of recovery. The operator should balance resources between these two types of patients while respecting, as far as possible, the waiting-time targets.

Two classical block scheduling heuristics have been proposed for online appointment booking problems (Petrovic et al. 2006): "just in time" (JIT) and "as soon as possible" (ASAP). The first schedules patients on their due dates, and the second schedules them on their release dates. Petrovic et al. (2006) combine the two heuristics: JIT for the high-priority patients and ASAP for the others. Klassen and Rohleder (2004) propose general heuristics for outpatient clinics. They compare different rules such as "First-call, first-appointment" and "Low variance clients at the beginning of the schedule." They show that the latter rule is the best for several objectives, such as the client waiting time. Patrick et al. (2008) propose a general approximate dynamic program for outpatient clinics. Sauré et al. (2012) have successfully applied this technique to a radiotherapy center. Legrain et al. (2014) solve the booking problem with an online clairvoyant algorithm.

However, none of these methods can deal with the appointment booking and scheduling problem. This is because if the pretreatment planning is not completed on time, the treatment may be postponed, and this may cause the cancellation of the linac appointment. The management aims to reduce pretreatment processing times and to avoid unnecessary linac cancellations.

We apply the OS algorithm presented in Figure 1 to the booking and scheduling of a radiotherapy center in Quebec, Canada. In Quebec, patients with cancer can wait a considerable time for treatment. The Quebec authorities have defined a target maximum delay of 28 days. However, the Quebec College of Physicians advises more specific targets. Palliative patients should start treatment less than three days after admission, whereas curative patients can wait 14 or 28 days depending on the type of cancer. Two tasks must be performed by two different dosimetrists before treatment can begin: the first is the preparation of the treatment and the second is its verification. Other minor tasks must be performed to complete the pretreatment; they are modeled through a set of fixed delays. The scheduling of the pretreatment can thus be viewed as a hybrid flow-shop with recirculation (Ruiz and Vázquez-Rodríguez 2010).

This work has been realized in collaboration with the Centre Intégré de Cancérologie de Laval (CICL). We have published an extended abstract (Legrain et al. 2015) on this application.

5.1 Online stochastic formulation

We now present a stochastic optimization model for the appointment booking and scheduling problem. The planning of the dosimetry and the linac appointments are represented by columns. On the arrival of patient j, the model infers the average cost of linac plans for a finite set Ω_j of future patient ($\mathcal{P}^\omega$) scenarios ω of probability p^ω. Each future patient set also contains the current patient j. Let $\mathcal{H}$ be the index set of the working days over the planning horizon, $\mathcal{B}$ the index set of Mondays, and $\mathcal{M}$ the set of available linacs. Let $\mathcal{S}_j$ be the index set of feasible linac appointment patterns for patient j, a_{ijk}^m the description of pattern $i \in \mathcal{S}_j$ ($= 1$ if the patient is treated on linac m on day k, and 0 otherwise), b_{ij} the day of the first treatment session in pattern $i \in \mathcal{S}_j$, and c_{ij} the cost of pattern $i \in \mathcal{S}_j$. This cost is a nonlinear combination of waiting times and deadline-violation penalties. The parameter r_{il} represents the end of the pretreatment for patient l in dosimetry planning pattern $i \in \mathcal{S}^D$. Let F_k^m be the number of available slots on linac m on day k, O_{day} the maximum daily number of overtime slots on linac m, O_{week} the maximum weekly number of overtime slots on linac m, and c^o the cost of an overtime slot. The variable x_{ij} is 1 if linac appointment pattern $i \in \mathcal{S}_j$ is allocated to new patient j and 0 otherwise; y_{il}^ω is 1 if linac appointment pattern $i \in \mathcal{S}_l$ is chosen for patient l in scenario $\omega \in \Omega_j$ and 0 otherwise; and v_i^ω is 1 if dosimetry planning pattern $i \in \mathcal{S}^D$ is chosen for all patients of scenario $\omega \in \Omega_j$ and 0 otherwise. Finally, z_{mk} is the number of overtime slots on linac m on day k.

$$\min \sum_{i \in \mathcal{S}_j} c_{ij} x_{ij} + \sum_{\omega \in \Omega_j} p^\omega [\sum_{l \in \mathcal{P}^\omega} \sum_{i \in \mathcal{S}_l} c_{il} y_{il}^\omega + \sum_{k \in \mathcal{H}} \sum_{m \in \mathcal{M}} c^o z_{mk}^\omega] \tag{30}$$

subject to

$$\sum_{i \in \mathcal{S}_j} x_{ij} + \sum_{i \in \mathcal{S}_j} y_{ij}^{\omega} = 1 \qquad (31)$$

$$\sum_{i \in \mathcal{S}_l} y_{il}^{\omega} = 1, \qquad \forall \omega \in \Omega_j, \forall l \in \mathcal{P}^{\omega} \backslash \{j\} \qquad (32)$$

$$\sum_{i \in \mathcal{S}^D} v_i^{\omega} = 1, \qquad \forall \omega \in \Omega_j \qquad (33)$$

$$\sum_{i \in \mathcal{S}_j} b_{ij} x_{ij} + \sum_{i \in \mathcal{S}_l} b_{ij} y_{ij}^{\omega} - \sum_{i \in \mathcal{S}^D} r_{ij} v_i^{\omega} \geq 0, \qquad \forall \omega \in \Omega_j \qquad (34)$$

$$\sum_{i \in \mathcal{S}_l} b_{il} y_{il}^{\omega} - \sum_{i \in \mathcal{S}^D} r_{il} v_i^{\omega} \geq 0, \qquad \forall \omega \in \Omega_j, \forall l \in \mathcal{P}^{\omega} \backslash \{j\} \qquad (35)$$

$$\sum_{i \in \mathcal{S}_j} a_{ijk}^m x_{ij} + \sum_{l \in \mathcal{P}^{\omega}} \sum_{i \in \mathcal{S}_l} a_{ilk}^m y_{il}^{\omega} \leq F_k^m + z_{mk}^{\omega}, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H}, \forall \omega \in \Omega_j \qquad (36)$$

$$\mathbb{1}_{\mathcal{P}_p}(j) \sum_{i \in \mathcal{S}_j} a_{ijk}^m x_{ij} + \sum_{l \in \mathcal{P}_p^{\omega}} \sum_{i \in \mathcal{S}_l} a_{ilk}^m y_{il}^{\omega} \geq z_{mk}^{\omega}, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H}, \forall \omega \in \Omega_j \qquad (37)$$

$$\sum_{k=b}^{b+4} z_{mk}^{\omega} \leq O_{week}, \qquad \forall m \in \mathcal{M}, \forall b \in \mathcal{B}, \forall \omega \in \Omega_j \qquad (38)$$

$$z_{mk}^{\omega} \leq O_{day}, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H}, \forall \omega \in \Omega_j \qquad (39)$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in \mathcal{S}_j \qquad (40)$$

$$y_{il}^{\omega} \in \{0,1\}, \qquad \forall l \in \mathcal{P}^{\omega}, \forall i \in \mathcal{S}_l, \forall \omega \in \Omega_j \qquad (41)$$

$$v_i^{\omega} \in \{0,1\}, \qquad \forall i \in \mathcal{S}^D, \forall \omega \in \Omega_j \qquad (42)$$

$$z_{mk}^{\omega} \geq 0, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H}, \forall \omega \in \Omega_j \qquad (43)$$

Constraints (31) and (32) ensure respectively that patient j and all future patients are scheduled on linacs. Constraints (33) choose a dosimetry schedule for each scenario ω. The columns x_{ij}, y_{il}^{ω} representing patient appointments on linacs are all inserted once at the beginning because the sets $\mathcal{S}_j$ and $\mathcal{S}_l$ are small. However, the columns v_i^{ω} are generated during the solution process because the set $\mathcal{S}^D$ is large. A genetic algorithm, presented in Appendix 8.1, is used for the column generation procedure. Constraints (34) and (35) are precedence constraints: they ensure respectively that patient j and all future patients have completed their pretreatment in time for their first linac treatment. Constraints (36) verify that the capacity (including the overtime) of each linac is not exceeded. Constraints (37) ensure that only palliative patients are scheduled in overtime slots ($\mathcal{P}_p^\omega$ is a subset of $\mathcal{P}^\omega$ containing the palliative patients). Constraints (38) and (39) bound the weekly and daily overtime on each linac. Constraints (40), (41), (42), and (43) are domain constraints. Finally, the objective (30) is divided into two parts: the cost of the plan for patient j and the average future cost of the linac plans.

5.2 Integer subproblems

The OS formulation is transformed as shown in the methodology presented in Section 4. The integer subproblems solve the booking and scheduling problem for a solution $\overline{x}_{ij}$ and for each scenario ω.

$$Q(\overline{x}_{ij}, \omega) \;=\; \min \sum_{l \in \mathcal{P}^\omega} \sum_{i \in \mathcal{S}_l} c_{il} y_{il}^\omega + \sum_{k \in \mathcal{H}} \sum_{m \in \mathcal{M}} c^o z_{mk}^\omega \tag{44}$$

subject to

$$\sum_{i \in \mathcal{S}_j} y_{ij}^\omega = 1 - \sum_{i \in \mathcal{S}_j} \overline{x}_{ij} \tag{45}$$

$$\sum_{i \in \mathcal{S}_l} y_{il}^\omega = 1, \qquad \forall l \in \mathcal{P}^\omega \backslash \{j\} \tag{46}$$

$$\sum_{i \in \mathcal{S}^D} v_i^\omega = 1, \tag{47}$$

$$\sum_{i \in \mathcal{S}_l} b_{ij} y_{ij}^\omega - \sum_{i \in \mathcal{S}^D} r_{ij} v_i^\omega \geq -\sum_{i \in \mathcal{S}_j} b_{ij} \overline{x}_{ij}, \tag{48}$$

$$\sum_{i \in \mathcal{S}_l} b_{il} y_{il}^\omega - \sum_{i \in \mathcal{S}^D} r_{il} v_i^\omega \geq 0, \qquad \forall \omega \in \Omega_j, \forall l \in \mathcal{P}^\omega \backslash \{j\} \tag{49}$$

$$\sum_{l \in \mathcal{P}^\omega} \sum_{i \in \mathcal{S}_l} a_{ilk}^m y_{il}^\omega \leq F_k^m + z_{mk}^\omega - \sum_{i \in \mathcal{S}_j} a_{ijk}^m \overline{x}_{ij}, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H} \tag{50}$$

$$\sum_{l \in \mathcal{P}_p^\omega} \sum_{i \in \mathcal{S}_l} a_{ilk}^m y_{il}^\omega \geq z_{mk}^\omega - \mathbb{1}_{\mathcal{P}_p}(j) \sum_{i \in \mathcal{S}_j} a_{ijk}^m \overline{x}_{ij}, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H} \tag{51}$$

$$\sum_{k=b}^{b+4} z_{mk}^\omega \leq O_{week}, \qquad \forall m \in \mathcal{M}, \forall b \in \mathcal{B} \tag{52}$$

$$z_{mk}^\omega \leq O_{day}, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H} \tag{53}$$

$$y_{il}^\omega \in [0,1], \qquad \forall l \in \mathcal{P}^\omega, \forall i \in \mathcal{S}_l \tag{54}$$

$$v_i^\omega \in [0,1], \qquad \forall i \in \mathcal{S}^D \tag{55}$$

$$z_{mk}^\omega \geq 0, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H} \tag{56}$$

5.3 Probabilistic feasibility cuts

Constraint (48) can be infeasible if new patient j is scheduled too early on the linacs. Let $\mathcal{D}$ be the singleton formed by this constraint. For each scenario ω, if $\overline{v}^\omega$ is the solution of the relaxed subproblems, we add the feasibility cuts $\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} \geq \sum_{i\in\mathcal{S}^D} r_{ij}\overline{v}_i^{\omega}$. The sum $\sum_{i\in\mathcal{S}^D} r_{ij}\overline{v}_i^{\omega} = \overline{r}_j^{\omega}$ is now a constant and represents the earliest starting time for a linac appointment for patient j in scenario ω. Furthermore, if $\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} \geq \overline{r}_j^{\omega_0}$ holds for a scenario ω_0, $\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} \geq \overline{r}_j^{\omega}$ will also hold for any scenario ω such that $\overline{r}_j^{\omega} \leq \overline{r}_j^{\omega_0}$. Consequently, adding just one constraint suffices to represent all the constraints associated with each scenario.

For a primal-ratio η, we thus choose the scenario $\omega_0 = \arg\min_{\omega\in\Omega_j}\{\overline{r}_j^{\omega}\Big|\, |\{\omega_1 \in \Omega_j | \overline{r}_j^{\omega_1} \leq \overline{r}_j^{\omega}\}| \geq \eta\, |\Omega_j|\}$ (where $|\,.\,|$ indicates cardinality). If $\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} \geq \overline{r}_j^{\omega_0}$ holds, $\mathbb{P}_{\Omega_j}\{\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} \geq \overline{r}_j^{\omega}\} \geq \eta$. The subset Ω_j^0 is thus equal to the singleton $\{\omega_0\}$ for the only constraint in $\mathcal{D}$.

5.4 Restricted stochastic matching problem

Let δ^{ω}, β_{mk}^{ω}, and γ_{mk}^{ω} be the dual variables associated respectively with the constraints (48), (50), and (51). The stochastic dual cost associated with pattern i of patient j is defined to be $\sum_{\omega\in\Omega_j} p^{\omega}[\delta^{\omega}b_{ij} + \sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{H}}(\beta_{mk}^{\omega} + \mathbb{1}_{\mathcal{P}_p}(j)\gamma_{mk}^{\omega})a_{ijk}^{m}]$. Therefore, the restricted stochastic matching problem is

$$Z^* = \min \sum_{i\in\mathcal{S}_j}\{c_{ij} - \sum_{\omega\in\Omega_j} p^{\omega}[\delta^{\omega}b_{ij} + \sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{H}}(\beta_{mk}^{\omega} + \mathbb{1}_{\mathcal{P}_p}(j)\gamma_{mk}^{\omega})a_{ijk}^{m}]\}x_{ij}$$

subject to

$$\begin{aligned}
\sum_{i\in\mathcal{S}_j} x_{ij} &= 1 \\
\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} &\geq \overline{r}_j^{\omega}, \qquad \forall\omega\in\Omega_j^0 \\
x_{ij} &\in \{0,1\}, \qquad \forall i\in\mathcal{S}_j
\end{aligned}$$

The appointment patterns for the linacs in set $\mathcal{S}_j$ are all feasible and thus respect constraints (36)–(39) for patient j. The restricted stochastic matching problem determines which feasible pattern i, with a starting time greater than $\overline{r}_j^{\omega_0}$, has the minimum expected cost Z^*.

5.5 Global online stochastic algorithm

Figure 2 illustrates the steps of the OS algorithm for this application. The information system must communicate the states of the linacs and the dosimetrists. The simulator generates new requests based on a probability distribution and schedules daily pending dosimetry tasks using a constraint program, presented in Appendix 8.2. This program also checks the feasibility of an allocation pattern during the filtering.

The online algorithm uses an online policy to book palliative patients. Since they have a high priority, the ASAP heuristic gives good results and avoids the need to solve the relaxed subproblems.

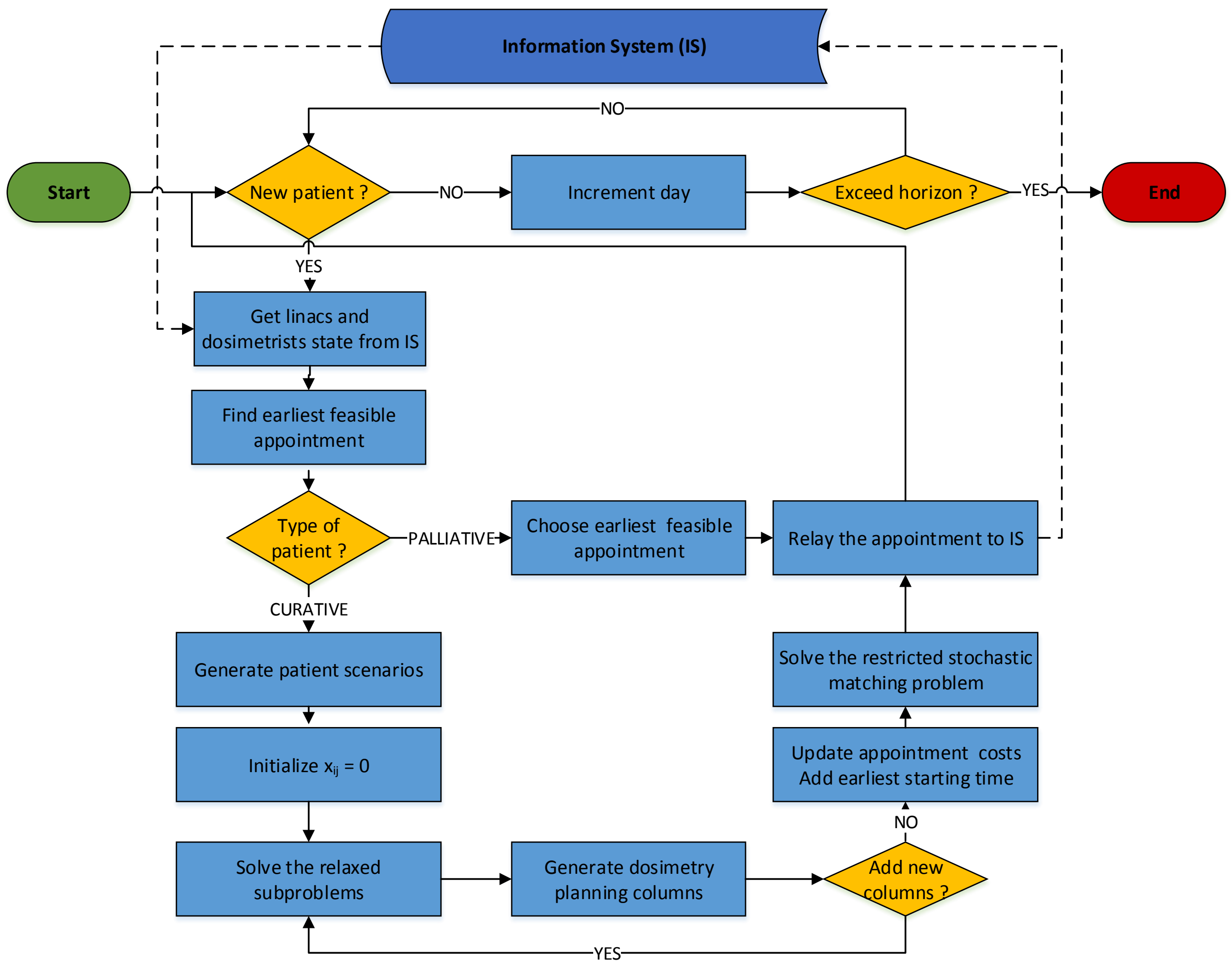
Information System (IS)
Start
New patient ?
NO
Increment day
Exceed horizon ?
YES
End
NO
YES
Get linacs and dosimetrists state from IS
Find earliest feasible appointment
Type of patient ?
PALLIATIVE
Choose earliest feasible appointment
Relay the appointment to IS
CURATIVE
Generate patient scenarios
Solve the restricted stochastic matching problem
Initialize $x_{ij} = 0$
Update appointment costs
Add earliest starting time
NO
Solve the relaxed subproblems
Generate dosimetry planning columns
Add new columns ?
YES

Figure 2: Flow chart of global CICL online stochastic algorithm.

5.6 Experiments

All the experiments were run over 8 threads on a computer with an Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz and 32 GB of memory. We used CPLEX and CP OPTIMIZER 12.6.

We first study the behavior of the algorithm and then present results for a real data set. The scenarios used for the online algorithms were drawn from the empirical distribution of the CICL. A large proportion (70%) of the curative patients are known in advance because they have already undergone surgery and/or chemotherapy in the center.

5.6.1 Sensitivity analysis

There are two linacs with 29 slots plus 3 overtime slots and 4 dosimetrists. Three statistics computed over 30 runs are used to analyze the algorithm presented in Figure 2:

- First appointment canceled: the number of patients for which the first treatment session is canceled because the pretreatment is late. This corresponds to the number of violated precedence constraints (34).

- Overdue: the number of patients for which the waiting-time target has not been met.
- Objective: the sum over all patients of deadline violations and waiting-time penalties.

The primal-ratio η can take into account the risk aversion of the operator in relation to the precedence constraints. Figure 3 shows the evolution of the three statistics as a function of the primal-ratio.

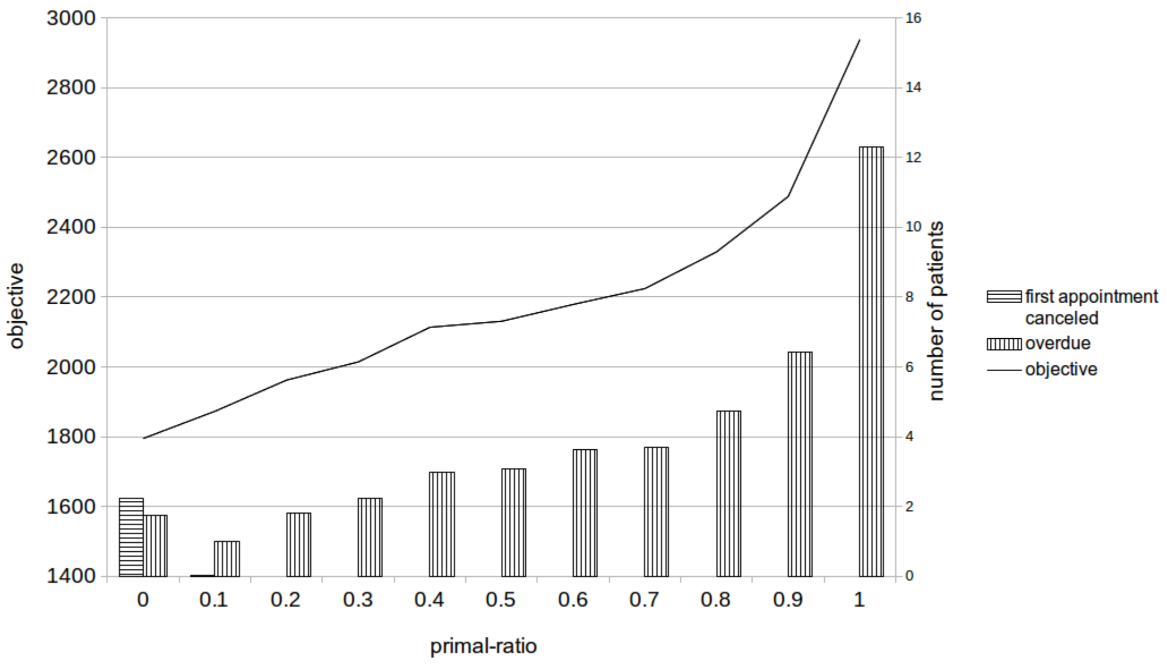

Figure 3: Analysis of the value of the primal-ratio.

The left axis gives the average value of the objective, and the right axis counts the average number of patients in each of the groups. As the primal-ratio increases, the objective value grows. Indeed, the more careful the operator, the later the patients will be given their first linac appointment. When the primal-ratio is high, the algorithm tends to delay curative patients to avoid canceling treatment sessions because of pretreatment delays. The objective function is also high because waiting time and deadline violations are penalized. In contrast, when the primal-ratio is low, the objective function is also low because the patients have short waiting times; however, some linac appointments will be canceled. Since this situation must be avoided, we set the primal-ratio to $\eta = 0.2$.

5.6.2 Results

We evaluate the approach on a large real instance from the CICL with 1529 patients over 248 working days. The center operates with 4 dosimetrists and 4 linacs. Table 1 compares three algorithms according to various criteria: 1) the number of violated precedence constraints (34) (i.e., first appointment canceled), 2) the number of targets not met (i.e., overdue), 3) the average waiting time, and 4) the number of overtime slots used. The CICL algorithm is the greedy procedure that is currently used at the facility. The online clairvoyant algorithm is the the procedure presented in (Legrain et al. 2014). These two algorithms, unlike the third, do not take into account the dosimetry planning and assume a fixed delay. Finally, the OS algorithm follows the procedure presented in Figure 2 and uses 8 scenarios (because the computer can run 8 threads simultaneously).

Table 1: Comparison of algorithms on a real instance.

	Appointment	Target not met			Average waiting time			Overtime
Algorithm	canceled	>3 days	>14 days	>28 days	3 days	14 days	28 days	slots
CICL	230	373 (25)	104	0	3.45	12.58	12.63	111
Online Clairvoyant	107	335 (25)	67 (1)	0	3.27	12.23	14.77	19
Online Stochastic	1	326 (24)	119 (5)	0	3.23	14.04	18.43	8

Table 1 shows that the OS algorithm outperforms the two other procedures. It has just one canceled appointment whereas the CICL procedure has 230 cancellations and the online clairvoyant has 107. Furthermore, the number of targets not met remains stable. The OS algorithm delays the curative patients to increase the likelihood that the pretreatment will be completed on time. The curative patients do not do as well on criteria 2) and 3), but the palliative patients do better on these criteria. Finally, the computational time for the OS algorithm is an average of 20 seconds per patient.

6 Application II: Task assignment and routing problem

The task assignment and routing problem arises in the management of warehouse operations (Gu et al. 2007). Orders arrive in the warehouse and are treated by the manager or the warehouse management system (WMS), which splits or groups them into pick-up tasks. Additional tasks, such as put-away, counting, and replenishing must also be performed. A route is composed of a starting location, various intermediate stops, and a final destination. Whenever a worker completes a task, the WMS must dynamically choose the next task for that worker.

In general, the tasks are given priorities, and the highest-priority tasks should be completed first. However, a few low-priority tasks may be interleaved to avoid unnecessary deadheads (i.e., trips without any goods). For example, a transition task may be inserted between the final destination of the previous task and the starting location of the next, resulting in more efficient routes.

The offline task assignment problem has been widely studied (Ernst et al. 2004). Corominas et al. (2006) propose a heuristic to assign tasks in the service industry once the shift schedule has been fixed. Bard and Wan (2006) develop a tabu search procedure for a similar problem. Their system has been tested on real data from a U.S. Postal Service mail processing and distribution center. Boyer et al. (2014) propose a grammar-based approach to include the task assignment problem at the shift scheduling level.

However, few studies investigate the online task assignment problem. For a warehouse, Rubrico et al. (2011) reschedule the current routes each time a new task arrives. They develop VRP heuristics to compute new routes as well as heuristics to adapt the current routes. These heuristics give good results when there is a balanced mix of static and dynamic tasks.

However, they are pure online algorithms: no stochastic information about future tasks is considered.

In this application, which has been realized in collaboration with JDA Software, we investigate the task assignment and routing problem in an OS fashion. The challenge in deciding the next task for each worker lies in finding a balance between minimizing deadheads and performing urgent tasks quickly enough. We choose to explicitly maximize the sum of the priorities of the completed tasks over the horizon, while the deadheads will be implicitly minimized through the generation of patterns.

6.1 Online stochastic formulation

In this application, the requests normally represent the tasks. However, an assignment becomes effective only when the employee has completed his/her previous nonpreemptive task. To avoid unnecessary computation, we can postpone the allocation until an employee is available. We define the requests as a sequence of employees that have completed an assignment (an employee will normally appear several times in this sequence). Consequently, there is a set $\mathcal{T}$ of waiting tasks, which might not be empty at the beginning of the horizon.

We present a stochastic optimization model for this task assignment and routing problem. When the jth request arrives, i.e., employee $r \in \mathcal{R}$ has finished an assignment, the WMS assigns a new task to this employee, and thus it dynamically builds a global set of efficient routes for all the workers. The model infers the average cost of completing the current routes for a finite set Ω_j of future task scenarios. Since solving a stochastic vehicle routing problem is computationally demanding, the algorithm solves this problem once for all the employees and stores their future assignments. When the jth request arrives, the algorithm either retrieves a stored assignment or solves the OS problem if the previously computed assignment has already been performed.

The scenario $\omega \in \Omega_j$, with probability p^ω, represents the set $\mathcal{T}^\omega$ of future tasks, which contains the set $\mathcal{T}$ (common to all the scenarios) of waiting tasks. Variable x_{ir} is 1 if waiting task $i \in \mathcal{T}$ with priority c_i is allocated to employee r, and 0 otherwise. Each new task must form the beginning of a future route $p \in \mathcal{S}_r$. This route is described by a pattern for employee r: parameter a_{irp} (b_{irp}) is 1 if task i is on route p (starts route p), and 0 otherwise. The variable y_i^ω is 1 if task $i \in \mathcal{T}^\omega$ is completed in scenario ω, and 0 otherwise; v_{rp}^ω is 1 if route p is allocated to employee r, and 0 otherwise.

$$\max \sum_{i \in \mathcal{T}} \sum_{r \in \mathcal{R}} c_i x_{ir} + \sum_{\omega \in \Omega_j} p^\omega \Big[\sum_{i \in \mathcal{T}^\omega} c_i y_i^\omega \Big] \tag{57}$$

subject to

$$\sum_{r\in\mathcal{R}} x_{ir} + y_i^\omega \le 1, \qquad \forall i \in \mathcal{T},\ \forall \omega \in \Omega_j \tag{58}$$

$$\sum_{p\in\mathcal{S}_r} v_{rp}^\omega \le 1, \qquad \forall r \in \mathcal{R},\ \forall \omega \in \Omega_j \tag{59}$$

$$\sum_{r\in\mathcal{R}} \sum_{p\in\mathcal{S}_r^\omega} a_{irp} v_{rp}^\omega \ge y_i^\omega, \qquad \forall i \in \mathcal{T}^\omega,\ \forall \omega \in \Omega_j \tag{60}$$

$$\sum_{p\in\mathcal{S}_r^\omega} b_{irp}^\omega v_{rp}^\omega \ge x_{ir}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T},\ \forall \omega \in \Omega_j \tag{61}$$

$$x_{ir} \in \{0,1\}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T} \tag{62}$$

$$y_i^\omega,\ v_{rp}^\omega \in \{0,1\}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T}^\omega,\ \forall \omega \in \Omega_j,\ \forall p \in \mathcal{S}_r^\omega \tag{63}$$

The objective (57) maximizes the sum of the priorities of the current and expected assignments. Constraints (58) verify that a potential next task, that must be in the set $\mathcal{T}$ of waiting tasks, is completed at most once in each scenario. Constraints (59) ensure that at most one route is assigned to each worker. The columns v_{rp}^ω are generated during the solution process because the sets $\mathcal{S}_r$ are large (see Appendix 8.1 for the column generation procedure). Constraints (60) link each route to the included tasks. Constraints (61) ensure that the next assignment of each employee starts a feasible route. Finally, Constraints (62) and (63) define x_{ir}, y_i^ω, and v_{rp}^ω as binary variables.

6.2 Integer subproblems

The OS formulation is transformed as shown in the methodology presented in Section 4. The integer subproblems solve a VRP for a solution $\overline{x}_{ir}$ and for each scenario ω.

$$Q(\overline{x}_{ir},\omega) = \max \sum_{i\in\mathcal{T}^\omega} c_i y_i^\omega \tag{64}$$

subject to

$$y_i^\omega \le 1 - \sum_{r\in\mathcal{R}} \overline{x}_{ir}, \qquad \forall i \in \mathcal{T} \tag{65}$$

$$\sum_{p\in\mathcal{S}_r} v_{rp}^\omega \le 1, \qquad \forall r \in \mathcal{R} \tag{66}$$

$$\sum_{r\in\mathcal{R}} \sum_{p\in\mathcal{S}_r^\omega} a_{irp} v_{rp}^\omega \ge y_i^\omega, \qquad \forall i \in \mathcal{T}^\omega \tag{67}$$

$$\sum_{p\in\mathcal{S}_r^\omega} b_{irp}^\omega v_{rp}^\omega \ge \overline{x}_{ir}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T} \tag{68}$$

$$y_i^\omega,\ v_{rp}^\omega \in \{0,1\}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T}^\omega,\ \forall p \in \mathcal{S}_r^\omega \tag{69}$$

6.3 Probabilistic feasibility cuts

Let $\mathcal{D}$ be the subset of constraints (68). The dual variables associated with these constraints are null if $\overline{x}_{ir} = 0$, i.e., task i is not allocated to employee r. If $\overline{v}_{rp}^\omega$ is the solution of the subproblem, we add the feasibility cut $\sum_{p\in\mathcal{S}_r^\omega} b_{irp}^\omega \overline{v}_{rp}^\omega \geq x_{ir}$ in the following restricted stochastic matching problem for each task i, each employee r, and each scenario ω. Since the variables x_{ij} are binary, we transform these cuts to $\lceil \sum_{p\in\mathcal{S}_r^\omega} b_{irp}^\omega \overline{v}_{rp}^\omega \rceil \geq x_{ir}$. Finally, these constraints prevent the assignment of a task that is not at the beginning of an active route (i.e., $\overline{v}_{rp}^\omega > 0$) in each solution of the relaxed subproblem associated with scenario $\omega \in \Omega_j$.

For a primal-ratio η, let $\mathcal{T}_r$ be the set of the feasible next tasks that are at the beginning of at least $\eta|\Omega_j|$ active routes of employee r. If the subset $\mathcal{T}_r$ is empty, we instead define $\mathcal{T}_r = \{i \in \arg\max_{i_0\in\mathcal{T}}(|\{\omega \in \Omega_j| \sum_{p\in\mathcal{S}_r^\omega} b_{i_0rp}^\omega \overline{v}_{rp}^\omega > 0\}|)\}$, which is the set of tasks that occur the most often at the beginning of an active route. The subset Ω_j^{ir} is thus equal to the singleton $\{\omega_{ir}| \sum_{p\in\mathcal{S}_r^\omega} b_{irp}^{\omega_{ir}} > 0, \text{if } i \in \mathcal{T}_r,\ \sum_{p\in\mathcal{S}_r^\omega} b_{irp}^{\omega_{ir}} = 0, \text{otherwise}\}$ for each employee r and each task i.

6.4 Restricted stochastic matching problem

Let β_i^ω and δ_i^ω be the dual variables associated respectively with constraints (65) and (68). The stochastic priority associated with task i for worker r is defined to be $\sum_{\omega\in\Omega_j} p^\omega[\beta_i^\omega + \delta_i^\omega]$. Therefore, the restricted stochastic matching problem is

$$Z^* = \max \sum_{r\in\mathcal{R}} \sum_{i\in\mathcal{F}_r} (c_i - \sum_{\omega\in\Omega_j} p^\omega[\beta_i^\omega + \delta_i^\omega])x_{ir} \tag{70}$$

subject to

$$\sum_{r\in\mathcal{R}} x_{ir} \leq 1, \qquad \forall i \in \mathcal{T} \tag{71}$$

$$\sum_{i\in\mathcal{T}} x_{ir} \leq 1, \qquad \forall r \in \mathcal{R} \tag{72}$$

$$x_{ir} \leq \lceil \sum_{p\in\mathcal{S}_r^\omega} b_{irp}^\omega \overline{v}_{rp}^\omega \rceil, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T},\ \forall \omega \in \Omega_j^{ir} \tag{73}$$

$$x_{ir} \in \{0,1\}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T} \tag{74}$$

Constraints (71) and (72) are added because this matching problem is solved once for all the employees. We explicitly maximize the total expected priority Z^*, which does not depend on the employee: the dual variables measure the expected reward of performing a task. We implicitly minimize the deadheads through the probabilistic feasibility cuts (73): based on the primal solution $\overline{v}_{rp}^\omega$, these constraints forbid certain next assignments for each employee.

6.5 Global online stochastic algorithm

Figure 4 illustrates the steps of the OS algorithm for this application. The information system must communicate the states of the employees and the tasks waiting in the queue. The simulator generates new requests based on the employee availability and updates the employee locations in the warehouse as well as the queue.

For each new request, the algorithm either assigns the stored next task to an employee if the previous reoptimization has already computed this assignment or reoptimizes the whole routing problem.

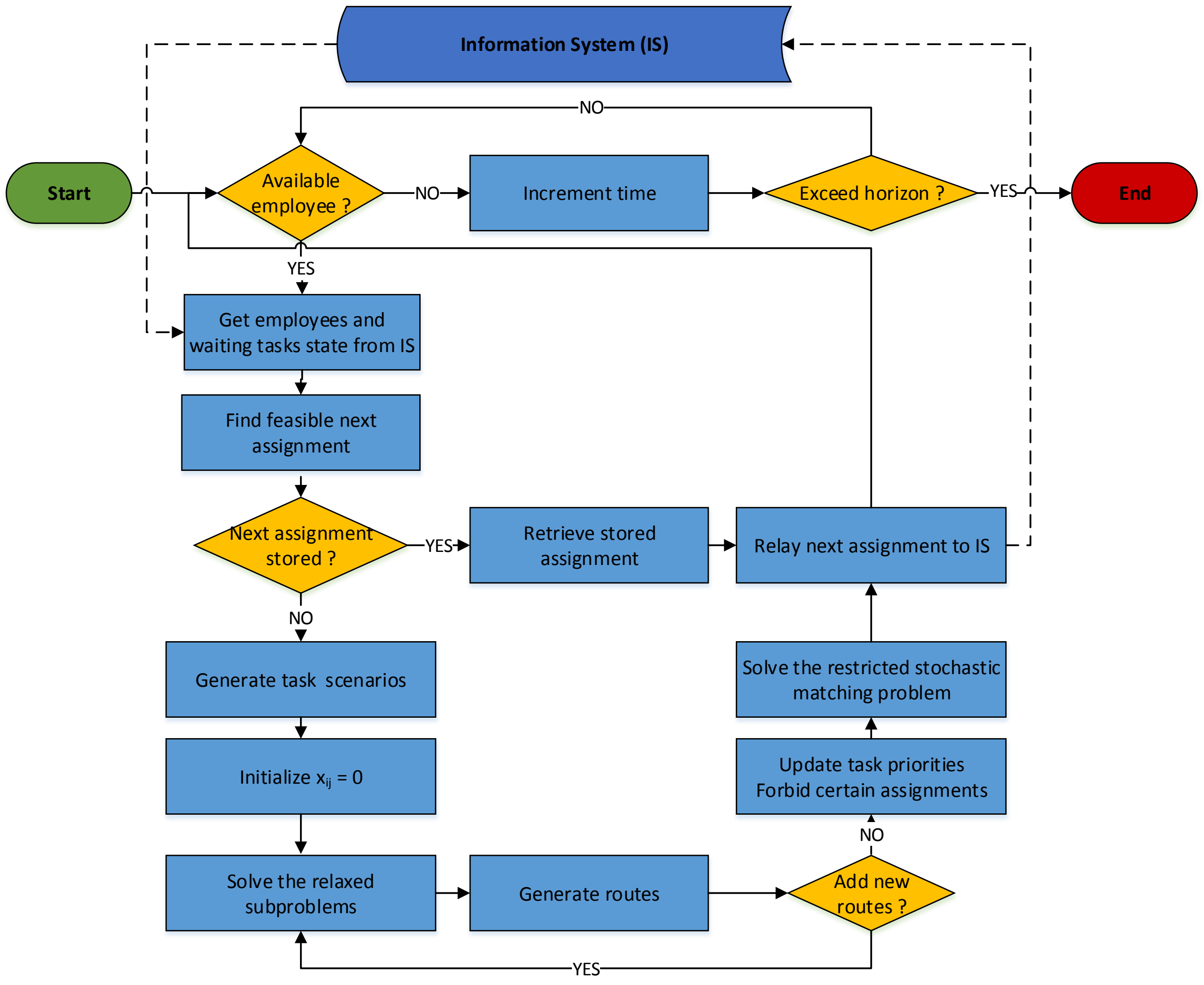
Information System (IS)
NO
Start
Available employee ?
NO
Increment time
Exceed horizon ?
YES
End
YES
Get employees and waiting tasks state from IS
Find feasible next assignment
Next assignment stored ?
YES
Retrieve stored assignment
Relay next assignment to IS
NO
Generate task scenarios
Solve the restricted stochastic matching problem
Initialize $x_{ij} = 0$
Update task priorities Forbid certain assignments
NO
Solve the relaxed subproblems
Generate routes
Add new routes ?
YES

Figure 4: Flow chart of global JDA online stochastic algorithm.

6.6 Experiments

The planning horizon is 2 hours, and the workers start and finish at the rest area. All the experiments were run over 8 threads on a server with an Intel(R) Xeon(R) E5530 CPU @ 2.40 GHz and 24 GB of memory. We use CPLEX 12.5.0 and set the CPU time to two hours because of the planning horizon.

We built instances based on real data sets from one of JDA Software's clients. The scenarios were based on a random task selection from the client database. The queue contains half of the total tasks at the beginning of the horizon, as in Rubrico et al. (2011).

6.6.1 Sensitivity analysis

We report the results of a sensitivity analysis based on an instance with 150 tasks and 6 employees. We generate 10 scenarios, and the primal-ratio η controls the quality of the solution. Figure 5 shows the evolution of the average objective (computed over 30 runs) as a function of the primal-ratio.

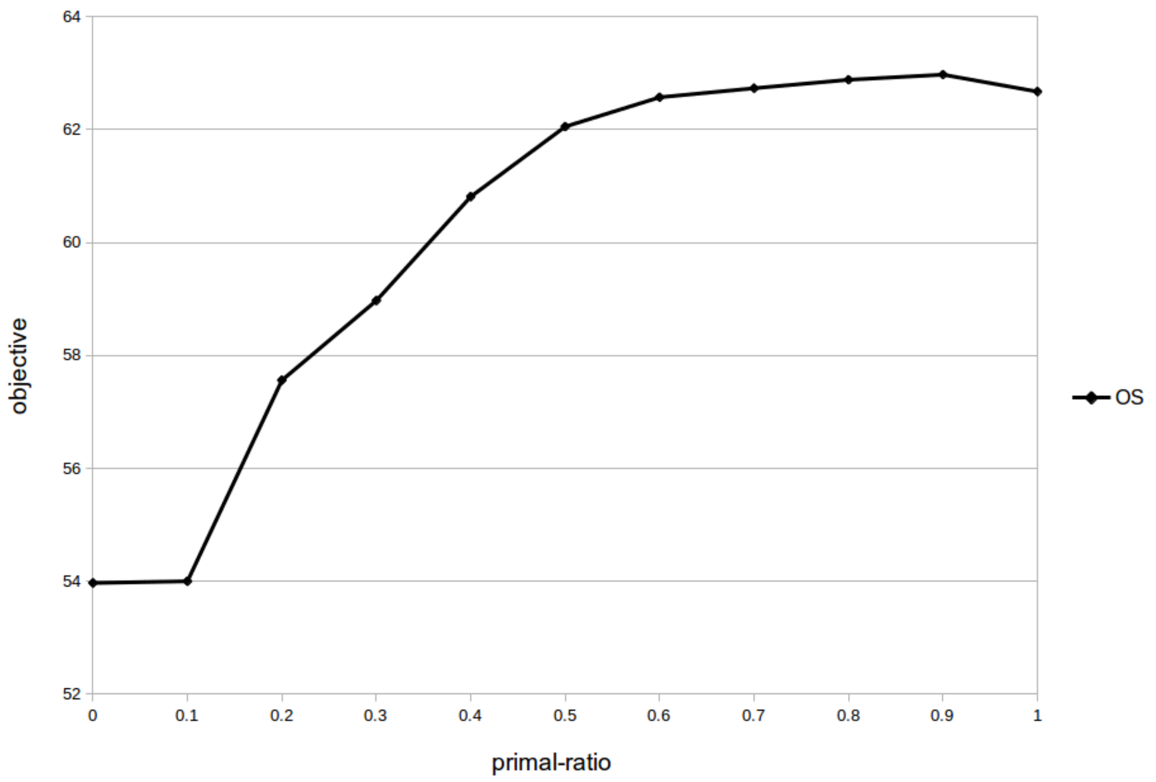

Figure 5: Analysis of the value of the primal-ratio.

As the primal-ratio grows, the objective value increases until $\eta = 0.5$ and then it starts to level off. This indicates that the primal solutions of the relaxed subproblems give more and more information about the efficiency of the routes. To avoid unnecessary primal noise, we set the primal-ratio to $\eta = 0.5$ for the results below.

6.6.2 Results

We compare the OS algorithm to a greedy heuristic similar to that used in the WMS of JDA Software. The JDA procedure chooses the maximum priority task in the same zone as the employee. Our algorithm uses five scenarios or all the available computational time (i.e., two hours).

Preliminary tests show that the OS algorithm performs poorly (i.e., worse than the JDA procedure) without the probabilistic feasibility cuts. Indeed, the one-iteration L-shaped procedure takes into account only the expected reward of a task and thus allocates the tasks with the highest expected priorities to the employees. Deadheads are implicitly minimized in the relaxed subproblems but not in the stochastic matching problem when the feasibility cuts are absent.

The instances used for the comparison contain from 13 to 26 employees and, in average, 35 tasks per employee. Figure 6 plots for each instance the improvement of the objective as a function of the increase in the number of completed tasks. The improvements are measured as relative gains of the OS algorithm against the JDA procedure. The OS algorithm clearly

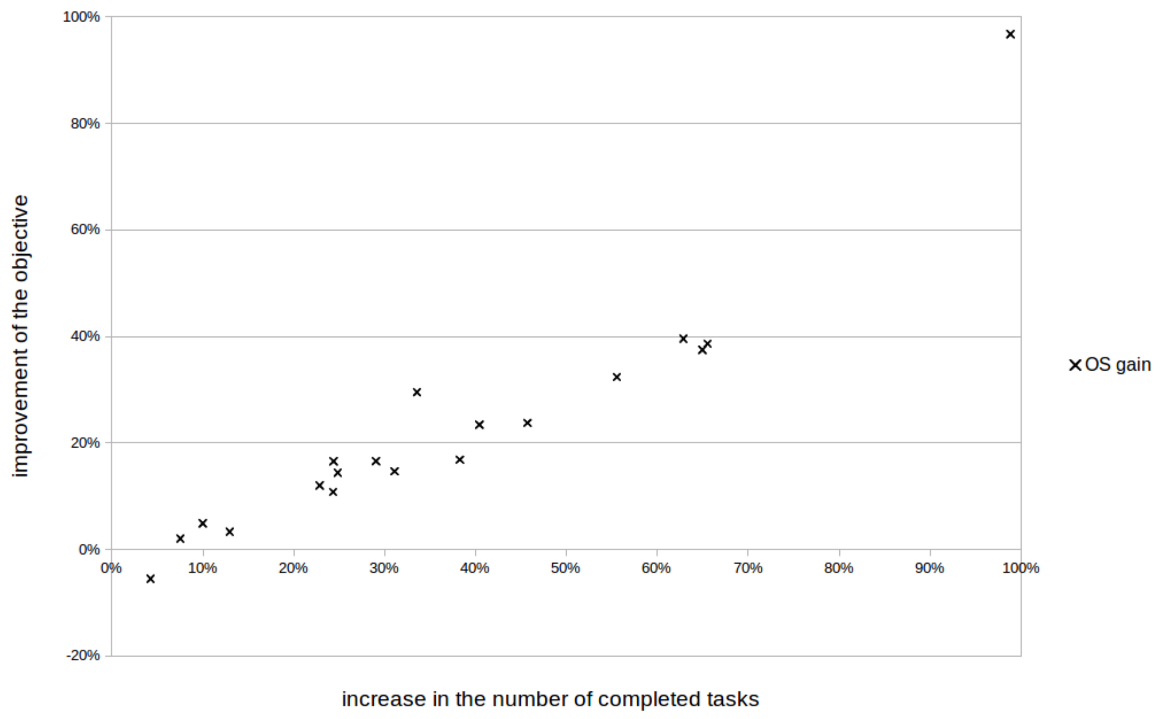

Figure 6: Gain of the OS algorithm on real instances.

outperforms the JDA algorithm, as, in average, it increases of 35% the number of completed tasks and raises of 20% the objective function. The observed improvements correspond to tremendous productivity gains for a WMS.

7 Conclusion

In this paper, we have proposed a mathematical-programming-based framework for general OS resource allocation problems. We model the problem as a resource allocation problem via Dantzig–Wolfe decomposition and then as a two-stage program with fixed recourse using classical stochastic optimization tools. Finally, we apply Benders decomposition. We build a one-iteration L-shaped procedure to quickly obtain dual information about the future resource load. We compute the linear relaxation of an integer subproblem (an offline resource allocation problem) for each scenario to infer this load. When we have insufficient information for a resource (e.g., the dual variables associated with the resource are null), we add probabilistic feasibility cuts based on primal solutions to the master problem to remove infeasible and nonoptimal decisions.

For each new request, the online algorithm either applies an online policy or reoptimizes the resource allocation problem. At each reoptimization, we solve a restricted stochastic matching problem (i.e., the master problem) to allocate the new request according to the stochastic information about the future resource load and about expected infeasible and nonoptimal decisions. Two applications have illustrated the modeling process and demonstrated the efficiency of the proposed framework on real data sets. The OS algorithm outperforms existing procedures for both applications.

We believe that many online decision problems could be solved by our algorithm. The main challenge is to model the problem as a resource allocation problem via a Dantzig–Wolfe decomposition. One can then apply the steps of the framework.

8 Appendices

This section presents the specialized algorithms for generating the columns and checking the feasibility of a given allocation pattern.

8.1 Specialized algorithms for column generation procedure

The appointment and booking problem needs a specialized column generation procedure, while the task assignment and routing problem is a classical VRP in its offline version.

The network for the latter problem can easily be described. The tasks are nodes with a serving time and a cost (i.e., the priority). The travel time between tasks is represented by free arcs. The last routes must finish before the end of the horizon. The difficulty of this problem lies in the size of the network, which is complete. The OS algorithm solves only a linear relaxation of this problem for each scenario. The problem is decomposed using column generation (Desaulniers et al. 2005). The master problem is solved using a linear programming solver, and the subproblems, which are just resource-constrained shortest path problems in the complete network, are solved via dynamic programming.

The columns v_i^ω of the appointment and booking problem are generated during the solution process using a genetic algorithm inspired by Bertel and Billaut (2004).

Algorithm 1 Genetic Algorithm

```
Population: list of chromosomes: $P := [\,]$
Initialization: add $N/4$ chromosomes built with dispatching rules and fill the rest with random chromosomes
while solving time $< T$ AND not enough different chromosomes with a negative reduced cost do
    for all chromosomes $c$ in population $P$ do
        Cyclic Crossover: with probability $p_c$, cross $c$ with a random different chromosome
        Mutation: with probability $p_m$, randomly swap two positions of $c$
        Intensification: with probability $p_l$, make a small local search with insertions on $c$
    end for
    Add all new chromosomes to $P$
    Selection: keep the best $N$ chromosomes
end while
```

The dosimetry scheduling problem has two tasks and a time window per job. These time windows correspond to fixed delays: they ensure that all the other pretreatment tasks are performed. The two tasks are the preparation and the verification of the treatment by a dosimetrist. Since the OS formulation follows a Dantzig–Wolfe scheme (Gélinas and Soumis

2005), this flow-shop problem minimizes the weighted completion time (due to the dual variables linked to constraints (34) and (35)). Algorithm 1 thus tries to build in time T a population P of N chromosomes with a negative reduced cost. A chromosome is a sequence of tasks and represents a dosimetry plan. A task occurs twice in a chromosome: the first appearance corresponds to the preparation of the dosimetry, and the second corresponds to the verification. A plan is made from a chromosome by simply scheduling each task in the chromosome as early as possible. The cyclic crossover is also presented in Bertel and Billaut (2004). Finally, the intensification phase aims to improve some of the chromosomes: it tests several insertions to see if they decrease the reduced cost.

8.2 Specialized algorithms for checking allocation pattern feasibility

The feasibility of a pattern allocation (i.e., the next task to perform) is easily checked for the task assignment and routing problem: the algorithm checks that the employee has time to return to the rest area after performing an assignment. However, for the appointment and booking problem, the algorithm must solve a flow-shop problem to determine the feasibility of the dosimetry plan induced by an allocation pattern (i.e., a linac appointment).

The genetic algorithm quickly computes several columns, but it is impossible to know if a dosimetry plan is infeasible or optimal. To check the feasibility of an allocation pattern and to schedule the daily tasks at the dosimetry, we introduce a constraint program. It can check feasibility and find better solutions, but it takes more computational time. Let N_d be the number of dosimetrists and $\mathcal{P}_k$ be the set of patients waiting for dosimetry on day k (they correspond to jobs in this flow-shop problem). The variables C_j represent the completion time of the dosimetry for job j. The activities are interval variables in constraint programming. They are defined by four linked variables: the beginning, the end, the length, and the presence of an interval. The variables t_{ij}, t^d_j, and t^d_{ij} are thus activities, and for job j they represent, respectively, the ith dosimetry task, the only dosimetry task performed by the dth dosimetrist, and the ith dosimetry task performed by the dth dosimetrist. Some activities are of course optional and have length zero; the compulsory tasks are the activities t_{ij}. Finally, we aim to minimize over all the patients the square of the tardiness, $(\max(0, C_j - b_j))^2$, and the weighted completion time, $w_j C_j$. The tardiness is more important because the dosimetry plan must be feasible. However, the weighted completion time should break any ties.

$$\min \sum_{j \in \mathcal{P}_k} [(\max(0, C_j - b_j))^2 + w_j C_j] \tag{75}$$

subject to

$$alternative(t_{ij}, d = 1 \dots N_d, t_{ij}^d), \qquad \forall i = 1 \dots T, \forall j \in \mathcal{P}_k \tag{76}$$

$$sequential\ machine(t_{ij}^d, i = 1 \dots T, j \in \mathcal{P}_k), \qquad \forall d = 1 \dots N_d \tag{77}$$

$$alternative(t_j^d, i = 1 \dots T, t_{ij}^d), \qquad \forall d = 1 \dots N_d, \forall j \in \mathcal{P}_k \tag{78}$$

$$t_{ij}.end \leq t_{(i+1)j}.begin, \qquad \forall i = 1 \dots T-1, \forall j \in \mathcal{P}_k \tag{79}$$

$$C_j \geq t_{Tj}.end, \qquad \forall j \in \mathcal{P}_k \tag{80}$$

$$t_{ij}, t_j^d, t_{ij}^d \in I_j, \qquad \forall j \in \mathcal{P}_k \tag{81}$$

$$C_j \in Days, \qquad \forall j \in \mathcal{P}_k \tag{82}$$

The alternative global constraints (76) ensure that only one dosimetrist performs the ith task for job j. The sequential resource constraints (77) ensure that each dosimetrist executes one task at a time. The global constraints (78) ensure that a dosimetrist completes at most one task for job j. Constraints (79) simply ensure that the tasks of a job are performed in the right sequence. Constraints (80) compute the completion time of each job j. Finally, constraints (81) and (82) describe the domains of the variables, where the set $Days$ represents the days of the planning horizon and the set I_j defines the discretized domain for each job (the possible ready and due dates are the bounds of the set).

References

Bard JF, Wan L (2006) The task assignment problem for unrestricted movement between workstation groups. *Journal of Scheduling* 9(4):315–341.

Bäuerle N, Rieder U (2011) *Markov Decision Processes with Applications to Finance* (Springer Science & Business Media).

Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4(1):238–252.

Bertel S, Billaut JC (2004) A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation. *European Journal of Operational Research* 159(3):651–662.

Birge JR, Louveaux F (2011) *Introduction to Stochastic Programming.* Springer Series in Operations Research and Financial Engineering (New York, NY, USA: Springer), 2nd edition, URL `http://dx.doi.org/10.1007/978-1-4614-0237-4`.

Boyer V, Gendron B, Rousseau LM (2014) A branch-and-price algorithm for the multi-activity multi-task shift scheduling problem. *Journal of Scheduling* 17(2):185–197.

Buchbinder N (2008) *Designing Competitive Online Algorithms via a Primal-Dual Approach.* Ph.D. thesis, Technion.

Ciocan DF, Farias VF (2012) Dynamic allocation problems with volatile demand. *Mathematics of Operations Research* 37(3):501–525.

Corominas A, Pastor R, Rodríguez E (2006) Rotational allocation of tasks to multifunctional workers in a service industry. *International Journal of Production Economics* 103(1):3–9.

Dantzig GB, Wolfe P (1960) Decomposition principle for linear programs. *Operations Research* 8(1):101–111.

Desaulniers G, Desrosiers J, Solomon MM (2005) *Column Generation* (Springer).

Ernst AT, Jiang H, Krishnamoorthy M, Sier D (2004) Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research* 153(1):3–27.

Feldman J, Henzinger M, Korula N, Mirrokni VS, Stein C (2010) Online stochastic packing applied to display ad allocation. *Algorithms–ESA 2010*, 182–194 (Springer).

Feldman J, Mehta A, Mirrokni V, Muthukrishnan S (2009) Online stochastic matching: Beating 1-1/e. *50th Annual IEEE Symposium on Foundations of Computer Science, 2009 (FOCS'09)*, 117–126 (IEEE).

Gélinas S, Soumis F (2005) *Dantzig-Wolfe Decomposition for Job Shop Scheduling* (Springer).

Gu J, Goetschalckx M, McGinnis LF (2007) Research on warehouse operation: A comprehensive review. *European Journal of Operational Research* 177(1):1–21.

Gupta D, Denton B (2008) Appointment scheduling in health care: Challenges and opportunities. *IIE transactions* 40(9):800–819.

Jaillet P, Lu X (2012) Near-optimal online algorithms for dynamic resource allocation problems. *arXiv preprint arXiv:1208.2596* .

Jaillet P, Lu X (2014) Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research* 39(3):624–646.

Karande C, Mehta A, Tripathi P (2011) Online bipartite matching with unknown distributions. *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, 587–596 (ACM).

Karp RM, Vazirani UV, Vazirani VV (1990) An optimal algorithm for on-line bipartite matching. *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, 352–358 (ACM).

Klassen KJ, Rohleder TR (2004) Outpatient appointment scheduling with urgent clients in a dynamic, multi-period environment. *International Journal of Service Industry Management* 15(2):167–186, URL `http://dx.doi.org/10.1108/09564230410532493`.

Legrain A, Fortin MA, Lahrichi N, Rousseau LM (2014) Online stochastic optimization of radiotherapy patient scheduling. *Health Care Management Science* 3(3):1–14.

Legrain A, Fortin MA, Lahrichi N, Rousseau LM, Widmer M (2015) Stochastic optimization of the scheduling of a radiotherapy center. *Journal of Physics: Conference Series*, volume 616, 012008 (IOP Publishing).

Legrain A, Jaillet P (2013) *Stochastic Online Bipartite Resource Allocation Problems* (CIRRELT).

Manshadi VH, Gharan SO, Saberi A (2012) Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research* 37(4):559–573.

Mehta A, Saberi A, Vazirani U, Vazirani V (2007) Adwords and generalized online matching. *Journal of the ACM (JACM)* 54(5).

Novoa C, Storer R (2009) An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research* 196(2):509–515.

Patrick J, Puterman ML, Queyranne M (2008) Dynamic multipriority patient scheduling for a diagnostic resource. *Operations Research* 56(6):1507–1525.

Petrovic S, Leung W, Song X, Sundar S (2006) Algorithms for radiotherapy treatment booking. *Proceedings of the 25th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG'2006), Nottingham, UK.*

Powell WB (2007) *Approximate Dynamic Programming: Solving the Curses of Dimensionality* (John Wiley & Sons).

Powell WB, Roy BV (2004) Approximate dynamic programming for high dimensional resource allocation problems. *Handbook of Learning and Approximate Dynamic Programming*, 261–280 (IEEE Press, Los Alamitos, CA).

Puterman ML (2014) *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (John Wiley & Sons).

Rubrico JI, Higashi T, Tamura H, Ota J (2011) Online rescheduling of multiple picking agents for warehouse management. *Robotics and Computer-Integrated Manufacturing* 27(1):62–71.

Ruiz R, Vázquez-Rodríguez JA (2010) The hybrid flow shop scheduling problem. *European Journal of Operational Research* 205(1):1–18.

Sauré A, Patrick J, Tyldesley S, Puterman ML (2012) Dynamic multi-appointment patient scheduling for radiation therapy. *European Journal of Operational Research* 223(2):573–584.

Slyke RV, Wets RJ (1969) L-shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal and Applied Mathematics* 638–663.

Van Hentenryck P, Bent R (2009) *Online Stochastic Combinatorial Optimization* (The MIT Press).