Computers & Operations Research 167 (2024) 106641

Contents lists available at ScienceDirect

Computers and Operations Research

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

Distributionally robust scheduling of stochastic knapsack arrivals

Hayo Bos [a,b,*], Richard J. Boucherie [b,c], Erwin W. Hans [b,d], Gréanne Leeftink [b,d]

[a] Diakonessenhuis Utrecht, P.O. Box 80250, 3508 TG Utrecht, The Netherlands
[b] Center for Healthcare Operations Improvement and Research, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
[c] Stochastic Operations Research, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
[d] Industrial Engineering and Business Information Systems, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Keywords:
Stochastic knapsack
Cyclic schedule
Scheduled arrivals
Distributionally robust optimization
Wasserstein distance
Master surgical schedule

ABSTRACT

This paper studies the discrete-time Stochastic Knapsack with Periodic Scheduled Arrivals (SKPSA). The goal is to find a schedule such that the capacity usage of the unconstrained cousin of the knapsack is as close as possible to a target utilization. We approximate the SKPSA with a Wasserstein distance based Distributionally Robust Optimization (DRO) model, resulting in the DRO-SKPSA. We present an algorithm that efficiently solves this model, and show that the DRO-SKPSA produces robust schedules. The problem arises in particular in healthcare settings in the development of Master Surgical Schedules (MSSs). We discuss managerial insights for MSSs with downstream capacity constraints.

1. Introduction

In this paper we consider the problem of developing a periodic schedule, dictating the arrivals to a discrete-time stochastic knapsack. More precisely, jobs of different classes are to be scheduled on multiple parallel machines. These jobs generate a random number of objects that occupy one or more shared resources of the knapsack for a random holding time. The objective is to find a schedule such that the occupancy of these resources is close to the target occupancy of the unconstrained cousin of the knapsack. We introduce the Stochastic Knapsack with Periodic Scheduled Arrivals (SKPSA), and its Distributionally Robust Optimization (DRO) approximation, the Distributionally Robust Stochastic Knapsack with Periodic Scheduled Arrivals (DRO-SKPSA).

The problem that motivated this research is a common one in healthcare. Surgical departments are well known to be one of the most expensive departments of hospitals, driving other departments such as wards. The Operating Rooms (ORs) correspond to the parallel machines, and the ward to the stochastic knapsack. The relationship between the surgical department and downstream wards has been studied in several ways. Vanberkel et al. (2011) were on of the first to explicitly relate the surgical schedule to bed occupancy on a tactical level by predicting bed occupancy given the Master Surgical Schedule (MSS). This has been further extended by Kortbeek et al. (2015) and Braaksma et al. (2021). The optimal assignment of surgical sessions in a cyclic schedule, known as the MSS, while accounting for downstream resource capacity is studied in, e.g., van Essen et al. (2014), Fuegener et al. (2014) and Schneider et al. (2020). Scheduling patients from a waiting list given the MSS while accounting for bed capacity is done in, e.g., Neyshabouri and Berg (2017) and Shehadeh and Padman (2021). As an MSS is a cyclic schedule, the problem of finding the best MSS, i.e., the Master Surgical Schedule Problem (MSSP), fits perfectly in the framework of a discrete-time stochastic knapsack with periodic scheduled arrivals.

There are also examples from other areas. Consider for instance a production planning problem where goods are produced on parallel production lines or ordered from external parties, and stock levels are not allowed to exceed limited warehouse capacity. If demand is uncertain, it is important to find a schedule that aims to make the most efficient use of existing capacity, without exceeding its limits. This problem is commonly referred to as the "Warehousing Scheduling Problem", see, e.g., Hariga and Jackson (1996). Another example arises from the service industry, in which a consultancy company has to select projects out of a number of opportunities. All projects demand resources from consultants, for a possibly random amount of time. The goal then is to find a subset of projects and to schedule their starting time while accounting for the availability of employees, which is an example of the Project Selection and Scheduling Problem as studied in, e.g., Drezet and Billaut (2008).

In the stochastic knapsack literature, two variants are considered: the *static* and the *non-static* stochastic knapsack. The static stochastic knapsack problem is to select a subset of items with random weights

* Corresponding author at: Center for Healthcare Operations Improvement and Research, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

E-mail address: h.bos-1@utwente.nl (H. Bos).

https://doi.org/10.1016/j.cor.2024.106641
Received 27 February 2023; Received in revised form 1 February 2024; Accepted 1 April 2024
Available online 2 April 2024
0305-0548/

or rewards such that the total reward is maximized while capacity is not exceeded. In a non-static stochastic knapsack, as defined in, e.g., Ross (1995) or Kleywegt and Papastavrou (2001), objects arrive to the knapsack according to a Poisson process and occupy a certain number of resources in the knapsack for a random amount of time. Applications of the non-static stochastic knapsack include loss networks (Kelly, 1991; Ross, 1995), revenue management and reward optimization (van Slyke and Young, 2000; Kleywegt and Papastavrou, 2001), and project scheduling (Lu et al., 1999).

We approach our problem with a discrete-time (non-static) stochastic knapsack and its unconstrained cousin. The decision maker has to decide on a cyclic schedule dictating the arrivals of objects to this knapsack. Related work on scheduling arrivals in the context of Queueing Theory includes Pegden and Rosenshine (1990) and Hassin and Mendel (2008). These authors' goal is to find the best inter arrival times for customers such that waiting time and availability cost are minimized. We focus on the occupancy of the knapsack resources. Also related are batch arrivals to infinite server queues (Pang and Whitt, 2012; Daw and Pender, 2019), and product form distributions for networks of discrete-time infinite server queues with batch arrivals (Walrand, 1983; Boucherie and Van Dijk, 1991). Their work differs from our work, as their objective is to describe the behavior of the queueing system with a random arrival process, while our goal is to find the best schedule of jobs generating the batch arrivals.

Our goal is to optimize the cyclic schedule of jobs on the parallel machines generating arrivals of objects in the knapsack, given the uncertainty in the knapsack's inflow and objects' length of stay. A commonly used method in optimization under uncertainty is Stochastic Programming (SP). One of the main drawbacks of this method, however, is the assumption on full knowledge about the underlying stochastic processes, which in many cases is not realistic. SP suffers from out-of-sample disappointment (Smith and Winkler, 2006), which refers to the predicted performance based on training data versus the actual performance based on out-of-sample data. An alternative is Robust Optimization (RO), which considers the worst case of random variables with respect to their uncertainty set. This approach is known to produce rather conservative solutions (Bertsimas et al., 2011). DRO overcomes the necessity of full probabilistic knowledge, as one optimizes with respect to the worst case distribution in a set of distributions. This is less conservative than optimization with respect to the worst case realization, as is done in RO (Esfahani and Kuhn, 2018).

Three commonly used methods in optimization under uncertainty are SP, RO and DRO. One of the main drawbacks of SP is the assumption that there is complete knowledge of the underlying stochastic processes, which in most cases is not realistic. Besides, SP suffers from out-of-sample disappointment (Smith and Winkler, 2006), which refers to the predicted performance based on training data versus the actual performance based on out-of-sample data. An alternative is RO, which considers the worst case of random variables with respect to their uncertainty set. This approach is known to produce rather conservative solutions (Bertsimas et al., 2011). DRO is often considered as a middle road between these two methods.

In DRO, originating from Scarf (1957), rather than assuming a probability distribution, one assumes a ambiguity set of many distributions, after which one optimizes with respect to the worst case distribution. This approach requires less rigid assumptions compared to SP, and produces less conservative solutions compared to RO (Esfahani and Kuhn, 2018). The method has gained renewed attention due to tractability results as presented in Delage and Ye (2010) and Esfahani and Kuhn (2018). Chen et al. (2020) recently presented a new model unifying both scenario tree-based stochastic optimization and DRO, together with their modeling tool RSOME: Robust Stochastic Optimization Made Easy. Concerning DRO with chance constraints, Liu et al. (2023) consider globalized distributionally robust counterpart, in which also distributions outside of the ambiguity set are explicitly addressed and violation of the constraint by those distributions is controlled. DRO with state-dependent ambiguity is applied in, e.g., Liu et al. (2022), who model demand uncertainty in a Facility Location problem with a state-dependent ambiguity set.

DRO has been applied in, e.g., facility location problems (Saif and Delage, 2021; Shehadeh and Sanci, 2021; Liu and Li, 2021), machine scheduling problems (Chang et al., 2019; Bruni et al., 2020), power grid operations (Wang et al., 2020) and healthcare logistics (Wang et al., 2019; Shehadeh et al., 2020; Shehadeh and Padman, 2021). Similar to the shape of the uncertainty set in RO, the type of ambiguity set plays a key role in DRO. Two common types of ambiguity are moment based ambiguity and distance based ambiguity. In the former, the ambiguity set consists of all distributions subject to one or more moment based constraints. In the latter, the ambiguity set contains all distributions within a distance from a reference distribution. In this paper, we use the distance based Wasserstein ambiguity set around the empirical distribution retrieved from data. We use this set for the following reasons. First, it uses all available data through the empirical distribution, instead of only using the sample moments. Second, Wasserstein based DRO generalizes other forms of decision making under uncertainty, such as Sample Average Approximation (SAA) and the expected value approach, facilitating easy comparison. Third, there are theoretical confidence results on containing the true distribution, and fourth Wasserstein DRO has tractable reformulations (Esfahani and Kuhn, 2018).

Closely related to our work is the work presented by Wang et al. (2019) and Shehadeh and Padman (2021). Our work differs in several ways. First, in Wang et al. (2019) a similar scheduling problem is considered, but the downstream resource usage is omitted. Second, both authors used an operational timescale, where we consider a tactical timescale. Third, the schedules considered in both works are non-cyclic, where we assume a cyclic schedule. Fourth, in both works a moment-based ambiguity set is used, whereas we use a distance based ambiguity set in our DRO approximation.

Our first contribution is the introduction of the discrete-time Stochastic Knapsack with Periodic Scheduled Arrivals (SKPSA). This model generalizes existing work on, e.g., master surgery scheduling with downstream capacity constraints. Our second contribution is the DRO-SKPSA, an approximation of the SKPSA with a distance based ambiguity set. We solve this approximation using an available algorithm form literature.

This paper is organized as follows. In Section 2 we formally introduce the SKPSA and in Section 3 the DRO-SKPSA as well as a column-and-constraint generation algorithm to solve our model. In Section 4 we provide computational results, and Section 5 provides the conclusions of our research and possible directions for future work.

2. Stochastic knapsack with periodic scheduled arrivals

We consider the discrete-time stochastic knapsack with periodic scheduled arrivals. In this model, a schedule of jobs generates objects routed to the knapsack, in which the objects have to complete their length of stay.

The arrival process of objects is generated by a repeating cyclic schedule of T time units, $\mathcal{T} = \{1, \dots, T\}$, that schedules jobs of classes $k \in \mathcal{K}$, $\mathcal{K} = \{1, \dots, K\}$, over M machines, $\mathcal{M} = \{1, \dots, M\}$. Let cycle s start at time $t = sT$, $s \in \{\dots, -1, 0, 1, \dots\}$. Let $x_{k,m,t} \in \{0, 1\}$ be 1 if we schedule job $k \in \mathcal{K}$ on machine $m \in \mathcal{M}$ at time $t \in \mathcal{T}$. A complete cycle of the cyclic schedule is then encoded by $\mathbf{x} \in \{0, 1\}^{KMT}$. A class-$k$ job takes one time unit to complete and generates a random number O_k of objects of class k that arrive instantaneously to the knapsack. We assume that for each $k \in \mathcal{K}$, the number of objects O_k are independent and identically distributed, with finite support $\{1, \dots, \overline{O}_k\}$. Let $p_{O_k}(o) = \mathbb{P}(O_k = o)$, $o = 1, \dots, \overline{O}_k$.

The knapsack consists of L resources, $\mathcal{L} = \{1, \dots, L\}$, where resource $\ell \in \mathcal{L}$ has a capacity of $c_{\ell,t}$ resource units at time t. We define the column vector $\mathbf{c}_\ell = [c_{\ell,1}, c_{\ell,2}, \dots, c_{\ell,T}]^\intercal$. Let $\mathcal{K}_\ell \subseteq \mathcal{K}$ denote the set of object classes using resource ℓ, $\ell \in \mathcal{L}$. An object of class $k \in \mathcal{K}_\ell$ uses b_k resource units of resource ℓ. We will number the elements of $\mathcal{K}_\ell$ as $\mathcal{K}_\ell(1), \dots, \mathcal{K}_\ell(|\mathcal{K}_\ell|)$. We assume that $\mathcal{K}_\ell \cap \mathcal{K}_{\bar{\ell}} = \emptyset$, $\ell \neq \bar{\ell}$, which means that each object class only uses one resource, and $\cup_\ell \mathcal{K}_\ell = \mathcal{K}$, implying all classes are assigned a resource. Let H_k denote the discrete random holding time during which an object of class k occupies these resource units. We assume that for each $k \in \mathcal{K}$, H_k are independent and identically distributed with finite support $\{1, \dots, \overline{H}_k\}$. Let $p_{H_k}(h) = \mathbb{P}(H_k = h)$, $h = 1, \dots, \overline{H}_k$.

The resource utilization at time t by the number of class-k objects in the knapsack, $u_{k,t} = \sum_{k \in \mathcal{K}_\ell} b_k n_k$, cannot exceed the available resource units at each time t. Thus, at time t the state space for the number n_k of objects of class k, $k \in \mathcal{K}$, is

$$\mathcal{S}_t := \left\{ \mathbf{n} = [n_1, n_2, \dots, n_K]^\intercal \in \mathbb{N}^K \mid \sum_{k \in \mathcal{K}_\ell} b_k n_k \leq c_{\ell,t},\ \ell \in \mathcal{L} \right\}, \tag{1}$$

with $\mathbb{N} = \{0, 1, \dots\}$. The evolution of the stochastic process, $\left\{ \widetilde{U}_{k,t},\ k \in \mathcal{K}, t \in \mathcal{T} \right\}$, recording the number of objects in the system is then determined by the distributions of O_k and H_k, $k \in \mathcal{K}$.

The unconstrained cousin of the stochastic knapsack described above has unlimited capacity, i.e., $c_{\ell,t} = \infty$, $\ell \in \mathcal{L}, t \in \mathcal{T}$. Our goal is to design the cyclic schedule such that the utilization of the resource units in the unconstrained cousin of the stochastic knapsack at state space $\mathcal{S}_t, t \in \mathcal{T}$, is as close as possible to the capacity of the stochastic knapsack as specified in (1). This requires a more detailed description of the state space.

Let $\theta_{k,1} = O_k$ denote the random number of class-k objects arriving to the knapsack upon completion of job k, and $\theta_{k,\tau+1}$ record the random number of class-k objects in the system τ time slots after its arrival time, $\tau = 1, \dots, \overline{H}_k - 1$. Let $\boldsymbol{\theta}_k = [\theta_{k,1},\ \theta_{k,2}, \dots, \theta_{k,\overline{H}_k}]^\intercal \in \mathbb{N}^{\overline{H}_k}$ denote the objects vector of a class k job. As class-k objects depart form the knapsack after completion of their holding time, it is immediate that $\overline{O}_k \geq \theta_{k,1} \geq \theta_{k,2} \geq \dots \geq \theta_{k,\overline{H}_k}$. We then have that $\boldsymbol{\theta}_k \in \Theta_k$, where

$$\begin{aligned} \Theta_k &:= \left\{ \boldsymbol{\theta}_k \in \mathbb{N}^{\overline{H}_k} \mid \theta_{k,1} \leq \overline{O}_k, \theta_{k,2} \leq \theta_{k,1}, \dots, \theta_{k,\overline{H}_k} \leq \theta_{k,\overline{H}_k - 1} \right\} \\ &= \left\{ \boldsymbol{\theta}_k \in \mathbb{N}^{\overline{H}_k} \mid \mathbf{A}_k \boldsymbol{\theta}_k \leq \mathbf{a}_k \right\},\ k \in \mathcal{K}, \end{aligned} \tag{2}$$

where $\mathbf{A}_k$ is a matrix with 1 on the diagonal and -1 on the sub diagonal, and $\mathbf{a}_k = \overline{O}_k \mathbf{e}_1$, where $\mathbf{e}_1$ is the first unit vector, with 1 in position 1, 0 elsewhere. Note that $\mathbf{A}_k$ is totally unimodular (TU) as it corresponds to the node-arc incidence matrix of a directed graph (Conforti et al., 2014, Theorem 4.9).

Let $\xi_\ell = [\boldsymbol{\theta}_{\mathcal{K}_\ell(1)}, \boldsymbol{\theta}_{\mathcal{K}_\ell(2)}, \dots, \boldsymbol{\theta}_{\mathcal{K}_\ell(|\mathcal{K}_\ell|)}]^\intercal$ denote the stacked random vector of $\boldsymbol{\theta}_k$, and $\mathbf{d}_\ell = [\mathbf{a}_{\mathcal{K}_\ell(1)}, \mathbf{a}_{\mathcal{K}_\ell(2)}, \dots, \mathbf{a}_{\mathcal{K}_\ell(|\mathcal{K}_\ell|)}]^\intercal$ the stacked vector of the $\mathbf{a}_k$, $k \in \mathcal{K}_\ell$. Let $\mathbf{D}_\ell$ be the block diagonal matrix with blocks $\mathbf{A}_{\mathcal{K}_\ell(1)}, \mathbf{A}_{\mathcal{K}_\ell(2)}, \dots, \mathbf{A}_{\mathcal{K}_\ell(|\mathcal{K}_\ell|)}$. Note that the matrix $\mathbf{D}_\ell$ consists of TU blocks and therefore is TU. Let ζ denote the stacked vector of ξ_ℓ, $\ell = 1, \dots, L$. We have that $\zeta \in \mathrm{Z}$, where

$$\mathrm{Z} := \left\{ \zeta = [\xi_1^\intercal, \dots, \xi_L^\intercal]^\intercal \in \mathbb{N}^{\sum_{\ell \in \mathcal{L}} \sum_{k \in \mathcal{K}_\ell} \overline{H}_k} \mid \mathbf{D}_\ell \xi_\ell \leq \mathbf{d}_\ell \right\}. \tag{3}$$

Let $U_{\ell,t}$ record the resource utilization of resource ℓ at time t. Then $U_{\ell,t}$ is obtained by adding all objects using resource ℓ at time $t \in \mathcal{T}$ from the current cycle 0, $r^c_{\ell,t}$, and from previous cycles $s \in \{\dots, -2, -1\}$, $r^p_{\ell,t}$. Thus

$$U_{\ell,t} = r^p_{\ell,t} + r^c_{\ell,t}, \tag{4}$$

where $r^c_{\ell,t}$ is obtained by considering all options for class-k jobs in the current schedule that may be present at time t, i.e.,

$$r^c_{\ell,t} = \sum_{k \in \mathcal{K}_\ell} \sum_{m=1}^{M} \sum_{j=1}^{t} x_{k,m,t-j+1} b_k \theta_{k,j}. \tag{5}$$

For previous cycles, we should only consider those cycles from which objects may be present at time t, i.e., if $\overline{H}_k - t > 0$. As a consequence, for class-k objects we must include cycles $s = -\lfloor \frac{\overline{H}_k - t - 1}{T} \rfloor - 1, \dots, -1$. Thus,

$$r^p_{\ell,t} = \sum_{k \in \mathcal{K}_\ell} \mathbb{1}_{\{\overline{H}_k - t > 0\}} \sum_{m=1}^{M} \sum_{i=0}^{\lfloor \frac{\overline{H}_k - t - 1}{T} \rfloor} \sum_{j=1}^{\min\{\overline{H}_k - t - iT, T\}} x_{k,m,T-j+1} b_k \theta_{k,t+j+iT}. \tag{6}$$

For fixed $k \in \mathcal{K}$ and $m \in \mathcal{M}$, each $\theta_{k,t}$, $t = 1, \dots, \overline{H}_k$ appears exactly once in (5), (6). Therefore, it follows that $U_{\ell,t}$ is a linear function of $\boldsymbol{\theta}$ with coefficients depending on b_k and $\mathbf{x}$. We collect the coefficients of each $\theta_{k,t}$, $t = 1, \dots, \overline{H}_k$. To this end, define the vectors

$$\mathbf{f}^c_{k,t}(\mathbf{x}) = \left[\sum_{m=1}^{M} \sum_{j=1}^{t} x_{k,m,t-j+1} \mathbb{1}(t = h),\ h = 1, \dots, \overline{H}_k \right]^\intercal,$$

$$\mathbf{f}^p_{k,t}(\mathbf{x}) = \left[\sum_{m=1}^{M} \sum_{i=0}^{\lfloor \frac{\overline{H}_k - t - 1}{T} \rfloor} \sum_{j=1}^{\min\{\overline{H}_k - t - iT, T\}} x_{k,m,T-j+1} \mathbb{1}(t = h),\ h = 1, \dots, \overline{H}_k \right]^\intercal,$$

$$\mathbf{g}_{\ell,t}(\mathbf{x}) = \left[\mathbf{f}^p_{\mathcal{K}_\ell(k),t}(\mathbf{x})^\intercal + \mathbb{1}_{\{\overline{H}_{\mathcal{K}_\ell(k)} - t > 0\}} \mathbf{f}^c_{\mathcal{K}_\ell(k),t}(\mathbf{x})^\intercal,\ k = \mathcal{K}_\ell(1), \dots, \mathcal{K}_\ell(|\mathcal{K}_\ell|) \right]^\intercal,$$

which allows us to give the following expression for $U_{\ell,t}$:

$$U_{\ell,t} = \sum_{k \in \mathcal{K}_\ell} \left(\mathbf{f}^p_{k,t}(\mathbf{x}) + \mathbb{1}_{\{\overline{H}_k - t > 0\}} \mathbf{f}^c_{k,t}(\mathbf{x}) \right)^\intercal (b_k \boldsymbol{\theta}_k) = \mathbf{g}_{\ell,t}(\mathbf{x})^\intercal (\mathbf{b}_\ell \odot \xi_\ell), \tag{7}$$

where $\mathbf{b}_\ell = \left[b_{\mathcal{K}_\ell(k)} \mathbf{1}_{\overline{H}_{\mathcal{K}_\ell(k)}},\ k = \mathcal{K}_\ell(1), \dots, \mathcal{K}_\ell(|\mathcal{K}_\ell|) \right]^\intercal$, with $\mathbf{1}_n$ the all-ones vector of size n, and $\odot$ denotes the Hadamard product. Let $\mathbf{U}_\ell = \left[U_{\ell,1}, \dots, U_{\ell,T} \right]^\intercal$, and $\mathbf{G}_\ell(\mathbf{x}) = [\mathbf{g}_{\ell,1}(\mathbf{x})^\intercal, \mathbf{g}_{\ell,2}(\mathbf{x})^\intercal, \dots, \mathbf{g}_{\ell,T}(\mathbf{x})^\intercal]^\intercal$, the matrix with rows $\mathbf{g}_{\ell,t}(\mathbf{x})^\intercal$, $t = 1, \dots, T$. This gives

$$\mathbf{U}_\ell = \mathbf{G}_\ell(\mathbf{x})(\mathbf{b}_\ell \odot \xi_\ell). \tag{8}$$

We are now ready to introduce the optimization model which determines the optimal schedule. The model searches for a schedule $\mathbf{x}$ such that over- and under-utilization of the knapsack, penalized by costs p_o and p_u respectively, is minimized. Let $\mathbf{y}_\ell \in \mathbb{R}^T$ denote an analytical variable representing the over or under utilization costs. The Stochastic Knapsack with Periodic Scheduled Arrivals (SKPSA) is formulated as follows:

$$\text{(SKPSA)} \quad \min_{\mathbf{x}, \mathbf{y}_1, \dots, \mathbf{y}_L} \sum_{\ell=1}^{L} \mathbf{y}_\ell^\intercal \mathbf{1}_T \tag{9a}$$

$$\text{s.t.} \quad p_o \left(\mathbf{c}_\ell - \mathbf{G}_\ell(\mathbf{x})(\mathbf{b}_\ell \odot \xi_\ell) \right) \leq \mathbf{y}_\ell, \qquad \ell \in \mathcal{L}, \tag{9b}$$

$$p_u \left(\mathbf{G}_\ell(\mathbf{x})(\mathbf{b}_\ell \odot \xi_\ell) - \mathbf{c}_\ell \right) \leq \mathbf{y}_\ell, \qquad \ell \in \mathcal{L}, \tag{9c}$$

$$\mathbf{x} \in \mathcal{X}, \tag{9d}$$

where

$$\mathcal{X} := \left\{ \mathbf{x} \in \{0, 1\}^{KMT} \mid \sum_{k \in \mathcal{K}} x_{k,m,t} \leq 1, m \in \mathcal{M}, t \in \mathcal{T} \right\}. \tag{10}$$

From Constraints (9b) and (9c) the under and over utilization of the knapsack's resources is obtained, which is in turn minimized by the objective function (9a). Note that $\mathcal{X}$ could be more restricted depending on the application. If, for instance, scheduling jobs on times $t \in \mathcal{T}$ of the cycle is prohibited, the set of feasible schedules would be smaller.

By analogy with van Essen et al. (2014), we can show that the SKPSA is NP-Hard by a reduction from 3-Partition. To this end, let $M = 3$, $L = 1$, and let the holding time be 1 for all jobs. Let $K = 3T$. Each job of class k is assumed to have a known outflow of o_k objects. Let w be an integer such that $\sum_{k=1}^{K} o_k = Tw$. Then determining whether there exists a schedule that requires p resource units is equivalent to determining whether there are T disjoint subsets $\overline{\mathcal{K}}_t \subset \{1, \dots, 3T\}$ such that $\sum_{k \in \overline{\mathcal{K}}_t} o_k = rw$. This is known as the 3-Partition problem, which is NP-Hard (Garey and Johnson, 1979).

3. Distributionally robust stochastic knapsack with periodic scheduled arrivals

This section presents the proposed DRO-SKPSA approximation of the SKPSA presented in Section 2. In Section 3.1 we provide a brief introduction to the DRO paradigm and how this is applied to our model. In Section 3.2 we provide a column-and-constraint generation based algorithm to solve our model efficiently.

3.1. DRO approximation of the SKPSA with wasserstein distance based ambiguity

Let $\mathcal{M}(\mathrm{Z})$ the probability space containing all distributions supported on Z. Let ζ', ζ'' be random variables with support Z. The Wasserstein distance (Kantorovich and Rubinstein, 1958) is a distance measure between probability distributions $\mathbb{P}$ and $\mathbb{Q}$, defined as $d_w : \mathcal{M}(\mathrm{Z}) \times \mathcal{M}(\mathrm{Z}) \mapsto \mathbb{R}_+$ with

$$d_w(\mathbb{P},\mathbb{Q}) := \inf \left\{ \int_{\mathrm{Z}\times\mathrm{Z}} \|\zeta' - \zeta''\|_p \Pi(d\zeta', d\zeta'') \;\middle|\; \begin{array}{l} \Pi \text{ is a joint} \\ \text{distribution of} \\ \zeta', \zeta'', \text{ with} \\ \text{marginals } \mathbb{P}, \mathbb{Q} \end{array} \right\},$$

where $\|\cdot\|_p$ is an arbitrary norm. The Wasserstein distance can be interpreted as the costs of moving "mass" from distribution $\mathbb{P}$ to distribution $\mathbb{Q}$, where the weights or costs are encoded by the norm $\|\cdot\|_p$. From now on, we assume that $p = \infty$.

The Wasserstein ambiguity set of a distribution $\mathbb{P}$ is defined as the set of all probability distributions in the Wasserstein ball around $\mathbb{P}$ (Esfahani and Kuhn, 2018):

$$\mathcal{P}(\epsilon, \mathrm{Z}) = \{\mathbb{Q} : \mathbb{Q}(\zeta \in \mathrm{Z}) = 1\} \cap \left\{\mathbb{Q} : d_w(\mathbb{P},\mathbb{Q}) \leq \epsilon\right\}, \tag{11}$$

where $\epsilon \in \mathbb{R}_+$ controls the radius of the Wasserstein ball.

Let $\hat{\mathrm{Z}} = \left\{\hat{\zeta}_1, \hat{\zeta}_2, \dots, \hat{\zeta}_N\right\}$, with $\hat{\zeta}_n = [\hat{\xi}_{1,n}, \dots, \hat{\xi}_{L,n}]$ a finite set of sample points from Z, indexed by $\mathcal{N} = \{1,\dots,N\}$. We now define

$$\hat{\mathbb{P}}_N(A) := \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}\left(\hat{\zeta}_n \in A\right), \quad \forall A \in \mathcal{F}, \tag{12}$$

where $\mathcal{F}$ is a Borel σ−algebra on Z. The Wasserstein ambiguity set $\mathcal{P}(\epsilon, \mathrm{Z})$ centered at $\hat{\mathbb{P}}_N$ is referred to as the data-driven Wasserstein ambiguity set (Mevissen et al., 2013), in which case $\mathcal{P}(\epsilon, \mathrm{Z})$ contains all distributions $\mathbb{Q}$ with $d_w(\hat{\mathbb{P}}_N, \mathbb{Q}) \leq \epsilon$, hence in loose terms all distributions "close" to the empirical distribution. In the DRO paradigm, we typically consider a problem of the form Esfahani and Kuhn (2018):

$$\min_{\mathbf{x}\in\mathcal{X}} \sup_{\mathbb{Q}\in\mathcal{P}(\epsilon,\mathrm{Z})} \mathbb{E}_{\zeta\sim\mathbb{Q}}\left(\phi(\mathbf{x},\zeta)\right), \tag{13}$$

where $\phi : \mathcal{X}, \mathrm{Z} \mapsto \mathbb{R}$ is commonly referred to as loss function.

We will now proceed with the DRO approximation of the SKPSA model with a data-driven Wasserstein ambiguity set. In the SKPSA, the only uncertain parameter is ζ. Therefore, the Distributionally Robust Stochastic Knapsack with Periodic Scheduled Arrivals (DRO-SKPSA) may be formulated as follows:

$$\text{(DRO-SKPSA)} \quad \min_{\mathbf{x}} \sup_{\mathbb{Q}\in\mathcal{P}(\epsilon,\mathrm{Z})} \mathbb{E}_{\zeta\sim\mathbb{Q}}\left(\min_{\mathbf{y}_1,\dots,\mathbf{y}_\ell} \sum_{\ell=1}^{L} \mathbf{y}_\ell^{\intercal}\mathbf{1}_T\right) \tag{14a}$$

$$\text{s.t.} \quad p_o\left(\mathbf{c}_\ell - \mathbf{G}_\ell(\mathbf{x})(\mathbf{b}_\ell \odot \xi_\ell)\right) \leq \mathbf{y}_\ell, \qquad \ell \in \mathcal{L}, \tag{14b}$$

$$p_u\left(\mathbf{G}_\ell(\mathbf{x})(\mathbf{b}_\ell \odot \xi_\ell) - \mathbf{c}_\ell\right) \leq \mathbf{y}_\ell, \qquad \ell \in \mathcal{L}, \tag{14c}$$

$$\mathbf{x} \in \mathcal{X}. \tag{14d}$$

Observe that all feasible $\mathbf{x}$ for the DRO-SKPSA are feasible for the SKPSA. Model (14) is not solvable in its current form. The next theorem provides a solvable reformulation of (14), and relies on (Esfahani and Kuhn, 2018, Corollary 5.4-(ii)). We chose to prove the required conditions to apply the corollary instead of copying the full proof.

Theorem 1. *If $\mathcal{P}(\epsilon, \mathrm{Z})$ is a Wasserstein ambiguity set, the DRO-SKPSA is equivalent to:*

$$\min_{\substack{\mathbf{x},\ \lambda\geq 0 \\ \mu_\ell, \gamma_{\ell,n,v}\geq 0}} \quad \lambda\epsilon + \frac{1}{N}\sum_{n=1}^{N}\mu_n \tag{15a}$$

$$\text{s.t.} \quad \sum_{\ell=1}^{L}\Big((\mathbf{v}_{\ell,v})^{\intercal}(\overline{\mathbf{G}}_\ell(\mathbf{x})(\mathbf{b}_\ell \odot \hat{\xi}_{\ell,n}) + \bar{\mathbf{c}}) + (\gamma_{\ell,n,v})^{\intercal}(\mathbf{d}_\ell - \mathbf{D}_\ell\hat{\xi}_{\ell,n})\Big) \leq \mu_n, \qquad \forall n \in \mathcal{N}, v \in \mathcal{I}, \tag{15b}$$

$$\sum_{\ell=1}^{L}\left(\|(\mathbf{D}_\ell)^{\intercal}\gamma_{\ell,n,v} - (\overline{\mathbf{G}}_\ell(\mathbf{x}))^{\intercal}\mathbf{v}_{\ell,v}\|_1\right) \leq \lambda, \quad \forall n \in \mathcal{N}, v \in \mathcal{I}, \tag{15c}$$

$$\mathbf{x} \in \mathcal{X}, \tag{15d}$$

where

$$\overline{\mathbf{G}}_\ell(\mathbf{x}) = \left[\mathbf{G}_\ell(\mathbf{x})^{\intercal}, -\mathbf{G}_\ell(\mathbf{x})^{\intercal}\right]^{\intercal}, \quad \bar{\mathbf{c}} = \left[-\mathbf{c}_\ell, \mathbf{c}_\ell\right],$$

and $\mu_\ell \in \mathbb{R}^N, \lambda \in \mathbb{R}_+, \gamma_{\ell,n,v} \in \mathbb{R}_+^{\sum_{k\in\mathcal{K}_\ell}\overline{H}_k}$.

Proof. We reformulate (14) into the following two-stage program with right-hand side uncertainty:

$$\min_{\mathbf{x}} \sup_{\mathbb{Q}\in\mathcal{P}(\epsilon,\mathrm{Z})} \mathbb{E}_{\zeta\sim\mathbb{Q}}\left(\sum_{\ell=1}^{L}\phi_\ell(\mathbf{x},\mathbf{y}_\ell,\xi_\ell)\right) \tag{16a}$$

$$\text{s.t.} \quad \mathbf{x} \in \mathcal{X}, \tag{16b}$$

with loss function

$$\phi_\ell(\mathbf{x},\mathbf{y}_\ell,\xi_\ell) := \min_{\mathbf{y}_\ell}\left\{\mathbf{y}_\ell^{\intercal}\mathbf{1}_T \;\middle|\; \begin{array}{l} p_u\left(\mathbf{G}_\ell(\mathbf{x})(\mathbf{b}_\ell \odot \xi_\ell) - \mathbf{c}_\ell\right) \leq \mathbf{y}_\ell, \\ p_o\left(\mathbf{c}_\ell - \mathbf{G}_\ell(\mathbf{x})(\mathbf{b}_\ell \odot \xi_\ell)\right) \leq \mathbf{y}_\ell \end{array}\right\}. \tag{17}$$

Let $\mathcal{W}$ denote the dual feasible set of (17) with vertices $\mathbf{w}_1,\dots,\mathbf{w}_{2^T}$, that is:

$$\mathcal{W} := \left\{\mathbf{w} \mid \left[\mathbf{I}_T, \mathbf{I}_T\right]\mathbf{w} = \mathbf{1}_{2T}, \mathbf{w} \geq 0\right\}, \tag{18}$$

with matrix $\left[\mathbf{I}_T, \mathbf{I}_T\right]$ consisting of two identity matrices of size T, stacked after one another. Note that $\mathcal{W}$ is identical for each $\ell \in \mathcal{L}$. We readily obtain that $\mathcal{W}$ is nonempty and compact, and that the number of vertices is finite.

Let $\mathcal{V}$ denote the dual feasible set of $\sum_{\ell=1}^{L}\phi_\ell(\mathbf{x},\mathbf{y}_\ell,\xi_\ell)$, with vertices $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{2^{LT}}$, indexed by $\mathcal{I} = \{1,\dots,2^{LT}\}$, that is

$$\mathcal{V} := \left\{\mathbf{v} = [\mathbf{w}_1^{\intercal}, \mathbf{w}_2^{\intercal}, \dots, \mathbf{w}_L^{\intercal}]^{\intercal} \mid \mathbf{w}_\ell \in \mathcal{W}, \ell = 1,\dots,L\right\}. \tag{19}$$

For $v \in \mathcal{I}$, $\ell \in \mathcal{L}$, let $\mathbf{v}_{\ell,v}$, denote the ℓ-th subvector $\mathbf{w}_\ell$ of $\mathbf{v}_v$.

The proof is completed following the proof of Esfahani and Kuhn (2018, Corollary 5.4-(ii)), observing that $\mathbf{D}_\ell$ is totally unimodular for all ℓ and that for $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ we have that $\|[\mathbf{x}_1^{\intercal}, \mathbf{x}_2^{\intercal}]^{\intercal}\|_1 = \|\mathbf{x}_1^{\intercal}\|_1 + \|\mathbf{x}_2^{\intercal}\|_1$. □

An important benefit from Wasserstein based DRO is that it generalizes an SAA model and an expected value approach. If we set $\epsilon = 0$ the SAA model is a special case of the DRO-SKPSA (Esfahani and Kuhn, 2018). Furthermore, if we use $\bar{\zeta}_\ell = \frac{1}{N}\sum_{n=1}^{N}\hat{\zeta}_{\ell,n}$, an expected value approach is derived where we only use averages as another special case of the DRO-SKPSA.

Remark 1 (*Separate Ambiguity Sets for Each Resource*)**.** A generalization of our formulation of the DRO-SKPSA is to model a separate ambiguity set $\mathcal{P}_\ell$ for each resource $\ell \in \mathcal{L}$, by which we could tune the Wasserstein radius ϵ for each resource independently reflecting the degree of confidence in the empirical data for each resource separately. This would, however, result in having to tune L hyper parameters ϵ_ℓ, which might be a cumbersome task.

3.2. Column-and-constraint generation algorithm

The DRO-SKPSA (15) is a Mixed Integer Linear Program (MILP), and can therefore be solved using off the shelve solvers. However, the problem grows exponentially in terms of constraints and variables. In this section, we present an algorithm exploiting the structure of our problem.

Let $\mathcal{V}$ and $\mathcal{I}$ be as defined in Theorem 1. Observe that the number of constraints and variables of (15) grows exponentially as $|\mathcal{I}| = 2^{LT}$. We will present an algorithm that solves (15) by only considering subsets of these constraints by iteratively adding them. The algorithm we present is derived from results in Saif and Delage (2021), who encounter the same challenge in terms of model growth after applying the tractability results as presented in Esfahani and Kuhn (2018).

We first reformulate the DRO-SKPSA as follows:

$$\min_{\substack{\mathbf{x},\ \lambda\geq 0\\ \mu_\ell,\gamma_{\ell,n,v}\geq 0}} \quad \lambda\epsilon + \frac{1}{N}\sum_{n=1}^{N}\mu_n \tag{20a}$$

$$\text{s.t.} \quad h_n(\mathbf{x},\lambda,\mathbf{v}) \leq \mu_n, \qquad n\in\mathcal{N},\ \mathbf{v}\in\mathcal{V}, \tag{20b}$$

$$\mathbf{x}\in\mathcal{X}, \tag{20c}$$

with $h_n : \mathcal{X},\mathbb{R},\mathcal{V}\mapsto\mathbb{R}$ defined as

$$h_n(\mathbf{x},\lambda,\mathbf{v}_{\ell,v}) := \min_{\gamma_{\ell,n,v}\geq 0}\sum_{\ell=1}^{L}\left((\mathbf{v}_{\ell,v})^\intercal(\overline{\mathbf{G}}_\ell(\mathbf{x})\hat{\xi}_{\ell,n} + \bar{\mathbf{c}}) + (\boldsymbol{\gamma}_{\ell,n,v})^\intercal(\mathbf{d}_\ell - \mathbf{D}_\ell\hat{\xi}_{\ell,n})\right) \tag{21a}$$

$$\text{s.t.} \quad \sum_{\ell=1}^{L}\left(\|(\mathbf{D}_\ell)^\intercal\boldsymbol{\gamma}_{\ell,n,v} - (\overline{\mathbf{G}}_\ell(\mathbf{x}))^\intercal\mathbf{v}_{\ell,v}\|_1\right) \leq \lambda. \tag{21b}$$

The value of $h_n(\mathbf{x},\lambda,\mathbf{v}_{\ell,v})$ is readily obtained as for fixed $\mathbf{x}$, (21) is a Linear Program.

We are now ready to solve (20),(21) by iteratively adding constraints (20b). To this end, let $\mathcal{I}_n \subseteq \mathcal{I},\ n\in\mathcal{N}$, denote (small) subsets of $\mathcal{I}$ obtained from Theorem 2, where each $\mathcal{I}_n$ corresponds to the subset of constraints (20b) which are indexed by n. Let $opt_r : \mathcal{I}_1,\dots,\mathcal{I}_N \mapsto \mathbb{R}$ denote the optimal value of the relaxation of the DRO-SKPSA obtained by considering constraints (20b) for which $v\in\mathcal{I}_n, n\in\mathcal{N}$. Let opt^* denote the optimal value of the DRO-SKPSA (20). The following theorem provides a lower and upper bound on $opt(\mathcal{I}_1,\dots,\mathcal{I}_n)$, and which indices of vertices to add to each $\mathcal{I}_n$.

Theorem 2. *Let $(\bar{\mathbf{x}},\bar{\lambda},\bar{\mu}_n)$ denote an incumbent solution obtained from solving the DRO-SKPSA (20) with only constraints (20b) for which $v\in\mathcal{I}_n, n\in\mathcal{N}$. Then*

$$opt_r\left(\mathcal{I}_1,\dots,\mathcal{I}_N\right) \leq opt_r\left(\mathcal{I}_1\cup\{v_1\},\dots,\mathcal{I}_N\cup\{v_N\}\right) \leq opt^* \leq \bar{\lambda}\epsilon + \frac{1}{N}\sum_{n=1}^{N}\mu_n^*(\bar{\mathbf{x}},\bar{\lambda}), \tag{22}$$

with

$$\mu_n^*(\bar{\mathbf{x}},\bar{\lambda}) = \max_{v\in\mathcal{I}_n} h_n(\mathbf{x},\lambda,\mathbf{v}_{\ell,v}), \quad v_{\ell,n} = \arg\max_{v\in\mathcal{I}_n} h_n(\mathbf{x},\lambda,\mathbf{v}_{\ell,v}). \tag{23}$$

Proof. The first and second inequality are trivial, as the first and second quantities are obtained from solving a relaxation of the DRO-SKPSA. The third inequality follows from the fact that $\bar{\mathbf{x}}$ and $\bar{\lambda}$ are both feasible for the DRO-SKPSA, and that for any feasible μ_n for the DRO-SKPSA, we have that $\mu_n \leq \mu_n^*(\bar{\mathbf{x}},\bar{\lambda})$. □

We are now ready to introduce a column-and-constraint algorithm to solve the DRO-SKPSA.

Algorithm 1 CCG Algorithm to solve DRO-SKPSA

1: **input** A feasible $\bar{\mathbf{x}}$, sample data $\hat{\zeta}_1,\hat{\zeta}_2,\dots,\hat{\zeta}_N$, Wasserstein radius $\epsilon\geq 0$, time limit T_{lim}, $\kappa\in[0,1]$.
2: **output** A solution $\mathbf{x}\in\mathcal{X}$, optimality gap.
3: **initialize** $\mathcal{I}_n \leftarrow \emptyset\ \forall n\in\mathcal{N}$, LB $\leftarrow -\infty$ and UB $\leftarrow \infty$, Gap $\leftarrow \infty$
4: **while** `CurrentTime` $< T_{lim}$ **and** Gap $< \kappa$ **do**
5:     **for** $n\in\mathcal{N}$ **do**
6:         Find a new $\mu_n^*(\bar{\mathbf{x}},\bar{\lambda})$ and v_n from (23) .
7:         Let $\mathcal{I}_n \leftarrow \mathcal{I}_n\cup\{v_n\}$.
8:     UB $\leftarrow \min\left\{\text{UB},\ \bar{\lambda}\epsilon + \frac{1}{N}\sum_{n=1}^{N}\mu_n^*(\bar{\mathbf{x}},\bar{\lambda})\right\}$.
9:     Solve DRO-SKPSA (20) with constraints for which $v\in\mathcal{I}_n$ to obtain a new $\bar{\mathbf{x}}$ and $\bar{\lambda}$, and $opt_r\left(\mathcal{I}_1,\dots,\mathcal{I}_N\right)$.
10:     LB $\leftarrow opt_r(\mathcal{I}_1,\dots,\mathcal{I}_N)$.
11:     Gap $\leftarrow \frac{\text{UB}-\text{LB}}{\text{UB}}$.
12: **end**

The algorithm is initialized with any feasible $\mathbf{x}\in\mathcal{X}$, sample data $\hat{\zeta}_1,\hat{\zeta}_2,\dots,\hat{\zeta}_N$, a Wasserstein radius ϵ and empty vertex sets $\mathcal{I}_n$. The algorithm iterates until it exceeds the given time limit T_{lim}, or until the relative optimality gap is smaller than κ. In each iteration, the algorithm obtains a new $\mu_n^*(\bar{\mathbf{x}},\bar{\lambda})$ and v_n from (23) in Step 6. If $\mu_n^*(\bar{\mathbf{x}},\bar{\lambda}) > \mu_n(\bar{\mathbf{x}},\bar{\lambda})$, the algorithm adds v_n to $\mathcal{I}_n$. If $\mu_n^*(\bar{\mathbf{x}},\bar{\lambda}) \leq \bar{\mu}_n(\bar{\mathbf{x}},\bar{\lambda})\ \forall v\in\mathcal{I}$, i.e., there is no vertex to add, we declare the current solution optimal. Index v_n is added to $\mathcal{I}_n$ in Step 7. In Step 8, the upper bound is updated using (22). In Step 9 we solve a relaxed DRO-SKPSA (20), by only considering constraints (20b) for $v\in\mathcal{I}_n$ to obtain a new incumbent solution $\bar{\mathbf{x}}$ and $\bar{\lambda}$ and $opt_r(\mathcal{I}_1,\dots,\mathcal{I}_N)$. The latter is used in Step 10 to update the lower bound. The iteration is concluded by updating the optimality gap in Step 11. As we are iteratively converging to the full model, and stop if no other vertex can be found in Step 6, the algorithm is guaranteed to converge. Observe that we can exploit parallel computing to find $\mu_n^*(\bar{\mathbf{x}},\bar{\lambda})$, which we found to work for instances of a reasonable size. For larger instances, we refer to a KKT based approach as presented in Saif and Delage (2021).

4. Numerical results

This section presents numerical results for several instances of the SKPSA. The results in Section 4.1 are based on a theoretical instance, whereas the results in Section 4.2 are obtained using an instance generator of the MSSP. For a definition of the MSSP see, e.g., Santos and Marques (2021). For both instance types, we solve the DRO-SKPSA using Algorithm 1. As the performance of Algorithm 1 is analyzed in Saif and Delage (2021), experiments with the sole purpose investigating the performance of Algorithm 1 are excluded. In all experiments, we aim to find the right value of the Wasserstein radius ϵ. The Wasserstein radius ϵ has to be chosen by the decision maker, which can be done using cross validation as suggested in Esfahani and Kuhn (2018). Note that for all instances we solve the DRO-SKPSA for $\epsilon = 0$, so that we also obtain the results for an SAA model. All results were obtained using a computing cluster with 1TB of memory, 32 CPUs and Gurobi 10.0.0 (Gurobi Optimization, 2022) as solver.

4.1. Results based on theoretical instances

In Section 4.1.1 we discuss the experimental setup for the theoretical instance of the SKPSA. In Section 4.1.2 we discuss the performance of the DRO-SKPSA approximation.

4.1.1. Set up of the SKPSA instance

We consider a cyclic schedule of $T = 7$ time units (days), in which jobs of $K = 16$ classes have to be scheduled on $M = 4$ machines. Each job generates at most $\overline{O}_k = 5$ objects. The knapsack consists of $L = 1$ resource, with capacity $c_1 = c_2 = \cdots = c_5 = 20$, and $c_6 = c_7 = 15$, to represent a capacity difference between week and weekend days.

Table 1
Beta-binomial distributions $i = 1, \dots, 4$.

	Beta-binomial distribution i			
	$i = 1$	$i = 2$	$i = 3$	$i = 4$
$\alpha_i = \beta_i$	100	10	1	0.10
Mean	2.50	2.50	2.50	2.50
Variance	1.27	1.49	2.92	5.41

Table 2
Beta-binomial distribution for each H_k, O_k, $k = 1, \dots, 16$.

k	1	2	3	4	5	6	7	8
O_k	$i = 1$	$i = 1$	$i = 1$	$i = 1$	$i = 2$	$i = 2$	$i = 2$	$i = 2$
H_k	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 1$	$j = 2$	$j = 3$	$j = 4$
k	9	10	11	12	13	14	15	16
O_k	$i = 3$	$i = 3$	$i = 3$	$i = 3$	$i = 4$	$i = 4$	$i = 4$	$i = 4$
H_k	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 1$	$j = 2$	$j = 3$	$j = 4$

We let $\overline{H}_k = 5$, and let under and over utilization of the knapsack be weighted equally by setting $p_u = p_o = 1$. The set of feasible schedules $\mathcal{X}$, see (10), consists of all schedules subject to three constraints. First, scheduling multiple jobs on one machine at the same time is forbidden; second, each job can be scheduled only once; third, jobs cannot be scheduled during the weekend corresponding to $t = 6$ or $t = 7$. Besides, we add symmetry breaking constraints as presented in, e.g., Santos and Marques (2021), which restricts the search space by eliminating symmetric schedules. This means that for this set of experiments $\mathcal{X}$ is defined as:

$$\mathcal{X} = \left\{ \mathbf{x} \in \{0,1\}^{KMT} \left| \begin{array}{l} \sum_{k\in\mathcal{K}} x_{k,m,t} \leq 1, m \in \mathcal{M}, t \in \mathcal{T}, \\ \sum_{k\in\mathcal{K}} \sum_{m\in\mathcal{M}} x_{k,m,t} = 0, \ t = 6,7, \\ \sum_{m\in\mathcal{M}} \sum_{t\in\mathcal{T}} x_{k,m,t} \leq 1, \ k \in \mathcal{K}, \\ x_{\overline{k},m,t} \leq \sum_{\left\{k\in\mathcal{K} | k \geq \overline{k}\right\}} x_{k,m,t}, \ \overline{k} \in \mathcal{K}, m \in \mathcal{M}, t \in \mathcal{T} \end{array} \right. \right\}. \tag{24}$$

We introduce four beta-binomial distributions $BetaBin_i(n, \alpha_i, \beta_i)$, $i = 1, \dots, 4$, with parameters α_i and β_i and support $0, \dots, n$. The mean of a beta-binomial distribution is given by $\frac{n\alpha_i}{\beta_i+\alpha_i}$ and the variance is given by $\frac{n\alpha_i\beta_i(\alpha_i+\beta_i+n)}{(\alpha_i+\beta_i)^2(\alpha_i+\beta_i+1)}$. We let $\alpha_i = \beta_i$ for all four distributions and $n = 5$, so that each beta-binomial distribution has equal mean of 2.50. The parameters of each distribution i are given in Table 1. We may now obtain 16 job classes combining for each job class $K = 1, \dots, 16$, a beta-binomial distribution $BetaBin_i$ for O_k and $BetaBin_j$ for H_k as shown in Table 2.

4.1.2. Computational results of the DRO-SKPSA approximation

In this section we present the results of the DRO-SKPSA algorithm on the theoretical instances based on the beta-binomial distributions. We obtain the solution for $\epsilon \in \{0, 0.1, \dots, 1\}$ using $N = 100$ samples, which we investigated to be a sufficiently high number, meaning that letting $N > 100$ does not change the solution. We allowed for a solving time of at most 10 h per instance. Table 3 presents the objective value, optimality gap, and solving time. It shows that Algorithm 1 is able to find close to optimal solutions with a largest optimality gap of 5.9%. The optimality gap is the largest for small values of ϵ. Note that the results in the row with $\epsilon = 0$ correspond to the SAA solution. The objective value increases as ϵ increases, which is because of a larger Wasserstein ball contains more distributions to account for. The objective values for $\epsilon = 0.1, \dots, 1$, presented in Table 3, are likely to be too conservative, as they are based on the worst case distribution

Table 3
Computational performance based on the theoretical instance.

ϵ	Objective value	Gap	Runtime
0.0	23.1	0.4%	11 049
0.1	31.8	5.9%	34 510
0.2	40.1	5.3%	34 509
0.3	47.7	2.3%	34 508
0.4	55.9	2.9%	34 507
0.5	63.4	2.0%	34 509
0.6	71.2	1.7%	34 508
0.7	79.0	1.5%	34 508
0.8	86.0	0.9%	34 507
0.9	93.8	1.5%	34 507
1.0	100.4	0.9%	34 507

of ξ_ℓ in $\mathcal{P}(\epsilon, \mathrm{Z})$. The objective value for $\epsilon = 0$, however, is likely to be too optimistic, because SP and therefore SAA suffers from out-of-sample disappointment, see Section 1. To choose ϵ, we need the out-of-sample performance of the solutions, as suggested by Esfahani and Kuhn (2018). We simulate the performance of these solutions using new, unseen data, to be able to predict the out-of-sample performance of the solutions and to provide the right value of ϵ.

Table 4 presents the out-of-sample results, based on 25 replications of 5 week simulations. These numbers are chosen such the confidence intervals do not get too narrow. The warm up period is 3 weeks, which, because the maximum holding time is 5 days, is a sufficiently long warm up period. Table 4 shows that the mean total costs over all replications do not significantly differ up to $\epsilon = 0.8$, and that the total mean costs are much lower than the reported objective values in Table 3. The mean overutilization costs however decrease significantly as ϵ increases, while the mean underutilization increase. This skewness is because of ξ_ℓ being from below and that therefore the worst case distribution in the Wasserstein ball rather causes overutilization than underutilization of capacity. The more robust solution thus tends to protect against overutilization of capacity. From a managerial point of view decision makers are likely to prefer underutilization of capacity over overutilization of capacity. Therefore, a higher value of ϵ might be favorable, since the total costs are not likely to be much higher. From this point of view, the best value of ϵ for this instance would therefore be $\epsilon = 0.6$. Also note the fact that the objective value of the solution of $\epsilon = 0$ in Table 3 is lower than the corresponding mean out-of-sample costs. The difference between these two quantities is the out-of-sample disappointment. Concluding, for this instance distributionally robust solutions protect against capacity overutilization without a significant increase of the total costs and do not suffer from out-of-sample disappointment.

4.2. Results for a master surgery scheduling problem

Our work is motivated by the Master Surgical Schedule Problem (MSSP) with downstream bed constraints, where surgical specialty sessions are to be scheduled in a cyclic MSS dictating the assignment of surgical sessions to shifts, where a shift is a combination of a day (part) and an OR. The MSS should be such that the resulting bed occupancy does not exceed ward capacity limitations too often. The MSSP can be modeled as a stochastic knapsack with scheduled arrivals by considering specialty sessions as jobs, and wards as knapsack resources. Each bed on a ward then corresponds to a resource unit. After completion of each job, a random number of patients is transported to the wards and is discharged after completion of a random length of stay.

We consider 5 ORs and 2 shifts per day, thus $M = 10$. The cycle length is $T = 7$ time units (days). We consider $S = 12$ specialties. The available bed capacity is $c_{\ell,t} = 20$, $\ell \in \mathcal{L}$, $t \in \mathcal{T}$. As in Section 4.1.2, we assume that $p_o = p_u = 1$. This corresponds to the setting of a medium sized hospital in The Netherlands. In the MSSP, specialties usually have to be scheduled multiple times per cycle. Therefore, one

Table 4
Out-of-sample performance of the theoretical instance.

ϵ	Mean total costs	95%-CI	Mean under utilization costs	95%-CI	Mean over utilization costs	95%-CI
0	27.8	(25.3, 30.2)	18.1	(16.0, 20.2)	9.5	(8.3, 10.8)
0.1	24.9	(22.7, 27.1)	15.5	(14.1, 17.0)	9.3	(8.5, 10.1)
0.2	27.5	(25.1, 29.8)	16.3	(14.8, 17.9)	11.1	(10.3, 11.9)
0.3	27.4	(24.9, 29.8)	18.4	(16.6, 20.2)	8.8	(7.4, 10.2)
0.4	26.2	(23.9, 28.6)	19.9	(18.1, 21.7)	6.2	(5.3, 7.0)
0.5	25.5	(23.2, 27.9)	18.7	(16.9, 20.5)	6.9	(6.5, 7.3)
0.6	24.0	(21.8, 26.3)	19.1	(17.4, 20.8)	5.0	(4.1, 6.0)
0.7	28.5	(26.1, 31.0)	25.0	(22.8, 27.1)	3.3	(3.0, 3.7)
0.8	29.9	(27.3, 32.5)	26.5	(24.1, 28.9)	3.4	(3.1, 3.6)
0.9	32.7	(30.0, 35.5)	30.0	(27.4, 32.6)	2.6	(2.1, 3.1)
1.0	30.5	(27.9, 33.1)	27.9	(25.6, 30.3)	2.5	(2.3, 2.7)

specialty maps to multiple jobs. Let $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_S\}$ denote the set of specialties. We define a demand δ_σ, and availability α_σ: specialty σ has to be scheduled at least δ_σ times per cycle, and at most α_σ times per cycle. Let $K = \sum_{\sigma\in\Sigma} \alpha_\sigma$, and denote $\mathcal{K}_\sigma \subseteq \mathcal{K}, \sigma \in \Sigma$, as the set of sessions belonging to specialty σ. In this context, O_k models the session outflow, and H_k the length of stay. All patients occupy one bed, thus $b_k = 1,\ k \in \mathcal{K}$, and all patients are routed to a single ward, hence $L = 1$. Compared to (24), we add the following constraints to account for the demand and availability of specialties:

$$\delta_\sigma \leq \sum_{m\in\mathcal{M}} \sum_{t\in\mathcal{T}} x_{k,m,t} \leq \alpha_\sigma,\ \ k \in \mathcal{K}_\sigma, \sigma \in \Sigma,$$

$$\sum_{m\in\mathcal{M}} x_{k,m,t} \leq \sum_{m\in\mathcal{M}} x_{k+1,m,t},\ \ k = 1, \dots, |\mathcal{K}_\sigma| - 1, \sigma \in \Sigma, t \in \mathcal{T}$$

The first constraint ensures that all specialties are planned between δ_σ and α_σ times per cycle. The second constraint ensures that, as surgical sessions of the same specialties are assumed to be independent, for fixed $\sigma \in \Sigma$, session $k+1 \in \mathcal{K}_\sigma$ can only be scheduled if session $k \in \mathcal{K}_\sigma$ is scheduled. This results in the following set of feasible schedules for the MSSP based experiments:

$$\mathcal{X}_{MSSP} = \left\{ \mathbf{x} \in \{0,1\}^{KMT} \;\middle|\; \begin{array}{l} \sum_{k\in\mathcal{K}} x_{k,m,t} \leq 1,\ \ m \in \mathcal{M}, t \in \mathcal{T}, \\ \delta_\sigma \leq \sum_{m\in\mathcal{M}} \sum_{t\in\mathcal{T}} x_{k,m,t} \leq \alpha_\sigma,\ \ \forall k \in \mathcal{K}_\sigma, \forall \sigma \in \Sigma, \\ \sum_{k\in\mathcal{K}} \sum_{m\in\mathcal{M}} x_{k,m,t} = 0,\ \ t = 6, 7, \\ \sum_{m\in\mathcal{M}} \sum_{t\in\mathcal{T}} x_{k,m,t} \leq 1,\ \ k \in \mathcal{K}, \\ x_{\bar{k},m,t} \leq \sum_{\{k\in\mathcal{K} | k \geq \bar{k}\}} x_{k,m,t},\ \ \bar{k} \in \mathcal{K}, m \in \mathcal{M}, t \in \mathcal{T}, \\ \sum_{m\in\mathcal{M}} x_{k,m,t} \leq \sum_{m\in\mathcal{M}} x_{k+1,m,t},\ \ k = 1, \dots, |\mathcal{K}_\sigma| - 1, \\ \qquad\qquad\qquad \sigma \in \Sigma, t \in \mathcal{T} \end{array} \right\}. \tag{25}$$

Instances are generated by a modified version of the instance generator presented in Santos and Marques (2021). We modified the aforementioned instance generator such that the available capacity is close to the required capacity, to give a realistic view of the robustness of the solutions. We let O_k be a truncated Poisson distributed random variable, and H_k a truncated geometrically distributed random variable, as is done in Santos and Marques (2021) for their MSSP instances. The maximum session outflow $\overline{O}_k$ and patient' length of stay $\overline{H}_k$ are instance dependent. For each instance, the demand δ_σ, the availability α_σ and the parameters of the outflow and holding time distributions differ through parameters η_σ and ψ_σ, which as in Santos and Marques (2021), are called the activity level and reference percentage, respectively. To prevent that two specialties with the same activity level have exactly the same probability distributions for the length of stay and outflow, we define a perturbed activity level $\bar{\eta}_\sigma$. Table 5 presents the instance's parameter settings, for a more detailed description of the parameters we refer to Santos and Marques (2021).

We solve the DRO-SKPSA for three instances generated by the instance generator, which is sufficient to give an idea of the behavior of the DRO-SKPSA applied on the MSS. In Tables 6–8 we present the mean simulated out-of-sample over and underutilization of one based on again 25 replications, with a three week warm up period. From these results we can draw a similar conclusion as in Section 4.1, i.e., that we improve the robustness, in terms of overutilization, of our schedule by increasing ϵ, although at a much faster rate. Contrary to the experiments in Section 4.1, the costs for this robustness is significantly higher than the costs of the SAA with $\epsilon = 0$. Thus, robustness comes at a price of higher underutilization of capacity. However, especially for the MSSP, having underutilization of beds is preferred over bed overutilization, as the latter could lead to patient refusals. Therefore, from a managerial point of view, the solution with $\epsilon = 0.05$ could in many cases still be favorable over the solution with $\epsilon = 0$, despite the higher mean total costs.

5. Concluding remarks

We considered the discrete-time Stochastic Knapsack with Periodic Scheduled Arrivals (SKPSA). Jobs are scheduled on parallel machines, where each job subsequently generates a random number of objects routed to a knapsack. The goal is to find a schedule of jobs on parallel machines, such that the utilization of the unconstrained cousin of the knapsack is as close as possible to the capacity. We introduced the SKPSA, and proposed a DRO based approximation, the DRO-SKPSA. Instead of assuming a distribution for the random job outflow and the holding time, we proposed a Wasserstein distance based DRO solution approach where we try to find a distributionally robust solution using empirical data. This is of practical relevance, as it usually is hard to fit the correct distribution on data, especially when data is scarce. This problem has many applications, for instance in healthcare. An example is the Master Surgical Schedule Problem (MSSP) with downstream capacity constraints. Aside from healthcare our model can be applied in the context of project selection and planning in for instance the consultancy industry.

We solved our model using recent results from Esfahani and Kuhn (2018) and a column-and-constraint generation algorithm based on Saif and Delage (2021). This approach finds good quality solutions. We showed that our approach indeed finds robust scheduled that protect against overutilization of the knapsack. To illustrate the practical use of our model, we solved several instances of the MSSP with downstream resource constraints.

This work presents a basis for future work. Our work formalizes existing work on, e.g., the MSSP, which allows for easy adaption of existing theory to other examples. One possible direction is to further incorporate methods from stochastic knapsack literature in the models presented in this paper. A limitation of our work is that our approach becomes unsuitable if the number of time periods increases, due to the rapidly increasing state space. Therefore, future work should focus on approaches that extend our work to larger state spaces. In our numerical experiments we only considered the case with one resource, i.e., $L = 1$. For $L \gg 1$, the solution algorithm has to consider many vertices

Table 5
Parameters of the instance generator.

Parameters	
Cycle length T	7
# Shifts per day M	5ORs $\times 2$ shifts $= 10$
Number of specialties S	12
Number of beds $c_t,\ t \in \mathcal{T}$	20
Activity level η_σ	$uniform(\{6, \dots, 10\})$
Reference percentage ψ_σ	$\frac{\eta_\sigma}{\sum_{\sigma' \in \Sigma} \eta_\sigma}$
Demand δ_σ	$\max(1, \lfloor \frac{\psi_\sigma \lvert B \rvert}{2} \rfloor)$
Availability α_σ	$uniform(\mathbb{N} \cap [\delta_\sigma, \lfloor 1.35\psi_\sigma \lvert B \rvert \rfloor])$
Perturbed activity level $\bar{\eta}_\sigma$	$uniform([\eta_\sigma - 0.5, \eta_\sigma + 0.5])$
Length of stay per session $H_k,\ k \in \mathcal{K}_\sigma$	Poisson with mean $\frac{9\bar{\eta}_\sigma + 10}{100}$, truncated at $\lceil \frac{1+\eta_\sigma}{5} \rceil$
Number of patients per session $O_k,\ k \in \mathcal{K}_\sigma$	Geometric with $p = \frac{7.75 - 0.7\bar{\eta}_\sigma}{12.5}$ truncated at $2 + 3\eta_\sigma$

Table 6
Results of the DRO-SKPSA for MSSP instance 1.

Instance 1									
ϵ	Objective	Gap	Solving time (s)	Mean total costs	95%-CI	Mean under utilization costs	95%-CI	Mean over utilization costs	95%-CI
0.00	24.7	10.3%	34 878	22.2	(20.3, 24.2)	15.3	(13.9,16.6)	7.0	(6.1, 7.8)
0.05	62.7	13.8%	34 757	32.0	(29.3, 34.7)	28.9	(26.4, 31.4)	3.1	(2.6, 3.6)
0.10	92.8	5.9%	34 751	39.2	(35.8, 42.6)	37.6	(34.2, 40.9)	1.7	(1.4, 2.0)
0.15	119.8	2.9%	34 772	38.7	(35.4, 41.9)	37.8	(34.6, 41.0)	0.8	(0.7, 1.0)
0.20	144.9	1.0%	34 745	54.1	(49.6, 58.6)	54.1	(49.6, 58.6)	0.0	(0.0, 0.0)
0.25	168.7	0.5%	18 214	61.1	(56.2, 66.0)	61.1	(56.2, 66.0)	0.0	(0.0, 0.0)
0.30	192.6	0.4%	24 355	59.7	(54.7, 64.7)	59.7	(54.7, 64.6)	0.0	(0.0, 0.1)
0.35	216.7	0.4%	19 387	62.8	(57.7, 67.9)	62.8	(57.7, 67.9)	0.1	(0.1, 0.1)
0.40	240.8	0.4%	13 327	60.8	(55.9, 65.7)	60.8	(55.9, 65.6)	0.1	(0.0, 0.1)
0.45	264.8	0.4%	10 195	60.1	(55.2, 64.9)	59.8	(55.0, 64.7)	0.2	(0.2, 0.3)
0.50	288.8	0.4%	11 774	61.9	(57.0, 66.9)	61.8	(56.9, 66.7)	0.1	(0.1, 0.1)

Table 7
Results of the DRO-SKPSA for MSSP instance 2.

Instance 2									
ϵ	Objective	Gap	Solving time (s)	Mean total costs	95%-CI	Mean under utilization costs	95%-CI	Mean over utilization costs	95%-CI
0.00	24.9	12.7%	34 917	29.4	(26.9, 31.9)	25.4	(23.2, 27.5)	4.0	(3.5, 4.5)
0.05	62.4	16.9%	34 794	34.4	(31.5, 37.3)	33.2	(30.4, 36.1)	1.2	(1.0, 1.4)
0.10	87.9	4.4%	34 814	49.1	(44.9, 53.3)	48.7	(44.5, 52.9)	0.4	(0.3, 0.5)
0.15	116.2	2.8%	34 805	63.2	(58.1, 68.2)	63.0	(58.0, 68.0)	0.2	(0.1, 0.2)
0.20	142.4	1.6%	34 783	64.3	(59.3, 69.4)	64.3	(59.3, 69.4)	0.0	(0.0, 0.0)
0.25	167.8	1.1%	34 754	73.2	(67.5, 78.8)	73.2	(67.5, 78.8)	0.0	(0.0, 0.0)
0.30	191.5	0.5%	34 747	71.9	(66.3, 77.5)	71.9	(66.3, 77.5)	0.0	(0.0, 0.0)
0.35	215.8	0.3%	31 478	70.3	(64.8, 75.7)	70.3	(64.8, 75.7)	0.0	(0.0, 0.0)
0.40	240.5	0.4%	23 663	72.7	(67.0, 78.3)	72.7	(67.0, 78.3)	0.0	(0.0, 0.0)
0.45	265.0	0.4%	22 618	72.6	(66.9, 78.2)	72.6	(66.9, 78.2)	0.0	(0.0, 0.0)
0.50	289.8	0.5%	26 852	72.5	(66.9, 78.1)	72.5	(66.9, 78.1)	0.1	(0.0, 0.3)

Table 8
Results of the DRO-SKPSA for MSSP instance 3.

Instance 3									
ϵ	Objective	Gap	Solving time (s)	Mean total costs	95%-CI	Mean under utilization costs	95%-CI	Mean over utilization costs	95%-CI
0.00	28.5	9.9%	36 252	28.1	(25.5, 30.7)	21.2	(19.0, 23.4)	6.9	(5.6, 8.1)
0.05	68.1	7.1%	37 579	32.5	(29.8, 35.3)	29.9	(27.3, 32.5)	2.6	(2.3, 3.0)
0.10	101.2	3.3%	37 537	42.0	(38.6, 45.3)	41.3	(37.9, 44.7)	0.6	(0.5, 0.7)
0.15	129.3	0.8%	37 047	61.1	(56.2, 66.0)	61.1	(56.2, 66.0)	0.0	(0.0, 0.1)
0.20	155.4	0.4%	10 223	64.6	(59.6, 69.6)	64.6	(59.6, 69.6)	0.0	(0.0, 0.0)
0.25	180.2	0.5%	6175	70.4	(64.9, 75.9)	70.4	(64.8, 75.9)	0.0	(0.0, 0.0)
0.30	204.4	0.4%	5676	70.8	(65.2, 76.3)	70.8	(65.2, 76.3)	0.0	(0.0, 0.0)
0.35	228.4	0.4%	5089	71.5	(65.9, 77.2)	71.5	(65.9, 77.1)	0.0	(0.0, 0.0)
0.40	252.8	0.4%	5834	70.9	(65.3, 76.5)	70.9	(65.3, 76.4)	0.0	(0.0, 0.0)
0.45	276.8	0.3%	5509	73.2	(67.5, 78.9)	73.2	(67.5, 78.9)	0.0	(0.0, 0.0)
0.50	301.4	0.5%	3213	73.2	(67.5, 78.9)	73.2	(67.5, 78.8)	0.0	(0.0, 0.0)

in each iteration, which might cause computational challenges. This poses a challenging problem for future research. Besides, we applied the Wasserstein ambiguity set because it generalizes other methods for optimization under uncertainty. It could be interesting to investigate the effect of considering other ambiguity sets as well. Another future extension could be the situation where objects arriving to the knapsack from a class-k job use multiple resources simultaneously, or subsequently. A last consideration are additional restrictions on the state space $\mathcal{S}_t$, such that when two jobs share the same knapsack resource, one of them is allowed only a fraction of the resource's capacity.

CRediT authorship contribution statement

Hayo Bos: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Richard J. Boucherie:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Formal analysis, Conceptualization. **Erwin W. Hans:** Writing – original draft, Supervision, Project administration, Funding acquisition, Conceptualization. **Gréanne Leeftink:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Methodology, Conceptualization.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research is funded by and done at Diakonessenhospital Utrecht, the Netherlands, in collaboration with the Center for Healthcare Operations Improvement and Research (CHOIR).

References

Bertsimas, D., Brown, D.B., Caramanis, C., 2011. Theory and applications of robust optimization. SIAM Rev. 53 (3), 464–501.
Boucherie, R.J., Van Dijk, N.M., 1991. Product forms for queueing networks with state-dependent multiple job transitions. Adv. in Appl. Probab. 23 (1), 152–187.
Braaksma, A., Kortbeek, N., Boucherie, R.J., 2021. Bed census predictions and nurse staffing. In: Handbook of Healthcare Logistics: Bridging the Gap Between Theory and Practice. Springer International Publishing, Cham, pp. 151–180.
Bruni, M., Khodaparasti, S., Demeulemeester, E., 2020. The distributionally robust machine scheduling problem with job selection and sequence-dependent setup times. Comput. Oper. Res. 123, 105017.
Chang, Z., Ding, J.-Y., Song, S., 2019. Distributionally robust scheduling on parallel machines under moment uncertainty. European J. Oper. Res. 272 (3), 832–846.
Chen, Z., Sim, M., Xiong, P., 2020. Robust stochastic optimization made easy with RSOME. Manage. Sci. 66 (8), 3329–3339.
Conforti, M., Cornuéjols, G., Zambelli, G., 2014. Integer programming. In: Graduate Texts in Mathematics, Springer International Publishing.
Daw, A., Pender, J., 2019. On the distributions of infinite server queues with batch arrivals. Queueing Syst. 91 (3–4), 367–401.
Delage, E., Ye, Y., 2010. Distributionally robust optimization under moment uncertainty with application to data-driven problems. Oper. Res. 58 (3), 595–612.
Drezet, L.-E., Billaut, J.-C., 2008. A project scheduling problem with labour constraints and time-dependent activities requirements. Int. J. Prod. Econ. 112 (1), 217–225, Special Section on Recent Developments in the Design, Control, Planning and Scheduling of Productive Systems.
Esfahani, P.M., Kuhn, D., 2018. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. Math. Programm. 171 (1-2), 115–166.
Fuegener, A., Hans, E.W., Kolisch, R., Kortbeek, N., Vanberkel, P.T., 2014. Master surgery scheduling with consideration of multiple downstream units. Eur. J. Oper. Res. 239 (1), 227–236.
Garey, M., Johnson, D., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. In: Mathematical Sciences Series, Freeman.
Gurobi Optimization, L., 2022. Gurobi Optimizer Reference Manual.
Hariga, M., Jackson, P., 1996. The warehouse scheduling problem: Formulation and algorithms. IIE Trans. 28 (2), 115–127.
Hassin, R., Mendel, S., 2008. Scheduling arrivals to queues: A single-server model with no-shows. Manage. Sci. 54 (3), 565–572.
Kantorovich, L., Rubinstein, G.S., 1958. On a space of totally additive functions. Vestnik Leningrad. Univ 13, 52–59.
Kelly, F., 1991. Loss Networks. Ann. Appl. Probab. 1 (3), 319–378.
Kleywegt, A.J., Papastavrou, J.D., 2001. The dynamic and stochastic knapsack problem with random sized items. Oper. Res. 49 (1), 26–41.
Kortbeek, N., Braaksma, A., Smeenk, F.H.F., Bakker, P.J.M., Boucherie, R.J., 2015. Integral resource capacity planning for inpatient care services based on bed census predictions by hour. J. Oper. Res. Soc. 66 (7), 1061–1076.
Liu, F., Chen, Z., Wang, S., 2023. Globalized distributionally robust counterpart. INFORMS J. Comput. 35 (5), 1120–1142.
Liu, T., Saldanha-da Gama, F., Wang, S., Mao, Y., 2022. Robust stochastic facility location: sensitivity analysis and exact solution. INFORMS J. Comput. 34 (5), 2776–2803.
Liu, T., Li, G., 2021. Robust recycling facility location with clustering. Comput. Oper. Res. 136, 105466.
Lu, L.L., Chiu, S.Y., Cox, L.A., 1999. Optimal project selection: Stochastic knapsack with finite time horizon. J. Oper. Res. Soc. 50 (6), 645–650.
Mevissen, M., Ragnoli, E., Yu, J.Y., 2013. Data-driven distributionally robust polynomial optimization. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (Eds.), In: Advances in Neural Information Processing Systems, vol. 26, Curran Associates, Inc., pp. 37–45.
Neyshabouri, S., Berg, B.P., 2017. Two-stage robust optimization approach to elective surgery and downstream capacity planning. Eur. J. Oper. Res. 260 (1), 21–40.
Pang, G., Whitt, W., 2012. Infinite-server queues with batch arrivals and dependent service times. Probab. Engrg. Inform. Sci. 26 (2), 197–220.
Pegden, C., Rosenshine, M., 1990. Scheduling arrivals to queues. Comput. Oper. Res. 17 (4), 343–348.
Ross, K.W., 1995. Multiservice Loss Models for Broadband Telecommunication Networks. Springer-Verlag.
Saif, A., Delage, E., 2021. Data-driven distributionally robust capacitated facility location problem. European J. Oper. Res. 291 (3), 995–1007.
Santos, D., Marques, I., 2021. Designing master surgery schedules with downstream unit integration via stochastic programming. European J. Oper. Res..
Scarf, H., 1957. A min-max solution of an inventory problem. Econometrica 25 (2), 352.
Schneider, A.J.T., van Essen, J.T., Carlier, M., Hans, E.W., 2020. Scheduling surgery groups considering multiple downstream resources. Eur. J. Oper. Res. 282 (2), 741–752.
Shehadeh, K.S., Cohn, A.E., Jiang, R., 2020. A distributionally robust optimization approach for outpatient colonoscopy scheduling. European J. Oper. Res. 283 (2), 549–561.
Shehadeh, K.S., Padman, R., 2021. A distributionally robust optimization approach for stochastic elective surgery scheduling with limited intensive care unit capacity. Eur. J. Oper. Res. 290 (3), 901–913.
Shehadeh, K.S., Sanci, E., 2021. Distributionally robust facility location with bimodal random demand. Comput. Oper. Res. 134, 105257.
Smith, J.E., Winkler, R.L., 2006. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. Manage. Sci. 52 (3), 311–322.
van Essen, J.T., Bosch, J.M., Hans, E.W., van Houdenhoven, M., Hurink, J.L., 2014. Reducing the number of required beds by rearranging the OR-schedule. OR Spectrum 36 (3), 585–605.
van Slyke, R., Young, Y., 2000. Finite horizon stochastic knapsacks with applications to yield management. Oper. Res. 48 (1), 155–172.
Vanberkel, P.T., Boucherie, R.J., Hans, E.W., Hurink, J.L., van Lent, W., van Harten, W.H., 2011. An exact approach for relating recovering surgical patient workload to the master surgical schedule. J. Oper. Res. Soc. 62 (10), 1851–1860.
Walrand, J., 1983. A discrete-time queueing network. J. Appl. Probab. 20 (4), 903–909.
Wang, Y., Yang, Y., Tang, L., Sun, W., Li, B., 2020. A wasserstein based two-stage distributionally robust optimization model for optimal operation of CCHP micro-grid under uncertainties. Int. J. Electr. Power Energy Syst. 119.
Wang, Y., Zhang, Y., Tang, J., 2019. A distributionally robust optimization approach for surgery block allocation. European J. Oper. Res. 273 (2), 740–753.