Robust Two-Stage Packing into Designated and Multipurpose Bins

Yacine Al Najjar, Noam Goldberg, Shlomo Karhi, Michael Poss

Robust Two-Stage Packing into Designated and Multipurpose Bins

Yacine Al Najjar * **Noam Goldberg** ** **Shlomo Karhi** ***
Michael Poss ****

* *Mathematical and Algorithmic Sciences Lab, Paris Research Center - Huawei Technologies Co. Ltd, France (e-mail: najjar@huawei.com).*
** *Bar-Ilan University, Ramat Gan 5290002, Israel (e-mail: noam.goldberg@biu.ac.il).*
*** *Bar-Ilan University, Ramat Gan 5290002, Israel (e-mail: shlomo.karhi@biu.ac.il).*
**** *LIRMM, Universitde Montpellier, CNRS, Montpellier, France (e-mail: michael.poss@lirmm.fr).*

Abstract: Multitype bin packing is a natural extension of the classical bin packing with applications to shipping using climate-controlled containers and plain dry containers. In transportation and other logistics applications there may be significant uncertainty with respect to the exact quantities of different variants of products (or item types) that may need to be shipped at the time when the containers and packaging are procured. In the current paper we model the problem as a robust two-stage two-item type bin packing problem. In the first stage bins of different types are acquired (e.g., reefer containers and dry containers). In the second stage the items are packed into bins. The bins that are secured in the first phase must allow for all of the items to be packed in the "worst-case" demand scenario. We first develop an algorithm for the robust two-stage two-item type bin packing problem with general item-number uncertainty sets and certain box uncertainty sets for item sizes (or equivalently two item sizes). We then consider the special case of identical (or unit) item sizes. In this special case we develop closed-form solutions for the optimal solution. Our closed-form solution reveals that it is optimal to use a number of multipurpose bins that is linear in the number of items. This is in contrast with solutions of the online and offline deterministic version of our problem that use at most one multipurpose bin. Finally, we consider computational methods that are efficient in practice for a generalization with unit item sizes but with an arbitrary number of item and bin types and arbitrary compatibility structures.

1. INTRODUCTION

Consider a two-stage problem in which there are demands for two types of items and these have to be packed using three types of bins; a bin type designated for each type of item and an additional multipurpose type of bin that can be used to pack both types of items. The bins are acquired before the demands, accordingly the quantities of the items $n = (n_1, n_2)$ are realized and become known. The objective is to minimize the costs while being able to meet every possible demand n in a (compact) set of possible scenarios $\mathcal{N}$ of the item demands. Some of our results may apply in particular to the case of finite uncertainty sets $\mathcal{N}$. This case includes the standard bin packing problem setting where the numbers of bins are natural numbers. Without loss of generality we assume that the type-1 bin cost is 1 unit and that α, β are the costs of type-2 bins and multipurpose (type-3) bins, respectively, satisfying $1 \leq \alpha \leq \beta < 1 + \alpha$.

The two-item type bin packing problem is introduced in Goldberg and Karhi [2017]. It is motivated by two types of items to be transported with different climate control settings. In Figure 1 there are two types of items – ones that must be refrigerated above the freezing point and ones that need to be transported frozen. Then the three container types are a refrigerated containers set to the corresponding temperatures of refrigerating, freezing, and a multi-temperature container that can simultaneously transport items with different temperature requirements (see for example Tassou et al. [2009]). Each bin type has a different cost and it is expected that more versatile bin types would cost more than ones that can contain only a few item types. In Goldberg and Karhi [2017] the online two-size two-item type problem is considered. A tight (optimal) absolute competitive ratio of 1.618 is proved for this problem. The bin packing problem is closely related to machine scheduling where similar multipurpose machines and machine compatibility restrictions have been extensively studied; see for example Azar et al. [1995], Shabtay and Karhi [2012], Karhi and Shabtay [2014].

Robust optimization is a useful mathematical modeling framework that aims to determine solutions that are immune to uncertainty while addressing issues of computational tractability; see Ben-Tal et al. [2009], Bertsimas and Sim [2003]. This modeling approach is also useful in cases where the uncertainty is not easily described or estimated by any particular probability distribution. In such cases, the number of items of each type may be defined by

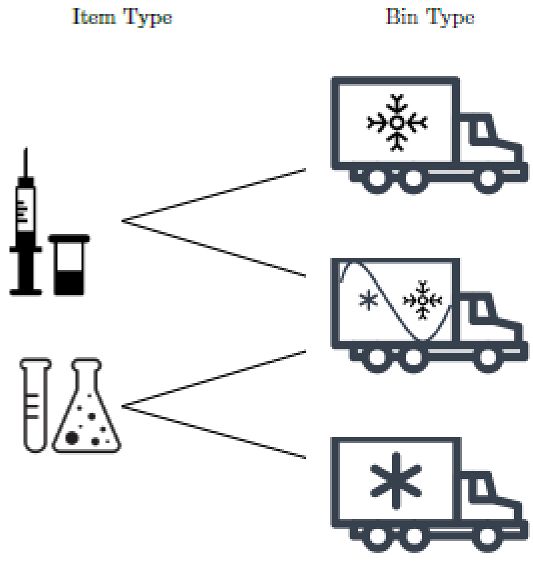

Fig. 1. Two item types and three bin types illustrated for shipping of medicine and vaccines that require refrigeration and freezing, respectively.

a given uncertainty set. The type of uncertainty set is typically a box, given by an interval for each (independent) input parameter, budgeted uncertainty (which restricts not only the deviation of each parameter independently but also the total deviation), or an ellipsoid. Bougeret et al. [2018] recently consider several variants of robust machine scheduling with different scheduling objectives (in the absence of multipurpose machines) in the robust optimization framework. Computational techniques such as branch-and-price algorithms are considered for a robust bin packing problem, in which item sizes are subject to uncertainty (or job processing times), in Song et al. [2018].

Multi-stage (also known as adjustable) robust optimization is an especially challenging setting in which the uncertainty is realized and decisions are made over two or more stages. The challenge remains also in the case of robust two-stage linear programs that are proven to be NP-hard [Ben-Tal et al., 2004]. In the following we consider particular two-stage robust formulations motivated by the container shipping application of multitype bin packing that has been introduced by Goldberg and Karhi [2017]. Multitype bin packing is extended to a two stage problem where the bins are ordered in the first stage and the items are produced and must be assigned to compatible bins in the second stage. The number of items of each type is subject to uncertainty in the first stage when the bins are ordered. For example, the uncertainty may be associated with the uncertain demand that each particular product variant faces before it is realized in the second stage.

2. A GENERAL ROBUST TWO-STAGE TWO-TYPE PROBLEM

In the general two-type problem, the number of bins of each type, as well as possibly the item sizes, are defined within given uncertainty sets. Suppose an arbitrary item number uncertainty set $\mathcal{N} \subseteq \mathbb{R}^2$. Then, in the most general case for each $n \in \mathcal{N}$, an uncertainty set is defined for the item sizes $\mathcal{A}(n) \subseteq \mathbb{R}_+^{n_1+n_2}$. The two-stage robust multipurpose bin packing problem is stated in the following.

Two-Stage Two-Type Robust Bin Packing
Given uncertainty (scenario) set $\mathcal{N}$ for the number of items of each type, and for each $n \in \mathcal{N}$, item size uncertainty set $\mathcal{A}(n)$, determine the number of designated type-1 and type-2 bins $y_1, y_2 \in \mathbb{N}$, respectively, and number of multipurpose bins $y_3 \in \mathbb{N}$ so that the cost of bins needed to store all of the items in the worst case scenario, $y_1 + \alpha y_2 + \beta y_3$ is minimized.

We focus on the particular case of a bin packing problem with two item types and item size sets $\mathcal{A}_1$ and $\mathcal{A}_2$ for type 1 and type 2 items, respectively. Accordingly $\mathcal{A}(n) = \mathcal{A}_1 \times \mathcal{A}_2$ for all $n \in \mathcal{N}$. We also model unit-size situations in which uncertainty sets are defined only for the number of items and the set $\mathcal{A}(n)$ is a singleton. A key observation is that the problem with hyper-cube item-size uncertainty (for each item type) reduces to a two-size problem. It is summarized in the following proposition.

Observation 1. Suppose a 2-type bin packing problem with uncertain item numbers $n = (n_1, n_2) \in \mathcal{N}$ and item sizes in hypercube sets $\mathcal{A}_1$ and $\mathcal{A}_2$ for type 1 and type 2 items, respectively. Then, the robust bin packing problem reduces to a two-size uncertain item number bin packing problem with sizes: $a_1 = \max_{a \in \mathcal{A}_1} ||a||$ and $a_2 = \max_{a \in \mathcal{A}_2} ||a||$.

In the following two sections we focus on the problem where each item type has a distinct size and then on the unit size case. Efficient algorithmic and closed-form results, respetively, are developed for these two settings. In Section 5 we consider computational solution schemes in a general setting with an arbitrary number of item types and type-compatibility structure with unit-item sizes and variable bin sizes for different bin types.

3. ROBUST TWO-STAGE TWO-SIZE TWO-TYPE PROBLEM

Now consider the two-item size case where type 1 items have size a_1 and type 2 items have size $a_2 \neq a_1$. The formulation (1) accounts for different item sizes in multipurpose bins using the set of all possible two-size feasible packing pattern set $\mathcal{P}$. For each $p \in \mathcal{P}$ and $i = 1, 2$, $z_i(p)$ denotes the number of items of type i placed in a multipurpose bin by a (feasible) packing pattern p. For each $i = 1, 2$, y_i denotes the number of designated bins of type i used, and $x(n)_i$ the number of items of type i packed into a bin a type i for $n \in \mathcal{N}$. Note that although there may be exponentially many patterns (even with a constant number of item sizes) this formulation has a tighter linear programming relaxation compared with a compact assignment-based formulation of bin packing. Letting λ_p denote the decision variable indicating how many bins of a pattern $p \in \mathcal{P}$ are opened then $y_3 = \sum_{p \in \mathcal{P}} \lambda_p$. Finally, the two-size robust two-stage integer program is given by the pattern based formulation

$$
\begin{aligned}
z^* \equiv \min \quad & y_1 + \alpha y_2 + \beta \sum_{p\in\mathcal{P}} \lambda_p && \text{(1a)}\\
\text{s.t.} \quad & x(n)_i \le \left\lfloor \frac{V}{a_i} \right\rfloor y_i && \forall n \in \mathcal{N}, i = 1,2 \quad \text{(1b)}\\
& \sum_{p\in\mathcal{P}} z_i(p)\lambda_p \ge n_i - x(n)_i && \forall n \in \mathcal{N}, i = 1,2. \quad \text{(1c)}\\
& 0 \le x(n)_i \le n_i, && \forall n \in \mathcal{N}, i = 1,2 \quad \text{(1d)}\\
& \lambda_p \in \mathbb{N} && p \in \mathcal{P}\\
& y_j \in \mathbb{N}. && j = 1,2,3
\end{aligned}
$$

Let us define

$$
n^{\max} \equiv \max_{n\in\mathcal{N}} \{n_1 + n_2\}, \qquad n_1^{\max} \equiv \max_{n\in\mathcal{N}} n_1, \quad \text{and} \quad n_2^{\max} \equiv \max_{n\in\mathcal{N}} n_2.
$$

For fixed numbers of designated bins, given by a pair of values, x_1 and x_2, a sequence of two-size problems can be solved to determine the number of multipurpose bins needed in the worst case scenario. To this end, let $\hat{y}_3(n_1, n_2)$ denote the minimum number of bins required to pack n_1, n_2 items of type 1 and type 2, respectively, in multipurpose bins. Then, the cost of a feasible solution given that x_1, x_2 are packed in designated bins of type 1 and 2, respectively, is given by

$$
f(x_1, x_2) = \left\lceil \frac{x_1}{\lfloor V/a_1 \rfloor} \right\rceil + \alpha \left\lceil \frac{x_2}{\lfloor V/a_2 \rfloor} \right\rceil + \beta \max_{n\in\mathcal{N}} \{\hat{y}_3(n_1 - x_1, n_2 - x_2)\}.
$$

Note that $\mathcal{N}$ in this expression can be effectively replaced by a subset of *nondominated* scenarios $\bar{\mathcal{N}} \subseteq \mathcal{N}$. In particular, $\bar{n} \in \bar{\mathcal{N}}$ implies that for all $n \in \mathcal{N}$ either $\bar{n}_1 \ge n_1$ or $\bar{n}_2 \ge n_2$. Conversely, $\bar{n} \in \mathcal{N}$ is said to *dominate* $n \in \mathcal{N}$ if $\bar{n} \ge n$. Let $\bar{f}(x_1, x_2)$ denote $f(x_1, x_2)$ with $\mathcal{N}$ replaced by $\bar{N}$. So, given this set a straightforward algorithm is given by

$$
z_A = \min_{x_1=0,\dots,n_1^{\max}, x_2=0,\dots,n_2^{\max}} \bar{f}(x_1, x_2). \tag{2}
$$

The subset of nondominated solutions can be computed in output-polynomial time Bökler et al. [2017]. The following proposition establishes the correctness and a running time complexity bound of the straightforward algorithm given by 2 in the case that the uncertainty (scenario set) $\mathcal{N}$ is either finite or polyhedral.

Proposition 1. Suppose that $\mathcal{N}$ is either a finite or polyhedral set. Then, the algorithm given by (2) outputs $z_A = z^*$, an optimal solution for (1), with a time complexity bound of $O(n_1^{\max} n_2^{\max} |\bar{\mathcal{N}}| \log n \log^2 V)$.

Proof. The algorithm enumerates for $i = 1,2$ all possible values of x_i in $\{0, \dots, n_i^{\max}\}$. For each value of (x_1, x_2), $\bar{f}(x_1, x_2)$ evaluates the cost of opening x_1 and x_2 type 1 and type 2, respectively, designated bins, and the number of multipurpose bins required to maintain feasibility in the worst case nondominated scenario (in the set $\bar{\mathcal{N}}$). In either case that $\mathcal{N}$ is finite or polyhedral, then the set $\bar{\mathcal{N}}$ is finite (in the case that $\mathcal{N}$ is polyhedral then $\bar{N}$ is a subset of the finite set of extreme points of $\mathcal{N}$). The standard two-size problem is solved by the algorithm in McCormick et al. [2001] whose complexity is $O(\log n \log^2 V)$, then the overall running time complexity of this straightforward algorithm is $O(n_1^{\max} n_2^{\max} |\bar{\mathcal{N}}| \log n \log^2 V)$. □

Note that the proof of the proposition in the case that $\mathcal{N}$ is polyhedral relies on the fact that $\bar{N}$ is a subset of the extreme points of $\mathcal{N}$, which is a finite set. Further, the set of nondominated points $\bar{\mathcal{N}}$ can be significantly smaller in cardinality than the set of extreme points of $\mathcal{N}$.

4. UNIT SIZE ITEMS

In the unit size case we can formulate the problem as a robust mathematical program

$$
\begin{aligned}
z^* \equiv \min \quad & y_1 + \alpha y_2 + \beta y_3 && \text{(3a)}\\
\text{s.t.} \quad & x(n)_{ii} + x(n)_{i3} = n_i && \forall n \in \mathcal{N}, i = 1,2 \quad \text{(3b)}\\
& x(n)_{ii} \le V y_i && \forall n \in \mathcal{N}, i = 1,2 \quad \text{(3c)}\\
& x(n)_{13} + x(n)_{23} \le V y_3 && \forall n \in \mathcal{N} \quad \text{(3d)}\\
& y_j, x_{ij} \in \mathbb{N} && i = 1,2, j = 1,2,3.
\end{aligned}
$$

Here, for $j = 1,2,3$, y_j is the number of bins of type j that are procured. For $i = 1,2$ and $j = 1,2,3$, $x(n)_{ij}$ is the number of items of type i assigned to bin type j in demand scenario $n = (n_1, n_2)$, where n_i is the demand for item type i.

Let $y_3 = \left\lceil \frac{n_1^{\max}}{V} \right\rceil - y_1$ and consider the function $f : \mathbb{R} \to \mathbb{R}$ given by

$$
\begin{aligned}
f(y_1) =& y_1 + (\beta - \alpha) y_3 + \alpha \max\left\{ \left\lceil \frac{n^{\max}}{V} \right\rceil - y_1, \left\lceil \frac{n_2^{\max}}{V} \right\rceil \right\}\\
=& (1 - \beta) y_1 + \alpha \max\left\{ \left\lceil \frac{n^{\max}}{V} \right\rceil, \left\lceil \frac{n_2^{\max}}{V} \right\rceil + y_1 \right\}\\
&+ (\beta - \alpha) \left\lceil \frac{n_1^{\max}}{V} \right\rceil.
\end{aligned}
$$

Lemma 1. Suppose that y^* is optimal for (3). Then, $z^* = y_1^* + \alpha y_2^* + \beta y_3^* = f(y_1^*)$.

Proof. The fact that y^* is feasible for (3), and in particular that it satisfies (3d), implies that $y_3^* \ge \left\lceil \frac{n_1^{\max}}{V} \right\rceil - y_1^*$. If $\beta > \alpha$, then over all y that are feasible for (3), a pair (y_2^*, y_3^*) that minimizes $f(y_1^*)$ must have $y_3^* \le \left\lceil \frac{n_1^{\max}}{V} \right\rceil - y_1^*$. To see this, suppose for the sake of contradiction that y^* is optimal for (3) and $y_3^* > \left\lceil \frac{n_1^{\max}}{V} \right\rceil - y_1^*$. Then, $y_3^* \ge \left\lceil \frac{n_1^{\max}}{V} \right\rceil - y_1^* + 1$. Letting y' be defined by $y'_i = \begin{cases} y_1^* & i = 1 \\ y_2^* + 1 & i = 2 \\ y_3^* - 1 & i = 3, \end{cases}$ evidently y' satisfies (3c)-(3d) and is therefore also feasible for (3). Further, the fact that $\beta > \alpha$ implies that $y'_1 + \alpha y'_2 + \beta y'_3 < z^*$, thereby establishing a contradiction. Then, feasibility, in particular the fact that y^* satisfies (3c) for $i = 2$, implies that $y_2^* = \max\left\{ \left\lceil \frac{n^{\max}}{V} \right\rceil - y_1^* - y_3^*, \left\lceil \frac{n_2^{\max}}{V} \right\rceil - y_3^* \right\}$.

Otherwise, if $\beta = \alpha$ then $y_2^* + y_3^* = \max\left\{ \left\lceil \frac{n^{\max}}{V} \right\rceil - y_1^*, \left\lceil \frac{n_2^{\max}}{V} \right\rceil \right\}$, and y^* must be optimal since $y_2^* + y_3^*$ is the minimum number of bins required to store the type 2 items. $\square$

The following proposition establishes that an optimal solution and optimal solution value are given by closed-form solutions. Note that we may assume $1 < \beta$ without loss of generality (otherwise we may use only multipurpose or designated bins, respectively), in addition to the assumption in Section 1 that $1 \leq \alpha \leq \beta < 1 + \alpha$.

Proposition 2. Suppose that $1 < \beta < \alpha + 1$. The optimal objective value of (3) is attained with some y^* such that $y_1^* = \lceil \frac{n^{\max}}{V} \rceil - \lceil \frac{n_2^{\max}}{V} \rceil$. Further, the optimal objective value

$$
\begin{aligned}
z^* =& (1-\beta)\left(\left\lceil \frac{n^{\max}}{V} \right\rceil - \left\lceil \frac{n_2^{\max}}{V} \right\rceil\right) + (\beta - \alpha)\left\lceil \frac{n_1^{\max}}{V} \right\rceil \\
&+ \alpha \left\lceil \frac{n^{\max}}{V} \right\rceil .
\end{aligned}
$$

Proof. Following Lemma 1 the optimal objective value of (3) is given by the minimum of f over the integers. Note that the function f is continuosly differentiable over $\left[0, \lceil \frac{n^{\max}}{V} \rceil - \lceil \frac{n_2^{\max}}{V} \rceil\right) \cup \left(\lceil \frac{n^{\max}}{V} \rceil - \lceil \frac{n_2^{\max}}{V} \rceil, \infty\right)$.

For $y_1 \in \left[0, \lceil \frac{n^{\max}}{V} \rceil - \lceil \frac{n_2^{\max}}{V} \rceil\right)$, $\frac{df}{dy_1}(y_1) = 1 - \beta < 0$ It follows that f is decreasing over the interval $\left[0, \lceil \frac{n^{\max}}{V} \rceil - \lceil \frac{n_2^{\max}}{V} \rceil\right)$.

For $y_1 \in \left(\lceil \frac{n^{\max}}{V} \rceil - \lceil \frac{n_2^{\max}}{V} \rceil, \infty\right)$, by our assumption on β it follows that $\frac{df}{dy_1}(y_1) = 1 - \beta + \alpha > 0$. It follows that f is increasing over $\left(\lceil \frac{n^{\max}}{V} \rceil - \lceil \frac{n_2^{\max}}{V} \rceil, \infty\right)$. Thus, $y_1^* = \lceil \frac{n^{\max}}{V} \rceil - \lceil \frac{n_2^{\max}}{V} \rceil \in \arg\min_{y_1 \geq 0} f(y_1)$. Since $y_1^* \in \mathbb{N} \subseteq \mathbb{R}_+$ it must also be that $y_1^* \in \arg\min_{y_1 \in \mathbb{N}} f(y_1)$. $\square$

It follows that in the unit size case only the three scenarios corresponding to $n_1^{\max}, n_2^{\max}$ and $n^{\max}$ need to be considered. This also applies to infinite uncertainty sets $\mathcal{N}$ such as in the following example.

Example 1. Consider an example where $\mathcal{N}$ is an ellipsoid

$$\mathcal{N} = \left\{ n \in \mathbb{R}^2 \;\middle|\; b_1^2 n_1^2 + b_2^2 n_2^2 = r^2, n_1, n_2 \geq 0 \right\}$$

We have that $n_1^{\max} = r/b_1$, $n_2^{\max} = r/b_2$. Further, for $(n_1, n_2) \in \mathcal{N}$ we have $n_2 = \sqrt{r^2 - b_1^2 n_1^2}/b_2$. $n^{\max}$ is given by the maximum of $n_1 + n_2 = n_1 + \sqrt{r^2 - b_1^2 n_1^2}/b_2$. Let $g(n_1) = n_1 + \sqrt{r^2 - b_1^2 n_1^2}/b_2$. Its maximum is attained as n_1 that satisfies $g'(n_1) = 1 - n_1/(b_2\sqrt{r^2 - b_1^2 n_1^2}) = 0$. So, $n_1 = \sqrt{b_2^2 r^2/(1 + b_2^2 b_1^2)}$, and $n^{\max} = g(n_1) = g(\sqrt{b_2^2 r^2/(1 + b_2^2 b_1^2)})$. In particular, for a circle ($b_1 = b_2 = 1$) this gives $n_1 = n_2 = r/\sqrt{2}$, and solving the problem for the circle-uncertainty set reduces to solving the problem with a set of three scenarios $\{(r, 0), (0, r), (r/\sqrt{2}, r/\sqrt{2})\}$ (although there are points of the circle that are not convex combinations of these 3 points). Then, $z^* = \beta\left(2\lceil \frac{r}{V} \rceil - \left\lceil \frac{\sqrt{2}r}{V} \right\rceil\right) + (\alpha + 1)\left(\left\lceil \frac{\sqrt{2}r}{V} \right\rceil - \lceil \frac{r}{V} \rceil\right)$.

Interestingly, in the robust two-stage problem the number of multipurpose bins that are acquired in an optimal solution depends on the number of items and the uncertainty set. In sharp contrast, in the offline deterministic unit-size problem as well as online two-size problems that are described in Goldberg and Karhi [2017], (optimal) solutions use at most one multipurpose bin (when $\beta > \alpha$).

5. ARBITRARY NUMBER OF BIN TYPES WITH UNIT ITEM SIZES AND VARIABLE-SIZE BINS

In this section we generalize the unit-item size problem by considering an arbitrary number of item types and bin types. We are given as data for the problem a set of item types S and a set of bin types T. Each item type $i \in S$ can only fit in a given subset $T(i) \subseteq T$ of bins types. We also note $S(j)$ the item types that can fit into bin j. Each bin of type $j \in T$ may contain up to V_j items, and its cost is c_j. The problem is to determine the number of bins of each type that must be ordered such that all possible realizations of the demand $n \in \mathcal{N} \subseteq \mathbb{R}^{|S|}$ can be satisfied at a minimum cost. We suppose that $\mathcal{N}$ is bounded from above, otherwise there may not exist any finite number of bins that can satisfy all possible demands $n \in N$. We model this problem as a mixed-integer linear program (MILP), which may contain an infinite number of variables and constraints,

$$z^* \equiv \min \quad \sum_{j \in M} c_j y_j \tag{4a}$$

$$\text{s.t.} \quad \sum_{j \in T(i)} x(n)_{ij} = n_i \qquad \forall i \in S, n \in \mathcal{N} \tag{4b}$$

$$\sum_{j \in S(j)} x(n)_{ij} \leq V_j y_j \qquad \forall j \in T, n \in \mathcal{N} \tag{4c}$$

$$
\begin{aligned}
& y_j \in \mathbb{N} && j \in T \\
& x(n)_{ij} \geq 0 && i \in S, j \in T, n \in \mathcal{N}.
\end{aligned}
$$

Here for each $j \in T$, y_j is a decision variable that represents the number of bins of type j that are ordered. x_{ij} is the number of items of type $i \in S$ that are stored in a bin of type $j \in T$. Constraints (4b) ensure that for each demand scenario $n \in \mathcal{N}$ and for each item type i, its demand n_i is fully assigned to some bin. Constraints (4c) ensure that the capacity of a bins is not exceeded in any scenario. Note that this problem generalizes notoriously hard (strongly NP-hard) weighted set cover and set partitioning problems; see for example Barnhart et al. [1996].

As discussed in Section 3, the notion of dominance can be used to consider a potentially small subset of set $\mathcal{N}$. Dominance may not suffice to make problem (4) tractable. To this end, we develop computational schemes that are efficient in practice. In particular, methods are proposed for iteratively generating certain critical subsets of demand vectors $\mathcal{N}^0 \subseteq \mathcal{N}$.

5.1 Iterative methods

As in the case of the two-item type problem, we expect that a small subset of $\mathcal{N}$ may suffice to determine an optimal decision variable vector y. Accordingly, we now introduce an iterative algorithm that at each iteration solves the problem (4) with only a subset $\mathcal{N}^0$ of all possible demand scenarios to obtain a solution $y^* \in \mathbb{N}^{|T|}$. Then, the iterative algorithm solves another separation problem, described below, to check if there is a demand scenario $n^* \in \mathcal{N}$ for which no feasible assignment of the items to the ordered bins corresponding to y^* can be found (a *violated demand scenario*). If such a violated scenario $n^* \in \mathcal{N}$ exists, then n^* is appended to the set $\mathcal{N}^0$ and an iteration is repeated. Otherwise, the problem is solved. This method is referred to as a *row and column generation* (RCG) algorithm, as each iteration adds variables $x(n^*)_{ij}$, for all $i \in S$ and $j \in T$, and associated constraints (4b)-(4c) to problem (4).

A variant of the general RCG method is a *row generation* (RG) algorithm. RG solves the problem in the y-space (initially with an empty set of the some of the constraints of formulation (4)), and at each iteration a cut violated by the current solution y^* is appended to the formulation. These two algorithms have been previously introduced for two-stage robust optimization in Ayoub and Poss. [2016] and Zeng and Zhao [2013]. We now propose how to determine a violated demand scenario $n^* \in \mathcal{N}$ and corresponding cut at each iteration where (4) is solved, with $\mathcal{N}$ replaced by some $\mathcal{N}^0 \subseteq N$.

For a given solution $y^* \in \mathbb{N}^{|T|}$ and demand scenario $n \in \mathcal{N}$, the ordered bins corresponding to y^* can satisfy the demand vector n if and only if the following problem has a zero optimal objective value,

$$\begin{aligned} \min \quad & \sum_{i \in S} \alpha_i && && \text{(5a)} \\ \text{s.t.} \quad & \alpha_i + \sum_{j \in T(i)} x_{ij} \geq n_i && \forall i \in S && \text{(5b)} \\ & \sum_{i \in S(j)} x_{ij} \leq V_j y_j^* && \forall j \in T && \text{(5c)} \\ & x \geq 0, \alpha \geq 0 \end{aligned}$$

By linear programming duality, the optimal solution of problem (5) has the same objective value as the dual problem,

$$\begin{aligned} \max \quad & \sum_{i \in S} n_i \pi_i - \mu_j V_j y_j^* && && \text{(6a)} \\ \text{s.t.} \quad & \pi_i - \mu_j \leq 0 && \forall i \in S, \forall j \in T(i) && \text{(6b)} \\ & \pi_i \leq 1 && \forall i \in S && \text{(6c)} \\ & \pi, \mu \geq 0. \end{aligned}$$

Thus, to check if a given solution y^* can cover all demand vectors in $\mathcal{N}$, we can solve problem (6) individually for every demand $n \in \mathcal{N}$. It may be possible only if $\mathcal{N}$ is finite. Alternatively, $n \in \mathcal{N}$ can be incorporated in the formulation as a vector of decision variables, resulting in a (nonconvex) quadratic program in the decision variables n, π, and μ,

$$\begin{aligned} w^* \equiv \max \quad & \sum_{i \in S} n_i \pi_i - \mu_j V_j y_j^* && && \text{(7a)} \\ \text{s.t.} \quad & \pi_i - \mu_j \leq 0 && \forall i \in S, \forall j \in T(i) && \text{(7b)} \\ & \pi_i \leq 1 && \forall i \in S && \text{(7c)} \\ & n \in \mathcal{N} && && \text{(7d)} \\ & \pi, \mu \geq 0. \end{aligned}$$

Let n^*, π^*, μ^* denote an optimal solution to (7).

Lemma 2. A bin vector $y \in \mathbb{N}^{|T|}$ is feasible for (4) if and only if

$$w^* = \sum_{i \in S} n_i^* \pi_i^* - \mu_j^* V_j y_j \leq 0. \tag{8}$$

Proof. If $\sum_{i \in S} n_i^* \pi_i^* - \mu_j^* V_j y_j \leq 0$, then it implies that the optimal objective value of (6) is non-positive for all $n \in \mathcal{N}$. Then, y is feasible for (4) together with some $x(n) \geq 0$, for each $n \in \mathcal{N}$. Otherwise $w^* > 0$, implying that n^* is a scenario in which, for all $x(n^*)$, y is infeasible for (4). □

If $w^* > 0$, then n^* is used a violated scenario for RCG (to be appended to the formulation) while (8) is a violated valid inequality that can be used with RG. The finite convergence of both methods is established by the following proposition.

Proposition 3. Algorithms RG and RCG converge in a finite number of iterations.

Proof. Following Lemma 2, each iteration either accepts a solution y^* (of (4) with $\mathcal{N}^0$ in place of N) or rejects it because $w^* > 0$, in which case either the inequality (8), or the set of variables and constraints associated to violated scenario n^* (RCG), are appended to the formulation. In either case, the previous solution y^* is no longer feasible for the problem. Then, note that a solution y that is optimal to (4) (together with some x) must satisfy for each $j \in T$, $y_j \leq \max_{n \in \mathcal{N}} ||n||_1$. Hence, at most a finite number of candidate solutions y^* is enumerated by each of the algorithms. □

5.2 *Budgeted uncertainty set*

Quadratic program (7) is nonconvex and accordingly it is hard to solve exactly in general. For some polytopes, problem (7) can be reformulated as a MILP that may be tractable to solve in practice. This is the case, for instance, when considering the following polytope

$$\mathcal{N}_\Gamma = \left\{ n \in \mathbb{N}^{|S|} \;\middle|\; \exists z \in [0,1]^{|S|} : \begin{smallmatrix} \forall i \in S: n_i = \overline{n}_i + \hat{n}_i z_i, \\ \sum_{i \in S} z_i \leq \Gamma \end{smallmatrix} \right\},$$

where $\overline{n}_i$ represents the nominal demand for item type i, $\hat{n}_i$ the maximum possible deviation of the demand from $\overline{n}_i$, and $\Gamma > 0$ is a parameter that controls the desired level of protection or immunity. The quadratic objective function (7a) can be rewritten in this case as

$$\sum_{i \in S} (\overline{n}_i + \hat{n}_i z_i) \pi_i - \mu_j V_j y_j^* = \sum_{i \in S} \overline{n}_i \pi_i + \sum_{i \in S} \hat{n}_i z_i \pi_i - \mu_j V_j y_j^*, \tag{9}$$

where only the second term is (bilinear) quadratic. One readily verifies (see for example Ayoub and Poss. [2016] for details) that there exists an optimal solution to (7) with $\mathcal{N} = \mathcal{N}_\Gamma$, for integer Γ, that satisfies $z \in \{0,1\}^{|S|}$. Therefore, we can linearize the product $z_i \pi_i$, for each $i \in S$, by introducing auxiliary variable η_i and the constraints

$$\eta_i \leq z_i, \qquad \eta_i \leq \pi_i, \qquad \eta_i \geq \pi_i + z_i - 1.$$

Notice that $\mathcal{N}_\Gamma$ satisfies

$$\mathcal{N}_\Gamma \subseteq \mathcal{N}_{\Gamma+1} \tag{10}$$

Further, for integer Γ the number of nondominated extreme points of $\mathcal{N}_\Gamma$ is equal to the number of subsets of S of cardinality Γ. Thus, the number of non-dominated extreme points of $\mathcal{N}_\Gamma$ is the binomial coefficient $\binom{|S|}{\Gamma}$. In particular, if $\Gamma = 0$, or $\Gamma = |S|$, then $\mathcal{N}_\Gamma$ contains one nondominated extreme point, either $\overline{n}$ or $\overline{n} + \hat{n}$, respectively.

Γ	0	1	2	3	4	5	6	7	8	9	10
Cost	77.5	108.8	118.9	126.4	132.1	136.7	139.8	141.9	144	145	145.6
Time RCG (s)	0.9	2.6	3.2	4.6	3.2	3.1	2.7	2.2	2.7	1.7	0.7
Iterations RCG	2.0	4.0	4.4	4.7	4.4	4.2	3.7	3.2	3.1	2.9	2.0
Time RG (s)	6.4	12.8	12.5	11.6	10.4	9	7.6	7.2	6.7	6.1	6.2
Iterations RG	11.6	13.4	13	12.2	11.2	10.9	10.1	12.2	10	10.2	10.9

Table 1. Numerical comparison of RG and RCG on randomly generated instances.

5.3 Numerical experiments

In our numerical experiments we considered randomly generated instances given the parameters $c_{\min} > 0, c_{\max} > c_{\min}$, $V_{\min} > 0$, $V_{\max} > V_{\min}$ and $p \in (0,1)$. For each bin type $j \in T$ we draw its cost c_j and its size V_j, uniformly from the intervals $[c_{\min}, c_{\max}]$ and $[V_{\min}, V_{\max}]$, respectively. For every item type $i \in S$, it is set to be in $S(j)$, for each $j \in T$, with a probability p. We consider a demand uncertainty polytope $\mathcal{N}_\Gamma$, generating $\overline{n}_i$ and $\hat{n}_i$ uniformly from an intervals $[\overline{n}_{\min}, \overline{n}_{\max}]$ and $[\hat{n}_{\min}, \hat{n}_{\max}]$, respectively, for parameters $\overline{n}_{\min}, \overline{n}_{\max}, \hat{n}_{\min}, \hat{n}_{\max}$.

Table 1 reports average numerical results obtained for 10 instances generated for the parameter values $|S| = 10$, $|T| = 15$, $c_{\min} = 1$, $c_{\max} = 10$, $V_{\min} = 1$, $V_{\max} = 10$, $p = 0.5$, $\overline{n}_{\min} = 1, \overline{n}_{\max} = 10, \hat{n}_{\min} = 1, \hat{n}_{\max} = 10$. We solved the instance for all possible values of Γ. It can be observed that, following the inclusion (10), the objective value of the optimal solution increases in Γ. The results of Table 1 indicate, unsurprisingly, that the extreme cases $\Gamma = 0$ and $\Gamma = 10$ are easiest to solve. It can also be observed that the solution time of the RG method exceeds that of the RCG method. It can be expected since the RCG algorithm appends more information to the master problem (violated scenarios including both rows and columns) at each iteration, compared with the RG algorithm that may append only violated cuts.

6. ONGOING AND FUTURE WORK

In the current paper we considered the robust two-stage bin packing problem focusing on the uncertainty of the number of items of different item types. For the special case of unit item sizes we proved closed-form solutions for the optimal solutions. These closed-form solutions provide insight and show that a large number of multipurpose bins is used to address the uncertainty compared to the deterministic case in which at most a single multipurpose bin is used. We developed an algorithm for the two size model for polyhedral or finite item number uncertainty (scenario) sets. The algorithm that we propose is polynomial in the number of items. In future work we are considering algorithms that scale logarithmically in the number of items and linearly in the number of nondominated scenarios. We also plan to extend the row and column generation scheme developed for arbitrary type compatibility structures for an even more general setting with arbitrary item sizes.

REFERENCES

Josette Ayoub and Michael Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Computational Management Science*, 13:219–239, 2016.

Yossi Azar, Joseph Naor, and Raphael Rom. The competitiveness of on-line assignments. *Journal of Algorithms*, 18(2):221–237, 1995.

Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1996.

Aharon Ben-Tal, Alexander Goryashko, Elana Guslitzer, and Arkadi Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98:49–71, 2003. ISSN 0025-5610. doi: 10.1007/s10107-003-0396-4.

Fritz Bökler, Matthias Ehrgott, Christopher Morris, and Petra Mutzel. Output-sensitive complexity of multiobjective combinatorial optimization. *J MultiCrit Decis Anal*, 24:25–36, 2017. ISSN 1057-9214. doi: 10.1002/mcda.1603.

Marin Bougeret, Artur Alves Pessoa, and Michael Poss. Robust scheduling with budgeted uncertainty. *Discrete Applied Mathematics*, 2018. ISSN 0166-218X. doi: https://doi.org/10.1016/j.dam.2018.07.001.

Noam Goldberg and Shlomo Karhi. Packing into designated and multipurpose bins: A theoretical study and application to the cold chain. *Omega*, 71:85–92, 2017.

Shlomo Karhi and Dvir Shabtay. Online scheduling of two job types on a set of multipurpose machines. *International Journal of Production Economics*, 150:155 – 162, 2014. ISSN 0925-5273. doi: http://dx.doi.org/10.1016/j.ijpe.2013.12.015.

Thomas S. McCormick, Scott R. Smallwood, and Frits C. R. Spieksma. A polynomial algorithm for multiprocessor scheduling with two job lengths. *Mathematics of Operations Research*, 26(1):31–49, 2001.

Dvir Shabtay and Shlomo Karhi. Online scheduling of two job types on a set of multipurpose machines with unit processing times. *Computers & Operations Research*, 39 (2):405–412, 2012.

Guopeng Song, Daniel Kowalczyk, and Roel Leus. The robust machine availability problem–bin packing under uncertainty. *IISE Transactions*, 50(11):997–1012, 2018.

Savvas Tassou, G. De-Lille, and Yunting Ge. Food transport refrigeration–approaches to reduce energy consumption and environmental impacts of road transport. *Applied Thermal Engineering*, 29(8-9):1467–1477, 2009.

Bo Zeng and Long Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Oper. Res. Lett.*, 41(5):457–461, 2013. doi: 10.1016/j.orl.2013.05.003.