Robust Two-Stage Packing into Designated and Multipurpose Bins
Yacine Al Najjar, Noam Goldberg, Shlomo Karhi, Michael Poss

To cite this version:
Yacine Al Najjar, Noam Goldberg, Shlomo Karhi, Michael Poss. Robust Two-Stage Packing into Designated and Multipurpose Bins. IFAC-PapersOnLine, Elsevier, 2019, 52 (13), pp.397-402. 10.1016/j.ifacol.2019.11.157. hal-02430107

HAL Id: hal-02430107
https://hal.archives-ouvertes.fr/hal-02430107
Submitted on 7 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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. Copyright ©2019 IFAC

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 \( \alpha, \beta \) 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...
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.

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

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 $A(n) \subseteq \mathbb{R}^{n_1+n_2}$. 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 $A(n)$, 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 $A_1$ and $A_2$ for type 1 and type 2 items, respectively. Accordingly $A(n) = A_1 \times A_2$ 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 $A(n)$ 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 $A_1$ and $A_2$ 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: $a_1 = \max_{a \in A_1} ||a||$ and $a_2 = \max_{a \in A_2} ||a||$.

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, respectively, 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 $P$. For each $p \in P$ 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 $x(n)$, 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 $\lambda_p$ denote the decision variable indicating how many bins of a pattern $p \in P$ are opened then $y_3 = \sum_{p \in P} \lambda_p$. Finally, the two-size robust two-stage integer program is given by the pattern based formulation.
and 2, respectively, is given by

\[ z^* \equiv \min \ y_1 + \alpha y_2 + \beta \sum_{p \in \mathcal{P}} \lambda_p \]

s.t.

\[ x(n)_i \leq \left(\frac{V}{a_i}\right) y_i \quad \forall n \in \mathcal{N}, i = 1, 2 \] (1a)

\[ \sum_{p \in \mathcal{P}} z_i(p) \lambda_p \geq n_i - x(n)_i \quad \forall n \in \mathcal{N}, i = 1, 2. \] (1b)

\[ 0 \leq x(n)_i \leq n_i, \quad \forall n \in \mathcal{N}, i = 1, 2 \] (1c)

\[ \lambda_p \in \mathbb{N} \quad p \in \mathcal{P} \]

\[ y_j \in \mathbb{N}. \quad j = 1, 2, 3 \] (1d)

Let us define

\[ n_{\max}^{1} \equiv \max_{n \in \mathcal{N}} \{n_1 + n_2\}, \quad n_{\max}^{1} \equiv \max_{n \in \mathcal{N}} n_1, \]

and

\[ n_{\max}^{2} \equiv \max_{n \in \mathcal{N}} n_2. \]

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

\[ f(x_1, x_2) = \left[ \frac{x_1}{\left(\frac{V}{a_1}\right)} \right] + \alpha \left[ \frac{x_2}{\left(\frac{V}{a_2}\right)} \right] + \beta \max_{n \in \mathcal{N}} \{\hat{y}_3(n_1 - x_1, n_2 - x_2)\}. \]

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

\[ z_A = \min_{x_1 = 0, \ldots, n_{\max}^{1}, x_2 = 0, \ldots, n_{\max}^{1}} f(x_1, x_2). \]

The subset of nondominated solutions can be computed in output-polynomial time Böcker 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 \( O(n_{\max}^{1} n_{\max}^{2} |\mathcal{N}'| \log n \log^2 V) \).

**Proof.** The algorithm enumerates for \( i = 1, 2 \) all possible values of \( x_1 \) in \( \{0, \ldots, n_{\max}^{1}\} \). For each value of \( (x_1, x_2) \), \( f(x_1, x_2) \) 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 \( \mathcal{N}' \)). In either case that \( \mathcal{N} \) is finite or polyhedral then \( \mathcal{N} \) 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 \( O(n_{\max}^{1} n_{\max}^{2} |\mathcal{N}'| \log n \log^2 V) \). 

Note that the proof of the proposition in the case that \( \mathcal{N} \) is polyhedral relies on the fact that \( \mathcal{N} \) is a subset of the extreme points of \( \mathcal{N} \), which is a finite set. Further, the set of nondominated points \( \mathcal{N}' \) 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

\[ z^* \equiv \min \ y_1 + \alpha y_2 + \beta y_3 \] (3a)

s.t. \[ x(n)_i + x(n)_i = n_i \quad \forall n \in \mathcal{N}, i = 1, 2 \] (3b)

\[ x(n)_i \leq V y_i \quad \forall n \in \mathcal{N}, i = 1, 2 \] (3c)

\[ x(n)_i + x(n)_i \leq V y_3 \quad \forall n \in \mathcal{N}, i = 1, 2 \] (3d)

\[ y_j, x_{ij} \in \mathbb{N} \quad i = 1, 2, j = 1, 2, 3 \]

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

Let \( y_3 = \left[ n_{\max}^{2} \right] - y_1 \) and consider the function \( f: \mathbb{R} \to \mathbb{R} \) given by

\[ f(y_1) = y_1 + (\beta - \alpha)y_3 + \alpha \max \left( \left[ \frac{n_{\max}^{1}}{V} \right] - y_1, \left[ \frac{n_{\max}^{2}}{V} \right] \right) \]

\[ = (1 - \beta)y_1 + \alpha \max \left( \left[ \frac{n_{\max}^{1}}{V} \right], \left[ \frac{n_{\max}^{2}}{V} \right] + y_1 \right) \]

\[ + (\beta - \alpha) \left[ \frac{n_{\max}^{1}}{V} \right]. \]

**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 \( y_3^* \geq \left[ \frac{n_{\max}^{2}}{V} \right] - y_1^* \).

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 \( y_3^* \leq \left[ \frac{n_{\max}^{2}}{V} \right] - y_1^* \).

To see this, suppose for the sake of contradiction that \( y^* \) is optimal for (3) and \( y_3^* > \left[ \frac{n_{\max}^{2}}{V} \right] - y_1^* \). Then, \( y_3^* \geq \left[ \frac{n_{\max}^{2}}{V} \right] - y_1^* + 1 \). Letting \( y^* \) be defined by \( y_i^* = \left\{ \begin{array}{ll} y_1^* & i = 1 \\ y_2^* + 1 & i = 2 \\ y_3^* - 1 & i = 3 \end{array} \right. \), evidently \( y^* \) satisfies (3c)-(3d) and is therefore also feasible for (3). Further, the fact that \( \beta > \alpha \) implies that \( y_1^* + y_2^* + \beta y_3^* < z^* \), thereby establishing a contradiction. Then, feasibility, in particular the fact that \( y^* \) satisfies (3c) for \( i = 2 \), implies that \( y_2^* = \max \left( \left[ \frac{n_{\max}^{1}}{V} \right] - y_1^* - y_3^*, \left[ \frac{n_{\max}^{2}}{V} \right] - y_3^* \right) \).

Otherwise, if \( \beta = \alpha \) then \( y_2^* + y_3^* = \max \left( \left[ \frac{n_{\max}^{1}}{V} \right] - y_1^*, \left[ \frac{n_{\max}^{2}}{V} \right] \right) \), and \( y^* \) must be
optimal since \( y_2^* + y_3^* \) is the minimum number of bins required to store the type 2 items.

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 \( 1 \leq \beta < 1 + \alpha \).

**Proposition 2.** Suppose that \( 1 < \beta < \alpha + 1 \). The optimal objective value of (3) is attained with some \( y^* \) such that \( y_i^* = \left\lfloor \frac{n_i^{\text{max}}}{V} \right\rfloor \). Further, the optimal objective value

\[
z^* = (1 - \beta) \left( \frac{n_1^{\text{max}}}{V} - \frac{n_2^{\text{max}}}{V} \right) + (\beta - \alpha) \left( \frac{n_1^{\text{max}}}{V} \right).
\]

**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 continuously differentiable over \( [0, \left\lfloor \frac{n_1^{\text{max}}}{V} \right\rfloor - \left\lfloor \frac{n_2^{\text{max}}}{V} \right\rfloor, \infty \). For \( y_1 \in [0, \left\lfloor \frac{n_1^{\text{max}}}{V} \right\rfloor - \left\lfloor \frac{n_2^{\text{max}}}{V} \right\rfloor, \infty \), by our assumption on \( \beta \) it follows that \( f \) is decreasing over the interval \( [0, \left\lfloor \frac{n_1^{\text{max}}}{V} \right\rfloor - \left\lfloor \frac{n_2^{\text{max}}}{V} \right\rfloor, \infty \). Thus, \( y_i^* = \left\lfloor \frac{n_i^{\text{max}}}{V} \right\rfloor - \left\lfloor \frac{n_i^{\text{max}}}{V} \right\rfloor \in \arg\min_{y_i \in \mathbb{N}} f(y_1) \). Since \( y_i^* \in \mathbb{N} \cap \mathbb{R}_+ \), it must also be that \( y_i^* \in \arg\min_{y_i \in \mathbb{N}} f(y_1) \).

It follows that in the unit size case only the three scenarios corresponding to \( n_1^{\text{max}}, n_2^{\text{max}} \) and \( n_1^{\text{max}} \) 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

\[ \mathcal{N} = \{ n \in \mathbb{R}^2 \mid b_1^n_1 + b_2^n_2 - r^2, n_1, n_2 \geq 0 \} \]

We have that \( n^{\text{max}} = r/b_1, n^{\text{max}} = r/b_2 \). Further, for \((n_1, n_2) \in \mathcal{N} \) we have \( n_2 = \sqrt{r^2 - b_1^n_1/b_2}, n^{\text{max}} \) is given by the maximum of \( n_1 + n_2 = n_1 + \sqrt{r^2 - b_1^n_1/b_2} \). Let \( g(n_1) = n_1 + \sqrt{r^2 - b_1^n_1/b_2} \). Its maximum is attained as \( n_1 \) that satisfies \( g'(n_1) = 1 - n_1/(b_2 \sqrt{r^2 - b_1^n_1/b_2}) \). So, \( n_1 = g(b_2 \sqrt{r^2/(1+b_2^n_1/b_2)}) \), and \( n^{\text{max}} = g(1) = g(\sqrt{r^2/(1+b_2^n_1/b_2)}) \). 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, \( z^* = 2 \left( \frac{2}{r} - \frac{\sqrt{r}}{V} \right) + (\alpha + 1) \left( \frac{\sqrt{r}}{V} \right) \).

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 Karri [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 bin 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 \( n \in \mathcal{N} \). We model this problem as a mixed-integer linear program (MILP), which may contain an infinite number of variables and constraints,

\[
z^* = \min \sum_{j \in M} c_j y_j \quad \text{s.t.} \quad \sum_{j \in T(i)} x(n)_{ij} = n_i \quad \forall i \in S, n \in \mathcal{N} \]

\[
\sum_{j \in T(j)} x(n)_{ij} \leq V_j y_j \quad \forall j \in T, n \in \mathcal{N} \]

\[
y_j \in \mathbb{N} \quad \forall j \in T, i \in S, n \in \mathcal{N} \]

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 bin 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 \( \mathcal{N} \). 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 \( y^0 \in \mathbb{N}^{|T|} \). 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 \( x(n^*), j \) 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 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 \( \mathcal{N} \)) 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 \( j \in T \), \( y_j \leq \max_{i \in \mathcal{N}} \|v_i\|_1 \). 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

\[
\mathcal{N}_\Gamma = \left\{ n \in \mathbb{N}^{|\mathcal{S}|} : \exists z \in [0, 1]^{|\mathcal{S}|} : \forall i \in \mathcal{S} : \pi_i + z_i \approx \hat{\pi}_i, \sum_{i \in \mathcal{S}} z_i \leq \Gamma \right\},
\]

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

\[
\sum_{i \in \mathcal{S}} (\pi_i + z_i - \hat{\pi}_i) - \mu_j V_j y_j = \sum_{i \in \mathcal{S}} \pi_i \xi_i + \sum_{i \in \mathcal{S}} \hat{\pi}_i z_i - \mu_j V_j y_j^*,
\]

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 \( \Gamma \), that satisfies \( z \in \{0, 1\}^{|\mathcal{S}|} \). Therefore, we can linearize the product \( z_i \pi_i \), for each \( i \in \mathcal{S} \), by introducing auxiliary variable \( \eta_i \) and the constraints

\[
\eta_i \leq z_i, \quad \eta_i \leq \pi_i, \quad \eta_i \geq \pi_i + z_i - 1.
\]

Notice that \( \mathcal{N}_\Gamma \) satisfies

\[
\mathcal{N}_\Gamma \subseteq \mathcal{N}_\Gamma + 1
\]

Further, for integer \( \Gamma \) the number of nondominated extreme points of \( \mathcal{N}_\Gamma \) is equal to the number of subsets of \( \mathcal{S} \) of cardinality \( \Gamma \). Thus, the number of non-dominated extreme points of \( \mathcal{N}_\Gamma \) is the binomial coefficient \( \binom{|\mathcal{S}|}{\Gamma} \). In particular, if \( \Gamma = 0 \), or \( \Gamma = |\mathcal{S}| \), then \( \mathcal{N}_\Gamma \) contains one nondominated extreme point, either \( \pi \) or \( \pi + \hat{\pi} \), respectively.
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 $\mathcal{A}_\Gamma$, generating $\pi_i$ and $\hat{n}_i$ uniformly from an interval $[\pi_{\min}, \pi_{\max}]$ and $[\hat{n}_{\min}, \hat{n}_{\max}]$, respectively, for parameters $\pi_{\min}, \pi_{\max}, \hat{n}_{\min}, \hat{n}_{\max}$.

Table 1 reports average numerical results obtained for 10 instances generated for the parameter values $|S| = 10, |T| = 15, c_{\min} = 1, c_{\max} = 10, V_{\min} = 1, V_{\max} = 10, p = 0.5, \pi_{\min} = 1, \pi_{\max} = 10, \hat{n}_{\min} = 1, \hat{n}_{\max} = 10$. We solved the instance for all possible values of $\Gamma$. It can be observed that, following the inclusion (10), the objective value of the optimal solution increases in $\Gamma$. 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.

|\(\Gamma\)| 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.

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


