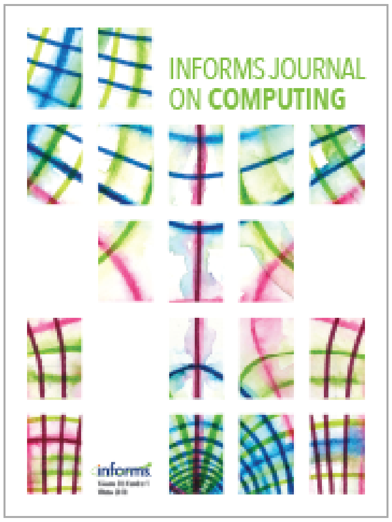


INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Branch and Price for Chance-Constrained Bin Packing

Zheng Zhang, Brian T. Denton, Xiaolan Xie

INFORMS JOURNAL ON COMPUTING
Vol. 32, No. 3, Summer 2020, pp. 547–564
ISSN 1091-9856 (print), ISSN 1526-5528 (online)
http://pubsonline.informs.org/journal/ijoc

Branch and Price for Chance-Constrained Bin Packing

Zheng Zhang,[a,b] Brian T. Denton,[b] Xiaolan Xie[c,d]

[a] Department of Service Science and Operations Management, School of Management, Zhejiang University, 310058 Hangzhou, China; [b] Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Michigan 48109; [c] Mines Saint-Etienne, Université Clermont Auvergne, CNRS, UMR 6158 LIMOS, Centre CIS, F-42023 Saint-Etienne, France; [d] Antai College of Economics and Management, Shanghai Jiao Tong University, 200030 Shanghai, China

Contact: zhazheng@umich.edu, http://orcid.org/0000-0002-6943-3723 (ZZ); btdenton@umich.edu, http://orcid.org/0000-0002-6372-6066 (BTD); xie@emse.fr, http://orcid.org/0000-0002-6579-1523 (XX)

Received: April 6, 2017
Revised: March 24, 2018; September 23, 2018; December 21, 2018; February 13, 2019
Accepted: February 15, 2019
Published Online in Articles in Advance: January 14, 2020

https://doi.org/10.1287/ijoc.2019.0894

Abstract. This article describes two versions of the chance-constrained stochastic bin-packing (CCSBP) problem that consider item-to-bin allocation decisions in the context of chance constraints on the total item size within the bins. The first version is a stochastic CCSBP (SP-CCSBP) problem, which assumes that the distributions of item sizes are known. We present a two-stage stochastic mixed-integer program (SMIP) for this problem and a Dantzig–Wolfe formulation suited to a branch-and-price (B&P) algorithm. We further enhance the formulation using coefficient strengthening and reformulations based on probabilistic packs and covers. The second version is a distributionally robust CCSBP (DR-CCSBP) problem, which assumes that the distributions of item sizes are ambiguous. Based on a closed-form expression for the DR chance constraints, we approximate the DR-CCSBP problem as a mixed-integer program that has significantly fewer integer variables than the SMIP of the SP-CCSBP problem, and our proposed B&P algorithm can directly solve its Dantzig–Wolfe formulation. We also show that the approach for the DR-CCSBP problem, in addition to providing robust solutions, can obtain near-optimal solutions to the SP-CCSBP problem. We implement a series of numerical experiments based on real data in the context of surgery scheduling, and the results demonstrate that our proposed B&P algorithm is computationally more efficient than a standard branch-and-cut algorithm, and it significantly improves upon the performance of a well-known bin-packing heuristic.

History: Accepted by Andrea Lodi, Area Editor for Design and Analysis of Algorithms.
Funding: This research was funded in part by the National Science Foundation Division of Civil, Mechanical and Manufacturing Innovation [Grant CMMI-0844511] (Denton). This research was also funded in part by the National Natural Science Foundation of China [Grants 71671111, 71801058, and 71432006].
Supplemental Material: The online supplement is available at https://doi.org/10.1287/ijoc.2019.0894.

Keywords: bin packing • chance constraints • branch and price • surgery scheduling

1. Introduction

Bin packing has been studied in several contexts, including computer scheduling (Coffman et al. 1978), internet advertising (Adler et al. 2002), and operating-room management (Denton et al. 2010). Stochastic versions of this problem typically assume that the item sizes are random variables, the items are allocated to bins before knowing the random outcome of their sizes, and the bins are *extensible* such that the bin size extends (at a cost) to the size necessary to contain the items. Thus, the problem becomes one of minimizing the fixed cost on the number of bins necessary to pack the items and the expected cost of extending the bins. In this article, we study a generalization of this problem, which we refer to as the *chance-constrained stochastic bin-packing* (CCSBP) problem in which there are chance constraints on the sum of the item sizes within each bin. Thus, the optimal allocation of items to bins is one that minimizes the expected cost while, at the same time, meeting all chance constraints.

The stochastic version of the CCSBP (SP-CCSBP) problem assumes that the distributions of item sizes are known. It can be formulated as a large-scale mixed-integer program (MIP) based on a finite number of scenarios for the items' sizes. However, this problem is difficult to solve for several reasons. First, it lacks convexity because of the binary item-to-bin allocation variables and the fact that chance constraints are generally nonconvex (Luedtke and Ahmed 2008). Second, the linear programming (LP) relaxation of the MIP is often weak because of the *big-M* coefficients that are introduced to model the chance constraints. Third, there are many identical solutions for problems in which there are some identical bins and items. To overcome these challenges, we propose a branch-and-price (B&P) approach that is tailored to this problem. We show that it eliminates symmetry and, most importantly, provides opportunities to exploit properties of the binary bin-packing problem that defines the *subproblem*. In addition to presenting a method for the SP-CCSBP problem, we also present a distributionally robust version of the CCSBP (DR-CCSBP) problem, which assumes that the distributions of item sizes are ambiguous. We approximate the DR-CCSBP problem as an MIP based on a closed-form expression for the worst-case violated probability concerning the chance constraints.

We use surgery scheduling in hospitals as a motivating example for our work. In this context, the bins are operating rooms (ORs) that are available for some nominal period during the day of surgery (e.g., 10 hours). The items represent surgeries, and the item sizes represent the surgery durations. Surgery scheduling is important in healthcare delivery because of the high cost of OR utilization and OR staff overtime. The chance constraints are particularly important to provide surgical teams with appropriate "end-of-the-day" guarantees. The SP-CCSBP problem is suitable for situations in which the decision maker trusts the estimated distributions of surgery durations, and the DR-CCSBP problem is suitable for cases in which the decision maker only trusts the estimated mean and variance of the surgery duration distribution. For example, this situation may arise when the sample size is insufficient to be confident about the complete distribution. In addition to this case, the DR-CCSBP problem is also much easier to solve, and thus, its solution approach can serve as an effective approximation approach for the SP-CCSBP problem when it is computationally intractable.

In this article, we focus on two main research questions. The first research question is: what are the advantages of using the B&P algorithm as a solution method to solve the chance-constrained bin-packing problem? We describe a number of properties of the problem that can be exploited to achieve computational advantages in our B&P algorithm implementation. We use numerical experiments based on examples of surgery scheduling to illustrate the performance of our approach compared with a standard branch-and-cut (B&C) approach. The second research question is: what is the value of the DR-CCSBP solution compared with the SP-CCSBP solution? We investigate the formulations for the SP-CCSBP problem and the DR-CCSBP problem, respectively, to compare their computational efficiency and the scheduling performance under ambiguity in distributions of item sizes.

The remainder of this article is organized as follows. Section 2 reviews the literature on the stochastic bin-packing problem and highlights the novel contributions of this article. Section 3 defines the stochastic version of the CCSBP problem and formulates it as a general two-stage stochastic-programming formulation with integer recourse. Section 4 reformulates the problem using Dantzig–Wolfe decomposition, presents a B&P algorithm, and analyzes theoretical properties of the formulation that can be exploited to accelerate the computation. Section 5 describes the DR version of the CCSBP problem and its approximation as an MIP. Section 6 presents computational results and managerial insights in the context of a surgery-scheduling case study. Finally, Section 7 concludes the paper with a summary of the most important findings and opportunities for future work.

2. Literature Review

This section reviews the literature on stochastic bin packing and relevant methods, including column generation, B&P, and distributionally robust optimization. We close this section with a description of the main contributions of this article to the literature.

2.1. Stochastic Bin Packing

A significant portion of the literature on stochastic bin-packing problems is in the context of surgery scheduling. For example, Denton et al. (2010) considered the stochastic extensible bin-packing problem in the context of allocating surgeries (items) to operating rooms (bins). The objective included a fixed cost for each utilized OR and a variable cost for expected overtime. They proposed a two-stage stochastic mixed-integer program (SMIP) to account for uncertainty in surgery durations and also a robust optimization model to account for ambiguous distributions when only lower and upper bounds are available on the surgery durations. Batun et al. (2011) considered a related problem in which some surgeries could be completed in parallel, in different ORs, by the same primary surgeon assisted by surgical fellows. In both cases, L-shaped methods were used to solve the problem with additional cuts to reduce symmetry among the first-stage decision variables.

The addition of chance constraints was motivated by the challenges of attributing a cost to OR overtime in practice because much of the cost is due to a loss of goodwill among nursing staff and is not attributable to the monetary cost of overtime alone (Deng et al. 2020). The addition of chance constraints, although attractive for practical reasons, creates computational challenges because the joint cumulative probability density function of item sizes is generally nonconvex except for certain cases, such as when all item sizes have a log-concave probability distribution (Prékopa 1995, theorem 10.2.1). However, surgery durations are more often associated with distributions that have a right tail, such as the lognormal distribution (Strum et al. 2003), which is not log-concave. Shylo et al. (2012) appears to be the first to consider a chance-constrained bin-packing problem. They assumed that the objective was to minimize the number of ORs to open while satisfying a set of probabilistic capacity constraints. Their chance-constrained problem was expressed as a series of MIPs based on a *normal approximation* of cumulative surgery durations. However, this approximation is not likely to be accurate when the number of surgeries is small (e.g., $n \leq 4$), which is common for many hospital inpatient surgery practices. Deng et al. (2020) considered a joint OR-scheduling problem similar to Batun et al. (2011) but for which the surgeon waiting times are subject to individual chance constraints and the overtime in all ORs is subject to a joint chance constraint. They proposed a two-stage SMIP with integer recourse based on a *sample average approximation* (SAA). Further, to speed up the computation, they used a cutting-plane algorithm and lifting methods based on problem decompositions on scenarios and ORs, respectively.

Lamiri et al. (2008) used column generation to solve OR-allocation problems combined with surgery-date decisions. In their work, uncertainty exists in the number of emergency surgeries that must share ORs with elective surgeries. They assumed that surgery durations are deterministic and solved the subproblem as deterministic knapsack problems using dynamic programming and the integer master problem using heuristics. In a related study, Wang et al. (2014) used column-generation approaches to solve an OR-allocation problem in which they assumed that surgery durations are stochastic. The OR completion times were chance-constrained, and the risk was evaluated by an independent *Monte-Carlo simulation* on each generated column. The master problem was solved by using heuristics to obtain integer solutions.

2.2. Branch and Price

Solution methods for SMIPs often employ Benders decomposition (Benders 1962) or extensions thereof. Examples in the bin-packing context include most of the previously cited references: Denton et al. (2010), Batun et al. (2011), and Deng et al. (2020). Unfortunately, these decompositions use a master problem whose LP relaxation is no stronger than that of the original model (Silva and Wood 2006), leading to a poor continuous relaxation of the formulation, especially when big-M coefficients exist, such as is the case for a common reformulation of chance-constrained problems as we show in Section 3. In contrast, the B&P algorithm solves a column-oriented reformulation of the model, also by a form of decomposition, but with a tighter relaxation bound. A comprehensive description of B&P algorithms is provided in Vanderbeck and Wolsey (1996), Barnhart et al. (1998), and Vanderbeck (2000). Here, we focus on B&P algorithms applied to SMIPs and literature related to our specific problem.

Silva and Wood (2006) surveyed B&P algorithms for two-stage SMIPs, such as the general allocation problem, the routing and scheduling problem, the crew-scheduling problem, and the integer multicommodity flow problem. They also implemented B&P algorithms for a stochastic facility-location problem and reported that this method could be orders of magnitude faster than solving the original problem by branch and bound (B&B). The B&P algorithm has also been used to solve capacitated lot-sizing problems and multistage stochastic capacity-planning problems by Degraeve and Jans (2007) and Singh et al. (2009), respectively. Dinh et al. (2018) proposed a B&P-and-cut algorithm for a chance-constrained vehicle-routing problem with correlations between random customer demands of normal distributions. Hashemi Doulabi et al. (2016) proposed a B&P-and-cut algorithm for an integrated operating room planning and scheduling problem in which the day of surgery, allocation of surgeries to ORs and the time slot for each surgery were integrally optimized. The subproblem was reformulated as a constraint-programming model enhanced by dominance rules to improve the computational efficiency. However, the authors did not consider the uncertainties in surgery durations and the associated chance constraints that raise the major challenge to our problem.

To the best of our knowledge, only Lulli and Sen (2004) incorporated chance constraints in a B&P method; the authors solved a multistage stochastic batch-sizing problem for both recourse and probabilistic constrained formulations. Their computational results showed that the B&P algorithm was faster than the B&B algorithm in the model without chance constraints; however, because of the challenges imposed by the special structure of chance constraints, the B&P algorithm failed to outperform the B&B algorithm.

Recently, Song et al. (2014) provided a *probabilistic cover*–based reformulation to explore the structure of chance-constrained binary packing problems with a single bin. Their reformulation casts the SMIP as an MIP based on probabilistic covers. They also used deterministic cover inequalities to perform approximate lifting of probabilistic cover inequalities. As we show in Section 4, some of the properties they describe can be exploited in a B&P approach for the SP-CCSBP problem.

2.3. Distributionally Robust Optimization

Distributionally robust stochastic programs focus on the worst-case result under a confidence region that defines how far the true item sizes may be from their point estimates. There are several types of expressions for the confidence region regarding moments (e.g., Scarf 1958), confidence regions of moments (e.g., Delage and Ye (2010)), and support (e.g., Denton et al. 2010). DR chance-constrained stochastic programs were considered by Jiang and

Guan (2016), who presented a series of stochastic programs depending on different types of confidence regions.

Several papers have recently addressed DR solutions in scheduling problems. Kong et al. (2013) considered a stochastic appointment-scheduling problem to determine arrival times for a sequence of customers. The item sizes are stochastic, and only the mean and covariance estimates are known. They formulated a copositive programming model to minimize the expected cost over a worst-case distribution in a region defined by moments and nonnegative support of item sizes. As the problem was not necessarily polynomial-time solvable, they further proposed a semidefinite programming relaxation as a solution approach. Mak et al. (2015) considered a similar problem except that item sizes were assumed to be independently distributed. With this assumption, they showed that the DR model could be formulated as a second-order conic program based on marginal moments, which can be solved more easily. They also presented a closed form of the worst-case distribution; however, their formulation requires the assumption that item sizes could take on negative values.

Deng et al. (2020) also considered a DR variant of the chance-constrained OR-scheduling problem to restrict the maximum risk of surgery delay and overtime. They used an ambiguity set based on statistical divergence functions, which is equivalent to a chance-constrained program evaluated on an empirical probability density function but with smaller risk tolerances. Once the risk tolerance is fixed, their DR model becomes an SAA to a chance-constrained program. Our DR model is similar to the first version of the DR chance-constrained binary programs as considered in Zhang et al. (2018). Our contributions to this aspect of our study are in the use of new methods for solving the problem and numerical experiments that we present. Zhang et al. (2018) formulated the DR model as a 0-1 second-order cone (SOC) formulation and utilized the submodularity of the 0-1 SOC constraints to derive inequalities to strengthen the formulation. In contrast, we approximated the problem as an MIP formulation and developed a B&P algorithm to solve the problem.

2.4. Contributions to the Literature

Our approach differs from the aforementioned literature in two main aspects: First, we present an "integrated" B&P algorithm that combines column generation with multiple techniques, including coefficient strengthening and subproblem reformulations based on probabilistic cover/pack constraints. We show that this integration significantly improves the computational performance of the algorithm over standard methods, such as the B&C algorithm. Second, we approximate the DR-CCSBP problem as an MIP that has significantly fewer integer variables compared with the SMIP for the SP-CCSBP problem. More importantly, we bridge between the two problems by proposing a method to compute an adjusted probability tolerance. As a result, the approach for the DR-CCSBP problem can give good approximate solutions to the SP-CCSBP problem with significantly less computational effort and information about item-size distributions.

3. Stochastic Version of the CCSBP Problem

We first consider a stochastic version of the CCSBP problem, which we refer to as the *SP-CCSBP* problem, which assumes that the distributions of item sizes are known. We formulate the SP-CCSBP problem as a two-stage SMIP based on an SAA (Luedtke and Ahmed 2008). There are n items to be allocated among m bins (we assume that $n > m$ to avoid the trivial case). There are K bin types that have chance constraints concerning a bin extension threshold, δ_k, and a probability tolerance, α_k, for each bin type $k = 1, \cdots, K$. There may also exist bins without any chance constraints, which we refer to as another bin type, $K + 1$. For simplicity of exposition and commonality in practice, we assume that all bins have the same bin size, T; however, this assumption can be relaxed. We let $\hat{T}_k = T + \delta_k$ represent the extended size for chance-constrained bins of type k. We assume that each bin must contain at least one item. We let $\mathcal{I}$ denote the set of items and $\mathcal{R}$ denote the set of bins. We let $\mathcal{R}_k$ denote the set of chance-constrained bins of type k, and m_k is the number of bins of type k. We let Ω denote the finite set of all possible scenarios based on the SAA. Finally, we let $\xi_i(\omega)$ denote the size of item i in scenario $\omega \in \Omega$, where ω indexes a finite set of realizations of the random outcome vector $\xi(\omega) = \{\xi_1(\omega), \cdots, \xi_n(\omega)\}$. Indices i and j index items, r indexes bins, k indexes bin types, and ω indexes scenarios.

3.1. Notation

We denote the set of nonnegative reals by $\mathbb{R}_+$, the probability of an event e by $\mathbb{P}(e)$, the cardinality of a set X by $|X|$, the ceiling of a faction x by $\lceil x \rceil$, and the vector or matrix form of a scalar x by $\mathbf{x}$.

We consider the following decision variables:

- First-stage decision variable $\mathbf{x} \in \{0, 1\}^{|\mathcal{I}| \times |\mathcal{R}|}$ such that $x_{ir} = 1$ if item i is allocated to bin r and $x_{ir} = 0$ otherwise.
- Second-stage decision variable $\mathbf{o} \in \mathbb{R}_+^{|\mathcal{R}| \times |\Omega|}$ such that $o_r(\omega)$ represents the extension for bin r in scenario ω.

The extension threshold of a chance-constrained bin of type k can be exceeded with at most probability α_k. Thus, chance constraints can be expressed as follows:

$$\mathbb{P}\left(\sum_{i\in\mathcal{I}} \xi_i(\omega)x_{ir} \le \hat{T}_k\right) \ge 1-\alpha_k,\ \forall k=1,\cdots,K,\ \forall r\in\mathcal{R}_k, \tag{1}$$

where $\sum_{i\in\mathcal{I}} x_{ir}\xi_i(\omega)$ is the total size of items in bin r in scenario ω.

Except some special cases, the function on the left-hand side of (1) is nonlinear and nonconvex. To express it as a linear function with the help of additional binary variables, we use the common approach of introducing auxiliary binary variables, $\mathbf{z}\in\{0,1\}^{(|\mathcal{R}_1|+\cdots+|\mathcal{R}_K|)\times|\Omega|}$, such that $z_{kr}(\omega)=1$ if bin r satisfies the chance constraint of type k in scenario ω and $z_{kr}(\omega)=0$ otherwise. The extensive formulation of the SP-CCSBP problem, which we refer to as (P), can be formulated as follows:

$$(\mathbf{P}):\min \sum_{r\in\mathcal{R}} \mathbb{E}_{\omega\in\Omega}[o_r(\omega)] \tag{2a}$$

$$\text{s.t.} \sum_{r\in\mathcal{R}} x_{ir}=1, \quad \forall i\in\mathcal{I} \tag{2b}$$

$$o_r(\omega) \ge \sum_{i\in\mathcal{I}} \xi_i(\omega)x_{ir} - T, \quad \forall r\in\mathcal{R}, \forall\omega\in\Omega \tag{2c}$$

$$\sum_{i\in\mathcal{I}} \xi_i(\omega)x_{ir} \le \hat{T}_k + M_{rk}(\omega)(1-z_{kr}(\omega)), \quad \forall k=1,\cdots,K, \forall r\in\mathcal{R}_k, \forall\omega\in\Omega \tag{2d}$$

$$\sum_{\omega\in\Omega} z_{kr}(\omega) \ge \lceil(1-\alpha_k)|\Omega|\rceil, \quad \forall k=1,\cdots,K, \forall r\in\mathcal{R}_k \tag{2e}$$

$$\mathbf{x},\mathbf{z}\ \text{binary}, \mathbf{o}\ge 0, \tag{2f}$$

where $M_{rk}(\omega)$ are big-M coefficients. The objective (2a) is to minimize the sum of the expected bin extension. Constraint (2b) ensures that each item is allocated to exactly one bin. Constraint (2c) determines the bin extension in each scenario. Constraints (2d) and (2e) are the linearization of chance constraints, where (2d) determines $z_{kr}(\omega)$ and (2e) enforces the probability tolerance.

Remark 1. Our formulation assumes that the number of bins used is fixed at m. This is consistent with the observations at our collaborating hospitals in which all ORs are utilized daily. This assumption is consistent with the first formulation of the extensible bin-packing problem presented by Dell'Olmo et al. (1998) but slightly different from Denton et al. (2010) in which the decision to utilize a bin is a binary decision variable. To consider the latter case, our formulation (as well as all formulations in Sections 4 and 5) should be revised with minor changes to account for additional binary variables for determining whether to open each bin. The extensive formulations (e.g., (P)) can be revised to be similar as the stochastic OR-allocation formulation of Denton et al. (2010). The Dantzig–Wolfe formulations, as we show in Remark 4, can also be revised with minor changes, and our proposed B&P algorithm can be directly applied to the revised formulations.

Remark 2. We assume that there exists a bin type (i.e., type $K+1$) without chance constraints in our description of the formulation. This assumption is not a requirement of the formulation. We allow this possibility to make the formulation general enough to accommodate the case of surgery scheduling, in which some ORs are planned to remain open as long as necessary to complete all cases.

Remark 3. There may exist inconsistency issues in two-stage, chance-constrained stochastic programs with recourse as reported by Takriti and Ahmed (2004) and Liu et al. (2016). Specifically, constraint (2d) only enforces $z_{rk}(\omega)$ to be zero when the chance constraint is not satisfied. The inconsistency could arise if $z_{rk}(\omega)$ is zero even if the chance constraint is satisfied. However, in formulation (P), this inconsistency does not occur because if it does (i.e., chance constraint (2e) is satisfied, $z_{kr}(\omega)=0$, and $\sum_i \xi_i(\omega)x_{ir}\le\hat{T}_k$ is true for some ω), then one can change $z_{kr}(\omega)$ from zero to one and change $z_{kr}(\omega')$ for some other ω' from one to zero without violating constraint (2e). It means that we remove constraint (2d), which was a binding constraint for ω', and thus, the objective function can be further reduced, that is, yielding a better solution. For the same reason, this inconsistency does not occur in the rest of the formulations with chance constraints in this article.

As noted earlier, our problem has symmetry concerning bins of the same type. Therefore, eliminating symmetric solutions may be beneficial to reduce the computation time. We add the following antisymmetry constraints, which are extended from Denton et al. (2010), to formulation (P):

$$\sum_{k=1}^{K+1}\sum_{r=\Gamma_k+1}^{\Upsilon_{ik}} x_{ir}=1, \qquad \forall i=1,\cdots,\max_k m_k \tag{3a}$$

$$\sum_{r'=\Gamma_k+r}^{\Upsilon_{ik}} x_{ir'} \le \sum_{j=r-1}^{i-1} x_{j,\Gamma_k+r-1}, \quad \forall k=1,\cdots,K, \forall i=2,\cdots,n, \quad \forall r=1,\cdots,\min(i,m_k), \tag{3b}$$

where $\Gamma_k=\sum_{v=1}^{k-1} m_v$ and $\Upsilon_{ik}=\min(\Gamma_k+i,\Gamma_{k+1})$. The constraint set (3) ensures that items are allocated to bins in lexicographic order. Specifically, constraint (3a) requires that item i should be allocated to one of the first i bins of a specific type; constraint (3b) requires that, if the $(\hat{r}-1)^{th}$ bin of a specific type is not allocated for any of the first $i-1$ items, then item i should not be allocated to the $\hat{r}^{th}$ bin of that type. Big-M coefficients usually present a challenge to solving the MIP. Strengthening the coefficients ahead of time has been shown to reduce the computation time considerably (Qiu et al. 2014). Given a particular scenario, ω', and a particular bin, r', of type k', the big-M coefficient, $M_{r'k'}(\omega')$, is valid if it is greater than or equal to

$$\bar{M}_{r'k'}(\omega') = \max \sum_{i' \in \mathcal{I}} x_{i'r'} \xi_{i'}(\omega') - \hat{T}_{k'},$$
$$\text{s.t. (2b)-(2e) are satisfied for all } i \in \mathcal{I}, \quad k = 1, \cdots, K, r \in \mathcal{R}_k, \omega \in \Omega,$$
$$\mathbf{x}, \mathbf{z} \text{ binary.}$$

In fact, $M_{r'k'}(\omega')$ achieves its lowest feasible value when it reaches $\bar{M}_{r'k'}(\omega')$. However, solving for $\bar{M}_{r'k'}(\omega')$ is equivalent to solving formulation (P) because they have the same number of integer decision variables and big-M formulated chance constraints. Instead, we relax the formulation and estimate M_{rk} for individual bins. The resulting problem is the same as a subproblem formulation of our Dantzig–Wolfe decomposition, so we present the coefficient strengthening later in Section 4.2.3.

Although formulation (2) can be strengthened as described, it is still difficult to solve. Our preliminary results show that the strengthened formulation is solvable via a sophisticated MIP solver, such as CPLEX 12.6, only for problem instances that are much smaller than our target application. The integer L-shaped method is also well known to be inefficient for formulations such as this because of weak optimality cuts resulting from big-M coefficients (Codato and Fischetti 2006). There exist recent adaptations of Benders decomposition to reformulate big-M coefficients (Codato and Fischetti 2006, Luedtke 2014, Liu et al. 2016); however, as we show, our B&P approach offers a means to decompose the extensive formulation of the SP-CCSBP problem into subproblems with special structures that can be exploited to achieve computational efficiency.

4. Branch-and-Price Algorithm

In this section, we present a B&P algorithm based on Dantzig–Wolfe formulations of the SP-CCSBP problem. Further, we present some analytic properties of the problem structure that can be used to accelerate the algorithm. The algorithm can also apply to solve the DR-CCSBP problem as discussed in Section 5.

4.1. Dantzig–Wolfe Decomposition

The Dantzig–Wolfe decomposition of the SP-CCSBP problem consists of a *master problem* and several *subproblems*, which are connected by *columns*. Each column, indexed by p, represents a candidate allocation to a bin that comes with the following information:

y_{ip}: binary parameter representing whether item i is in column p.

c_{kp}: binary parameter representing whether the total item size in column p is chance-constrained for bins of type k.

$\bar{o}_p$: expected bin extension for column p over all scenarios $\omega \in \Omega$.

4.1.1. Master Problem. For a given pool ($\mathcal{S}$) of columns, the master problem selects columns from the pool to minimize the total expected bin extensions, and the subproblem generates new columns that are added to the pool until the relaxation of the master problem is solved to optimality.

The decision variables in the master problem are as follows:

- $\lambda \in \{0,1\}^{|\mathcal{S}|}$ such that $\lambda_p = 1$ if column p is selected and $\lambda_p = 0$ otherwise.

We formulate the master problem (MP) as the following mixed-integer program:

$$(\mathbf{MP}): \min_{\lambda \in \{0,1\}^{|\mathcal{S}|}} \sum_{p \in \mathcal{S}} \bar{o}_p \lambda_p \tag{4a}$$
$$\text{s.t.} \sum_{p \in \mathcal{S}} y_{ip} \lambda_p = 1, \quad \forall i \in \mathcal{I} \tag{4b}$$
$$\sum_{p \in \mathcal{S}} c_{kp} \lambda_p = m_k, \quad \forall k = 1, \cdots, K \tag{4c}$$
$$\sum_{p \in \mathcal{S}} \lambda_p = m. \tag{4d}$$

The objective (4a) is to minimize the total expected bin extensions. Constraint (4b) ensures that each item is allocated to a bin. Constraint (4c) ensures that the total number of chance-constrained columns equals the number of chance-constrained bins of each type. Constraint (4d) limits the number of selected columns to be equal to the total number of bins.

When $\mathcal{S}$ is the complete pool of all possible columns, formulation (MP) represented by (4) is equivalent to the extensive formulation (P) represented by (2) because any feasible allocation of items to bins in (2) can be regarded as feasible columns in (4); conversely, any feasible solution of (4) can be converted to a feasible allocation in (2). Unfortunately, (MP) is, generally speaking, intractable because of a large number of columns in $\mathcal{S}$ and the integer restriction of the solution. Thus, we relax λ to be $\lambda_p \geq 0$, $\forall p \in \mathcal{S}$, and we refer to the resulting relaxed master problem as (RMP). Moreover, we consider a restricted pool with a limited number of columns, and we refer to the restricted (RMP) as (R^2MP). Next, we show how we can iteratively solve (R^2MP) and its subproblems to find an optimal solution to (MP).

4.1.2. Subproblem. For a given pool $\mathcal{S}$, we solve (R^2MP) and obtain the following dual solutions:

π_i: dual price corresponding to item i being allocated to a bin (constraint (4b)).

ϕ_k: dual price corresponding to all bin extensions of type k being chance-constrained (constraint (4c)).

φ: dual price corresponding to the total number of bins being equal to m (constraint (4d)).

Once (R^2MP) is solved using an initial restricted pool of columns, the preceding dual solutions are passed to the subproblems for generating new columns to $\mathcal{S}$. The subproblem has the following decision variables:

- first-stage decision variable $\mathbf{Y} \in \{0,1\}^{|\mathcal{I}|}$ such that $Y_i = 1$ if item i is selected in the new column and $Y_i = 0$ otherwise.
- first-stage decision variable $\mathbf{V} \in \{0,1\}^K$ such that $V_k = 1$ if the new column satisfies the chance constraint of type k and $V_k = 0$ otherwise.
- second-stage decision variable $\mathbf{O} \in \mathbb{R}_+^{|\Omega|}$ such that $O(\omega)$ is the bin extension of the new column in scenario ω.

The generic subproblem is to generate a new column that maximizes the sum of dual prices minus the expected bin extension of the column. When the difference is positive, the new column is improving to (R^2MP); otherwise, the (RMP) is already solved to optimality. (R^2MP) and the generic subproblem are solved iteratively until the generic subproblem cannot generate any improving column.

The generic subproblem is generally difficult to solve because it has chance constraints that are represented by the decision variables, V_k. Because this subproblem may be solved many times before the SP-CCSBP problem is solved to optimality, the efficiency of solving this subproblem is a key factor that determines the performance of the B&P algorithm. To address this point, we discuss four different formulations of the generic subproblem in Sections 4.2 and 4.3.

Remark 4. Our Dantzig–Wolfe decomposition is based on a fixed number of ORs that are assumed in Remark 1. However, our formulations can be adapted to consider a variable number of ORs with opening costs by making the following change:

- Remove constraint (4d) from (MP) and set φ as the unit cost of opening an OR for a day for the subproblem formulations.

The B&P algorithm as proposed in Section 4.4 can be directly applied to the revised Dantzig–Wolfe formulations for both the SP-CCSBP and the DR-CCSBP problems.

4.2. Scenario-Based Subproblem

In this section, we formulate the generic subproblem based on the scenarios, including an extensive formulation that solves the subproblem once and a bin type–specific formulation that solves an independent subproblem for each type of bin.

4.2.1. Extensive Formulation of the Scenario-Based Subproblem. We introduce auxiliary decision variables, $\mathbf{Z} \in \{0,1\}^{K\times|\Omega|}$, such that $Z_k(\omega) = 1$ if the to-be-generated column satisfies the chance constraint of type k in scenario ω and $Z_k(\omega) = 0$ otherwise. We then formulate the extensive formulation of the scenario-based subproblem, (SSP), as the following two-stage SMIP:

$$(\mathbf{SSP}): \zeta^{SSP} = \max \sum_{i\in\mathcal{I}} \pi_i Y_i + \sum_{k=1}^{K} \phi_k V_k + \varphi - \mathbb{E}[O(\omega)] \tag{5a}$$

$$\text{s.t.} \sum_{k=1}^{K} V_k \le 1 \tag{5b}$$

$$\sum_{i\in\mathcal{I}} \xi_i(\omega) Y_i - O(\omega) \le T, \quad \forall \omega \in \Omega \tag{5c}$$

$$\sum_{i\in\mathcal{I}} \xi_i(\omega) Y_i \le \hat{T}_k + \dot{M}_k(\omega)(1 - Z_k(\omega)), \quad \forall k = 1,\cdots,K, \forall \omega \in \Omega \tag{5d}$$

$$\sum_{\omega\in\Omega} Z_k(\omega) - \lceil (1-\alpha)|\Omega| \rceil V_k \ge 0, \quad \forall k = 1,\cdots,K$$

$$\mathbf{Y}, \mathbf{V}, \mathbf{Z} \text{ binary}, \mathbf{O} \ge 0, \tag{5f}$$

where $\dot{M}(\omega)$ are big-M coefficients. The objective (5a) maximizes the sum of dual prices minus the expected bin extension associated with the column. Constraint (5b) ensures that the new column can be associated with at most one bin type. Constraint (5c) determines the bin extension in each scenario. Constraints (5d) and (5e) jointly enforce that the column satisfies the chance constraint for type k bins; (5d) determines $Z_k(\omega)$, and (5e) enforces the probability tolerance.

After (SSP) is solved, and if the objective value $\zeta^{SSP} > 0$, it generates an improving column, $\hat{p}$, to (R^2MP) such that $y_{i\hat{p}} = Y_i^*, \forall i \in \mathcal{I}$, $c_{k\hat{p}} = V_k^*, \forall k = 1,\cdots,K$, and $\bar{o}_{\hat{p}} = \mathbb{E}[O^*(\omega)]$, where $\mathbf{Y}^*$, $\mathbf{V}^*$, and $\mathbf{O}^*$ are the optimal solutions of (SSP).

Remark 5. The big-M coefficient, $\dot{M}_k(\omega)$, must be at least $\sum_{i\in\mathcal{I}} \xi_i(\omega) - \hat{T}_k$ for all $k = 1,\cdots,K$, and $\omega \in \Omega$ because constraint (5d) should be satisfied when all $Z_k(\omega)$ are equal to zero. However, when the column is chance-constrained, this big-M coefficient becomes unnecessarily large, which degrades the performance of the (SSP) formulation.

4.2.2. Type-Specific Formulation of the Scenario-Based Subproblem. According to Remark 5, if the column is fixed to satisfy a specific type of chance constraints, the big-M coefficients could be strengthened more tightly. Thus, we reformulate (SSP) separately into $K+1$ subproblems in which the $k^{th}(k=1,\cdots,K)$ subproblem, which we refer to as (SSP_k), generates a new column for a chance-constrained bin of type k. We formulate (SSP_k) as the following two-stage SMIP:

$$(\mathbf{SSP_k}):\ \zeta_k^{SSP} = \max \sum_{i\in\mathcal{I}} \pi_i Y_i + \phi_k + \varphi - \mathbb{E}[O(\omega)] \tag{6a}$$

$$\text{s.t. (5c) is satisfied for all } \omega\in\Omega, \text{ and}$$

$$\sum_{i\in\mathcal{I}} \xi_i(\omega) Y_i \le \hat{T}_k + \tilde{M}_k(\omega)(1-Z(\omega)), \quad \forall \omega\in\Omega \tag{6b}$$

$$\sum_{\omega\in\Omega} Z(\omega) \ge \lceil (1-\alpha_k)|\Omega| \rceil, \tag{6c}$$

$$\mathbf{Y},\mathbf{Z} \text{ binary}, \mathbf{O}\ge 0,$$

where $Z(\omega)$ are auxiliary decision variables that are specific to (SSP_k), $Z(\omega)=1$ if this column satisfies the chance constraint in scenario ω and $Z(\omega)=0$ otherwise, and $\tilde{M}_k(\omega)$ are big-M coefficients (strengthening of which is discussed in Section 4.2.3).

The $(K+1)^{th}$ subproblem, ($\text{SSP}_{\text{K+1}}$), generates a column unrestricted by chance constraints. It can be formulated as follows:

$$(\mathbf{SSP_{K+1}}):\ \zeta_{K+1}^{SSP} = \max \sum_{i=1}^{n} \pi_i Y_i + \varphi - \mathbb{E}[O(\omega)],$$

$$\text{s.t. (5c) is satisfied for all } \omega\in\Omega,$$

$$\mathbf{Y} \text{ binary}, \mathbf{O}\ge 0.$$

After solving the $K+1$ subproblems, we may have multiple improving columns for (R^2MP). There exist several approaches regarding managing these columns, such as adding the first improving column, the most improving column, or all of these columns (Lamiri et al. 2008). If none of the columns is improving to (R^2MP), it means the (RMP) is already solved to optimality.

4.2.3. Coefficient Strengthening. Given a particular scenario ω' and a particular bin type k', the big-M coefficient, $\tilde{M}_{k'}(\omega')$, is valid if it is greater than or equal to

$$\bar{M}_{k'}(\omega') = \max \sum_{i'\in\mathcal{I}} Y_{i'}\xi_{i'}(\omega') - \hat{T}_{k'},$$
$$\text{s.t. (6b)-(6c) are satisfied for all } \omega\in\Omega,$$
$$\mathbf{Y},\mathbf{Z} \text{ binary}. \tag{7}$$

We can also use $\bar{M}_{k'}(\omega')$ to choose the big-M coefficients for formulation (P); however, determining the exact value of $\bar{M}_{k'}(\omega')$ is computationally intensive. There exist several approaches in the literature that efficiently compute good upper bounds of $\bar{M}_{k'}(\omega')$. Qiu et al. (2014) solved the LP relaxation of (7) and iteratively improved the big-M coefficients using heuristics. Song et al. (2014) solved scenario-decomposed problems of (7) and used a quantile solution to approximate the upper bound of $\bar{M}_{k'}(\omega')$. It was suggested that the approach of Song et al. (2014) is more efficient and yields coefficients of similar quality to Qiu et al. (2014). Therefore, we apply the approach of Song et al. (2014) and determine $\tilde{M}_{k'}(\omega')$ by the following steps:

Step 1. For each scenario ω, we solve the following MIP:

$$\eta_{\omega'}^{k'}(\omega) := \max \sum_{i'\in\mathcal{I}} Y_{i'}\xi_{i'}(\omega') - \hat{T}_{k'}$$
$$\text{s.t.} \sum_{i\in\mathcal{I}} Y_i\xi_i(\omega) \le \hat{T}_{k'}, \mathbf{Y} \text{ binary}. \tag{8}$$

Step 2. Sort $\{\eta_{\omega'}^{k'}(\omega)\}$ in a nondecreasing order of ς such that $\eta_{\omega'}^{k'}(\varsigma_1) \le \eta_{\omega'}^{k'}(\varsigma_2) \le \cdots \eta_{\omega'}^{k'}(\varsigma_{|\Omega|})$.

Step 3. Determine $\tilde{M}_{k'}(\omega') = \eta_{\omega'}^{k'}(\varsigma_{\lceil \alpha_k|\Omega|\rceil+1})$.

4.3. Cover-Based Subproblem

In this section, we show two other subproblem formulations based on a collection of item sets that are labeled as "satisfying" or "violating" the chance constraints. These formulations do not have the scenario-based integer decision variables.

We first define a probability distribution function Φ^k on a given set $\mathcal{A}$ of items:

$$\Phi^k(\mathcal{A}) = \mathbb{P}\left(\sum_{i\in\mathcal{A}} \xi_i > \hat{T}_k\right).$$

Given a probability tolerance, α_k, a set $\mathcal{C}_k$ is called a *probabilistic cover* if $\Phi^k(\mathcal{C}_k) > \alpha_k$, whereas a set $\mathcal{P}_k$ is called a *probabilistic pack* if $\Phi^k(\mathcal{P}_k) \le \alpha_k$. We define $\mathcal{Q}(\mathcal{P}_k)$ as a *complementary set* of $\mathcal{P}_k$ such that $\Phi^k(\mathcal{P}_k \cup i) > \alpha_k, \forall i\in\mathcal{Q}(\mathcal{P}_k)$. Similar definitions can be found in Song et al. (2014).

4.3.1. Extensive Formulation of the Cover-Based Subproblem. The first of the two subproblem formulations in this section, which we refer to as (CSP), solves a relaxed subproblem multiple times until a feasible solution is found. Given the collection of covers (denoted by $\mathfrak{C}_k$) and the collection of packs (denoted by $\mathfrak{P}_k$), the extensive formulation of the cover-based subproblem is

$$(\mathbf{CSP}): \zeta^{CSP} = \max \sum_{i\in\mathcal{I}} \pi_i Y_i + \sum_{k=1}^{K} \phi_k V_k + \varphi - \mathbb{E}[O(\omega)] \tag{9a}$$

$$\text{s.t. (5b), (5c) are satisfied for all } \omega \in \Omega, \text{and}$$

$$\sum_{i\in\mathcal{Q}(\mathcal{P}_k)} Y_i \le \hat{M}_{\mathcal{P}_k}(|\mathcal{P}_k| - \sum_{i\in\mathcal{P}_k} Y_i + 1 - V_k), \quad \forall k = 1,\cdots,K, \forall \mathcal{P}_k \in \mathfrak{P}_k, \tag{9b}$$

$$\sum_{i\in\mathcal{C}_k} Y_i \le |\mathcal{C}_k| - 1 + \hat{M}_{\mathcal{C}_k}(1 - V_k), \quad \forall k = 1,\cdots,K, \forall \mathcal{C}_k \in \mathfrak{C}_k, \tag{9c}$$

$$\mathbf{Y}, \mathbf{V} \text{ binary}, \mathbf{O} \ge 0, \tag{9d}$$

where $\hat{M}_{\mathcal{P}_k}$ and $\hat{M}_{\mathcal{C}_k}$ are big-M coefficients, which can be bounded by $\hat{M}_{\mathcal{P}_k} \ge n - m + 1 - |\mathcal{P}_k|$, and $\hat{M}_{\mathcal{C}_k} \ge n - m + 2 - |\mathcal{C}_k|$, respectively. The objective (9a) maximizes the sum of dual prices minus the cost of the expected bin extension. Constraints (9b) and (9c) jointly enforce the relaxed chance constraints; constraint (9b) ensures that, if $V_k = 1$ and all items in $\mathcal{P}_k$ are selected, then one cannot select any item from $\mathcal{Q}(\mathcal{P}_k)$, and constraint (9c) ensures that, if $V_k = 1$, then one cannot select all items from $\mathcal{C}_k$. After (CSP) is solved, we verify the feasibility of chance constraints using the scenarios of item sizes. If the solution is feasible, and $\zeta^{CSP} > 0$, then CSP generates an improving column, $\hat{p}$, to (R^2MP) such that $y_{i\hat{p}} = Y_i^*, \forall i \in \mathcal{I}$, $c_{k\hat{p}} = V_k^*, \forall k = 1,\cdots,K$, and $\bar{o}_{\hat{p}} = \mathbb{E}[O^*(\omega)]$, where $\mathbf{Y}^*$, $\mathbf{V}^*$, and $\mathbf{O}^*$ are the optimal solutions of (CSP). However, if the solution is infeasible concerning the chance constraints, we generate new covers and packs to $\mathfrak{C}_k$ and $\mathfrak{P}_k$, respectively, and repeat solving the updated (CSP) until its solution becomes feasible.

Remark 6. There are many ways to generate cover sets and pack sets. In this article, they are generated dynamically as follows: $\mathfrak{C}_k$ and $\mathfrak{P}_k$ are empty sets initially. Suppose we obtain a solution with $V_k^* = 1$ and $\mathbb{P}(\sum_i Y_i^* \xi_i(\omega) \le \hat{T}_k) < 1 - \alpha_k$; we automatically find a cover set, $\bar{\mathcal{C}}_k = \{i \mid Y_i^* = 1\}$. Regarding the cover set ($\mathfrak{C}_k$), we add a subset of $\bar{\mathcal{C}}_k$, $\hat{\mathcal{C}}_k$, which is the *minimal* cover such that $\Phi^k(\hat{\mathcal{C}}_k) > \alpha_k$ and $\Phi^k(\hat{\mathcal{C}}_k \setminus i) \le \alpha_k, \forall i \in \hat{\mathcal{C}}_k$. Regarding the pack set ($\mathfrak{P}_k$), we add all packs ($\hat{\mathcal{P}}_k$) in $\bar{\mathcal{C}}_k$ such that $\Phi^k(\hat{\mathcal{P}}_k) \le \alpha_k$ unless they already exist in $\mathfrak{P}_k$. Although we did not explore this option, another approach is to add the *maximal* pack (Song et al. 2014), which might be more efficient in cases in which the number of packs is very large.

Next we have the following theorem (all proofs of theorems, lemmas, and corollaries are relegated to Section 1 of the online supplement).

Theorem 1. *(CSP) is a relaxation of (SSP) such that $\zeta^{CSP} \ge \zeta^{SSP}$ when a partial collection of sets $\mathcal{C}_k$ are considered; moreover, the relaxation is tight such that $\zeta^{CSP} = \zeta^{SSP}$ as long as $\mathbb{P}(\sum_i Y_i^* \xi_i(\omega) \le \hat{T}_k) \ge 1 - \alpha_k$ for any k with $V_k^* = 1$, where $\mathbf{Y}^*$ and $\mathbf{V}^*$ are solutions of (CSP).*

Theorem 1 can be interpreted to mean that we can use (CSP) to solve the subproblem. If the solution of (CSP) is feasible to the chance constraints, then it is naturally optimal to the generic subproblem. Therefore, using (CSP) to solve the subproblem can guarantee the optimality of (RMP). The (CSP) formulation with a restricted collection of sets $\mathcal{C}_k$ and $\mathcal{P}_k$ is easier to solve than (SSP) because it does not have second-stage binary decision variables. If the solution is infeasible to the chance constraints, we add packs and covers as illustrated by Remark 6 and repeat solving (CSP) until its solution becomes feasible. Therefore, the packs and covers in (CSP) are accumulated, and the resulting formulation becomes increasingly powerful in generating a feasible solution over iterations (the computational performance of the cover- and scenario-based subproblem formulations is compared in Section 6.3.2).

Remark 7. According to Theorem 1, the pack constraint (9b) is not needed to yield a feasible solution concerning the chance constraints. We include this constraint in (CSP) because it significantly improves the computational efficiency.

4.3.2. Type-Specific Formulation of the Cover-Based Subproblem.

Our final subproblem formulation is to reformulate (CSP) separately into $K + 1$ subproblems, where the $k^{th}(k = 1,\cdots,K)$ subproblem, which we refer to as ($\mathrm{CSP_k}$), generates a new column for a chance-constrained bin of type k. We formulate it as the following two-stage SMIP:

$$(\mathbf{CSP_k}): \zeta_k^{CSP} = \max \sum_{i\in\mathcal{I}} \pi_i Y_i + \phi_k + \varphi - \mathbb{E}[O(\omega)] \tag{10a}$$

$$\text{s.t. (5c) is satisfied for all } \omega \in \Omega, \text{ and}$$

$$\sum_{i\in\mathcal{Q}(\mathcal{P}_k)} Y_i \le \hat{M}_{\mathcal{P}_k}(|\mathcal{P}_k| - \sum_{i\in\mathcal{P}_k} Y_i), \quad \forall \mathcal{P}_k \in \mathfrak{P}_k, \tag{10b}$$

$$\sum_{i\in\mathcal{C}_k} Y_i \le |\mathcal{C}_k| - 1, \quad \forall \mathcal{C}_k \in \mathfrak{C}_k, \tag{10c}$$

$$\mathbf{Y} \text{ binary}, \mathbf{O} \ge 0. \tag{10d}$$

The last subproblem, ($\mathrm{CSP_{K+1}}$), is the same as ($\mathrm{SSP_{K+1}}$) as discussed in Section 4.2.2.

4.4. Branch-and-Price Algorithm

In this section, we present a summary of the B&P algorithm that finds an optimal integer solution for the SP-CCSBP problem (a detailed description of the B&P algorithm is available in Section 2 of the online supplement). In each major iteration of the algorithm, we select and fathom a node on the list, and a node represents an instance of (RMP). We let $\bar{\lambda}$ be its optimal solution, $\zeta^{RMP}(\bar{\lambda})$ be the objective value of (RMP) associated to this node, λ^* be the current best integer solution, and UB be the current upper bound of the optimal objective value. We briefly summarize the key steps of the B&P algorithm including node selection and branching decisions.

• If $\zeta^{RMP}(\bar{\lambda}) \geq UB$, then no feasible solution can have objective value smaller than UB, so we fathom this selected node.

• If $\bar{\lambda}$ is integer for all $p \in \mathcal{P}$ and $\zeta^{RMP}(\bar{\lambda}) < UB$, reset UB to $\zeta^{RMP}(\bar{\lambda})$ and reset λ^* to $\bar{\lambda}$.

• Finally, if $\zeta^{RMP}(\bar{\lambda}) < UB$ and $\bar{\lambda}$ is fractional for some $p \in \mathcal{P}$, we select a pair of items (i, j) from a column with fractional $\bar{\lambda}_p$ according to the method as described in Section 4.4.1: place two new "child" nodes on the list, one with constraint $Y_i = Y_j$ appended (i and j must appear in the same bin) and the other with constraint $Y_i + Y_j \leq 1$ appended (i and j must *not* appear in the same bin). Then select a child node that has the smallest objective value among all unexplored nodes on the list, using column generation to solve the (RMP) to the optimal continuous solution (i.e., a new node is added to the list).

Remark 8. We use a pair-based branching strategy that determines whether two particular items are allocated in the same bin or not (Ryan and Foster 1981). This strategy has recently been suggested by Hashemi Doulabi et al. (2016) for its efficiency in generating better integer columns than other branching strategies in solving operating-room-scheduling problems.

4.4.1. Generating an Item Pair. There are many ways to generate an item pair. A pair is *feasible* if the two items therein are different and the pair has not yet appeared in the parent nodes. Because of the variation of dual prices, some columns may have the same y_{ip} but different c_{kp}; however, because these columns have the same set of items, they can be converted to satisfy chance constraints associated with the same bin type. We let $\mathbb{O}_p$ denote the set of all columns (including column p) that have the same elements of y_{ip} as column p. With this definition, we have the following lemma:

Lemma 1. *If the following condition,*

$$\sum_{q \in \mathbb{O}_p} \lambda_q = 0 \text{ or } 1$$

holds for each column $p \in \mathcal{S}$, then λ can be converted as an integer solution.

We let $\mathbb{O}$ denote a collection of columns with $0 < \lambda_p < 1$ and $\sum_{q \in \mathbb{O}_p} \lambda_q < 1$. We define S_q as the set of selected items in a given column, q. If $\mathbb{O}$ is empty, according to Lemma 1, we already have an integer solution; otherwise, we search for a feasible pair for branching according to the following theorem.

Theorem 2. *A column $p \in \mathbb{O}$ must contain a feasible pair if $S_p \not\subset S_q$, $\forall q \neq p$, and $q \in \mathbb{O}$.*

Theorem 2 can be interpreted to mean that, given a nonempty set $\mathbb{O}$, the existence of a feasible pair of items in a column p such that $p = \arg_q \max_{q \in \mathbb{O}} |S_q|$ is guaranteed.

5. Distributionally Robust Version of the CCSBP Problem

In this section, we consider a DR version of the CCSBP (DR-CCSBP) problem, which assumes that the distributions of item sizes are ambiguous. We first present a closed-form expression for the worst-case violated probability, that is, the probability of a bin exceeding its extension threshold under the worst-case distribution, and then we approximate the DR-CCSBP problem as an MIP formulation based on this expression.

5.1. DR-CCSBP Problem with Moment Information

We relax the assumption that the distribution of each item size is known, and we consider the case in which only the mean μ_i and variance σ_i^2 of the item sizes are known, which can be expressed by the following ambiguity set:

$$\mathcal{D} = \left\{\mathbb{D} \in \Sigma : \mathbb{E}[\xi_i(\omega)] = \mu_i, \mathbb{E}\left[\xi_i(\omega)^2\right] = \mu_i^2 + \sigma_i^2, \forall i \in \mathcal{I}\right\},$$

where Σ represents the entire family of probability distributions.

The DR-CCSBP problem assumes that the extension threshold of each chance-constrained bin can be exceeded with at most probability α_k under the worst-case distribution in the ambiguity set, $\mathcal{D}$. We define additional decision variables, $\hat{\mathbf{o}} \in \mathbb{R}_+^{|\mathcal{R}|}$, such that $\hat{o}_r$ is the extension of bin r. The DR-CCSBP problem can be expressed as follows:

$$\min \sum_{r \in \mathcal{R}} \hat{o}_r \tag{11a}$$

$$\text{s.t. } \hat{o}_r \geq \sum_{i \in \mathcal{I}} x_{ir}\mu_i - T, \qquad \forall r \in \mathcal{R} \tag{11b}$$

$$\sum_{r \in \mathcal{R}} x_{ir} = 1, \qquad \forall i \in \mathcal{I} \tag{11c}$$

$$\inf_{\mathbb{D} \in \mathcal{D}} \mathbb{P}\left(\sum_{i \in \mathcal{I}} x_{ir}\xi_i(\omega) \leq \hat{T}_k\right) \geq 1 - \alpha_k, \qquad \forall k = 1, \cdots, K, \forall r \in \mathcal{R}_k \tag{11d}$$

$$\mathbf{x} \text{ binary}, \hat{\mathbf{o}} \geq 0. \tag{11e}$$

The objective (11a) is to minimize the total extension of bins based on the mean value of item sizes. We consider the mean-based bin extension because this simplifies the formulation, allowing us to approximate the problem as an MIP. Constraint (11b) determines the extension of each bin. Constraint (11c) ensures that each item is allocated to exactly one bin. Constraint (11d) defines the DR chance constraints.

The main challenge for solving formulation (11) is that constraint (11d) is no longer well defined by SAA, which involves sampling from a distribution that is fixed in advance. However, we can employ the existing result in Calafiore and El Ghaoui (2006) to approximate constraint (11d) as a closed-form expression.

To approximate constraint (11d), we first consider an artificial item size, ξ, with mean μ_ξ and variance σ_ξ^2. This item exclusively occupies a bin with extended size, $\hat{T}_\xi$. The DR chance constraint associated with ξ can be expressed as follows:

Theorem 3 (Calafiore and El Ghaoui 2006, theorem 3.1). *For any $\alpha \in (0,1)$, the DR chance constraint*

$$\inf_{\mathbb{D}\in\mathcal{D}} \mathbb{P}\left(\xi \le \hat{T}_\xi\right) \ge 1-\alpha \tag{12}$$

is equivalent to the following constraint:

$$\mu_\xi + \sqrt{\frac{1-\alpha}{\alpha}}\sigma_\xi \le \hat{T}_\xi. \tag{13}$$

With Theorem 3, we are able to reformulate constraint (12) as follows:

$$\mu_\xi \le \hat{T}_\xi, \tag{14a}$$
$$\frac{1-\alpha}{\alpha}\sigma_\xi^2 \le \left(\hat{T}_\xi - \mu_\xi\right)^2. \tag{14b}$$

We approximate (14b) as linear constraints with mixed-integer decision variables and the DR-CCSBP problem as an MIP formulation. For each bin r, we let ξ_r represent the aggregate size of items allocated to bin r, that is, $\xi_r = \sum_{i\in\mathcal{I}} \xi_i x_{ir}$. Note that this is an approximation of the true DR chance constraints as defined in (11d) because we consider the moment constraints related to the aggregated item size rather than individual item sizes. As a result, $\mathcal{D}$ is relaxed, and thus, the solution is more conservative; however, we show in Section 6 that this formulation gives very good solutions. We further assume that item sizes are mutually independent so that the variance of ξ_r can be calculated as $\sigma_{\xi_r}^2 = \sum_{i\in\mathcal{I}} \sigma_i^2 x_{ir}$. We introduce new decision variables, $\mathbf{q} \in \{0,1\}^{|\mathcal{I}|\times|\mathcal{I}|\times|\mathcal{R}|}$, such that $q_{ijr} = 1$ if both items i,j are in bin r, and $q_{ijr} = 0$ otherwise. Although q_{ijr} are binary decision variables, they can be naturally relaxed to continuous nonnegative decision variables. We approximate the DR-CCSBP problem as an MIP as shown in the following theorem:

Theorem 4. *When item sizes are mutually independent, formulation (11) of the DR-CCSBP problem can be approximated as the following MIP formulation:*

$$(\textbf{ROP}): \min \sum_{r\in\mathcal{R}} \hat{o}_r \tag{15a}$$

$$\text{s.t. (11b), (11c) } \textit{satisfied for all } r\in\mathcal{R}, i\in\mathcal{I}, \textit{ and}$$

$$\sum_{r\in\mathcal{I}} \mu_i x_{ir} \le \hat{T}_k, \quad \forall k=1,\cdots,K, \forall r\in\mathcal{R}_k \tag{15b}$$

$$\sum_{i\in\mathcal{I}} \Lambda_{ik} x_{ir} - 2\sum_{i,j\in\mathcal{I}, j>i} \mu_i\mu_j q_{ijr} \le \hat{T}_k^2, \quad \forall k=1,\cdots,K, \forall r\in\mathcal{R}_k \tag{15c}$$

$$x_{ir} + x_{jr} - q_{ijr} \le 1, \quad \forall \textit{distinct } i,j\in\mathcal{I}, \forall r\in\mathcal{R} \tag{15d}$$

$$x_{ir} - q_{ijr} \ge 0, \quad \forall \textit{distinct } i,j\in\mathcal{I}, \forall r\in\mathcal{R} \tag{15e}$$

$$x_{jr} - q_{ijr} \ge 0, \quad \forall \textit{distinct } i,j\in\mathcal{I}, \forall r\in\mathcal{R} \tag{15f}$$

$$\mathbf{x} \textit{ binary}, \ \mathbf{q}, \hat{\mathbf{o}} \ge 0, \tag{15g}$$

where $\Lambda_{ik} = \frac{1-\alpha_k}{\alpha_k}\sigma_i^2 + 2\hat{T}_k\mu_i - \mu_i^2$.

Remark 9. Regarding the Dantzig–Wolfe formulation of the DR-CCSBP problem, the master problem can be formulated the same as (MP); however, the subproblem formulation is different, which is given in Section 3 of the online supplement. The B&P algorithm as proposed in Section 4.4 can directly apply for the Dantzig–Wolfe formulation of the DR-CCSBP problem.

5.2. Symmetry in the DR-CCSBP Problem

In this subsection, we address the symmetry in the DR-CCSBP problem. The DR-CCSBP problem has symmetric solutions concerning identical bins and identical items. The identical items in the DR-CCSBP problem denote the items with the same mean and variance of item sizes (note that the identical items in the SP-CCSBP problem denote the items with the same samples of item sizes, and thus, they do not exist because item sizes are independently sampled for all items).

We address the symmetry issues from both formulation and algorithm levels. For the extensive formulation of the DR-CCSBP problem, that is, (ROP), the symmetry issues can be directly addressed from the formulation level. Specifically, we can add constraint (3) to eliminate symmetric solutions concerning identical bins and add the following constraint to break the symmetry associated with identical items:

$$\sum_{q=1}^{r} x_{jq} \leq \sum_{q=1}^{r} x_{iq}, \forall i<j, \mu_i = \mu_j, \sigma_i^2 = \sigma_j^2, \ \forall r, \tag{16}$$

where constraint (16) ensures that identical items are allocated to bins in a lexicographic order.

For the Dantzig–Wolfe decomposition, the formulation can automatically break the symmetry associated with identical bins; however, it cannot break the symmetry of identical items from the formulation level because the allocation of items is generated independently for each bin in the subproblem. Instead, we address this symmetry issue from the algorithm level. Specifically, the symmetry issue associated with identical items appears in the form of redundant nodes in the branch-and-bound tree. We address this issue by considering an approach to detect and prune the redundant node. The details of this approach are available in Section 4 of the online supplement.

5.3. Approximating the SP-CCSBP Solution Using DR Optimization

In general, formulation (ROP) for the DR-CCSBP problem is much easier to solve than formulation (P) for the SP-CCSBP problem because it has a significantly fewer number of binary decision variables, which makes it possible to serve as a useful approximation to the latter for very large-scale problems. However, it is well known that the DR solution tends to be much more conservative than the stochastic solution. Specifically, regarding a given probability tolerance, α_k, for the chance constraints of type k, the DR-CCSBP problem may generate a solution with a "real" violated probability that is much smaller than α_k (which means the bin is overprotected).

In this section, we consider choosing an adjusted probability tolerance, $\hat{\alpha}_k$ ($\hat{\alpha}_k \geq \alpha_k, \forall k$), in formulation (ROP) so that solving this formulation with $\hat{\alpha}_k$ can generate a solution with similar performance to that of the SP-CCSBP solution with probability tolerance α_k regarding the real violated probabilities.

Similar to Section 5.1, we consider an artificial item size, ξ, with mean μ_ξ and variance σ_ξ^2 that exclusively occupies a bin with extended size, $\hat{T}_\xi$. Under this setting, we define

$$\alpha_\xi^{WC} = \sup_{\mathbb{D}\in\mathcal{D}_\xi} \mathbb{P}\left(\xi \geq \hat{T}_\xi\right), \tag{17}$$

where $\mathcal{D}_\xi$ is the ambiguity set of the distribution of ξ defined by its mean and variance.

It has been shown that the general chance-constrained program can be reformulated based on SAA with the approximation error converging to zero as the sample size increases to infinity (Pagnoncelli et al. 2009, proposition 2.1). This means any probability distribution for the chance-constrained program can be replaced by a discrete distribution with sufficiently large-sized samples, including the worst-case distribution in our case. Moreover, we have the following result on the worst-case distribution of ξ.

Lemma 2. *There exists a discrete distribution of ξ with at most three points that maximizes α_ξ^{WC}.*

Then, based on Lemma 2, we can further show a closed-form expression of α_ξ^{WC} by the following theorem.

Theorem 5. *Given an item size with mean μ_ξ and variance σ_ξ^2, and a bin size $\hat{T}_\xi$, the probability α_ξ^{WC} as defined by* Equation (17) *can be expressed as follows:*

$$\alpha_\xi^{WC} = \begin{cases} 1, & \text{if } \mu_\xi > \hat{T}_\xi, \\ \mu_\xi/\hat{T}_\xi, & \text{if } \mu_\xi \leq \hat{T}_\xi, \mu_\xi^2 + \sigma_\xi^2 > \mu_\xi \hat{T}_\xi, \\ \dfrac{\sigma_\xi^2}{\sigma_\xi^2 + \left(\hat{T}_\xi - \mu_\xi\right)^2}, & \text{if } \mu_\xi \leq \hat{T}_\xi, \mu_\xi^2 + \sigma_\xi^2 \leq \mu_\xi \hat{T}_\xi. \end{cases} \tag{18}$$

Moreover, the following corollary reveals the worst-case violated probability.

Corollary 1. *The probability α_ξ^{WC} is continuous with respect to μ_ξ, σ_ξ^2, and $\hat{T}_\xi$, respectively, and $\alpha_\xi^{WC} = \hat{\alpha}_\xi^{WC}$, where $\hat{\alpha}_\xi^{WC} = \sup_{\mathbb{D}\in\mathcal{D}_\xi} \mathbb{P}(\xi > \hat{T}_\xi)$.*

Note that $\hat{\alpha}_\xi^{WC}$ is the *real* worst-case violated probability for ξ.

Using the closed-form expression of α_ξ^{WC} in Theorem 5, we can approximate the SP-CCSBP solution by solving formulation (ROP) with an adjusted probability tolerance. Setting this tolerance exactly could be difficult because it depends on the allocation decisions. However, there are several ways to set the tolerance approximately. Specifically, we consider the following approaches:

- Aggregate-item approach: we consider the artificial item to be an *aggregate* of all items in $\mathcal{I}$, and thus, we set $\mu_\xi = \sum_{i\in\mathcal{I}} \mu_i$, $\sigma_\xi^2 = \sum_{i\in\mathcal{I}} \sigma_i^2$. We set $\hat{T}_\xi$ to be the $(1-\alpha_k)$-percentile value of a normal distribution with μ_ξ and σ_ξ^2. We use the normal distribution because the size of the aggregate item is the sum of multiple ($n > 10$) independent item sizes. Finally, we set $\hat{\alpha}_k = \alpha_\xi^{WC}$ as the probability tolerance for formulation (ROP), where α_ξ^{WC} is determined by Theorem 5.
- Average-item approach: we consider the artificial item to be an *average* item that takes the average mean and the average variance across all items, that is, $\mu_\xi = \frac{\sum_{i\in\mathcal{I}} \mu_i}{n}$, $\sigma_\xi^2 = \frac{\sum_{i\in\mathcal{I}} \sigma_i^2}{n}$. Based on μ_ξ and σ_ξ^2, we set $\hat{T}_\xi$ to be the $(1-\alpha_k)$-percentile value of the specific distribution type of the SP-CCSBP problem (note that this approach assumes all item sizes have the same distribution type). Finally, we set $\hat{\alpha}_k = \alpha_\xi^{WC}$.

Our experiments show that these approximation approaches have similar performance regarding the expected bin extension and the violated probability (see details in Section 5 of the online supplement), but the *average-item* approach can obtain a violated probability that is slightly closer to the probability tolerance. Therefore, in our computational experiments, we use the average-item approach.

6. Computational Results

In this section, we test our proposed approaches on instances of the SP-CCSBP and DR-CCSBP problems. Because there are no standard test instances for this problem, we construct formulation instances in the context of surgery scheduling based on real data, and we use these instances to evaluate the performance of the B&P algorithm for the SP-CCSBP and DR-CCSBP problems. In this context, the bins are ORs that are available for some nominal period during the day of surgery (e.g., 10 hours). The items represent surgeries, and the item sizes represent the surgery durations. Figure 1 illustrates the problem with a small example that has three ORs and eight surgeries.

This is an important application because of the high cost of OR utilization and OR staff overtime (Deng et al. 2020). The latter is associated with anxiety and poor morale among OR nurses and, ultimately, a high turnover rate among nurses (e.g., see a report by Nursing Solutions 2016). This is particularly important because there is a severe shortage of OR nurses nationwide. To address concerns about overtime, some hospital administrators are implementing a rotating schedule. Under this paradigm, a portion of ORs has low overtime so that nurses allocated to the OR have predictable shift completion times. Staff then rotate among ORs from day to day so that all staff benefits from predictable shift durations on some portion of the surgery days.

6.1. Parameter Estimation

We use data from the inpatient surgical suite at Ruijin Hospital in Shanghai, China. The data are classified into three sets based on the number of surgeries on a single day on which sets 1–3 contain instances of small, medium, and large sizes, respectively. We set $\mathcal{R} = \{1, \cdots, 9\}$, $T = 10$, $K = 2$, and $\alpha_k = 10\%, \forall k = 1, \cdots, K$. For the chance-constrained ORs, we further set $\mathcal{R}_1 = \{1, 2, 3\}$, $\hat{T}_1 = 10$, $\mathcal{R}_2 = \{4, 5, 6\}$, and $\hat{T}_2 = 12$. The details of the parameter estimation are available in Section 6 of the online supplement.

6.2. Experimental Setting

Our main purposes of the experiments are to investigate the following:

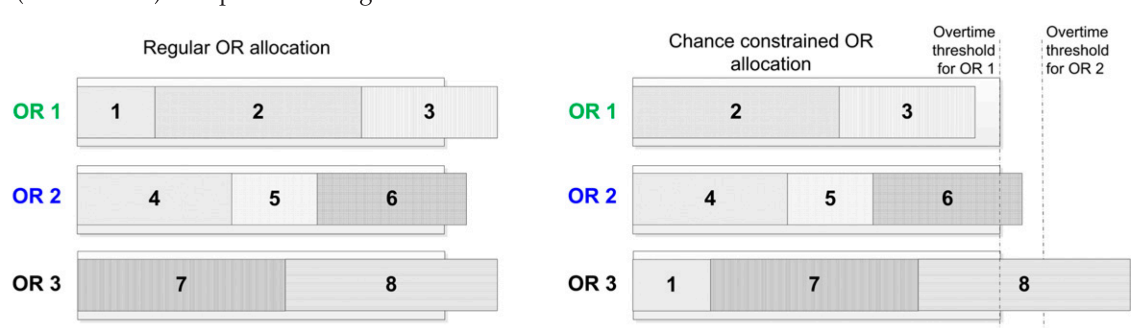
1. How the B&P algorithm performs on solving the SP-CCSBP problem compared with a B&C algorithm and which of the subproblem formulations performs best.
2. How the B&P algorithm performs at solving the DR-CCSBP problem compared with a standard B&C algorithm and how the redundant node-detection method performs.
3. How the proposed formulations perform in real practice at reducing the overtime of surgery scheduling compared with a *heuristic bin-packing solution* and how well the chance constraints are fulfilled by these solutions.

We compare our B&P algorithm with a standard B&C algorithm and a well-known longest-processing-time (LPT) first assignment heuristic. The details of the experimental setting are available in Section 7 of the online supplement. To compare the computational performance, we randomly chose 10 instances from each of the three instance sets for a total of 30 instances. The complete test instances are available as a separate online supplement.

6.3. Computational Performance of Methods for the SP-CCSBP Problem

In this section, we investigate the performance of the scenario-based subproblem formulations, the cover-based subproblem formulations, and the overall performance of the B&P algorithm compared with the B&C algorithm on solving the SP-CCSBP problem.

Figure 1. (Color online) Comparison of Regular OR Allocation and Chance-Constrained OR Allocation for a Small Example



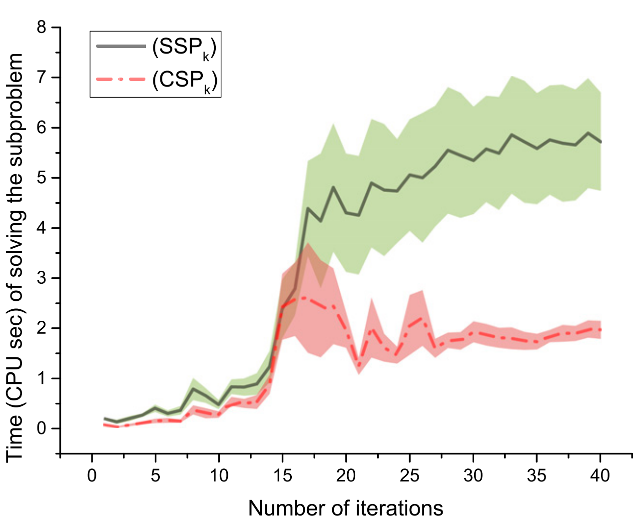
6.3.1. Selection of the Subproblem Formulations. We have many choices to implement the subproblem formulation in the B&P framework. Specifically, for the scenario-based subproblem formulation, we investigated the choices between the type-specific formulation and the extensive formulation; the choices between selecting all improving columns and selecting the first improving column; and the choices of big-M coefficients between without strengthening, with relaxed strengthening, and with exact strengthening. For the cover-based subproblem formulation, we investigated the choices between the type-specific formulation and the extensive formulation, the choices between selecting all improving columns and selecting the first improving column, and the choices between including and not including the probabilistic pack inequalities. We provide the main results here, but further details are available in Section 8 of the online supplement.

• The type-specific scenario-based subproblem formulation with selecting all columns and the relaxed strengthening of big-M coefficients had the best performance, which we refer to as (SSP_k).

• The extensive cover-based subproblem formulation with including the probabilistic pack inequalities had the best performance, which we refer to as (CSP). Another type-specific cover-based subproblem formulation with selecting all columns and probabilistic pack inequalities had comparable performance with (CSP), which we refer to as (CSP_k).

6.3.2. Performance Comparison of Methods for the SP-CCSBP Problem. We first compare the solution time for the subproblems in each iteration of the B&P algorithm for (SSP_k) and (CSP_k). In Figure 2, the solution time for (SSP_k) is the time spent to solve the $K+1$ type-specific subproblems. The time for (CSP_k) includes the time spent to solve the $K+1$ type-specific subproblems, check the feasibility and possibly add pack and cover constraints and repeatedly solve the subproblem until the solution becomes feasible.

Figure 2. (Color online) The Mean Solution Time (CPU Seconds) for the Subproblem in Each Iteration of the B&P Algorithm for (SSP_k) and (CSP_k), Respectively, Where the 95% Confidence Intervals Are Generated Based on the Results of 10 Instances



As we observe from Figure 2, the solution time for the subproblems (CSP_k) at beginning iterations is similar to the solution time for (SSP_k), but after 20 iterations, (CSP_k) becomes significantly faster than (SSP_k). This is because (CSP_k) has many fewer binary decision variables; moreover, its pack and cover constraints can be accumulated over iterations, and thus, the formulation is strengthened more as the iterations continue.

We then compare the performance of the B&C algorithm and the B&P algorithms based on (SSP_k), (CSP_k), and (CSP), respectively. The B&C algorithm used the same big-M coefficients as (SSP_k), and (SSP_k) applied the same column-management policy as (CSP_k). In Table 1, we report the relaxation gap (i.e., the gap between the objective values of formulations with binary and continuous decision variables, respectively), the solution time, the number of nodes explored, and the optimality gap. For simplicity, we omit the time for strengthening coefficients (less than four seconds). The following observations can be made from Table 1:

1. The Dantzig–Wolfe formulation (MP) as included in the B&P algorithms has a much tighter relaxation gap than the extensive formulation (P) of the SP-CCSBP problem as included in the B&C algorithm (1.2% versus 88.6% on average across all instances), which explains why the B&P algorithms generally outperformed the B&C algorithm.

2. From the results of (SSP_k) and (CSP_k), we observed that solving cover-based subproblem formulations is more efficient than the scenario-based subproblem formulations. This is because (SSP_k) spent more time solving the subproblem (see Figure 2). Note that, in instances of set 3, (SSP_k) spent less solution time than (CSP_k) because it quickly solved a few instances, but for most instances, the algorithm terminated at the time limit (3,600 seconds) with much larger optimality gaps than (CSP_k).

3. The B&P algorithm based on (CSP) had the best overall performance across all instance sets with an average solution time of 1,564 seconds and mean optimality gap of 1.1% for instances of set 3.

6.4. Computational Performance of Methods for the DR-CCSBP Problem

In this section, we investigate the performance of the B&P algorithm for the DR-CCSBP problem, the redundant node-detection method, and approximating the SP-CCSBP solution using DR optimization. First, we compared the B&P algorithm for the DR-CCSBP

Table 1. Overall Performance of Selected Methods for the SP-CCSBP Problem

		Relaxation gap		Solution time (CPU seconds)		Number of nodes explored		Optimality gap		
Set	Method	Average, %	Maximum, %	Average	Maximum	Average	Maximum	N_{opt}	Average, %	Maximum, %
1	B&C	100	100	1,865.5	> 3,600	702K	1,335K	7	20.9	74.7
	(SSP_k)	0.2	2.4	7.0	19.3	15	99	10	0.0	0.0
	(CSP_k)	0.2	2.4	4.7	13.4	66	660	10	0.0	0.0
	(CSP)	6.5	51.7	2.9	7.7	98	941	10	0.0	0.0
2	B&C	95.5	100	3,600	> 3,600	113K	205K	0	88.3	94.3
	(SSP_k)	0.0	0.1	160.9	511.2	100K	204K	10	0.0	0.0
	(CSP_k)	0.0	0.1	78.9	119.1	55K	106K	10	0.0	0.0
	(CSP)	0.1	1.2	77.0	149.5	95K	241K	10	0.0	0.0
3	B&C	70.2	97.4	3,600	> 3,600	64K	99K	0	70.7	90.4
	(SSP_k)[a]	NA	NA	2,662.2	> 3,600	1,841K	3,624K	3	60.2	100
	(CSP_k)	1.1	3.5	3,263.0	> 3,600	2,268K	5,050K	1	1.1	3.5
	(CSP)	2.0	6.4	1,564.4	> 3,600	2,521K	6,637K	8	1.1	6.4

Notes. $K = 1{,}000$. N_{opt} is the number of instances that were solved to optimality. When calculating the average, the solution time of the unsolved instance was considered as 3,600 seconds.

[a]Because the lower bound costs (i.e., relaxed master problems) of six instances were not solved within 3,600 seconds, we considered their relaxation gap as "NA" (unavailable) and optimality gap as 100%.

problem with and without the redundant node-detection method and the B&C algorithm that was enhanced by constraints (3) and (16) to break the symmetry of both identical bins and items. We report the relaxation gap, the solution time, the number of nodes explored, and the optimality gap in Table 2. Note that, in this subsection, we omitted the results of set 1 because the computation time is negligible (less than five seconds) for all methods.

As can be observed from Table 2, the B&C algorithm generally outperformed the B&P algorithms across the experimented instances, which can be explained by the following reason: the symmetry of identical items was entirely broken in the B&C algorithm by constraint (16), and the symmetry (to some extent) still exists in the B&P algorithms.

Another observation is that the redundant node-detection method in some instances of set 2 helped in speeding up the B&P algorithm, but in many other instances, B&P-D was no faster than B&P. As we discuss in Section 4.2 of the online supplement, the performance of the detection method is highly affected by the dynamic process of pair generation and node selection that is independent of the redundant node detection. As a result, many unnecessary nodes were generated before the redundant node appears. Identifying a more efficient way to address the symmetrical issue of identical items in the B&P algorithm is an important direction for future work.

In general, the DR-CCSBP problem is much easier to solve than the SP-CCSBP problem. As a result, the former could potentially serve as an approximation for the latter problem to avoid the long computation time necessary for large problem instances. We evaluated the maximum violated probability and expected overtime of the SP-CCSBP and DR-CCSBP solutions under 20,000 independently generated scenarios of various distributions, including lognormal (LOGN), uniform (UNIF), and gamma (GAMM). (Note that surgery durations in UNIF were truncated at zero, which only affects a negligibly small number of surgeries.) We set the probability tolerance for the SP-CCSBP problem as

Table 2. Overall Performance of Different Methods for the DR-CCSBP Problem

		Relaxation gap		Solution time (CPU seconds)		Number of nodes explored		Optimality gap		
Set	Method	Average, %	Maximum, %	Average	Maximum	Average	Maximum	N_{opt}	Average, %	Maximum, %
2	B&C	99.8	100.0	110.8	295.9	60K	130K	10	0.0	0.0
	B&P	36.0	93.1	1,824.5	> 3,600	295K	703K	5	0.6	2.7
	B&P-D	36.0	93.1	1,592.9	> 3,600	265K	574K	6	0.4	2.7
3	B&C	88.6	100.0	931.0	2337.5	194K	488K	10	0.0	0.0
	B&P	35.3	51.3	1,213.9	> 3,600	339K	903K	7	0.3	1.3
	B&P-D	35.3	51.3	1,220.7	> 3,600	420K	1,303K	7	0.3	1.3

Notes. $K = 1{,}000$. N_{opt} is the number of instances that were solved to optimality. B&P is the standard B&P algorithm, and B&P-D is the B&P algorithm combined with the redundant node detection method.

Table 3. The Maximum Violated Probability ($\bar{\alpha}$) and Mean Total Expected Overtime ($\bar{O}$)

		SP-CCSBP solution with α_k		DR-CCSBP solution with α_k		DR-CCSBP solution with $\hat{\alpha}_k$	
Set	Distribution type	$\bar{\alpha}$, %	$\bar{O}$	$\bar{\alpha}$, %	$\bar{O}$	$\bar{\alpha}$, %	$\bar{O}$
2	LOGN	11.7	3.4	1.1	9.6	10.0	4.0
	UNIF	12.4	3.1	0.0	9.6	10.5	3.8
	GAMM	12.2	3.3	0.6	9.6	10.4	3.9
3	LOGN	14.9	12.3	1.2	22.9	10.0	13.5
	UNIF	18.2	12.3	0.0	23.0	10.2	13.6
	GAMM	15.8	12.3	0.8	23.0	10.1	13.6

Notes. The results are evaluations based on 20,000 independently sampled scenarios. $\bar{\alpha}$ is the maximum violated probability across all chance-constrained bins and 10 instances. $\alpha_k = 10\%$, and $\hat{\alpha}_k$ was set as in Section 5.3.

$\alpha_k = 10\%$, and we set the adjusted probability tolerance for the DR-CCSBP problem according to the method as discussed in Section 5.3. The results are reported in Table 3. From this table, we can make the following observations:

1. The SP-CCSBP solution demonstrates moderate infeasibility concerning the chance constraint. There are multiple potential reasons for this. The first reason is that we only used 500 scenarios for optimization, which resulted in a 1.7%–4.9% gap in achieving the probability tolerance. The second potential reason is due to intentional use of the "wrong" distribution type. In addition to the limited number of scenarios sampled for optimization, the intentional use of the wrong distribution could also have caused the increased gap to 2.2%–8.2%.

2. The DR-CCSBP solution with α_k is too conservative. However, when the probability tolerance is adjusted to $\hat{\alpha}_k$, the DR-CCSBP problem finds competitive solutions to the SP-CCSBP problem. Moreover, the violated probability under the DR-CCSBP solution was much closer to the probability tolerance than under the SP-CCSBP solution. A plausible reason is that the DR-CCSBP solution takes advantage of modeling the worst-case chance constraint to obtain robust outcomes, which is well aligned with the rationale behind using a robust optimization formulation.

6.5. Scheduling Performance of the Proposed Formulations

Now we investigate how much our proposed formulations can improve upon the LPT heuristic in surgery-scheduling practice. We evaluated the violated probability and the expected overtime reduction of their solutions over the LPT heuristic in surgery-scheduling practice. We tested 10 instances in sets 2 and 3, respectively, and the surgery durations for evaluation were bootstrapped from the historical data of the same surgery type. We report the results in Table 4.

As can be observed from Table 4, the SP-CCSBP solution significantly reduced the expected overtime by 19.8%–26.5% compared with the LPT heuristic although the violated probability was slightly higher (≤1.6%) than the heuristic. The DR-CCSBP solution also reduced overtime significantly, and it had a smaller violated probability than the heuristic at the same time. Because the SP-CCSBP solution achieved the minimum overtime, it is favored when small extension of the violated probability is acceptable; on the other hand, the DR-CCSBP solution is favored when the violated probability must be strictly controlled.

7. Conclusion

This article addresses methods for the stochastic bin-packing problem with chance constraints. We formulated the SP-CCSBP problem as a two-stage SMIP formulation that is similar to other previously proposed formulations. We addressed the computational challenges for this formulation by using a Dantzig–Wolfe formulation that can be solved using a B&P algorithm. The algorithm was further improved by reformulating the subproblem of the Dantzig–Wolfe formulation based on probabilistic packs and covers. A series of computational results show that both the Dantzig–Wolfe formulation and the subproblem reformulation considerably reduce the computation

Table 4. Scheduling Performance of the Proposed Formulations Compared with the LPT Heuristic

Set	Solution	$\bar{\alpha}$,%	$\bar{O}$	Expected overtime reduction compared with LPT, %
2	LPT with α_k	10.3	4.6	—
	SP-CCSBP with α_k	11.9	3.3	26.5
	DR-CCSBP with $\hat{\alpha}_k$	10.0	3.9	15.5
3	LPT with α_k	13.8	15.4	—
	SP-CCSBP with α_k	14.9	12.3	19.8
	DR-CCSBP with $\hat{\alpha}_k$	10.0	13.6	11.6

Notes. The results are evaluations based on 20,000 scenarios bootstrapped from the historical data of surgery durations. $\alpha_k = 10\%$. $\hat{\alpha}_k$ was set as in Section 5.3. $\bar{\alpha}$ is the maximum violated probability across all chance-constrained bins and 10 instances, and $\bar{O}$ is the average total expected overtime.

time required to solve large-scale problem instances that arise in practice.

We also presented a DR-CCSBP problem under conditions of ambiguous distributions of item sizes for which only the mean and variance of the true distribution are known. Based on a closed-form expression, we approximated the DR chance constraints as a linear set of constraints and approximated the DR-CCSBP problem as an MIP with significantly fewer integer decision variables than the SMIP for the SP-CCSBP problem. Our B&P algorithm can also be applied to solve the DR-CCSBP problem. Moreover, we showed that a robust solution with similar overtime of the SP-CCSBP solution can be achieved by adjusting the probability tolerance of the DR-CCSBP problem. Thus, for very large problems, the DR-CCSBP problem may serve as a useful approximation to the SP-CCSBP problem.

Future work will study multistage chance-constrained bin-packing problems with item-to-bin allocation decisions made before the start of the day and dynamic reallocation decisions throughout the horizon over which the random item size outcomes are observed. The problem formulation likely requires nonanticipative constraints that depend on the scenario and the dynamic state of the bin-packing system. As a result, it is necessary to study useful formulations that accurately express the feasibility of decisions over time and for which optimal or good approximate solutions can be obtained efficiently. The work presented in this article lays the foundation for these future studies.

Acknowledgments

The authors are grateful for the comments of two anonymous reviewers and an associate editor whose comments helped improve this article significantly. The authors thank Siqian Shen at the University of Michigan and Jiefang Shen at Ruijin Hospital for helpful discussion.

References

Adler M, Gibbons PB, Matias Y (2002) Scheduling space-sharing for Internet advertising. *J. Scheduling* 5(2):103–119.

Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* 46(3):316–329.

Batun S, Denton BT, Huschka TR, Schaefer AJ (2011) Operating room pooling and parallel surgery processing under uncertainty. *INFORMS J. Comput.* 23(2):220–237.

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

Calafiore GC, El Ghaoui L (2006) On distributionally robust chance-constrained linear programs. *J. Optim. Theory Appl.* 130(1): 1–22.

Codato G, Fischetti M (2006) Combinatorial Benders' cuts for mixed-integer linear programming. *Oper. Res.* 54(4):756–766.

Coffman EG Jr, Garey MR, Johnson DS (1978) An application of bin-packing to multiprocessor scheduling. *SIAM J. Comput.* 7(1): 1–17.

Degraeve Z, Jans R (2007) A new Dantzig-Wolfe reformulation and branch-and-price algorithm for the capacitated lot-sizing problem with setup times. *Oper. Res.* 55(5):909–920.

Delage E, Ye Y (2010) Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Oper. Res.* 58(3):595–612.

Dell'Olmo P, Kellerer H, Speranza MG, Tuza Z (1998) A 1312 approximation algorithm for bin packing with extendable bins. *Inform. Processing Lett.* 65(5):229–233.

Deng Y, Shen S, Denton B (2020) Chance-constrained surgery planning under uncertain or ambiguous surgery duration. *INFORMS J. Comput.* Forthcoming.

Denton BT, Miller AJ, Balasubramanian HJ, Huschka TR (2010) Optimal allocation of surgery blocks to operating rooms under uncertainty. *Oper. Res*. 58(4-part-1) 802–816.

Dinh T, Fukasawa R, Luedtke J (2018) Exact algorithms for the chance-constrained vehicle routing problem. *Math. Programming* 172(1–2):105–138.

Hashemi Doulabi SH, Rousseau L-M, Pesant G (2016) A constraint-programming-based branch-and-price-and-cut approach for operating room planning and scheduling. *INFORMS J. Comput.* 28(3):432–448.

Jiang R, Guan Y (2016) Data-driven chance constrained stochastic program. *Math. Programming* 158(1–2):291–327.

Kong Q, Lee C-Y, Teo C-P, Zheng Z (2013) Scheduling arrivals to a stochastic service delivery system using copositive cones. *Oper. Res.* 61(3):711–726.

Lamiri M, Xie X, Zhang S (2008) Column generation approach to operating theater planning with elective and emergency patients. *IIE Trans.* 40(9):838–852.

Liu X, Küçükyavuz S, Luedtke J (2016) Decomposition algorithms for two-stage chance-constrained programs. *Math. Programming* 157(1):219–243.

Luedtke J (2014) A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Math. Programming* 146(1–2):219–244.

Luedtke J, Ahmed S (2008) A sample approximation approach for optimization with probabilistic constraints. *SIAM J. Optim.* 19(2):674–699.

Lulli G, Sen S (2004) A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. *Management Sci.* 50(6):786–796.

Mak H-Y, Rong Y, Zhang J (2015) Appointment scheduling with limited distributional information. *Management Sci.* 61(2): 316–334.

Nursing Solutions Inc. (2016) 2016 national healthcare and RN retention report. Report, Nursing Solutions, Inc., East Petersburg, PA.

Pagnoncelli, BK, Ahmed S, Shapiro A (2009) Sample average approximation method for chance constrained programming: Theory and applications. *J. Optim. Theory Appl.* 142(2):399–416.

Prékopa A (1995) *Stochastic Programming* (Kluwer Academic Publishers, Dordrecht, Netherlands).

Qiu F, Ahmed S, Dey SS, Wolsey LA (2014) Covering linear programming with violations. *INFORMS J. Comput.* 26(3): 531–546.

Ryan DM, Foster BA (1981) An integer programming approach to scheduling. Wren A, ed. *Computer Scheduling of Public Transport, Urban Passenger Vehicle and Crew Scheduling* (North Holland Publishing Company, Amsterdam), 269–280.

Scarf H (1958) A min-max solution of an inventory problem. Arrow KJ, Karlin S, Scarf H, eds. *Studies in the Mathematical Theory of Inventory and Production* (Stanford University Press, Stanford, CA), 201–209.

Shylo OV, Prokopyev OA, Schaefer AJ (2012) Stochastic operating room scheduling for high-volume specialties under block booking. *INFORMS J. Comput.* 25(4):682–692.

Silva EF, Wood RK (2006) Solving a class of stochastic mixed-integer programs with branch and price. *Math. Programming* 108(2–3): 395–418.

Singh KJ, Philpott AB, Wood RK (2009) Dantzig-Wolfe decomposition for solving multistage stochastic capacity-planning problems. *Oper. Res.* 57(5):1271–1286.

Song Y, Luedtke JR, Küçükyavuz S (2014) Chance-constrained binary packing problems. *INFORMS J. Comput.* 26(4):735–747.

Strum DP, May JH, Sampson AR, Vargas LG, Spangler WE (2003) Estimating times of surgeries with two component procedures: Comparison of the lognormal and normal models. *Anesthesiology* 98(1):232–240.

Takriti S, Ahmed S (2004) On robust optimization of two-stage systems. *Math. Programming* 99(1):109–126.

Vanderbeck F (2000) On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Oper. Res.* 48(1):111–128.

Vanderbeck F, Wolsey LA (1996) An exact algorithm for IP column generation. *Oper. Res. Lett.* 19(4):151–159.

Wang Y, Tang J, Fung RYK (2014) A column-generation-based heuristic algorithm for solving operating theater planning problem under stochastic demand and surgery cancellation risk. *Internat. J. Production Econom.* 158:28–36.

Zhang Y, Jiang R, Shen S (2018) Ambiguous chance-constrained binary programs under mean-covariance information. *SIAM J. Optim.* 28(4):2922–2944.