The Stochastic Bilevel Continuous Knapsack Problem with Uncertain Follower's Objective

Christoph Buchheim[1] · Dorothee Henke[1] · Jannik Irmai[2]

Abstract

We consider a bilevel continuous knapsack problem where the leader controls the capacity of the knapsack, while the follower chooses a feasible packing maximizing his own profit. The leader's aim is to optimize a linear objective function in the capacity and in the follower's solution, but with respect to different item values. We address a stochastic version of this problem where the follower's profits are uncertain from the leader's perspective, and only a probability distribution is known. Assuming that the leader aims at optimizing the expected value of her objective function, we first observe that the stochastic problem is tractable as long as the possible scenarios are given explicitly as part of the input, which also allows to deal with general distributions using a sample average approximation. For the case of independently and uniformly distributed item values, we show that the problem is #P-hard in general, and the same is true even for evaluating the leader's objective function. Nevertheless, we present pseudo-polynomial time algorithms for this case, running in time linear in the total size of the items. Based on this, we derive an additive approximation scheme for the general case of independently distributed item values, which runs in pseudo-polynomial time.

Keywords Bilevel optimization · Stochastic optimization · Complexity

✉ Jannik Irmai
  jannik.irmai@tu-dresden.de

  Christoph Buchheim
  christoph.buchheim@math.tu-dortmund.de

  Dorothee Henke
  dorothee.henke@math.tu-dortmund.de

[1] Department of Mathematics, TU Dortmund University, Dortmund, Germany

[2] Faculty of Computer Science, Dresden University of Technology, Dresden, Germany

1 Introduction

In many real-world optimization problems, more than one decision-maker is involved. Often, decisions are taken in a hierarchical order: the first actor takes a decision that determines the feasible set and objective function of the second actor, whose decision in turn may influence the objective value of the first actor. Formally, such problems may be modeled as *bilevel* optimization problems, where the problem solved by the first actor is called the *upper level problem* and the one of the second actor the *lower level problem*. When such bilevel problems are considered from a game-theoretic perspective, the two decision-makers are often called *leader* and *follower*, a terminology we use throughout this paper. Typical applications of bilevel optimization arise when the general rules of a system (e.g., an energy market or a transport system) are determined by one actor (e.g., some regulatory commission or some large logistics company), while the other actors (e.g., energy producers or subcontractors) try to optimize their own objectives within the rules of the system. For general introductions to bilevel optimization, we refer to [5, 7, 8].

Bilevel optimization problems often turn out to be NP-hard. This is the case, in general, even when all constraints and objective functions are linear [11]. A notable exception is the bilevel continuous knapsack problem [8]: here, the leader controls the capacity of the knapsack, while the follower chooses a feasible packing maximizing his own profit. The leader's aim is to optimize an objective function that is linear in the chosen capacity and in the follower's chosen solution. However, the leader's item values may differ from the follower's item values. This problem can be solved efficiently by first sorting the items according to the follower's profits and then enumerating the capacities corresponding to the total size of each prefix in this ordering; see Sect. 2.1 for more details. Applications of the bilevel knapsack problem considered here arise in, e.g., revenue management [1].

In practice, however, it is very likely that the leader does not know the follower's subproblem exactly. It is thus natural to combine bilevel optimization with optimization under uncertainty. To the best of our knowledge, uncertain bilevel optimization problems were first considered in [15]. Recently, Burtscheidt and Claus [4] investigated stochastic bilevel linear optimization problems, dealing in particular with structural properties of such problems. A thorough review of literature on stochastic bilevel linear optimization can be found in [12].

Regarding the bilevel continuous knapsack problem under uncertainty, the *robust* optimization approach has been investigated in depth in [2]. It is assumed that the vector of follower's item values is unknown to the leader. This implies that the follower's order of preference for the items is now uncertain. However, according to the robust optimization paradigm, the leader knows a so-called *uncertainty set* containing all possible (or likely) realizations of this vector. The aim is to find a capacity leading to an optimal worst-case objective value over all these realizations. Among other things, it is shown that the resulting problem is still tractable under discrete or interval uncertainty—the latter case being nontrivial here, while it turns out to be NP-hard for budgeted uncertainty (which is sometimes called Gamma uncertainty) and for ellipsoidal uncertainty, among others. Complexity questions for general robust bilevel optimization problems have been settled in [3].

In the following, we consider the *stochastic* bilevel continuous knapsack problem. The follower's item values are still unknown to the leader, but now given by probability distributions. Instead of the worst case, we are interested in optimizing the expected value. The problem can be written as:

$$
\begin{aligned}
\max \quad & \mathbb{E}_{\mathbf{c}}\left(d^\top x^{\mathbf{c}} - \delta b\right) \\
\text{s.t.} \quad & b \in [b^-, b^+] \\
& x^{\mathbf{c}} \in \underset{}{\operatorname{argmax}} \quad \mathbf{c}^\top x \\
& \qquad\qquad \text{s.t.} \quad a^\top x \le b \\
& \qquad\qquad\qquad x \in [0, 1]^n,
\end{aligned} \tag{SP}
$$

where b denotes the capacity determined by the leader and $x \in [0, 1]^n$ are the optimization variables of the follower. As $\mathbf{c}$ is a random vector in the stochastic setting, the same is true for the follower's optimum solution $x^{\mathbf{c}}$ and the leader's objective value $d^\top x^{\mathbf{c}} - \delta b$. In the latter, the vector d contains the leader's item values and δ denotes the cost for each unit of capacity provided by the leader. The leader's aim is now to optimize the expected objective value. For the sake of simplicity, we do not distinguish between the optimistic and the pessimistic view (which are the two standard ways to handle ambiguous follower's optimal solutions) in this formulation. In Sect. 2.2, we argue why we may assume uniqueness of the follower's optimal solution almost surely in the stochastic setting. Therefore, the results presented in this paper hold in both the optimistic and the pessimistic setting.

1.1 Outline and Overview of Results

After discussing the deterministic problem version and introducing notation and basic results in Sect. 2, we start investigating the computational complexity of the stochastic optimization problem (SP), which of course depends strongly on the underlying probability distribution of $\mathbf{c}$. Under the assumption that all possible realizations of the follower's objective vector are given explicitly as part of the input, together with their probabilities, we observe that the stochastic problem can be solved efficiently; see Sect. 3. Using standard methods, this result could be used to design a sample average approximation scheme for arbitrary distributions.

Our main results apply to the case of independently distributed item values. In the most basic setting, each item value is distributed uniformly on either a finite set or an interval. In contrast to the setting of Sect. 3, in the discrete case, the input here contains the finite sets for all items, but not each of the (exponentially many) possible realizations explicitly. Even in this basic setting of independently and uniformly distributed item values, we show that the stochastic problem turns out to be #P-hard; see Sect. 4.1. It is thus unlikely that an efficient algorithm exists for solving the problem exactly. In fact, even the computation of the objective value resulting from a given capacity choice is #P-hard in these cases, and the same is true for finding a multiplicative approximation for any desired factor. However, all results only show weak #P-hardness. In fact, we also devise a pseudo-polynomial algorithm for the mentioned cases in Sect. 4.2, running in time linear in the total size of all items.

Finally, in Sect. 5, we consider general distributions with independent item values, given only by oracles for the cumulative distribution functions and the quantile functions. Assuming that these oracles can be queried in constant time, we devise an algorithm for solving (SP) with an arbitrarily small additive error $\varepsilon > 0$. The running time of this algorithm is pseudo-polynomial in the problem data and linear in $1/\varepsilon$. The idea of this approach is to approximate the given distribution by a componentwise discrete distribution and then to apply the main ideas used for the pseudo-polynomial algorithm of Sect. 4.2.

In Sect. 6, we summarize the main results and close the paper with a few remarks and a discussion of related questions.

2 Preliminaries

We start with basic observations concerning the stochastic bilevel continuous knapsack problem. We first have a closer look at the deterministic problem variant in Sect. 2.1. Subsequently, in Sect. 2.2, we introduce notation and list some basic observations concerning the stochastic problem (SP).

2.1 The Underlying Certain Problem

As mentioned in the introduction, the deterministic version of the bilevel continuous knapsack problem can be solved efficiently. This is explained in [8], but for the convenience of the reader and since our algorithms for the stochastic case build on this, we now describe the solution approach in more detail. The deterministic bilevel continuous knapsack problem can be formulated as follows, using the same notation as in the stochastic problem (SP):

$$
\begin{aligned}
\max \quad & d^\top x - \delta b \\
\text{s.t.} \quad & b \in [b^-, b^+] \\
& x \in \operatorname*{argmax} \ c^\top x \\
& \qquad \text{s.t.} \quad a^\top x \le b \\
& \qquad \qquad x \in [0, 1]^n \ .
\end{aligned}
\tag{P}
$$

The leader's only variable is $b \in \mathbb{R}$, which can be considered the knapsack's capacity. The follower's variables are $x \in \mathbb{R}^n$, i.e., the follower fills the knapsack with a subset of the objects, where also fractions are allowed. The item sizes $a \in \mathbb{R}^n_{\geq 0}$, the follower's item values $c \in \mathbb{R}^n$, the capacity bounds $b^-, b^+ \in \mathbb{R}_{\geq 0}$ as well as the leader's item values $d \in \mathbb{R}^n$ and a scalar $\delta \geq 0$ are given. The latter can be thought of as a price the leader has to pay for providing one unit of knapsack capacity. For the following, we define $A := \sum_{i=1}^n a_i$ and assume that $a > 0$ and $0 \le b^- \le b^+ \le A$. Moreover, since we are mostly interested in complexity results, we assume throughout that $a \in \mathbb{N}^n$ and $d \in \mathbb{Z}^n$. Finally, we will use the notation $[k] := \{1, \ldots, k\}$ for $k \in \mathbb{N}$ and $[0] := \emptyset$.

As usual in bilevel optimization, we have to be careful in case the follower's optimal solution is not unique. In this case, the model (P) is not well defined. The standard

approach is to distinguish between the optimistic setting, in which the follower always chooses one of his optimal solutions that is best possible for the leader, and the pessimistic setting, where the follower chooses an optimal solution that is worst possible for the leader. The former case is equivalent to considering the follower's variables x being under the leader's control as well. However, regarding the results presented in this paper, there are no relevant discrepancies between the two cases. For the sake of simplicity and since we can also make this assumption almost surely in the stochastic setting later on (see Sect. 2.2), we assume in this section that the follower's profits $\frac{c_i}{a_i}$, $i \in [n]$, are pairwise distinct and nonzero, so that his optimal solution is unique for any capacity b.

Indeed, the follower in (P) solves a continuous knapsack problem with fixed capacity b. This can be done, for example, using Dantzig's algorithm [6]: by first sorting the items, we may assume

$$\frac{c_1}{a_1} > \cdots > \frac{c_{n'}}{a_{n'}} > 0 > \frac{c_{n'+1}}{a_{n'+1}} > \cdots > \frac{c_n}{a_n}, \tag{1}$$

for some $n' \in \{0, \ldots, n\}$. The idea is then to pack all items with positive profit into the knapsack, in this order, until it is full. More formally, if $A' := \sum_{i=1}^{n'} a_i \leq b$, all items with positive profit can be taken, so an optimum solution is $x_i = 1$ for $i \in [n']$ and $x_i = 0$ else. Otherwise, we consider the critical item

$$k := \min \left\{ i \in [n']: \ \sum_{j=1}^{i} a_j > b \right\},$$

and an optimum solution is given by

$$x_i := \begin{cases} 1 & \text{for } i \in \{1, \ldots, k-1\} \\ \frac{1}{a_k} \left(b - \sum_{j=1}^{k-1} a_j \right) & \text{for } i = k \\ 0 & \text{for } i \in \{k+1, \ldots, n\}. \end{cases} \tag{2}$$

We now turn to the leader's perspective. As only the critical item k, but not the ordering (1) depends on b, the leader can compute the described order of items once and then consider the behavior of the follower's optimum solution x when b changes. Every x_i in (2) is a continuous piecewise linear function in b, of the form:

$$x_i(b) := \begin{cases} 0 & \text{for } b \in \left[0, \sum_{j=1}^{i-1} a_j\right] \\ \frac{1}{a_i} \left(b - \sum_{j=1}^{i-1} a_j \right) & \text{for } b \in \left[\sum_{j=1}^{i-1} a_j, \sum_{j=1}^{i} a_j\right] \\ 1 & \text{for } b \in \left[\sum_{j=1}^{i} a_j, A\right] \end{cases} \tag{3}$$

for $b \in [0, A]$ and for $i \in [n']$, and constantly zero for $i > n'$. The leader's objective function f is given by the corresponding values $d^\top x(b) - \delta b$ and thus corresponds to

a weighted sum of the functions $x_i(b)$ for $i \in [n']$ and δb:

$$
f(b) = \begin{cases} \sum_{j=1}^{i-1} d_j + \frac{d_i}{a_i} \left(b - \sum_{j=1}^{i-1} a_j \right) - \delta b & \text{for } b \in [\sum_{j=1}^{i-1} a_j, \sum_{j=1}^{i} a_j], \\ & i \in [n'] \\ \sum_{j=1}^{n'} d_j - \delta b & \text{for } b \in [A', A]. \end{cases} \tag{4}
$$

Note that this piecewise linear function is well-defined and continuous with vertices in the points $b = \sum_{j=1}^{i} a_j, i \in [n']$, in which the critical item changes from i to $i+1$. The leader has to maximize f over the range $[b^-, b^+] \subseteq [0, A]$. As f is piecewise linear, it suffices to evaluate it at the boundary points b^- and b^+ and at all feasible vertices, i.e., at all points $b = \sum_{j=1}^{i} a_j$ for $i \in [n']$ such that $b \in [b^-, b^+]$. By computing $f(b)$ incrementally, Problem (P) can be solved in $\mathcal{O}(n \log n)$ time, which is the time needed for sorting.

2.2 Basic Definitions and Observations

In the stochastic version of the problem, the vector $\mathbf{c}$ of follower's item values is seen as a random variable having a known distribution. The follower's optimum solution $x^{\mathbf{c}}(b)$ and the leader's objective value $f^{\mathbf{c}}(b) = d^\top x^{\mathbf{c}}(b) - \delta b$ depend on the realization of $\mathbf{c}$ and hence are also random variables. The leader optimizes the expected value $\mathbb{E}_{\mathbf{c}}(f^{\mathbf{c}}(b))$.

If two different items have the same profit or an item has profit zero, the follower's optimal solution might be ambiguous. For simplicity, we assume throughout this paper that the profits of two different items almost surely disagree and that the profit of each item is almost surely nonzero, i.e.,

$$
\mathbb{P}(\mathbf{c}_i/a_i = \mathbf{c}_j/a_j) = 0 \quad \text{and} \quad \mathbb{P}(\mathbf{c}_i = 0) = 0 \quad \text{for } i, j \in [n], i \neq j. \tag{5}
$$

If $\mathbf{c}$ follows a continuous distribution, i.e., its cumulative distribution function is continuous, this assumption is always satisfied. In case of a discrete distribution with finite support, it can be obtained, if necessary, by a small perturbation of the entries in the support of $\mathbf{c}$. Using an appropriate perturbation, both the optimistic and the pessimistic setting can be modeled. In particular, we do not need to distinguish between these two settings in the following because under Assumption (5), the follower's optimal solution is almost surely unique.

For fixed c, we have seen in Sect. 2.1 that x_i^c for $i \in [n]$ and f^c are piecewise linear functions in b. These functions do not depend on c directly, but only on the implied order of the items when the latter are sorted according to the values c_i/a_i. Hence, the expected values

$$
\hat{x}_i(b) := \mathbb{E}_{\mathbf{c}}(x_i^{\mathbf{c}}(b)) \text{ for } i \in [n] \text{ and} \tag{6}
$$

$$
\hat{f}(b) := \mathbb{E}_{\mathbf{c}}(f^{\mathbf{c}}(b)) = d^\top \hat{x}(b) - \delta b \tag{7}
$$

can be seen as expected values with respect to a probability distribution on all permutations of the items $1, \ldots, n$. As the number of permutations is finite, this implies that the functions $\hat{x}_i$ and $\hat{f}$ are finite convex combinations of functions x_i^c and f^c, respectively, for appropriate values of c. In particular, they are piecewise linear functions again. Since we assume the item sizes a to be integral, the vertices of these functions all lie on integer points $b \in \{0, \ldots, A\}$ because this holds for the functions defined in (3) and (4).

This gives rise to a general algorithmic scheme for solving the stochastic problem (SP): enumerate all permutations π of $[n]$ and compute the corresponding leader's objective functions f^π as in (4), together with the probabilities p_π that the values $\mathbf{c}_i/a_i$ are sorted decreasingly when permuted according to π. Finally, sum all piecewise linear functions $p_\pi f^\pi$ to determine the leader's objective $\hat{f}$, and maximize $\hat{f}$ over $b \in [b^-, b^+]$.

We emphasize that, depending on the given probability distribution of $\mathbf{c}$, it might be nontrivial to compute the probabilities p_π in general. Moreover, due to the exponential number of permutations, this approach does not yield a polynomial time algorithm in general. In fact, we will show that such an efficient algorithm cannot exist for some probability distributions unless $P = NP$. However, for distributions with finite support, enumerating all values in the support yields an efficient algorithm, as we show in the next section. Although the above algorithm is not efficient for other distributions, it will be useful for our proofs to know the structure of the follower's optimum solutions and the leader's objective function as described above.

Besides the piecewise linear functions defined in (6) and (7), we will also make use of the values

$$\Delta \hat{x}_i(b) := \hat{x}_i(b) - \hat{x}_i(b-1) \text{ for } i \in [n] \text{ and} \tag{8}$$

$$\hat{f}'(b) := \hat{f}(b) - \hat{f}(b-1) = d^\top \Delta \hat{x}(b) - \delta \tag{9}$$

for $b \in [A]$. The values $\Delta \hat{x}_i(b)$ describe the expected amount of item i that will be added when increasing the capacity from $b-1$ to b. Together with δ and the leader's item values d, they yield the slope $\hat{f}'$ of the leader's objective function. Note that, by integrality of a, the functions $\hat{x}_i$ and $\hat{f}$ are linear on $[b-1, b]$. Whenever we deal with slopes of piecewise linear functions, in a point where the function is nondifferentiable, this refers to the slope of the linear piece directly left of this point, i.e., we always consider left derivatives.

For some probability distributions, it will turn out that not only the optimization in (SP), but also the computation of the values $\hat{x}_i(b)$ and $\Delta \hat{x}_i(b)$ is hard in general. However, we will devise pseudo-polynomial time algorithms in these cases that compute all values $\Delta \hat{x}_i(b)$, from which one can solve (SP) in pseudo-polynomial time as well; see Sect. 4.2.

3 Distributions with Finite Support

Assuming that $\mathbf{c}$ has a finite support U, i.e., that there exists a finite set U of possible follower's objectives c which occur with probabilities $0 < p_c \leq 1$, respectively, the leader's objective is given as a finite sum of the piecewise linear functions $p_c f^c$, where f^c is defined as in (4). Note that the definition in (4) depends on the order of the items given by the follower's preferences, which in turn depends on c.

Similar to the algorithm described in Sect. 2.2, the following algorithm solves Problem (SP) for distributions with finite support: for every $c \in U$, use the algorithm described in Sect. 2.1 to compute the piecewise linear function f^c in $\mathcal{O}(n \log n)$ time, and multiply each function f^c by the factor p_c. Then, maximize the resulting weighted sum, which is a piecewise linear function again. Note that the sum has $\mathcal{O}(|U|n)$ linear segments and that it can be computed by sorting the vertices of all functions and traversing them from left to right while keeping track of the sum of the active linear pieces. This is possible in a running time of $\mathcal{O}(|U|n \log(|U|n))$. Thus, we obtain

Theorem 1 *Assume that $\mathbf{c}$ is distributed on a finite set U and that the input consists of U together with the corresponding probabilities. Then, Problem (SP) can be solved in $\mathcal{O}(|U|n \log(|U|n))$ time.*

The result of Theorem 1 suggests to address the general problem (SP), with an arbitrary underlying distribution of $\mathbf{c}$, by means of *sample average approximation*: for a given number $N \in \mathbb{N}$, first compute N samples $c^{(1)}, \ldots, c^{(N)}$ of the random variable $\mathbf{c}$. Then, apply the algorithm of Theorem 1 to the uniform distribution over the finite set $\{c^{(1)}, \ldots, c^{(N)}\}$ and let ϑ_N be the resulting optimal value (which is a random variable again). Using general results from [16], one can show that ϑ_N almost surely converges to the optimal value of (SP) for $N \to \infty$, and a similar statement holds for the set of optimal solutions; see [17]. The only assumption needed here is that sampling of $\mathbf{c}$ is possible.

4 Componentwise Uniform Distributions

In this section, we consider the version of (SP) where the distribution of $\mathbf{c}$ is uniform on a product of either finite sets or continuous intervals. Equivalently, each component of $\mathbf{c}$ is drawn independently and according to some (discrete or continuous) uniform distribution. For both the discrete and the continuous case, we will show that (SP) cannot be solved efficiently unless $P = NP$. However, we will devise pseudo-polynomial time algorithms with a running time linear in the total item size A. The algorithm for the discrete case solves the problem not only for uniform, but also for arbitrary componentwise distributions with finite support.

Note that the results presented in Sect. 3 are not applicable to the discrete *independent* case discussed here because the support U, which contains all possible combinations of item values, is exponential in the number of items and hence in the problem input.

4.1 Hardness Results

Our first aim is to show that (SP) is #P-hard in case of componentwise uniform distributions. The class #P contains all counting problems associated with decision problems belonging to NP, or, more formally, all problems that ask for computing the number of accepting paths in a polynomial time nondeterministic Turing machine. Using a natural concept of efficient reduction for counting problems, one can define a counting problem to be #P-hard if every problem in #P can be reduced to it. A polynomial time algorithm for a #P-hard counting problem can only exist if P$=$NP. In the following proofs, we will use the #P-hardness of the problem #Knapsack, which asks for the number of feasible solutions of a given binary knapsack instance [9].

In stochastic optimization with continuous distributions, problems often turn out to be #P-hard, and this is often even true for the evaluation of objective functions containing expected values. For an example, see [10], from where we also borrowed some ideas for the following proofs.

Theorem 2 *Problem* (SP) *with a discrete componentwise uniform distribution of* **c** *is* *#P-hard.*

Proof We show the result by a reduction from #Knapsack. More precisely, for some given $a^* \in \mathbb{N}^m$ and $b^* \in \{0, 1, \ldots, \sum_{i=1}^m a_i^*\}$, we will prove that one can compute

$$\#\{x \in \{0, 1\}^m : a^{*\top} x \leq b^*\}$$

in polynomial time if the following instances of (SP) can be solved in polynomial time. In case $b^* = \sum_{i=1}^m a_i^*$, this is clear, so from now on, we assume that $b^* < \sum_{i=1}^m a_i^*$.

We define a family of instances of (SP), parameterized by $\tau \in [-1, 1]$: each of the instances has $n := m + 1$ items, where

$$(a_1, \ldots, a_m, a_{m+1}) := (a_1^*, \ldots, a_m^*, \textstyle\sum_{i=1}^m a_i^*) \text{ and}$$
$$(d_1, \ldots, d_m, d_{m+1}) := ((1 + \tau) \cdot a_1, \ldots, (1 + \tau) \cdot a_m, (-1 + \tau) \cdot a_{m+1}).$$

We set $\delta := 0$, $b^- := 0$ and $b^+ := a_{m+1} = \sum_{i=1}^m a_i$, and assume

$$(\mathbf{c}_1, \ldots, \mathbf{c}_{m+1}) \sim \mathcal{U}\{\varepsilon, 1\}^{m+1}$$

with

$$\varepsilon := \tfrac{1}{2a_{m+1}} > 0.$$

The proof consists of two main steps. First, we investigate the structure of the leader's objective functions for the described instances and show that by determining the slope of any of them at $b = b^*$, up to a certain precision, we can compute

$$\#\{x \in \{0, 1\}^m : a^{*\top} x \leq b^*\}.$$

In the second step, we show how to determine this slope up to the required precision by solving a polynomial number of these instances in a bisection algorithm.

As described in Sect. 2.2, the leader's objective function can be thought of as a weighted sum of piecewise linear functions corresponding to the permutations induced by different choices of c, with weights being the probabilities of the permutations, respectively. For fixed c, consider the set

$$I_c := \left\{ i \in [m] \mid \frac{c_i}{a_i} > \frac{c_{m+1}}{a_{m+1}} \right\}$$

of items the follower would choose before item $m + 1$. The corresponding piecewise linear function f^c first has slope $1 + \tau$ and then slope $-1 + \tau$, since $\frac{d_i}{a_i} = 1 + \tau$ for all $i \in [m]$, and $\frac{d_{m+1}}{a_{m+1}} = -1 + \tau$. The order of the items in I_c does not matter to the leader because they all result in the same slope in her objective. The slope changes from $1 + \tau$ to $-1 + \tau$ at $b = \sum_{i \in I_c} a_i$. The slope would change back to $1 + \tau$ at $b = \sum_{i \in I_c} a_i + a_{m+1} \geq b^+$, but this is outside of the range of the leader's objective.

The actual leader's objective is now a weighted sum of such functions. For obtaining the weights, we only need to know the probabilities for different sets I_c, because all c resulting in the same I_c also result in the same piecewise linear function f^c. The probability distribution is chosen such that $I_c = [m]$ occurs with probability $\frac{1}{2} + \frac{1}{2^{m+1}}$, while each other set $I_c \subset [m]$ has probability $\frac{1}{2^{m+1}}$: first, if $c_{m+1} = \varepsilon$, we certainly have $I_c = [m]$ because $\frac{c_i}{a_i} \geq \frac{\varepsilon}{a_i} > \frac{\varepsilon}{a_{m+1}}$ holds with probability 1, for all $i \in [m]$. On the other hand, if $c_{m+1} = 1$, then each item $i \in [m]$ is contained in I_c with probability exactly $\frac{1}{2}$. Indeed, $c_i = 1$ means $\frac{c_i}{a_i} = \frac{1}{a_i} > \frac{1}{a_{m+1}}$, hence $i \in I_c$, whereas $c_i = \varepsilon$ means $\frac{c_i}{a_i} = \frac{\varepsilon}{a_i} \leq \frac{1}{2a_{m+1}} < \frac{1}{a_{m+1}}$, hence $i \notin I_c$. Thus, the leader's objective function $\hat{f}_\tau$ is given as:

$$\hat{f}_\tau = \frac{1}{2} f_{[m],\tau} + \frac{1}{2^{m+1}} \sum_{M \subseteq [m]} f_{M,\tau},$$

where $f_{M,\tau}$ is the function that has slope $1 + \tau$ for $b \in [0, \sum_{i \in M} a_i]$ and slope $-1 + \tau$ afterward. It follows that, for any $b \in [b^+]$,

$$\hat{f}_\tau'(b) = \frac{1}{2}(1 + \tau) + \frac{1}{2^{m+1}} \left((-1 + \tau) \cdot \#\left\{ M \subseteq [m] : \sum_{i \in M} a_i < b \right\} \right.$$

$$\left. + (1 + \tau) \cdot \#\left\{ M \subseteq [m] : \sum_{i \in M} a_i \geq b \right\} \right)$$

$$= -\frac{1}{2^m} \#\left\{ M \subseteq [m] : \sum_{i \in M} a_i \leq b - 1 \right\} + 1 + \tau$$

$$= -\frac{1}{2^m} \#\left\{ x \in \{0, 1\}^m : a^\top x \leq b - 1 \right\} + 1 + \tau. \tag{10}$$

This shows that by computing $\hat{f}'_\tau(b^* + 1)$ for any fixed τ, we can determine the number

$$\#\{x \in \{0, 1\}^m : a^{*\top}x \le b^*\} = 2^m(1 + \tau - \hat{f}'_\tau(b^* + 1)).$$

It is even enough to compute an interval of length less than $\frac{1}{2^m}$ containing $\hat{f}'_\tau(b^* + 1)$ because the number of feasible knapsack solutions is an integer and this gives an interval of length less than 1 in which it must lie. This concludes the first step of our proof.

In the second step, we will describe a bisection algorithm to compute $\hat{f}'_0(b^* + 1)$ up to the required precision. We know that $\hat{f}'_0(b^* + 1) \in [-1, 1]$ as it is a convex combination of values -1 and 1. Starting with $s_0^- := -1$ and $s_0^+ := 1$, we iteratively halve the length of the interval $[s_k^-, s_k^+]$ by setting either $s_{k+1}^- := s_k^-$ and $s_{k+1}^+ := \frac{1}{2}(s_k^- + s_k^+)$ or $s_{k+1}^- := \frac{1}{2}(s_k^- + s_k^+)$ and $s_{k+1}^+ := s_k^+$. After $m + 2$ iterations, we have an interval of length $\frac{1}{2^{m+1}}$ containing $\hat{f}'_0(b^* + 1)$, which allows to compute $\#\{x \in \{0, 1\}^m : a^{*\top}x \le b^*\}$.

It remains to show how to determine whether $\hat{f}'_0(b^* + 1) \le \frac{1}{2}(s_k^- + s_k^+)$ or $\hat{f}'_0(b^* + 1) \ge \frac{1}{2}(s_k^- + s_k^+)$, in order to choose the new interval. To this end, we first maximize f_τ for $\tau := -\frac{1}{2}(s_k^- + s_k^+)$ over $[b^-, b^+]$. This can be done by solving (SP) for the corresponding instance, which by our assumption is possible in polynomial time. Suppose the maximum is attained at b_{k+1}. As a weighted sum of concave functions, $\hat{f}_\tau$ is concave, and hence, we know that $\hat{f}'_\tau(b) \ge 0$ for all $b < b_{k+1}$, and $\hat{f}'_\tau(b) \le 0$ for all $b \ge b_{k+1}$. From (10), one can conclude that $\hat{f}'_\tau(b) = \hat{f}'_0(b) + \tau$ for all $\tau \in [-1, 1]$ and all $b \in [b^+]$. We derive that $\hat{f}'_0(b^* + 1) \ge -\tau = \frac{1}{2}(s_k^- + s_k^+)$ if $b^* + 1 < b_{k+1}$, and $\hat{f}'_0(b^* + 1) \le -\tau = \frac{1}{2}(s_k^- + s_k^+)$ otherwise. □

Theorem 3 *Problem* (SP) *with a continuous componentwise uniform distribution of* **c** *is #P-hard.*

Proof The result can be shown by a similar proof to the one of Theorem 2: Instead of the discrete distribution used before, the continuous distribution

$$(\mathbf{c}_1, \ldots, \mathbf{c}_m) \sim \mathcal{U} \prod_{i=1}^m \left[\frac{a_i}{2a_{m+1}}, \frac{3a_i}{2a_{m+1}}\right]$$

is considered, while fixing $c_{m+1} = 1$. The sets $I_\mathbf{c}$ are defined as before, and it can be shown that each set has probability $\frac{1}{2^m}$: if $\mathbf{c}_i \in (\frac{2a_i}{2a_{m+1}}, \frac{3a_i}{2a_{m+1}}]$, we have that $\frac{\mathbf{c}_i}{a_i} > \frac{1}{a_{m+1}}$, hence $i \in I_\mathbf{c}$, while $\mathbf{c}_i \in [\frac{a_i}{2a_{m+1}}, \frac{2a_i}{2a_{m+1}})$ implies $\frac{\mathbf{c}_i}{a_i} < \frac{1}{a_{m+1}}$, so that $i \notin I_\mathbf{c}$. Both events have probability 1/2. The rest of the proof is analogous, except for a small change in the computation in (10) due to the slightly different probabilities. □

Note that c_{m+1} is fixed in the proof of Theorem 3. In order to avoid this, one can consider $\mathbf{c}_{m+1} \sim \mathcal{U}[1 - \varepsilon, 1 + \varepsilon]$ instead, for sufficiently small $\varepsilon > 0$.

Theorem 4 *Evaluating the objective function of Problem* (SP) *with a discrete or continuous componentwise uniform distribution of* **c** *is #P-hard.*

Proof Using the same construction and notation as in the preceding proofs, we have shown that computing $\hat{f}_0'(b^* + 1) = \hat{f}_0(b^* + 1) - \hat{f}_0(b^*)$ is #P-hard. It follows that also evaluating $\hat{f}_0$ is #P-hard. □

Note that the proof of Theorem 4, together with (7) and (9), implies that already the computation of the values $\hat{x}_i(b)$ and $\Delta \hat{x}_i(b)$ for some given $i \in [n]$ and $b \in [A]$ can be #P-hard. Moreover, the constructions in the proofs of Theorem 2 and 3 show that all stated hardness results still hold when $\delta = 0$. One can show that, since all follower's item values in these constructions are positive, the hardness result also holds when assuming $d \geq 0$ and $\delta > 0$; see [2].

Considering these results, it is a natural question whether (SP) can at least be approximated efficiently. Assuming $b^- = 0$, this question is well defined, since all optimal values are nonnegative then, due to $\hat{f}(0) = 0$. However, it is easy to derive the following negative result, which excludes the existence of any polynomial time (multiplicative) approximation algorithm, unless P = NP.

Theorem 5 *For Problem* (SP) *with a discrete or continuous componentwise uniform distribution of* **c** *and with* $b^- = 0$*, it is #P-hard to decide whether the optimal value is zero.*

Proof For any given instance of (SP) and any $K \in \mathbb{R}_{\geq 0}$, we can efficiently construct a new instance of (SP) by adding an item $n + 1$ with leader's value $d_{n+1} = -K + \delta$, size $a_{n+1} = 1$, and any distribution of $\mathbf{c}_{n+1}$ guaranteeing $\mathbb{P}(\mathbf{c}_{n+1} > \mathbf{c}_i/a_i) = 1$ for all $i \in [n]$. Let $\hat{f}$ and $\hat{f}_K$ denote the objective functions of the original and the extended problem, respectively. By construction, the follower will always choose item $n + 1$ first in the extended instance. Thus, $\hat{f}_K(b) = -Kb \leq 0$ for $b \in [0, 1]$ and $\hat{f}_K(b) = \hat{f}(b - 1) - K$ for $b \in [1, b^+ + 1]$. In summary, we can polynomially reduce the decision of

$$\max_{b \in [0, b^+]} \hat{f}(b) \leq K \tag{11}$$

to the decision of

$$\max_{b \in [0, b^+ + 1]} \hat{f}_K(b) \leq 0.$$

Since the computation of an optimal solution for the original instance of (SP) can be polynomially reduced to problems of type (11), using a bisection algorithm, the desired result now follows from Theorems 2 and 3. □

4.2 Pseudo-Polynomial Time Algorithms

We now present pseudo-polynomial time algorithms for the stochastic bilevel continuous knapsack problem in the #P-hard cases addressed in the previous section. These algorithms are based on a dynamic programming approach. As discussed in Sect. 2.2, the leader's objective function is piecewise linear with vertices at integral positions

$b \in [A]$, since the item sizes a are assumed to be integral. Therefore, we could solve the problem by evaluating $\hat{f}(b)$ for $\mathcal{O}(A)$ many values of b, and A has polynomial size in the numerical values of the sizes a. The quest for a pseudo-polynomial algorithm thus reduces to the computation of $\hat{f}(b)$ for given $b \in [A]$. However, we have seen in the previous section that also the latter task is #P-hard in general.

Actually, as mentioned, even the computation of $\Delta \hat{x}_i(b)$ for given $i \in [n]$ and $b \in [A]$ can be #P-hard. In the following, we will present algorithms for computing $\Delta \hat{x}$ in pseudo-polynomial time for the distributions addressed above, i.e., for item values that are componentwise uniformly distributed on a finite set or on a closed interval. From $\Delta \hat{x}$ one can compute $\hat{x}$ using formula (8) and with that evaluate $\hat{f}$ for all integral capacities using formula (7), which takes pseudo-polynomial time $\mathcal{O}(nA)$.

In order to compute $\Delta \hat{x}$, we first define the auxiliary function

$$g_i(b, I) := \mathbb{P}\left(\mathbf{c}_i > 0 \text{ and } \sum_{j \in I, \, \mathbf{c}_j/a_j > \mathbf{c}_i/a_i} a_j = b \right),$$

for $i \in [n]$, $I \subseteq [n] \setminus \{i\}$, and $b \in \mathbb{Z}$, i.e., the probability that item i is profitable and that the total size of all items in I that the follower prefers over item i is exactly b. We then have

$$\Delta \hat{x}_i(b) = \frac{1}{a_i} \sum_{r=1}^{a_i} g_i\left(b - r, [n] \setminus \{i\}\right) \tag{12}$$

for all $i \in [n]$ and $b \in [A]$. Indeed, for $r \in [a_i]$, the value $g_i(b - r, [n] \setminus \{i\})$ describes the probability that a percentage of exactly r/a_i of item i is packed when the capacity is b, and hence $(r - 1)/a_i$ when the capacity is $b - 1$. The sum thus represents the probability that the percentage of item i being packed is increased by $1/a_i$ when increasing the capacity from $b - 1$ to b. Hence, the right hand side agrees with $\hat{x}_i(b) - \hat{x}_i(b - 1) = \Delta \hat{x}_i(b)$.

Note that $\mathbf{c}_j/a_j > \mathbf{c}_i/a_i$ and $\mathbf{c}_i > 0$ in the definition of $g_i(b, I)$ could be replaced equivalently by $\mathbf{c}_j/a_j \geq \mathbf{c}_i/a_i$ and $\mathbf{c}_i \geq 0$ by Assumption (5), and the same is true in the following whenever comparing profits of different items.

Determining all probabilities $g_i(b, [n] \setminus \{i\})$ would thus allow us to compute $\Delta \hat{x}$ and hence to solve Problem (SP) in pseudo-polynomial time. To this end, for $i \in [n]$, $I \subseteq [n] \setminus \{i\}$, $b \in \mathbb{Z}$, and $\gamma \in \mathbb{R}$, we next define

$$h_i(b, I, \gamma) := \mathbb{P}\left(\sum_{j \in I, \, \mathbf{c}_j/a_j > \gamma/a_i} a_j = b \right), \tag{13}$$

i.e., the probability that the total size of all items in I with profit larger than $\frac{\gamma}{a_i}$ is exactly b. For the following, let $\mathrm{supp}^+(\mathbf{c}_i) := \mathrm{supp}(\mathbf{c}_i) \cap \mathbb{R}_{>0}$.

From now on we assume componentwise independent item values. Under this assumption, for item values with finite support, we then have

$$g_i(b, I) = \sum_{\gamma \in \text{supp}^+(\mathbf{c}_i)} \mathbb{P}(\mathbf{c}_i = \gamma) \cdot h_i(b, I, \gamma), \tag{14}$$

while for absolutely continuously distributed item values, we have

$$g_i(b, I) = \int_{\text{supp}^+(\mathbf{c}_i)} p_i(\gamma) \cdot h_i(b, I, \gamma) \, d\gamma, \tag{15}$$

where $p_i(\gamma)$ denotes the probability density function of the random variable $\mathbf{c}_i$. For all $i \in [n]$ and $\gamma \in \mathbb{R}$, it is easy to verify that

$$h_i(b, \emptyset, \gamma) = \begin{cases} 1 & \text{for } b = 0 \\ 0 & \text{for } b \neq 0 \end{cases}$$

and, if $j \in I$ and $i \notin I$,

$$\begin{aligned} h_i(b, I, \gamma) = & \, \mathbb{P}(\mathbf{c}_j/a_j > \gamma/a_i) \cdot h_i(b - a_j, I\backslash\{j\}, \gamma) \\ & + (1 - \mathbb{P}(\mathbf{c}_j/a_j > \gamma/a_i)) \cdot h_i(b, I\backslash\{j\}, \gamma). \end{aligned} \tag{16}$$

This recursive formula enables an incremental computation of $h_i(b, [n]\backslash\{i\}, \gamma)$ for given $i \in [n]$, $\gamma \in \mathbb{R}$, and $b \in \{0, \dots, A\}$. We emphasize that only $\mathcal{O}(n)$ many subsets I of $[n]$ need to be considered for this, as, for the recursion (16), it suffices to choose one element j to be removed from I. Using this, we can now develop pseudo-polynomial time algorithms for computing $\Delta \hat{x}$ in case of independently and uniformly distributed item values, with supports being either finite sets or continuous intervals.

4.2.1 Componentwise Uniform Distributions with Finite Support

In this section, we assume that all item values are independently and discretely distributed with finite support. More specifically, for $i \in [n]$, the value of item i has $m_i \in \mathbb{N}$ different realizations $c_i^1, \dots, c_i^{m_i}$ with probabilities $p_i^1, \dots, p_i^{m_i}$, respectively. Let $m := \max\{m_1, \dots, m_n\}$ denote the maximum number of different values any item can take.

Lemma 1 *All probabilities $\mathbb{P}(\mathbf{c}_j/a_j > c_i^k/a_i)$ for $i, j \in [n]$, $i \neq j$, and $k \in [m_i]$ can be computed in time $\mathcal{O}(m^2 n^2)$.*

Proof Each such probability can be computed in time $\mathcal{O}(m)$ as:

$$\mathbb{P}(\mathbf{c}_j/a_j > c_i^k/a_i) = \sum_{\ell \in [m_j], \, c_j^\ell/a_j > c_i^k/a_i} p_j^\ell,$$

and the claim follows.   □

Lemma 2 *Let $i \in [n]$ and $\gamma \in \mathbb{R}$. Given the probabilities $\mathbb{P}(\mathbf{c}_j/a_j > \gamma/a_i)$ for all $j \in [n]\backslash\{i\}$, the probabilities $h_i(b, [n]\backslash\{i\}, \gamma)$ for all $b \in \{0, \dots, A - a_i\}$ can be computed in time $\mathcal{O}(nA)$.*

Proof We apply the recursive formula (16) in order to compute the desired probabilities $h_i(b, [n]\backslash\{i\}, \gamma)$. More specifically, setting $I := \emptyset$ and $b_{max} := 0$ at the beginning, we iterate over all $j \in [n]\backslash\{i\}$ in an arbitrary order. For each such j, we first compute

$$
\begin{aligned}
h_i(b, I \cup \{j\}, \gamma) = {} & \mathbb{P}(\mathbf{c}_j/a_j > \gamma/a_i) \cdot h_i(b - a_j, I, \gamma) \\
& + (1 - \mathbb{P}(\mathbf{c}_j/a_j > \gamma/a_i)) \cdot h_i(b, I, \gamma)
\end{aligned}
$$

for all $b \in \{0, \dots, b_{max} + a_j\}$, with $h_i(b, I, \gamma) = 0$ for $b \notin \{0, \dots, b_{max}\}$, and then set $I := I \cup \{j\}$ and $b_{max} := b_{max} + a_j$. After the last iteration, we then have computed $h_i(b, [n]\backslash\{i\}, \gamma)$ for all $b \in \{0, \dots, A - a_i\}$. There are $\mathcal{O}(n)$ iterations and each iteration can be executed in time $\mathcal{O}(A)$.   □

Theorem 6 *For item values that are independently distributed on finite sets, Problem (SP) can be solved in time $\mathcal{O}(m^2 n^2 + m n^2 A)$.*

Proof By Lemmas 1 and 2, all values $h_i(b, [n]\backslash\{i\}, c_i^k)$ for $i \in [n]$, $k \in [m_i]$, and $b \in \{0, \dots, A\}$ can be computed in time $\mathcal{O}(m^2 n^2 + m n^2 A)$. Next, all probabilities $g_i(b, [n]\backslash\{i\})$ can be computed in time $\mathcal{O}(mnA)$ according to (14). Finally, all values $\Delta \hat{x}_i(b)$ for $i \in [n]$ and $b \in [A]$ can be computed in time $\mathcal{O}(nA)$ using $\Delta \hat{x}_i(1) = \frac{1}{a_i} g_i(0, [n]\backslash\{i\})$ and

$$
\Delta \hat{x}_i(b) = \Delta \hat{x}_i(b - 1) + \frac{1}{a_i} \left(g_i(b, [n]\backslash\{i\}) - g_i(b - a_i, [n]\backslash\{i\}) \right)
$$

for $b \in \{2, \dots, A\}$, which follows from (12). From $\Delta \hat{x}$ one can compute $\hat{x}$ in time $\mathcal{O}(nA)$ using (8) and with that evaluate $\hat{f}(b)$ for all $b \in [A]$ in time $\mathcal{O}(nA)$ using (7).   □

In case the number of possible realizations for each component of $\mathbf{c}$ is bounded by a constant, e.g., when we have $m = 2$ as in the proof of Theorem 2, the running time stated in Theorem 6 simplifies to $\mathcal{O}(n^2 A)$.

4.2.2 Componentwise Continuous Uniform Distributions

In this section, we assume that the value $\mathbf{c}_i$ of item $i \in [n]$ is distributed uniformly on the continuous interval $[c_i^-, c_i^+]$ with $c_i^- < c_i^+$. The key difference to the discrete case discussed in the previous section is that we compute the probabilities $h_i(b, I, \gamma)$ not for fixed γ, but as functions in γ. The involved probabilities $\mathbb{P}(\mathbf{c}_j/a_j > \gamma/a_i)$ are piecewise linear in γ. As a result, $h_i(b, I, \gamma)$ is a piecewise polynomial function in γ and the expected values $\Delta \hat{x}$ can be computed as integrals over piecewise polynomial functions.

Consider $V := \{\max\{0, \frac{c_i^-}{a_i}\}, \max\{0, \frac{c_i^+}{a_i}\} \mid i \in [n]\}$ and set $r := |V|$. Let $v_1, \dots, v_r \in V$ with $v_1 < \cdots < v_r$ be the ascending enumeration of all elements in V. For $i, j \in [n]$, $i \neq j$, and $\gamma \in \mathbb{R}$, we have

$$\mathbb{P}(\mathbf{c}_j/a_j > \gamma/a_i) = \begin{cases} 1 & \text{for } \gamma \leq \frac{a_i}{a_j} c_j^- \\ \frac{c_j^+ - \frac{a_j}{a_i}\gamma}{c_j^+ - c_j^-} & \text{for } \frac{a_i}{a_j} c_j^- \leq \gamma \leq \frac{a_i}{a_j} c_j^+ \\ 0 & \text{for } \gamma \geq \frac{a_i}{a_j} c_j^+. \end{cases}$$

In particular, $\mathbb{P}(\mathbf{c}_j/a_j > \gamma/a_i)$, as a function in γ, is linear on each interval $[a_i v_k, a_i v_{k+1}]$.

Lemma 3 *Let $i \in [n]$ and $k \in [r-1]$ be given. Then, the coefficients of the polynomials $h_i(b, [n]\backslash\{i\}, \gamma)$ on the interval $(a_i v_k, a_i v_{k+1}] \subseteq supp^+(\mathbf{c}_i)$, for all $b \in \{0, \dots, A - a_i\}$, can be computed in time $\mathcal{O}(n^2 A)$.*

Proof Similarly to the proof of Lemma 2, we can compute $h_i(b, [n]\backslash\{i\}, \gamma)$ by iteratively applying the recursive formula (16). The only difference is that each application of the recursive formula involves two multiplications of a polynomial $h_i(b, I, \gamma)$ of degree $\mathcal{O}(n)$ with a linear function and the summation of the resulting polynomials. This can be done in time $\mathcal{O}(n)$ and the claim follows. □

Theorem 7 *For item values that are independently and uniformly distributed on continuous intervals, Problem (SP) can be solved in time $\mathcal{O}(n^4 A)$.*

Proof By Lemma 3, the piecewise polynomial functions $h_i(b, [n]\backslash\{i\}, \gamma)$ can be computed for all $i \in [n]$, all $b \in \{0, \dots, A - a_i\}$, and all of the at most $2n$ intervals $(a_i v_k, a_i v_{k+1}] \subseteq supp^+(\mathbf{c}_i)$ in time $\mathcal{O}(n^4 A)$. Using (15), all probabilities $g_i(b, [n]\backslash\{i\})$ can be obtained in time $\mathcal{O}(n^3 A)$ by computing $\mathcal{O}(n)$ integrals over polynomials of degree $\mathcal{O}(n)$ for each $i \in [n]$ and $b \in [A]$. As in Theorem 6, the claim follows. □

From the proofs of Lemma 3 and Theorem 7, it follows easily that a pseudo-polynomial algorithm exists for each class of continuous distributions such that the entries $\mathbf{c}_i$ are independently distributed and such that each $\mathbf{c}_i$ has a piecewise polynomial density function, assuming that the latter is given explicitly as part of the input. In particular, each entry $\mathbf{c}_i$ may have a support consisting of a finite union of bounded closed intervals such that $\mathbf{c}_i$ induces a uniform distribution on each of these intervals.

Moreover, the approach can deal with item values $\mathbf{c}_i$ that are given as weighted sums of independently and uniformly distributed random variables on continuous intervals, as long as the number of summands is fixed. Indeed, if some given distributions have piecewise polynomial density functions, then the density function of their sum (assuming independence) is again piecewise polynomial, and the number of polynomial pieces is bounded by the product of the numbers of pieces of the original density functions. In particular, this applies to the case where each component independently follows an Irwin–Hall distribution.

4.3 Correlated Distributions

The fact that (SP) can be solved in pseudo-polynomial time for uniform independently distributed item values raises the question of how much the complexity increases when considering *correlated* item values. This question is only well defined when we restrict ourselves to specific classes of distributions and when we specify how exactly the input is given. As an example, one may consider uniform distributions on general polytopes, instead of boxes as in Theorems 3 and 7. For this case, one can show that no pseudo-polynomial algorithm in the total item size A can exist unless $P = NP$, since the problem is already hard for unit sizes. However, the hardness derives from the complexity that can be modeled into the polytope, rather than from the stochastic optimization task itself, and the problem may turn tractable again if the numbers appearing in the description of the polytope are polynomially bounded; see [14] for details.

5 Additive Approximation Scheme for Componentwise Distributions

Building on the results of the previous section, we next devise a fully pseudo-polynomial time additive approximation scheme for Problem (SP) for arbitrary absolutely continuous distributions with independent components $\mathbf{c}_i$. This approach can be easily adapted to deal with discrete distributions (with finite or infinite support) as well. Recall that the case of finite support was settled by Theorem 6, but the running time of the corresponding algorithm depends on the size of the supports. It may thus be desirable to approximate a discrete distribution by another discrete distribution with a smaller support in order to obtain a faster running time, even if this comes with a small error.

For each $i \in [n]$, we assume that the distribution of $\mathbf{c}_i$ is given by an oracle for its cumulative distribution function, defined as:

$$F_{\mathbf{c}_i}(\gamma) := \mathbb{P}(\mathbf{c}_i \leq \gamma),$$

as well as an oracle for its quantile function

$$Q_{\mathbf{c}_i}(p) := \inf \{\gamma \in \mathbb{R} \mid p \leq F_{\mathbf{c}_i}(\gamma)\}.$$

For convenience, we set $F_{\mathbf{c}_i}(\infty) = 1$. If $F_{\mathbf{c}_i}$ is invertible, we have $Q_{\mathbf{c}_i} = F_{\mathbf{c}_i}^{-1}$.
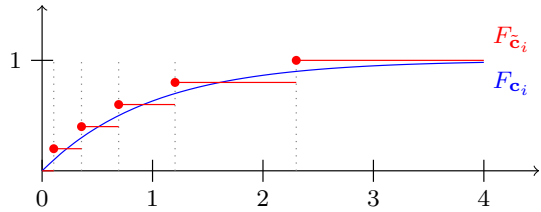
Starting with some desired additive accuracy $\varepsilon > 0$, we set

$$m := \left\lceil \tfrac{1}{\varepsilon}(n-1)AD \right\rceil + 1$$

with $D := \sum_{j=1}^{n} |d_j|$. The idea is to approximate each $\mathbf{c}_i$ by a new random variable $\tilde{\mathbf{c}}_i$ having a uniform distribution on the finite set $\{\tilde{c}_i^1, \ldots, \tilde{c}_i^m\}$, where

$$\tilde{c}_i^k := Q_{\mathbf{c}_i}\left(\tfrac{k-1/2}{m}\right)$$

Fig. 1 Approximation of $\mathbf{c}_i \sim \mathrm{Exp}(1)$ by $\tilde{\mathbf{c}}_i$, for $m = 5$

for $k \in [m]$; see Fig. 1 for an illustration.

Let $\tilde{h}_i(b, I, \gamma)$ be defined as in (13), but for $\tilde{\mathbf{c}}_i$ instead of $\mathbf{c}_i$. Then, for each $i \in [n]$ and $b \in \{0, \ldots, A - a_i\}$, the probabilities $\tilde{h}_i(b, I, \gamma)$ form a piecewise constant function in γ with all discontinuities belonging to the set

$$J_i := \left\{ \frac{a_i}{a_j} \tilde{c}_j^k \mid j \in [n] \backslash \{i\},\ k \in [m] \right\}.$$

By Lemma 2, for each fixed $i \in [n]$ and $\gamma \in \mathbb{R}$, the values $\tilde{h}_i(b, [n]\backslash\{i\}, \gamma)$ for all $b \in \{0, \ldots, A-a_i\}$ can be computed in time $\mathcal{O}(nA)$, given the probabilities $\mathbb{P}(\tilde{\mathbf{c}}_j/a_j > \gamma/a_i)$ for all $j \in [n]\backslash\{i\}$. Hence, the computation for all points $\gamma \in J_i$ can be done in time $\mathcal{O}(mn^2A)$. Each of the probabilities $\mathbb{P}(\tilde{\mathbf{c}}_j/a_j > \gamma/a_i)$ can be computed in constant time using the oracle for $F_{\mathbf{c}_j}$, since $F_{\tilde{\mathbf{c}}_j}(\gamma \frac{a_j}{a_i}) = 1 - \mathbb{P}(\tilde{\mathbf{c}}_j/a_j > \gamma/a_i)$ can be obtained from $F_{\mathbf{c}_j}(\gamma \frac{a_j}{a_i})$ by rounding it to the closest multiple of $1/m$, rounding up in case of a tie; compare Lemma 4. In summary, the computation of $\tilde{h}_i(b, [n]\backslash\{i\}, \gamma)$ for all $i \in [n]$, all $b \in \{0, \ldots, A\}$, and all $\gamma \in J_i$ takes time $\mathcal{O}(mn^3A)$.

Now let $j_1, \ldots, j_r$ be the elements of $J_i \cap \mathbb{R}_{>0}$ in ascending order, and set $j_0 = 0$ and $j_{r+1} = \infty$. Denoting the probability density function of $\mathbf{c}_i$ by p_i again, we set

$$\tilde{g}_i(b, I) := \int_0^\infty p_i(\gamma) \cdot \tilde{h}_i(b, I, \gamma)\, d\gamma = \sum_{k=0}^r \int_{j_k}^{j_{k+1}} p_i(\gamma) \cdot \tilde{h}_i(b, I, j_k)\, d\gamma$$

$$= \sum_{k=0}^r \tilde{h}_i(b, I, j_k)(F_{\mathbf{c}_i}(j_{k+1}) - F_{\mathbf{c}_i}(j_k)).$$

Using the oracle for evaluating $F_{\mathbf{c}_i}$, we can thus compute $\tilde{g}_i(b, [n]\backslash\{i\})$ for all $i \in [n]$ and all $b \in \{0, \ldots, A\}$ from the relevant values of $\tilde{h}_i$ in time $\mathcal{O}(mn^2A)$. The total time needed to compute the required values $\tilde{g}_i(b, [n]\backslash\{i\})$ is thus given as $\mathcal{O}(mn^3A)$. Proceeding exactly as in the proof of Theorem 6, based on the values $\tilde{g}_i(b, [n]\backslash\{i\})$ instead of $g_i(b, [n]\backslash\{i\})$, we can finally compute some $\tilde{b} \in [b^-, b^+]$ which maximizes the resulting objective function. The entire algorithm runs in pseudo-polynomial time

$$\mathcal{O}(mn^3A) = \mathcal{O}\left(\tfrac{1}{\varepsilon}n^4A^2D\right).$$

We claim that the computed solution is actually an ε-approximate solution for the original problem, in the additive sense. To show this, we first observe that the cumulative distribution functions of $\mathbf{c}_i$ and $\tilde{\mathbf{c}}_i$ have a pointwise difference of at most $\frac{1}{2m}$.

Lemma 4 *Let $i \in [n]$. Then, $||F_{\mathbf{c}_i} - F_{\tilde{\mathbf{c}}_i}||_\infty \leq \frac{1}{2m}$.*

Proof Let $k \in [m]$. For any $\beta \geq \tilde{c}_i^k$, we derive from the definition of $Q_{\mathbf{c}_i}$ that there exists $\gamma \leq \beta$ such that $\frac{k-1/2}{m} \leq F_{\mathbf{c}_i}(\gamma)$, and hence $\frac{k-1/2}{m} \leq F_{\mathbf{c}_i}(\beta)$ by monotonicity of $F_{\mathbf{c}_i}$. For any $\beta < \tilde{c}_i^k$, the definition of $Q_{\mathbf{c}_i}$ directly implies that $\frac{k-1/2}{m} > F_{\mathbf{c}_i}(\beta)$. In particular, for $k \in [m-1]$ and $\beta \in [\tilde{c}_i^k, \tilde{c}_i^{k+1})$, we obtain $F_{\tilde{\mathbf{c}}_i}(\beta) = \frac{k}{m}$ and $F_{\mathbf{c}_i}(\beta) \in [\frac{k-1/2}{m}, \frac{k+1/2}{m})$, and hence $|F_{\mathbf{c}_i}(\beta) - F_{\tilde{\mathbf{c}}_i}(\beta)| \leq \frac{1}{2m}$. For $\beta \in [0, \tilde{c}_i^1)$, we have $F_{\tilde{\mathbf{c}}_i}(\beta) = 0$ and $F_{\mathbf{c}_i}(\beta) \leq \frac{1}{2m}$, while $\beta \in [\tilde{c}_i^m, \infty)$ implies $F_{\tilde{\mathbf{c}}_i}(\beta) = 1$ and $F_{\mathbf{c}_i}(\beta) \geq \frac{m-1/2}{m} = 1 - \frac{1}{2m}$. $\qquad\square$

Lemma 5 *Let b^* be an optimizer of* (SP) *for the original distribution of $\mathbf{c}$ and let $\tilde{b}$ be computed as described above. Then, $|\hat{f}(\tilde{b}) - \hat{f}(b^*)| \leq \varepsilon$.*

Proof For $i \in [n]$, $I \subseteq [n] \backslash \{i\}$, and $\gamma \in \mathbb{R}$, define

$$\delta_i(I, \gamma) := \max_b |h_i(b, I, \gamma) - \tilde{h}_i(b, I, \gamma)|.$$

Then, using (16) for any $j \in I$, we obtain

$$\delta_i(I, \gamma) = \max_b |(1 - F_{\mathbf{c}_j}(\gamma \tfrac{a_j}{a_i}))h_i(b - a_j, I \backslash \{j\}, \gamma) + F_{\mathbf{c}_j}(\gamma \tfrac{a_j}{a_i})h_i(b, I \backslash \{j\}, \gamma)$$
$$- (1 - F_{\tilde{\mathbf{c}}_j}(\gamma \tfrac{a_j}{a_i}))\tilde{h}_i(b - a_j, I \backslash \{j\}, \gamma) - F_{\tilde{\mathbf{c}}_j}(\gamma \tfrac{a_j}{a_i})\tilde{h}_i(b, I \backslash \{j\}, \gamma)|$$
$$= \max_b |(1 - F_{\tilde{\mathbf{c}}_j}(\gamma \tfrac{a_j}{a_i}))(h_i(b - a_j, I \backslash \{j\}, \gamma) - \tilde{h}_i(b - a_j, I \backslash \{j\}, \gamma))$$
$$+ F_{\tilde{\mathbf{c}}_j}(\gamma \tfrac{a_j}{a_i})(h_i(b, I \backslash \{j\}, \gamma) - \tilde{h}_i(b, I \backslash \{j\}, \gamma))$$
$$+ (F_{\tilde{\mathbf{c}}_j}(\gamma \tfrac{a_j}{a_i}) - F_{\mathbf{c}_j}(\gamma \tfrac{a_j}{a_i}))(h_i(b - a_j, I \backslash \{j\}, \gamma) - h_i(b, I \backslash \{j\}, \gamma))|$$
$$\leq \max_b (1 - F_{\tilde{\mathbf{c}}_j}(\gamma \tfrac{a_j}{a_i})) \cdot |h_i(b - a_j, I \backslash \{j\}, \gamma) - \tilde{h}_i(b - a_j, I \backslash \{j\}, \gamma)|$$
$$+ F_{\tilde{\mathbf{c}}_j}(\gamma \tfrac{a_j}{a_i}) \cdot |h_i(b, I \backslash \{j\}, \gamma) - \tilde{h}_i(b, I \backslash \{j\}, \gamma)|$$
$$+ \underbrace{|F_{\tilde{\mathbf{c}}_j}(\gamma \tfrac{a_j}{a_i}) - F_{\mathbf{c}_j}(\gamma \tfrac{a_j}{a_i})|}_{\leq \frac{1}{2m} \text{ (Lemma 4)}} \cdot \underbrace{|h_i(b - a_j, I \backslash \{j\}, \gamma) - h_i(b, I \backslash \{j\}, \gamma)|}_{\leq 1}$$
$$\leq \max_b (1 - F_{\tilde{\mathbf{c}}_j}(\gamma \tfrac{a_j}{a_i})) \cdot \delta_i(I \backslash \{j\}, \gamma) + F_{\tilde{\mathbf{c}}_j}(\gamma \tfrac{a_j}{a_i}) \cdot \delta_i(I \backslash \{j\}, \gamma) + \tfrac{1}{2m}$$
$$= \delta_i(I \backslash \{j\}, \gamma) + \tfrac{1}{2m},$$

which by induction implies $\delta_i(I, \gamma) \leq \frac{n-1}{2m}$ for all $I \subseteq [n] \backslash \{i\}$ and all $\gamma \in \mathbb{R}$. We thus obtain

$$|g_i(b, I) - \tilde{g}_i(b, I)| \leq \int_0^\infty p_i(\gamma)|h_i(b, I, \gamma) - \tilde{h}_i(b, I, \gamma)| \, d\gamma$$
$$\leq \frac{n-1}{2m} \int_0^\infty p_i(\gamma) \, d\gamma = \frac{n-1}{2m}$$

for all $i \in [n]$, $I \subseteq [n] \setminus \{i\}$, and $b \in \{0, \ldots, A\}$. It follows from (12) that the resulting additive error in $\Delta \hat{x}_i(b)$ is at most $\frac{n-1}{2m}$ and thus the additive error in $\hat{x}_i(b)$ is at most $b \frac{n-1}{2m}$ for all $i \in [n]$ using (8). Finally, the additive error in $\hat{f}(b)$, for any b, is bounded by

$$\sum_{i=1}^{n} |d_i| b \frac{n-1}{2m} \leq \frac{(n-1)AD}{2m} \leq \tfrac{1}{2}\varepsilon,$$

which implies the desired result.   □

It is easy to see that the proof of Lemma 5 also works when all or some components of **c** follow a discrete distribution. For this, it suffices to adapt the definition of $\tilde{g}_i$ and the estimation of its error. Altogether, we thus obtain

Theorem 8 *Assume that the components of* **c** *are independently distributed and that the distribution of each component* $\mathbf{c}_i$ *is given by oracles for its cumulative distribution function and its quantile function. Then, there exists a fully pseudo-polynomial time additive approximation scheme for* (SP).

6 Conclusion

We have settled the complexity status of the stochastic bilevel continuous knapsack problem with uncertain follower's item values for different types of distributions. In case of a distribution with finite and explicitly given support, the problem is tractable. If the item values are independently and uniformly distributed, the problem is #P-hard in general, both for continuous and discrete distributions, but we devise pseudo-polynomial algorithms for both cases. Finally, we present a fully pseudo-polynomial additive approximation scheme for the case of arbitrary distributions with independent item values.

For the (single-level) binary knapsack problem, it is a classical result that the well-known pseudo-polynomial algorithm can be turned into an FPTAS by a natural rounding approach [13]. Unfortunately, such an approach fails for the pseudo-polynomial algorithm presented in Sect. 4.2. In fact, by Theorem 5, an FPTAS for the stochastic bilevel continuous knapsack problem under the distributions considered in Sect. 4 cannot exist, at least not in the multiplicative sense.

Finally, several interesting variants of the stochastic bilevel continuous knapsack problem arise when replacing the expected value in (SP) by some other risk measure, e.g., by a higher moment or a combination of different moments. This would allow to take also the variance into account, as is common in mean-risk optimization. Also other risk measures such as the conditional value at risk may be considered. For a collection of related results along the lines of Sects. 3 and 4.1, see [17].

Data Availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Brotcorne, L., Hanafi, S., Mansi, R.: A dynamic programming algorithm for the bilevel knapsack problem. Oper. Res. Lett. **37**(3), 215–218 (2009). https://doi.org/10.1016/j.orl.2009.01.007
2. Buchheim, C., Henke, D.: The robust bilevel continuous knapsack problem with uncertain coefficients in the follower's objective. J. Glob. Optim. (2022). https://doi.org/10.1007/s10898-021-01117-9
3. Buchheim, C., Henke, D., Hommelsheim, F.: On the complexity of robust bilevel optimization with uncertain follower's objective. Oper. Res. Lett. **49**(5), 703–707 (2021). https://doi.org/10.1016/j.orl.2021.07.009
4. Burtscheidt, J., Claus, M.: Bilevel linear optimization under uncertainty. In: Dempe, S., Zemkoho, A. (eds.) Bilevel Optimization: Advances and Next Challenges, pp. 485–511. Springer (2020). https://doi.org/10.1007/978-3-030-52119-6_17
5. Colson, B., Marcotte, P., Savard, G.: An overview of bilevel optimization. Ann. Oper. Res. **153**(1), 235–256 (2007). https://doi.org/10.1007/s10479-007-0176-2
6. Dantzig, G.B.: Discrete-variable extremum problems. Oper. Res. **5**(2), 266–277 (1957). https://doi.org/10.1287/OPRE.5.2.266
7. Dempe, S.: Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. Optimization **52**(3), 333–359 (2003). https://doi.org/10.1080/0233193031000149894
8. Dempe, S., Kalashnikov, V., Pérez-Valdés, G.A., Kalashnykova, N.: Bilevel Programming Problems. Springer (2015). https://doi.org/10.1007/978-3-662-45827-3
9. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co Ltd (1979). https://doi.org/10.1090/S0273-0979-1980-14848-X
10. Hanasusanto, G.A., Kuhn, D., Wiesemann, W.: A comment on "computational complexity of stochastic programming problems". Math. Program. **159**, 557–569 (2016). https://doi.org/10.1007/s10107-015-0958-2
11. Hansen, P., Jaumard, B., Savard, G.: New branch-and-bound rules for linear bilevel programming. SIAM J. Sci. Stat. Comput. **13**(5), 1194–1217 (1992). https://doi.org/10.1137/0913069
12. Henkel, C.: An algorithm for the global resolution of linear stochastic bilevel programs. Ph.D. thesis, University of Duisburg-Essen (2014)
13. Ibarra, O.H., Kim, C.E.: Fast approximation algorithms for the knapsack and sum of subset problems. J. ACM **22**(4), 463–468 (1975). https://doi.org/10.1145/321906.321909
14. Irmai, J.: The Stochastic Bilevel Selection Problem. Master's thesis, TU Dortmund University (2021)
15. Patriksson, M., Wynter, L.: Stochastic mathematical programs with equilibrium constraints. Oper. Res. Lett. **25**(4), 159–167 (1999). https://doi.org/10.1016/S0167-6377(99)00052-8
16. Shapiro, A., Dentcheva, D., Ruszczynski, A.: Lectures on Stochastic Programming. Society for Industrial and Applied Mathematics (2009). https://doi.org/10.1137/1.9780898718751

17. Warwel, L.V.: The Stochastic Bilevel Continuous Knapsack Problem. Master's thesis, TU Dortmund University (2020)

Publisher's Note   Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.