# Semi-Infinite Relaxations for the Dynamic Knapsack Problem with Stochastic Item Sizes


Daniel Blado  Weihong Hu  Alejandro Toriello

H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology

Atlanta, Georgia 30332

{deblado, weihongh} *at* gatech *dot* edu  atoriello *at* isye *dot* gatech *dot* edu


August 20, 2015


**Abstract**

We consider a version of the knapsack problem in which an item size is random and revealed only when the decision maker attempts to insert it. After every successful insertion the decision maker can choose the next item dynamically based on the remaining capacity and available items, while an unsuccessful insertion terminates the process. We propose a new semi-infinite relaxation based on an affine value function approximation, and show that an existing pseudo-polynomial relaxation corresponds to a non-parametric value function approximation. We compare both theoretically to other relaxations from the literature and also perform a computational study. Our main empirical conclusion is that our new relaxation provides tight bounds over a variety of different instances and becomes tighter as the number of items increases.


## 1 Introduction

The deterministic knapsack problem is one of the fundamental discrete optimization models studied by researchers in operations research, computer science, industrial engineering, and management science for many decades. It arises in a variety of applications, and also appears as a sub-problem or sub-structure in more complex optimization problems and algorithms. Relaxations of the knapsack problem have in particular been studied both as benchmarks for the problem itself, and also within general mixed-integer programming to derive valid inequalities. Work in this vein includes classical studies on valid inequalities for the knapsack polytope, such as covers and lifted covers (see [31] and references therein), and more recent results concerning extended formulations, relaxation schemes and extension complexity, e.g. [5, 33].

Knapsack problems under uncertainty have also received attention, both to model resource allocation applications with uncertain parameters, and also as substructures of more general discrete optimization models under uncertainty, such as stochastic integer programs [35]. Specifically, recent trends in both methodology and application have focused attention on models in which the uncertain data is not revealed at once after an initial decision stage, but rather is dynamically revealed over time based on the decision maker's choices; such models have applications in scheduling [11], equipment replacement [12] and machine learning [17, 18, 26], to name a few.

The model we study here is a knapsack problem with stochastic item sizes and this dynamic revealing of information: The decision maker has a list of available items, but only has a probability distribution for each item's size. Each size is revealed or realized only after the decision maker attempts to insert it, and the insertion is successful (and the process continues) only if the size is less than or equal to the remaining capacity in the knapsack. This dynamic paradigm contrasts with more static approaches, such as a chance-constrained model in which the decision maker chooses an entire set of items whose total size fits in the knapsack with at least a pre-specified probability [16].

Providing the decision maker with the flexibility to observe sizes as they are realized possibly increases the attainable expected value while satisfying the knapsack capacity with certainty. However, this additional model flexibility also implies additional complexity from both a practical and theoretical point of view; a feasible solution to this problem comes in the form of a policy that must prescribe what to do under any potential circumstance, rather than simply a subset of items. This additional difficulty has motivated work to both design efficient policies with good performance, and also to devise reasonably tight, yet tractable relaxations. Our results focus mostly on the latter question, and consist of the following main contributions:

*i*) We introduce a semi-infinite relaxation for the problem under arbitrary item size distributions, based on an affine value function approximation of the linear programming encoding of the problem's dynamic program. We show that the number of constraints in this relaxation is at worst countably infinite, and is polynomial in the input for distributions with finite support (assuming the distributions are part of the input).

*ii*) When item sizes have integer support, we show that a non-parametric value function approximation gives the relaxation from [26], which has pseudo-polynomially many variables and constraints.

*iii*) We theoretically and empirically compare these relaxations to others from the literature and show that both are quite tight. In particular, our new relaxation is notably tighter than a variety of benchmarks and compares favorably to the theoretically stronger pseudo-polynomial relaxation when this latter bound can be computed.

Our computational study employs a variety of policies related to or derived from various relaxations. Our results also show that even quite simple policies perform very well, especially as the number of items grows. More generally, our results may indicate a way to derive relaxations for more complex stochastic integer programs with dynamic aspects, such as those studied in [41].

The remainder of the paper is organized as follows. We conclude this section with a brief literature review. Section 2 formulates the problem and handles preliminaries. Section 3 introduces the semi-infinite relaxation and proves its structural results. Section 4 then discusses deriving the stronger relaxation when item sizes have integer support. Section 5 explains how to extend our methods to a more general model where an item's value may be stochastic and correlated to its size. Section 6 outlines the results of our empirical study, and Section 7 concludes. An Appendix contains detailed computational results.

### 1.1 Literature Review

In its full generality, this problem was first proposed and studied by [9, 11], though earlier research had studied the problem specifically with exponential item size distributions [12]. The computer science community has focused on problems of this kind, developing bounding techniques and approximation algorithms; in addition to [9, 11], other results in this vein include [4, 10, 17, 18, 26].

The knapsack problem and its generalizations have been studied for half a century or more, with many applications in areas as varied as budgeting, finance and scheduling; see [21, 28]. Knapsack problems under uncertainty have specifically received attention for several decades; [21, Chapter

14] surveys some of these results. For general packing under uncertainty see [10, 41]. As with optimization under uncertainty in general, models and solution approaches can be split into those that choose an a priori solution, sometimes also called *static* models [29], and models that dynamically choose items based on realized parameters, also called *adaptive* [10, 11]. Different authors have also studied uncertainty in different components of the problem. For example, a priori or static models with uncertain item values include [7, 19, 30, 36, 38], static models with uncertain item sizes include [15, 16, 22, 23], and [29] study a static model with uncertainty in both value and size. Dynamic or adaptive models for knapsacks with uncertain item sizes include the previously mentioned work [4, 9, 11, 12, 17, 18], while [20] study a dynamic model with uncertain item values. Other variants include *stochastic and dynamic models* [24, 25, 32] in which items are not available ahead of time but arrive dynamically according to a stochastic process.

The idea of obtaining relaxations of dynamic programs using value function approximations in the Bellman recursion dates back to [34, 40]. The technique gained wider use within the operations research community beginning with [1, 8], to obtain relaxations and also corresponding policies. It has since then been applied in a variety of stochastic dynamic programming models with discrete structure, such as inventory routing [2] and the traveling salesman problem [39].

## 2 Problem Formulation

Let $N := \{1, \ldots, n\}$ be a set of items. For each item $i \in N$ we have a non-negative random variable $A_i$ with known distribution representing its size, and a deterministic value $c_i > 0$. Item sizes are independent, and we can accommodate random values by using their expectation, as long as size and value are independent for each item. Section 5 below discusses how to extend our techniques to the case when an item's size and value may be correlated; see also [17, 18, 26]. We have a knapsack of deterministic capacity $b > 0$, and we would like to maximize the expected total value of inserted items. An item's size is realized when we choose to insert it, and we receive its value only if the knapsack's remaining capacity is greater than or equal to the realized size. Given any remaining capacity $s \in [0, b]$, we may choose to insert any available item, and the decision is irrevocable; see [17, 18, 26] for models that allow preemption. If the insertion is unsuccessful, i.e. the realized size is greater than the remaining capacity, the process terminates.

The problem can be modeled as a *dynamic program* (DP). The classical DP formulation for the deterministic knapsack [13] chooses an arbitrary ordering of the items and evaluates them one at a time, deciding whether to insert each one or not. However, to respond to realized item sizes it may be necessary to consider all available items together without imposing an order. We therefore use a more general DP formulation with state space given by $(M, s)$, where $\varnothing \neq M \subseteq N$ represents items available to insert and $s \in [0, b]$ is the remaining knapsack capacity. The optimal expected value is $\upsilon^*_N(b)$, where the optimal value function $\upsilon^*$ is defined recursively as

$$\upsilon^*_M(s) := \max_{i \in M} \big\{ \mathsf{P}(A_i \leq s)(c_i + \mathsf{E}[\upsilon^*_{M \setminus i}(s - A_i) | A_i \leq s]) \big\}, \tag{1}$$

and we take $\upsilon^*_\varnothing(s) := 0$. The *linear programming* (LP) formulation of this equation system is

$$\min_{\upsilon} \ \upsilon_N(b) \tag{2a}$$

$$\text{s.t. } \upsilon_{M \cup i}(s) - \mathsf{P}(A_i \leq s)\mathsf{E}[\upsilon_M(s - A_i) | A_i \leq s] \geq c_i \mathsf{P}(A_i \leq s), \quad \forall\, i \in N, M \subseteq N \setminus i, s \in [0, b] \tag{2b}$$

$$\upsilon \geq 0. \tag{2c}$$

In this doubly infinite LP the domain of each $\upsilon_M : [0, b] \to \mathbb{R}_+$ is an appropriate functional space [3].

**Notation** To alleviate the notational burden in the remainder of the paper, we identify singleton sets with their unique element when there is no danger of confusion. We denote an item size's cumulative distribution function by $\mathsf{F}_i(s) := \mathsf{P}(A_i \le s)$ for $i \in N$, and its complement by $\bar{\mathsf{F}}_i(s) := \mathsf{P}(A_i > s)$. Similarly, the quantity $\tilde{\mathsf{E}}_i(s) := \mathsf{E}[\min\{s, A_i\}]$ is the *mean truncated size* of item $i \in N$ at capacity $s \in [0, b]$ [9, 11, 41], and features prominently in our discussion. Intuitively, when the knapsack's remaining capacity is $s$, we should not care about item $i$'s distribution above $s$, since any realization of greater size results in the same outcome – an unsuccessful insertion.

## 3 Semi-Infinite Bound

The stochastic knapsack problem contains its deterministic counterpart as a special case, and is therefore at least NP-hard. Moreover, [41] shows that several variants of the problem are in fact PSPACE-hard. In general, therefore, we cannot expect to solve the LP (2) directly. However, any feasible $\upsilon$ provides an upper bound $\upsilon_N(b)$ on the optimal expected value. One possibility is to approximate the value function with an affine function,

$$\upsilon_M(s) \approx qs + r_0 + \sum_{i \in M} r_i, \tag{3}$$

where $r \in \mathbb{R}_+^{N \cup 0}$ and $q \in \mathbb{R}_+$. In this approximation, $q$ is the marginal value of the remaining knapsack capacity, $r_0$ represents the intrinsic value of having the knapsack available, and each $r_i$ represents the intrinsic value of having item $i \in M$ available to insert.

**Proposition 3.1.** *The best possible bound given by approximation* (3) *is the solution of the semi-infinite linear program*

$$\min_{q,r} \; qb + r_0 + \sum_{i \in N} r_i \tag{4a}$$

$$\text{s.t. } q\tilde{\mathsf{E}}_i(s) + r_0 \bar{\mathsf{F}}_i(s) + r_i \ge c_i \mathsf{F}_i(s), \qquad \forall\, i \in N, s \in [0, b] \tag{4b}$$

$$r, q \ge 0. \tag{4c}$$

*Proof.* Using (3),

$$\begin{aligned}
&\upsilon_{M \cup i}(s) - \mathsf{P}(A_i \le s)\mathsf{E}[\upsilon_M(s - A_i) | A_i \le s] \\
&= qs + r_0 + \sum_{j \in M \cup i} r_j - \mathsf{F}_i(s)\mathsf{E}\left[q(s - A_i) + r_0 + \sum_{j \in M} r_j \middle| A_i \le s\right] \\
&= qs\bar{\mathsf{F}}_i(s) + q\mathsf{F}_i(s)\mathsf{E}[A_i | A_i \le s] + r_0\bar{\mathsf{F}}_i(s) + r_i + \bar{\mathsf{F}}_i(s) \sum_{j \in M} r_j \\
&= q\tilde{\mathsf{E}}_i(s) + r_0\bar{\mathsf{F}}_i(s) + r_i + \bar{\mathsf{F}}_i(s) \sum_{j \in M} r_j \\
&\ge q\tilde{\mathsf{E}}_i(s) + r_0\bar{\mathsf{F}}_i(s) + r_i,
\end{aligned}$$

with equality holding when $M = \varnothing$ or $\bar{\mathsf{F}}_i(s) = 0$. □

**Example 3.2** (Deterministic Knapsack)**.** Suppose the item sizes are deterministic, so the problem becomes the well-known deterministic knapsack. Let $a_i \in [0, b]$ be item $i$'s size; we then have

$$q\tilde{\mathsf{E}}_i(s) + r_0\bar{\mathsf{F}}_i(s) + r_i = \begin{cases} qs + r_0 + r_i, & s < a_i \\ qa_i + r_i, & s \geq a_i. \end{cases}$$

When $s < a_i$, constraints (4b) are dominated by non-negativity since $c_i\mathsf{F}_i(s) = 0$, and hence we can set $r_0 = 0$. The constraints for all $s \geq a_i$ map to a single deterministic constraint, and we obtain the LP

$$\begin{aligned} \min_{q,r} \ & qb + \sum_{i \in N} r_i \\ \text{s.t. } & qa_i + r_i \geq c_i, && \forall\, i \in N \\ & r, q \geq 0. \end{aligned}$$

This is the dual of the deterministic knapsack's LP relaxation. Our bound therefore generalizes this LP relaxation to the dynamic setting with stochastic item sizes.

To solve (4), we must efficiently manage the uncountably many constraints. For each item $i \in N$, the separation problem is

$$\max_{s \in [0,b]} \big\{ (r_0 + c_i)\mathsf{F}_i(s) - q\tilde{\mathsf{E}}_i(s) \big\}. \tag{5}$$

The CDF $\mathsf{F}_i$ is upper semi-continuous, and the mean truncated size function $\tilde{\mathsf{E}}_i$ is continuous, concave and non-decreasing, so the maximum is always attained. Efficient separation then depends on the item's distribution.

**Proposition 3.3.** *If $\mathsf{F}_i$ is piecewise convex in the interval $[0, b]$, we can solve the separation problem* (5) *by examining only values corresponding to the CDF's breakpoints between convex intervals.*

*Proof.* Because of the concavity of $\tilde{\mathsf{E}}_i$, if $\mathsf{F}_i$ is convex, the most violated inequality will always be at $s \in \{0, b\}$. More generally, if the CDF is piecewise convex, within each convex interval the most violated inequality will be at the endpoints. □

Even if the CDF is not piecewise convex, it is almost everywhere differentiable [37, Theorem 3.4]. Therefore, we can still partition $[0, b]$ into at most a countable number of segments within which it is either convex or concave. By Proposition 3.3, we only need to check the endpoints of any convex segment. We may assume without loss of generality that the CDF is differentiable within each concave segment (since otherwise we can further partition the segment).

**Proposition 3.4.** *Within a segment $(\underline{s}, \hat{s}) \subseteq [0, b]$ where $\mathsf{F}_i$ is concave and differentiable,* (5) *can be solved by evaluating $\underline{s}$, $\hat{s}$ and all solutions to*

$$(r_0 + c_i)\frac{d}{ds}\mathsf{F}_i(s) = q\bar{\mathsf{F}}_i(s) \qquad s \in (\underline{s}, \hat{s}). \tag{6}$$

*Proof.* Let $g(s) := (r_0 + c_i)\mathsf{F}_i(s) - q\tilde{\mathsf{E}}_i(s)$. Then

$$\begin{aligned} g(s) &= (r_0 + c_i + qs)\mathsf{F}_i(s) - q\mathsf{F}_i(s)\mathsf{E}[A_i | A_i \leq s] - qs \\ &= (r_0 + c_i + qs)\mathsf{F}_i(s) - q\int_0^s a d\mathsf{F}_i(a) - qs. \end{aligned}$$

It follows that $g$ is differentiable when $\mathsf{F}_i$ is differentiable. Deriving with respect to $s$,

$$\begin{aligned}\frac{d}{ds}g(s) &= (r_0 + c_i)\frac{d}{ds}\mathsf{F}_i(s) + qs\frac{d}{ds}\mathsf{F}_i(s) + q\mathsf{F}_i(s) - qs\frac{d}{ds}\mathsf{F}_i(s) - q \\ &= (r_0 + c_i)\frac{d}{ds}\mathsf{F}_i(s) + q\mathsf{F}_i(s) - q = (r_0 + c_i)\frac{d}{ds}\mathsf{P}(A_i \le s) - q\bar{\mathsf{F}}_i(s). \qquad \square\end{aligned}$$

Even lacking piecewise convexity in the CDF, it may be possible to efficiently account for all constraints. We discuss some specific distributions next.

**Example 3.5** (Finite Distribution)**.** Suppose $A_i$ can take on a finite number of possible values $\{a_k\}_{k=1}^K$, where $0 \le a_1 < \cdots < a_K$. In this case, the CDF is piecewise constant, and thus piecewise convex, so the constraints can be modeled explicitly as long as $K$ is considered part of the problem input.

**Example 3.6** (Uniform Distribution)**.** Suppose $A_i$ is uniformly distributed between $[\underline{a}, \hat{a}]$, where $0 \le \underline{a} < \hat{a} \le b$. (The requirement $\hat{a} \le b$ is for ease of exposition.) $\mathsf{F}_i$ is again piecewise convex, and we obtain

$$(r_0 + c_i)\mathsf{F}_i(s) - q\tilde{\mathsf{E}}_i(s) = \begin{cases} -qs \le 0, & s \in [0, \underline{a}) \\ \frac{1}{\hat{a}-\underline{a}}\big(\frac{1}{2}qs^2 + s(r_0 + c_i - q\hat{a}) + \frac{1}{2}q\underline{a}^2 - (r_0 + c_i)\underline{a}\big), & s \in [\underline{a}, \hat{a}) \\ r_0 + c_i - \frac{1}{2}q(\hat{a} + \underline{a}), & s \in [\hat{a}, b]. \end{cases}$$

Therefore the most violated inequality is always at $s \in \{0, \hat{a}\}$. For $s = 0$, the inequality is dominated by the non-negativity constraints, so we only need to add the constraint $\frac{1}{2}q(\hat{a} + \underline{a}) + r_i \ge c_i$; we can once again set $r_0 = 0$.

**Example 3.7** (Exponential and Geometric Distributions)**.** If $A_i$ is exponentially distributed with rate $\lambda > 0$, $\mathsf{F}_i$ is concave. Nevertheless, we get

$$(r_0 + c_i)\mathsf{F}_i(s) - q\tilde{\mathsf{E}}_i(s) = \left(r_0 + c_i - \frac{q}{\lambda}\right)(1 - e^{-\lambda s}),$$

which is maximized at $s \in \{0, b\}$. As before, the case $s = 0$ is dominated by non-negativity, so we only add the constraint $\frac{1}{\lambda}q(1 - e^{-\lambda b}) + r_0 e^{-\lambda s} + r_i \ge c_i(1 - e^{-\lambda b})$; it can be shown that $r_0 = 0$ here as well without loss of optimality. An analogous argument shows that only the inequalities at $s \in \{0, b\}$ are necessary when $A_i$ follows a geometric distribution.

**Example 3.8** (Conditional Normal Distribution)**.** Suppose $A_i$ follows a normal distribution with mean $\mu \ge 0$ and standard deviation $\sigma > 0$, conditioned on being non-negative. $\mathsf{F}_i$ is then convex in $[0, \mu]$ and concave thereafter. Moreover, it is straightforward to see that $(r_0 + c_i)\mathsf{F}_i(s) - q\tilde{\mathsf{E}}_i(s)$ is convex in $[0, \mu + q\sigma^2/(r_0 + c_i)]$ and concave afterwards. Because this function's limit as $s \to \infty$ is $r_0 + c_i - q\mathsf{E}[A_i]$, it must be increasing in $[\mu + q\sigma^2/c_i, \infty)$. It follows that the most violated inequality is always at $s \in \{0, b\}$, so we only add the constraint (4b) for $s = b$. As with the other examples where this is the only constraint needed, it can be shown that $r_0 = 0$ without loss of optimality.

The next example shows that $r_0$ can drastically affect the bound given by (4).

**Example 3.9** (Bernoulli Distribution)**.** Suppose the knapsack has unit capacity, and each item has unit value and size following a Bernoulli distribution with parameter $p \in (0, 1)$. From Example 3.5, each item $i$ has constraints only at $s \in \{0, 1\}$. Suppose we impose $r_0 = 0$; then for any $n \ge 1$, the (restricted) optimal solution of (4) is $\hat{r}_i = c_i\mathsf{F}_i(0) = 1 - p$ for each $i \in N$ and $\hat{q} = (1 - \hat{r}_i)/\tilde{\mathsf{E}}_i(1) = 1$, yielding the objective $\sum_{i \in N} \hat{r}_i + \hat{q} = 1 + n(1-p)$. On the other hand, the optimal value for any $n$ is bounded above by the expected number of Bernoulli trials before the second success, which is

$$p^2 \sum_{k=0}^{\infty} (k+1)^2 (1-p)^k = \frac{2-p}{p}.$$

Once we include $r_0$ in (4), the optimal solution becomes $r_0^* = c_i \mathsf{F}_i(0)/\bar{\mathsf{F}}_i(0) = (1-p)/p$, $q^* = c_i \mathsf{F}_i(1)/\tilde{\mathsf{E}}_i(1) = 1/p$ and $r_i^* = 0$ for all $i \in N$, yielding an objective value of $(2-p)/p$, which is asymptotically tight.

## 3.1 Primal Relaxation

The finite-support dual of (4) yields a "relaxed primal", and gives further insight into the approximation:

$$\max_x \sum_{i \in N} \sum_{s \in [0,b]} c_i x_{i,s} \mathsf{F}_i(s) \tag{7a}$$

$$\text{s.t.} \sum_{i \in N} \sum_{s \in [0,b]} x_{i,s} \tilde{\mathsf{E}}_i(s) \le b \tag{7b}$$

$$\sum_{i \in N} \sum_{s \in [0,b]} x_{i,s} \bar{\mathsf{F}}_i(s) \le 1 \tag{7c}$$

$$\sum_{s \in [0,b]} x_{i,s} \le 1, \qquad \forall\, i \in N \tag{7d}$$

$$x \ge 0, \quad x \text{ has finite support.} \tag{7e}$$

This is a two-dimensional, semi-infinite, fractional *multiple-choice knapsack problem* [21], also called a fractional knapsack problem with *generalized upper bound constraints* (see e.g. [31]). The model has the following interpretation: For any feasible policy, $x_{i,s}$ represents the probability the policy attempts to insert item $i$ when $s$ capacity remains; clearly, the probability of attempting to insert $i$ at any point cannot exceed 1 (7d). Similarly, there cannot be more than one failed insertion (7c). Finally, for an attempted insertion, if the item's size exceeds the remaining capacity $s$, suppose we count this remaining capacity as a "fractional" insertion; then the total expected size the policy inserts, including any "fractionally" inserted size, does not exceed the knapsack's capacity (7b).

**Lemma 3.10.** *Problem* (7) *is a strong dual for problem* (4)*.*

*Proof.* By [14, Theorems 5.3 and 8.4], (7) is a strong dual if the cone of valid inequalities of (4), the *characteristic cone*, is closed. This cone is closed if for each $i \in N$ the set of inequalities implied by (4b) and the non-negativity constraints (4c) is closed. This is equivalent to the following set being closed,

$$\operatorname{conv}\big\{\big(\tilde{\mathsf{E}}_i(s), \bar{\mathsf{F}}_i(s), 1, c_i \mathsf{F}_i(s)\big) : s \in [0,b]\big\} + \{(\theta,0,0,0) : \theta \ge 0\} + \{(0,\theta,0,0) : \theta \ge 0\}$$
$$+ \{(0,0,\theta,0) : \theta \ge 0\} + \{(0,0,0,-\theta) : \theta \ge 0\},$$

where the sum is a Minkowski sum. The first set in the sum, which we denote $Q$ for convenience, represents all non-trivial valid inequalities for item $i \in N$ that do not weaken any coefficient, re-scaled so $r_i$'s coefficient is one. The remaining sets represent any potential weakening of the inequality, either by increasing a left-hand side coefficient, or by decreasing the right-hand side.

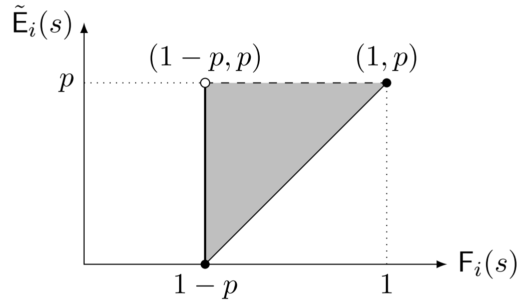


Figure 1: Two-dimensional projection of possible inequality (4b) coefficients for $q$ (vertical axis) versus right-hand side (horizontal axis) when $A_i$ has a Bernoulli distribution with parameter $p$. The thick solid line and black dots represent all possible coefficient values, and the dark gray triangle represents the convex hull of these values. This set does not include the white dot nor the dashed line and is therefore not closed.

Note that $Q$ by itself is not necessarily closed; see Figure 1 for an example. We will construct a convergent sequence in $Q$ and show that its limit can be achieved, perhaps by weakening a stronger inequality. For $t \in \mathbb{N}$, let $(\rho_k^t)$ and $(s_k^t)$ for $k = 1, \ldots, 4$ respectively be a sequence of convex multiplier 4-tuples and knapsack capacity 4-tuples yielding a convergent sequence

$$\left(\sum_k \rho_k^t \tilde{\mathsf{E}}_i(s_k^t), \sum_k \rho_k^t \bar{\mathsf{F}}_i(s_k^t), 1, c_i \sum_k \rho_k^t \mathsf{F}_i(s_k^t)\right) \to (\ell_q, \ell_{r_0}, 1, \ell_{\mathrm{RHS}}) \qquad \text{as } t \to \infty.$$

($Q$ is at most three-dimensional, so each convex combination requires at most four terms.) By iteratively replacing the sequence with a subsequence if necessary, we may assume $s_k^t \to \hat{s}_k$ and $\rho_k^t \to \hat{\rho}_k$ for each $k$. Then

$$\ell_q = \sum_k \hat{\rho}_k \tilde{\mathsf{E}}_i(\hat{s}_k), \qquad \ell_{r_0} \geq \sum_k \hat{\rho}_k \bar{\mathsf{F}}_i(\hat{s}_k) \qquad \ell_{\mathrm{RHS}} \leq c_i \sum_k \hat{\rho}_k \mathsf{F}_i(\hat{s}_k),$$

where we respectively use the continuity, lower semi-continuity and upper semi-continuity of $\tilde{\mathsf{E}}_i$, $\bar{\mathsf{F}}_i$ and $\mathsf{F}_i$. We can then recover the limit inequality by weakening $r_0$'s coefficient or the right hand side if necessary. □

We next compare (7) to a bound from the literature. The following linear knapsack relaxation appeared in [11]:

$$\max_x \sum_{i \in N} c_i x_{i,b} \mathsf{F}_i(b) \tag{8a}$$

$$\text{s.t.} \sum_{i \in N} x_{i,b} \tilde{\mathsf{E}}_i(b) \leq 2b \tag{8b}$$

$$0 \leq x_{i,b} \leq 1, \qquad i \in N. \tag{8c}$$

Even though this formulation only has one variable per item, we keep the two-index notation for consistency. The variables also have similar interpretations; $x_{i,b}$ in (8) represents the probability that a policy attempts to insert an item at any point.

**Theorem 3.11.** *The optimal value of* (7) *is less than or equal to the optimal value of* (8).

Intuitively, (8) seems weaker because it must double the knapsack capacity. In fact, for certain distributions, such as the ones covered in Examples 3.2, 3.6, 3.7 and 3.8, (7) is simply (8) with the original capacity of $b$.

*Proof.* Multiplying constraint (7c) by $b$ and adding it to constraint (7b), we can relax (7) to

$$\begin{aligned}
\max_x \quad & \sum_{i \in N} \sum_{s \in [0,b]} c_i x_{i,s} \mathsf{F}_i(s) \\
\text{s.t.} \quad & \sum_{i \in N} \sum_{s \in [0,b]} x_{i,s} \big( \tilde{\mathsf{E}}_i(s) + b \bar{\mathsf{F}}_i(s) \big) \leq 2b \\
& \sum_{s \in [0,b]} x_{i,s} \leq 1, && \forall\, i \in N \\
& x \geq 0, \quad x \text{ has finite support.}
\end{aligned}$$

The proof is finished by showing that $\tilde{\mathsf{E}}_i(s) + b\bar{\mathsf{F}}_i(s) \geq \tilde{\mathsf{E}}_i(b)$ for any $s \in [0,b]$, because after applying this further relaxation the optimal solution would have $x_{i,s} = 0$ for $s \neq b$. Indeed,

$$\begin{aligned}
\tilde{\mathsf{E}}_i(s) + b\bar{\mathsf{F}}_i(s) &= \mathsf{F}_i(s)\mathsf{E}[A_i | A_i \leq s] + s\bar{\mathsf{F}}_i(s) + b\big(\bar{\mathsf{F}}_i(s) - \bar{\mathsf{F}}_i(b)\big) + b\bar{\mathsf{F}}_i(b) \\
&= \mathsf{F}_i(s)\mathsf{E}[A_i | A_i \leq s] + s\bar{\mathsf{F}}_i(s) + b\big(\mathsf{F}_i(b) - \mathsf{F}_i(s)\big) + b\bar{\mathsf{F}}_i(b) \\
&\geq \mathsf{F}_i(s)\mathsf{E}[A_i | A_i \leq s] + \big(\mathsf{F}_i(b) - \mathsf{F}_i(s)\big)\mathsf{E}[A_i | s < A_i \leq b] + b\bar{\mathsf{F}}_i(b) \\
&= \tilde{\mathsf{E}}_i(b),
\end{aligned}$$

where in the inequality we use $s\bar{\mathsf{F}}_i(s) \geq 0$ and $b \geq \mathsf{E}[A_i | s < A_i \leq b]$. □

**Corollary 3.12** ([11, Theorem 4.1])**.** *The multiplicative gap between the optimal value of the stochastic knapsack problem $v^*_N(b)$ and the bound given by (4) and (7) is at most $32/7 \approx 4.57$.*

Example 3.2 shows that the relaxation (7) reduces to the deterministic knapsack's LP relaxation when item sizes are deterministic. This LP's gap is well known to be two [21, 28], and thus (7)'s gap cannot be less than two.

[11] also present a stronger polymatroid relaxation which has constraints similar to (8) applied to every subset of items. We are not able to prove that (7) dominates this bound; however, we discuss an empirical comparison of the two bounds in Section 6.

# 4 A Stronger Relaxation of Pseudo-Polynomial Size

Item sizes may have integer support in many cases. The knapsack capacity $b$ can then be taken to be integer as well, and it may be small enough that enumerating all possible integers up to it is computationally tractable. If both assumptions hold, we can produce better value function approximations of pseudo-polynomial size. For a state $(M, s)$ with $s \in \mathbb{Z}_+$, consider now the approximation

$$v_M(s) \approx \sum_{i \in M} r_i + \sum_{\sigma=0}^{s} w_\sigma, \tag{9}$$

where $r \in \mathbb{R}^N_+$ and $w \in \mathbb{R}^{b+1}_+$; the $r_i$'s have the same interpretation from before as intrinsic values of each item, and each $w_\sigma$ represents the incremental intrinsic value of having $\sigma$ capacity left instead of $\sigma - 1$. For a fixed $M$, this approximation allows a completely arbitrary non-decreasing function of the capacity $s$; in particular, we can recover (3) by setting $w_0 = r_0$ and $w_\sigma = q$ for $\sigma > 0$, and this shows that (9) can produce a tighter relaxation.

**Proposition 4.1** ([26])**.** *The model*

$$\max_x \sum_{i\in N}\sum_{s=0}^{b} c_i x_{i,s}\mathsf{F}_i(s) \tag{10a}$$

$$\text{s.t.} \sum_{i\in N}\sum_{s=\sigma}^{b} x_{i,s}\bar{\mathsf{F}}_i(s-\sigma) \le 1, \qquad \sigma = 0,\dots,b \tag{10b}$$

$$\sum_{s=0}^{b} x_{i,s} \le 1, \qquad i \in N \tag{10c}$$

$$x \ge 0 \tag{10d}$$

*gives an upper bound for the optimal value $v_N^*(b)$ when item sizes have integer support.*

The decision variables here have an identical interpretation to (7); $x_{i,s}$ is the probability the policy attempts to insert item $i$ when $s$ capacity remains in the knapsack. The probability of attempting to insert $i$ still cannot exceed 1 (10c). Similarly, the $\sigma$-th unit of capacity can be used at most once (10b). While this result is known from [26], our interpretation of the bound as arising from the approximation (9) is new.

*Proof.* Substituting (9) into (2b), we obtain

$$\begin{aligned} &v_{M\cup i}(s) - \mathsf{F}_i(s)\mathsf{E}[v_M(s-A_i)|A_i \le s] \\ &= \sum_{j\in M\cup i} r_j + \sum_{\sigma\le s} w_\sigma - \mathsf{F}_i(s)\sum_{j\in M} r_j - \sum_{s'\le s}\Big[\big(\mathsf{F}_i(s') - \mathsf{F}_i(s'-1)\big)\sum_{\sigma\le s-s'} w_\sigma\Big] \\ &= r_i + \bar{\mathsf{F}}_i(s)\sum_{j\in M} r_j + \sum_{\sigma\le s} w_\sigma\bar{\mathsf{F}}_i(s-\sigma) \ge r_i + \sum_{\sigma\le s} w_\sigma\bar{\mathsf{F}}_i(s-\sigma) \ge c_i\mathsf{F}_i(s), \end{aligned}$$

where as before the first inequality holds at equality when $M = \varnothing$ or $\bar{\mathsf{F}}_i(s) = 0$. The best bound from an approximation given by (9) satisfying these conditions is thus

$$\min_{r,w} \sum_{i\in N} r_i + \sum_{\sigma=0}^{b} w_\sigma \tag{11a}$$

$$\text{s.t. } r_i + \sum_{\sigma=0}^{s} w_\sigma\bar{\mathsf{F}}_i(s-\sigma) \ge c_i\mathsf{F}_i(s), \qquad i \in N, s = 0,\dots,b \tag{11b}$$

$$r, w \ge 0, \tag{11c}$$

precisely the dual of (10). (Because item sizes have integer support, the number of constraints in this model can be taken as finite, and thus classical LP duality applies.) □

The interpretation of (10) via the value function approximation (9) also allows us to compare it to another pseudo-polynomial bound from the literature. The following relaxation appeared in [17, 18]:

$$\max_x \sum_{i\in N}\sum_{s=0}^{b} c_i x_{i,s}\mathsf{F}_i(s) \tag{12a}$$

$$\text{s.t.} \sum_{i \in N} \sum_{s=\sigma}^{b} x_{i,s} \tilde{\mathsf{E}}_i(b-\sigma) \le 2(b-\sigma), \qquad \sigma = 0, \ldots, b \tag{12b}$$

$$\sum_{s=0}^{b} x_{i,s} \le 1, \qquad i \in N \tag{12c}$$

$$x \ge 0. \tag{12d}$$

Intuitively, this formulation applies the idea for (8) not only for the full capacity $b$, but also by assuming the knapsack has $\sigma$ fewer units of capacity for every $\sigma = 0, \ldots, b$.

**Theorem 4.2.** *The optimal value of* (10) *is less than or equal to the optimal value of* (12)*.*

This theorem is a stronger version of a similar result in [26], which showed that (10) is tighter than (12) in a worst-case sense.

*Proof.* Augment approximation (9) with redundant linear splines at every integer capacity, yielding

$$\upsilon_M(s) \approx \sum_{\sigma=0}^{s} q_\sigma (s-\sigma)_+ + \sum_{i \in M} r_i + \sum_{\sigma=0}^{s} w_\sigma,$$

where $q \ge 0$. These new functions cannot improve the approximation, since for any $M$ (9) already captures an arbitrary non-decreasing function of capacity. Nevertheless, adding these redundant variables makes the proof simpler. Following a similar argument to Propositions 3.1 and 4.1, this approximation results in the relaxation

$$\begin{aligned}
\max_x \quad & \sum_{i \in N} \sum_{s=0}^{b} c_i x_{i,s} \mathsf{F}_i(s) \\
\text{s.t.} \quad & \sum_{i \in N} \sum_{s=\sigma}^{b} x_{i,s} \tilde{\mathsf{E}}_i(s-\sigma) \le b - \sigma, && \sigma = 0, \ldots, b \\
& \sum_{i \in N} \sum_{s=\sigma}^{b} x_{i,s} \bar{\mathsf{F}}_i(s-\sigma) \le 1, && \sigma = 0, \ldots, b \\
& \sum_{s=0}^{b} x_{i,s} \le 1, && i \in N \\
& x \ge 0,
\end{aligned}$$

which is equivalent to (10) because the first set of constraints is redundant. The proof now follows by applying the argument from Theorem 3.11 to every $\sigma = 0, \ldots, b$. □

# 5 Correlated Item Values

Our formulation so far only allows an item's value to be random if it is independent of the size, by using its expectation as a deterministic value. A more general setting studied in the literature includes for each item $i \in N$ a random value $C_i$ that may be correlated to its size $A_i$, where we now require knowledge of the joint distribution over $(A_i, C_i)$. (Value-size pairs remain independent across items.) To simplify exposition, we assume throughout this section that each of these distributions has finite support.

Under these more general assumptions, the LP formulation (2) becomes

$$
\begin{aligned}
\min_{v} \quad & v_N(b) \\
\text{s.t.} \quad & v_{M\cup i}(s) - \mathsf{F}_i(s)\mathsf{E}[v_M(s-A_i)|A_i \le s] \ge \mathsf{F}_i(s)\mathsf{E}[C_i|A_i \le s], \\
& \quad \forall\, i \in N, M \subseteq N \setminus i, s \in [0,b] \\
& v \ge 0,
\end{aligned}
$$

and the DP recursion defining the optimal value function $v^*$ is analogous. Similarly, the value function approximations (3) and (9) remain the same, and yield analogous relaxations to (7) and (10) respectively where the objective function coefficient for each variable $x_{i,s}$ is now the item's conditional expected value $\mathsf{F}_i(s)\mathsf{E}[C_i|A_i \le s]$. Assuming item sizes have integer support, there is no substantive change to model (10), and this more general version is already treated in [17, 18, 26].

For the affine approximation, however, the relaxation

$$
\begin{aligned}
\max_{x} \quad & \sum_{i\in N}\sum_{s\in[0,b]} x_{i,s}\mathsf{F}_i(s)\mathsf{E}[C_i|A_i \le s] \\
\text{s.t.} \quad & \sum_{i\in N}\sum_{s\in[0,b]} x_{i,s}\tilde{\mathsf{E}}_i(s) \le b \\
& \sum_{i\in N}\sum_{s\in[0,b]} x_{i,s}\bar{\mathsf{F}}_i(s) \le 1 \\
& \sum_{s\in[0,b]} x_{i,s} \le 1, && \forall\, i \in N \\
& x \ge 0, \quad x \text{ has finite support},
\end{aligned}
$$

has the slightly altered separation problem

$$
\max_{s\in[0,b]} \left\{ \mathsf{F}_i(s)\big(r_0 + \mathsf{E}[C_i|A_i \le s]\big) - q\tilde{\mathsf{E}}_i(s) \right\}
$$

for every item $i \in N$. Separation now also depends on the conditional expected value function $s \mapsto \mathsf{F}_i(s)\mathsf{E}[C_i|A_i \le s]$. If size-value pairs have finite support, this function is piecewise constant, and its breakpoints occur in the same points as the CDF $\mathsf{F}_i$. Therefore, at optimality the relaxation will only have positive $x_{i,s}$ values for those $s$ where $A_i$ has probability mass, just as in the case where value is deterministic.

# 6 Computational Experiments

We next present the setup and results of a series of experiments intended to compare the upper bounds presented in the previous sections and benchmark them against various policies related to the bounds.

## 6.1 Bounds and Policies

We first describe each of the bounds and policies we investigated. We tested the bounds given by (7), which we refer to as MCK (for multiple-choice knapsack), and (10), which we call PP (for pseudo-polynomial). To include a bound independent of our techniques, we also computed a simulation-based *perfect information relaxation* (PIR) [6], obtained by repeatedly simulating a realization of each item's size and solving the resulting deterministic knapsack problem, then computing the sample mean of the optimal value across all realizations; this estimated quantity is an upper bound because it allows the decision maker earlier access to the uncertain data, i.e. it violates non-anticipativity. For this and all other simulations we used 400 realizations. We did not include bounds (8) and (12) in light of Theorems 3.11 and 4.2.

We also considered the following bound from [11]:

$$\begin{aligned}
\max_x \; & \sum_{i \in N} c_i x_{i,b} \mathsf{F}_i(b) \\
\text{s.t.} \; & \sum_{i \in J} x_{i,b} \tilde{\mathsf{E}}_i(b) \leq 2b\Big(1 - \prod_{i \in J}\big(1 - \tilde{\mathsf{E}}_i(b)/b\big)\Big), && J \subseteq N \\
& 0 \leq x_{i,b} \leq 1, && i \in N.
\end{aligned}$$

By employing an appropriate variable substitution, this LP can be recast as a linear polymatroid optimization problem and solved with a greedy algorithm. This bound clearly dominates (8), and [11] also show that it has a worst-case multiplicative gap of 4 with the optimal value $v^*_N(b)$. We haven't yet been able to show an analogue of Theorem 3.11, so we planned to also include this bound in the experiments. However, after preliminary tests, this bound did significantly worse than MCK; it was always at least 14% worse than the best comparable bound (either MCK or PIR), and was often 40%-60% worse. We therefore did not include it in the larger set of experiments.

As for policies, we considered several derived from the various bounds. Arguably the simplest policy for this problem is a *greedy* policy, which attempts to insert items in non-increasing order of their profitability ratio at full capacity, $c_i \mathsf{F}_i(b)/\tilde{\mathsf{E}}_i(b)$, the ratio of expected value to mean truncated size. In addition to its appealing simplicity, this policy is motivated by various theoretical results. First, it generalizes the deterministic knapsack's greedy policy, which is well-known to have a worst-case multiplicative gap of 1/2 under a simple modification [28]. Also, [12] showed that this policy is in fact optimal when item sizes follow exponential distributions. Finally, [11] analyzed a modified version of it with a simple randomization and showed that it achieves a worst-case multiplicative gap of 7/32 (this is the basis for the analysis of (8)). We also implemented an *adaptive greedy* version of the policy that does not fix an ordering of the items, but rather at every encountered state $(M, s)$ computes the profitability ratios at current capacity $c_i \mathsf{F}_i(s)/\tilde{\mathsf{E}}_i(s)$ for remaining items $i \in M$ and chooses a maximizing item.

In addition to yielding bounds by restricting (2), the value function approximations (3) and (9) can of course be used to construct policies, by substituting them into the DP recursion (1). We refer to these two policies as the *MCK and PP dual policies*, to match the bound names. The MCK dual policy uses an optimal solution $(q^*, r^*)$ to (4) to choose an item; at state $(M, s)$, the policy chooses

$$\arg\max_{i \in M} \Big\{ \mathsf{F}_i(s)\Big(c_i + r^*_0 + \sum_{k \in M \setminus i} r^*_k + q^*\big(s - \mathsf{E}[A_i | A_i \leq s]\big)\Big)\Big\}.$$

Similarly, the PP dual policy uses an optimal solution $(r^*, w^*)$ to (11), and at state $(M, s)$ chooses

$$\arg\max_{i \in M} \Big\{ \mathsf{F}_i(s)\Big(c_i + \sum_{k \in M \setminus i} r^*_k\Big) + \sum_{\sigma = 0}^{s} w^*_\sigma \mathsf{F}_i(s - \sigma)\Big\};$$

recall that this bound assumes item sizes have integer support.

Though we investigated both bounds, we did not implement the MCK dual policy, because this policy actually exhibits quite undesirable behavior. Specifically, suppose item sizes are deterministic; then (7) becomes the deterministic knapsack's linear relaxation, and its optimal solution has items set to 1 based on a non-increasing order of the deterministic profitability ratio $c_i/a_i$, with at most one fractional item (the one that fills the knapsack's capacity). In this case, it is not difficult to show that the MCK dual policy is actually indifferent between all items with positive value in the optimal solution of (7). While this lack of distinction between items is not as problematic in the deterministic case (as all items set to 1 would always fit), the policy exhibits analogous behavior for other item size distributions for which (4) has $r_0 = 0$ at optimality, such as uniform distributions, if all sizes are less than $b$ with certainty. This undesirable behavior was also reflected in preliminary results, where the MCK dual policy performed poorly. We therefore did not include it in further experiments.

## 6.2 Data Generation and Parameters

To our knowledge, there is no available test bed of stochastic knapsack instances; however, there are various sources of deterministic instances or instance generators available. Therefore, to obtain instances for our experiments, we used deterministic knapsack instances as a "base" from which we generated stochastic instances. From each deterministic instance we generated eight stochastic ones by varying the item size distribution and keeping all other parameters. If a particular deterministic instance's item $i$ had size $a_i$ (always assumed to be an integer), we generated the following four continuous distributions:

**E** Exponential with rate $1/a_i$.

**U1** Uniform between $[0, 2a_i]$.

**U2** Uniform between $[a_i/2, 3a_i/2]$.

**N** Normal with mean $a_i$ and standard deviation $a_i/3$, conditioned on being non-negative.

Similarly, we generated four discrete distributions:

**D1** 0 or $2a_i$ each with probability 1/2.

**D2** 0 with probability 1/3 or $3a_i/2$ with probability 2/3.

**D3** 0 or $2a_i$ each with probability 1/4, $a_i$ with probability 1/2.

**D4** 0, $a_i$ or $3a_i$ each with probability 1/5, $a_i/2$ with probability 2/5.

Note that all distributions are designed so an item's expected size equals $a_i$. Since the PP bound and dual policy assume integer support, we could only test them on the second set of instances. To ensure integer support for instances of type D2 and D4, after generating the deterministic instance we doubled all item sizes $a_i$ and the knapsack capacity.

The deterministic base instances came from two data sources. We took eight small instances from `http://people.sc.fsu.edu/~jburkardt/datasets/knapsack_01/knapsack_01.html`; these range from five to twenty-five items. We generated 40 larger instances using the "advanced" instance generator from `www.diku.dk/~pisinger/codes.html` (see [27]). The generator is a C++ script that takes in five arguments: number of items, range of coefficients, type, instance number, number of tests in series. The last two input parameters are used to adjust the problem fill rate, that is, the ratio between the sum of all item sizes and capacity; we set these to maintain a fill rate in $[2, 5]$. The "type" parameter refers to the relationship between item sizes and profits. We used two types; in the first, sizes and values are uncorrelated; in the second, sizes and values are "strongly correlated". (The generator's authors observe that deterministic instances tend to be more difficult when sizes and values are correlated.) We generated 10 uncorrelated instances with 100 items, 10 uncorrelated instances with 200 items, 10 strongly correlated instances with 100 items, and 10 strongly correlated instances with 200 items. For these 40 generated instances, we re-scaled the capacity to 1000, and scaled and rounded the item sizes accordingly; we performed this normalization for consistency, since the the dimension of (10) depends on the knapsack capacity and thus influences the computing of the PP bound.

We used CPLEX 12.6.1 for all LP solves, running on a MacBook Pro with OS X 10.7.5 and a 3.06 GHz Intel Core 2 Duo processor. To estimate the PIR bound and all the policies' expected values, we used the sample mean from 400 simulated knapsack instances. For all tests on instances with the conditional normal distribution, we simulated sizes according to a normal distribution with mean $a_i$ and standard deviation of $a_i/3$. Whenever a simulated item size was negative, we changed it to 0. Although this procedure does not exactly model the conditional normal distribution, the changes in the simulated instances are minor given that the probability of being non-negative is approximately 0.999.

We intended to test the PP bound and dual policy on all instances with discrete distributions, but encountered computational difficulty. Even for smaller instances, a naive implementation of (10) would run out of memory. We therefore implemented a column generation algorithm, but even this took a significant amount of time per instance. Roughly speaking, D1 instances were the easiest to solve (usually between 60 and 90 minutes), then D3 (120 to 150 minutes), then D2 (4.5 to 6.5 hours), and D4 instances were the most difficult (12 to 16 hours or even more); the increased computation time required for D2 and D4 instances can partly be explained by the need to double the knapsack capacity and thus the number of variables and constraints. We therefore chose a subset of the instances to test; of the small instances, we tested all except *p08*, since this instance has a very large capacity. From the larger instances, we chose four each of the uncorrelated and strongly correlated instances with 100 items. From all of these base instances, we tested the PP bound and dual policy on all four discrete instance types, D1 through D4. Table 5 in the Appendix includes computation times for the larger instances.

### 6.3 Summary and Results

Tables 1 and 2 contain a summary of our experiments for the different bounds and policies. Table 1 excludes the PP bound and dual policy, but covers all tested instances, while Table 2 includes the PP bound and dual policy but covers only the instances in which these were investigated. The tables are interpreted as follows. For each instance, we choose the smallest bound as baseline, and divide all bounds and policy expected values by this baseline. The first set of columns presents the geometric mean of this ratio, calculated over all instances represented in that row. We show the ratios as percentages for ease of reading; thus, policy ratios should be less than or equal to 100%, while bound ratios should be greater than or equal to 100%. The one exception is the instances with exponentially distributed sizes (type E); because we know from [12] that the greedy policy is optimal, we use this value as a baseline. Also, for these instances the profitability ratio is invariant with respect to remaining capacity, and thus the greedy and adaptive greedy policies are equivalent; hence we do not report adaptive greedy performance for these instances.

For the second set of columns, we count the number of successes – one among the bounds and one among the policies – and divide by the total number of instances represented in that row. A success for a particular instance indicates the bound with the smallest ratio and the policy with the largest ratio. If two ratios are within 0.1% of each other, we consider them equivalent; thus, the presented success rates for each row do not necessarily sum to 100%.

From the results we see that MCK is exclusively better than PIR in the summary statistics;

Table 1: Summary of all tested instances, excluding PP bound and dual policy.

| Distribution | Base | PIR | MCK | Greedy | Adapt. | PIR Success | MCK Success | Greedy Success | Adapt. Success |
|---|---|---|---|---|---|---|---|---|---|
| E | small | 147.59% | 104.13% | 100.00% | - | 0.00% | 100.00% | 100.00% | - |
| | 100cor | 183.62% | 100.07% | 100.00% | - | 0.00% | 100.00% | 100.00% | - |
| | 100uncor | 121.62% | 100.55% | 100.00% | - | 0.00% | 100.00% | 100.00% | - |
| | 200cor | 188.47% | 100.00% | 100.00% | - | 0.00% | 100.00% | 100.00% | - |
| | 200uncor | 121.90% | 100.28% | 100.00% | - | 0.00% | 100.00% | 100.00% | - |
| U1 | small | 126.74% | 100.37% | 89.69% | 89.31% | 12.50% | 87.50% | 100.00% | 87.50% |
| | 100cor | 154.20% | 100.00% | 98.78% | 98.78% | 0.00% | 100.00% | 100.00% | 100.00% |
| | 100uncor | 111.28% | 100.00% | 99.07% | 99.07% | 0.00% | 100.00% | 100.00% | 100.00% |
| | 200cor | 158.23% | 100.00% | 99.55% | 99.56% | 0.00% | 100.00% | 90.00% | 100.00% |
| | 200uncor | 111.96% | 100.00% | 99.70% | 99.70% | 0.00% | 100.00% | 100.00% | 100.00% |
| U2 | small | 112.55% | 100.44% | 86.95% | 89.31% | 12.50% | 87.50% | 12.50% | 87.50% |
| | 100cor | 123.53% | 100.00% | 98.53% | 98.65% | 0.00% | 100.00% | 80.00% | 100.00% |
| | 100uncor | 103.14% | 100.00% | 98.93% | 99.36% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 200cor | 126.15% | 100.00% | 99.20% | 99.30% | 0.00% | 100.00% | 60.00% | 100.00% |
| | 200uncor | 103.23% | 100.00% | 99.44% | 99.75% | 0.00% | 100.00% | 0.00% | 100.00% |
| N | small | 116.11% | 100.32% | 87.56% | 89.48% | 12.50% | 87.50% | 12.50% | 87.50% |
| | 100cor | 126.58% | 100.00% | 98.81% | 98.96% | 0.00% | 100.00% | 50.00% | 90.00% |
| | 100uncor | 104.31% | 100.00% | 99.14% | 99.48% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 200cor | 128.85% | 100.00% | 99.42% | 99.53% | 0.00% | 100.00% | 50.00% | 100.00% |
| | 200uncor | 104.41% | 100.00% | 99.64% | 99.90% | 0.00% | 100.00% | 0.00% | 100.00% |
| D1 | small | 111.00% | 101.67% | 75.04% | 78.46% | 50.00% | 50.00% | 12.50% | 87.50% |
| | 100cor | 174.50% | 100.00% | 95.31% | 97.15% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 100uncor | 121.89% | 100.00% | 96.91% | 97.85% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 200cor | 180.87% | 100.00% | 97.70% | 98.79% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 200uncor | 124.18% | 100.00% | 98.55% | 99.04% | 0.00% | 100.00% | 0.00% | 100.00% |
| D2 | small | 111.59% | 100.68% | 83.79% | 86.73% | 12.50% | 87.50% | 0.00% | 100.00% |
| | 100cor | 152.61% | 100.00% | 96.81% | 97.88% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 100uncor | 115.43% | 100.00% | 98.03% | 98.92% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 200cor | 155.03% | 100.00% | 98.30% | 98.97% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 200uncor | 116.48% | 100.00% | 98.82% | 99.43% | 0.00% | 100.00% | 0.00% | 100.00% |
| D3 | small | 120.37% | 100.74% | 83.55% | 87.22% | 12.50% | 87.50% | 0.00% | 100.00% |
| | 100cor | 156.19% | 100.00% | 97.47% | 98.75% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 100uncor | 114.73% | 100.00% | 98.24% | 98.86% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 200cor | 160.73% | 100.00% | 98.85% | 99.53% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 200uncor | 115.86% | 100.00% | 99.19% | 99.54% | 0.00% | 100.00% | 0.00% | 100.00% |
| D4 | small | 122.67% | 100.00% | 80.52% | 82.91% | 0.00% | 100.00% | 12.50% | 87.50% |
| | 100cor | 185.47% | 100.00% | 95.98% | 97.26% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 100uncor | 121.38% | 100.00% | 97.29% | 97.77% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 200cor | 195.01% | 100.00% | 97.84% | 98.58% | 0.00% | 100.00% | 0.00% | 100.00% |
| | 200uncor | 123.26% | 100.00% | 98.42% | 98.76% | 0.00% | 100.00% | 0.00% | 100.00% |

Table 2: Summary of instances selected for PP bound.

| Distribution | Base | MCK | Greedy | Adapt. | PP Dual | Greedy Success | Adapt. Success | PP Dual Success |
|---|---|---|---|---|---|---|---|---|
| D1 | small | 109.57% | 79.11% | 82.82% | 85.33% | 0.00% | 28.57% | 71.43% |
| | 100cor | 100.00% | 95.78% | 97.59% | 97.22% | 0.00% | 75.00% | 25.00% |
| | 100uncor | 100.00% | 97.22% | 97.95% | 95.24% | 0.00% | 100.00% | 0.00% |
| D2 | small | 104.82% | 86.40% | 89.24% | 87.98% | 0.00% | 42.86% | 57.14% |
| | 100cor | 100.08% | 97.37% | 98.39% | 97.84% | 0.00% | 75.00% | 75.00% |
| | 100uncor | 100.05% | 98.57% | 99.29% | 97.72% | 0.00% | 100.00% | 0.00% |
| D3 | small | 104.65% | 85.55% | 89.42% | 92.15% | 0.00% | 28.57% | 85.71% |
| | 100cor | 100.01% | 97.64% | 98.70% | 99.00% | 0.00% | 25.00% | 75.00% |
| | 100uncor | 100.01% | 98.26% | 98.83% | 97.63% | 0.00% | 100.00% | 0.00% |
| D4 | small | 110.77% | 88.09% | 90.60% | 91.14% | 0.00% | 28.57% | 71.43% |
| | 100cor | 101.54% | 96.98% | 98.03% | 97.90% | 0.00% | 75.00% | 50.00% |
| | 100uncor | 100.76% | 98.02% | 98.43% | 96.90% | 0.00% | 100.00% | 0.00% |

the success rates demonstrate that there are few cases in which PIR is better (mostly in the small instances) but even here PIR is much worse than MCK on average. While PIR is sometimes a good bound, e.g. for uncorrelated instances of type U2, it can often be much worse than MCK, as much as 80% or 90% worse for correlated instances of type D4, for example. We conjecture that MCK's better performance is due in part to an averaging effect: Assuming a large enough fill rate (recall the large instances maintain a fill rate between 2 and 5), individual items influence the solution less as the number of items increases. Whereas MCK uses expected values, PIR is allowed to observe realizations and thus choose each realization's more valuable items. When the number of items is large, this additional information may give the decision maker too much power and thus weaken the bound.

For the bounds reported in Table 2, we focus on comparing MCK to PP. We explain at the start of Section 4 that PP is always less than or equal to MCK; therefore, we report here only MCK as a percentage of PP. In contrast to the wide gaps we sometimes see between PIR and MCK, MCK is very close to PP even though the latter bound employs a much larger number of variables and constraints and is computationally much more demanding. Interestingly, PP seems to offer the most benefit in smaller instances, where MCK can be as much as 10% weaker on average. Conversely, the bounds are quite close in the larger instances; MCK was within 1% of PP for all but one, where the gap was 1.54%. This seems to match the original intent of PP, which was to consider instances in which $b$ is small and can be taken explicitly as part of the input [26].

As for policies, the adaptive greedy policy is in general better than the greedy policy. Setting aside instances of type E, where greedy is optimal and the two are equivalent, adaptive greedy is roughly equivalent to greedy for instances of type U1 and U2, and noticeably better than greedy for type N and for all instances with discrete distributions. This result is in line with what we expect, as adaptive greedy should be more robust to the variation in realized item sizes. However, we also note that the gap between greedy and adaptive greedy seems to decrease as the number of items increases; the experiments thus suggest that the greedy policy is sufficient when the number of items is large enough. The PP dual policy has mixed results compared to the greedy policies. It performs better than adaptive greedy on small instances, but is worse on the larger instances, similarly to what we see with the MCK and PP bounds.

In general, our results indicate that small instances might be harder, in the sense that the simple MCK bound and greedy policies perform better as the number of items grows, while the more complex PP bound and dual policy appear to offer the most benefit when the number of items is small. Of course, if an instance is small enough, it may be possible to directly solve the recursion (1), at least when sizes have integer support. It is thus in the "medium" instance size range that PP may be most useful.

# 7 Conclusions

We have studied a dynamic version of the knapsack problem with stochastic item sizes originally formulated in [11, 12], and proposed a semi-infinite, multiple-choice linear knapsack relaxation. We have shown how both this and a stronger pseudo-polynomial relaxation from [26] arise from different value function approximations being imposed on the doubly-infinite LP formulation of the problem's DP recursion. Our theoretical analysis shows that these bounds are stronger than comparable bounds from the literature, while our computational study indicates that the multiple-choice knapsack relaxation is quite strong in practice and in fact becomes tighter as the number of items increases.

Our results motivate additional questions. In particular, the fact that the simplest bound and policy that we tested grow better as the number of items increases suggests it may be possible to perform an asymptotic analysis of the two and perhaps show that they are optimal as the item number tends to infinity, under appropriate assumptions. On the other hand, our results for the smaller instances also show that even the tightest bound and best-performing policy can leave significant gaps to close. This motivates the investigation of strengthened relaxations, perhaps analogously to a classical cutting plane approach for deterministic knapsack problems. However, deriving such inequalities is not obvious in our context. Finally, our techniques point to a general procedure to obtain relaxations for dynamic integer programs with stochastic variable coefficients, such as the multi-row knapsack models studied in [41].

## Acknowledgements

A. Toriello thanks Will Haskell and Sheldon Ross for many helpful discussions when this research was in a preliminary stage.

## References

[1] D. Adelman, *Price-Directed Replenishment of Subsets: Methodology and its Application to Inventory Routing*, Manufacturing and Service Operations Management **5** (2003), 348–371.

[2] ———, *A Price-Directed Approach to Stochastic Inventory/Routing*, Operations Research **52** (2004), 499–514.

[3] E.J. Anderson and P. Nash, *Linear Programming in Infinite-Dimensional Spaces*, John Wiley & Sons, Inc., Chichester, England, 1987.

[4] A. Bhalgat, A. Goel, and S. Khanna, *Improved Approximation Results for Stochastic Knapsack Problems*, Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2011, pp. 1647–1665.

[5] D. Bienstock, *Approximate Formulations for* 0-1 *Knapsack Sets*, Operations Research Letters **36** (2008), 317–320.

[6] D.B. Brown, J.E. Smith, and P. Sun, *Information Relaxations and Duality in Stochastic Dynamic Programs*, Operations Research **58** (2010), 785–801.

[7] R.L. Carraway, R.L. Schmidt, and L.R. Weatherford, *An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns*, Naval Research Logistics **40** (1993), 161–173.

[8] D.P. de Farias and B. van Roy, *The Linear Programming Approach to Approximate Dynamic Programming*, Operations Research **51** (2003), 850–865.

[9] B.C. Dean, M.X. Goemans, and J. Vondrák, *Approximating the Stochastic Knapsack Problem: The Benefit of Adaptivity*, Proceedings of the 45th Annual IEEE Symposium on the Foundations of Computer Science, IEEE, 2004, pp. 208–217.

[10] ———, *Adaptivity and Approximation for Stochastic Packing Problems*, Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2005, pp. 395–404.

[11] ———, *Approximating the Stochastic Knapsack Problem: The Benefit of Adaptivity*, Mathematics of Operations Research **33** (2008), 945–964.

[12] C. Derman, G.J. Lieberman, and S.M. Ross, *A Renewal Decision Problem*, Management Science **24** (1978), 554–561.

[13] P.C. Gilmore and R.E. Gomory, *The Theory and Computation of Knapsack Functions*, Operations Research **14** (1966), 1045–1074.

[14] M.A. Goberna and M.A. López, *Linear Semi-Infinite Optimization*, Wiley Series in Mathematical Methods in Practice, John Wiley & Sons, Chichester, England, 1998.

[15] A. Goel and P. Indyk, *Stochastic load balancing and related problems*, Proceedings of the 40th Annual IEEE Symposium on the Foundations of Computer Science, IEEE, 1999, pp. 579–586.

[16] V. Goyal and R. Ravi, *A PTAS for Chance-Constrained Knapsack Problem with Random Item Sizes*, Operations Research Letters **38** (2010), 161–164.

[17] A. Gupta, R. Krishnaswamy, M. Molinaro, and R. Ravi, *Approximation Algorithms for Correlated Knapsacks and Non-Martingale Bandits*, Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science, IEEE, 2011, pp. 827–836.

[18] ———, *Approximation Algorithms for Correlated Knapsacks and Non-Martingale Bandits*, Preprint available on-line at `arxiv.org/abs/1102.3749`, 2011.

[19] M. Henig, *Risk criteria in a stochastic knapsack problem*, Operations Research **38** (1990), 820–825.

[20] T. Ilhan, S.M.R. Iravani, and M.S. Daskin, *The Adaptive Knapsack Problem with Stochastic Rewards*, Operations Research **59** (2011), 242–248.

[21] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*, Springer-Verlag, Berlin, 2004.

[22] J. Kleinberg, Y. Rabani, and É. Tardos, *Allocating Bandwidth for Bursty Connections*, Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, Association for Computing Machinery, 1997, pp. 664–673.

[23] ———, *Allocating Bandwidth for Bursty Connections*, SIAM Journal on Computing **30** (2000), 191–217.

[24] A. Kleywegt and J.D. Papastavrou, *The Dynamic and Stochastic Knapsack Problem*, Operations Research **46** (1998), 17–35.

[25] ———, *The Dynamic and Stochastic Knapsack Problem with Random Sized Items*, Operations Research **49** (2001), 26–41.

[26] W. Ma, *Improvements and Generalizations of Stochastic Knapsack and Multi-Armed Bandit Algorithms*, Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2014, pp. 1154–1163.

[27] S. Martello, D. Pisinger, and P. Toth, *Dynamic Programming and Strong Bounds for the* 0-1 *Knapsack Problem*, Management Science **45** (1999), 414–424.

[28] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley & Sons, Ltd., Chichester, England, 1990.

[29] Y. Merzifonluoğlu, J. Geunes, and H.E. Romeijn, *The static stochastic knapsack problem with normally distributed item sizes*, Mathematical Programming **134** (2012), 459–489.

[30] H. Morita, H. Ishii, and T. Nishida, *Stochastic linear knapsack programming problem and its applications to a portfolio selection problem*, European Journal of Operational Research **40** (1989), 329–336.

[31] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, 1999.

[32] J.D. Papastavrou, S. Rajagopalan, and A. Kleywegt, *The Dynamic and Stochastic Knapsack Problem with Deadlines*, Management Science **42** (1996), 1706–1718.

[33] S. Pokutta and M. van Vyve, *A Note on the Extension Complexity of the Knapsack Polytope*, Operations Research Letters **41** (2013), 347–350.

[34] P.J. Schweitzer and A. Seidmann, *Generalized Polynomial Approximations in Markovian Decision Processes*, Journal of Mathematical Analysis and Applications **110** (1985), 568–582.

[35] S. Sen, *Algorithms for Stochastic Mixed-Integer Programming Models*, Discrete Optimization (K. Aardal, G.L. Nemhauser, and R. Weismantel, eds.), Handbooks in Operations Research and Management Science, vol. 12, Elsevier, 2005, pp. 515–558.

[36] M. Sniedovich, *Preference order stochastic knapsack problems: methodological issues*, Journal of the Operational Research Society **31** (1980), 1025–1032.

[37] E.M. Stein and R. Shakarchi, *Real Analysis: Measure Theory, Integration, and Hilbert Spaces*, Princeton Lectures in Analysis, vol. III, Princeton University Press, Princeton, New Jersey, 2006.

[38] E. Steinberg and M.S. Parks, *A preference order dynamic program for a knapsack problem with stochastic rewards*, Journal of the Operational Research Society **30** (1979), 141–147.

[39] A. Toriello, W.B. Haskell, and M. Poremba, *A Dynamic Traveling Salesman Problem with Stochastic Arc Costs*, Operations Research **62** (2014), 1107–1125.

[40] M.A. Trick and S.E. Zin, *Spline Approximations to Value Functions: A Linear Programming Approach*, Macroeconomic Dynamics **1** (1997), 255–277.

[41] J. Vondrák, *Probabilistic Methods in Combinatorial and Stochastic Optimization*, Ph.D. thesis, Massachusetts Institute of Technology, 2005.

## Appendix

The following five tables present the raw data we used to calculate the summaries in Tables 1 and 2. Tables separate instances by their size, small instances followed by 100-item instances under continuous distributions, 100-item instances with discrete distributions, then 200-item instances under continuous and discrete distributions. Table 5 includes running times for the PP bound and dual policy; these are in seconds.

Table 3: Small instances.

| Instance | Items | Distribution | PIR | MCK | Greedy | Adapt. | PP | PP Dual Policy |
|---|---|---|---|---|---|---|---|---|
| p01 | 10 | E | 469.43 | 308.38 | 304.32 | - | | |
| | | U1 | 407.46 | 309.02 | 289.47 | 289.47 | | |
| | | U2 | 334.49 | 309.02 | 281.94 | 287.36 | | |
| | | N | 347.11 | 308.58 | 284.20 | 287.63 | | |
| p02 | 5 | E | 73.35 | 52.23 | 49.23 | - | | |
| | | U1 | 66.76 | 52.63 | 44.18 | 44.18 | | |
| | | U2 | 59.72 | 52.63 | 43.62 | 45.48 | | |
| | | N | 61.42 | 52.52 | 44.63 | 45.20 | | |
| p03 | 6 | E | 214.84 | 159.17 | 154.52 | - | | |
| | | U1 | 194.27 | 160.00 | 140.09 | 140.09 | | |
| | | U2 | 175.05 | 160.00 | 128.40 | 133.66 | | |
| | | N | 177.41 | 159.76 | 132.24 | 133.11 | | |
| p04 | 7 | E | 152.80 | 105.64 | 104.62 | - | | |
| | | U1 | 133.61 | 105.70 | 87.11 | 87.11 | | |
| | | U2 | 121.54 | 107.55 | 93.31 | 98.58 | | |
| | | N | 122.19 | 97.76 | 79.26 | 88.35 | | |
| p05 | 8 | E | 1174.89 | 1151.23 | 990.27 | - | | |
| | | U1 | 1155.69 | 1190.00 | 1006.87 | 1006.87 | | |
| | | U2 | 1148.92 | 1190.00 | 955.95 | 960.33 | | |
| | | N | 1157.76 | 1187.67 | 976.79 | 976.39 | | |
| p06 | 7 | E | 2878.60 | 1785.59 | 1687.88 | - | | |
| | | U1 | 2541.71 | 1786.50 | 1633.31 | 1633.31 | | |
| | | U2 | 2162.80 | 1786.50 | 1506.54 | 1524.70 | | |
| | | N | 2211.48 | 1783.92 | 1522.41 | 1534.91 | | |
| p07 | 15 | E | 2260.73 | 1461.48 | 1476.47 | - | | |
| | | U1 | 1980.37 | 1461.50 | 1390.88 | 1390.88 | | |
| | | U2 | 1748.72 | 1461.50 | 1364.71 | 1388.83 | | |
| | | N | 1784.32 | 1459.35 | 1391.59 | 1409.44 | | |
| p08 | 24 | E | 20824162.16 | 13580702.9 | 13373492.25 | - | | |
| | | U1 | 18268194.11 | 13580982.52 | 13220859.93 | 12445090.33 | | |
| | | U2 | 15901300.28 | 13580982.52 | 12868614.76 | 12674256.09 | | |
| | | N | 16185888.15 | 13560987.98 | 12987982.5 | 13169542.03 | | |
| p01 | 10 | D1 | 487.98 | 394.52 | 296.54 | 321.36 | 385.83 | 315.38 |
| | | D2 | 429.27 | 352.02 | 300.94 | 308.37 | 346.27 | 307.38 |
| | | D3 | 424.15 | 337.77 | 296.20 | 307.71 | 327.87 | 304.54 |
| | | D4 | 488.86 | 345.97 | 301.53 | 313.49 | 334.23 | 314.79 |
| p02 | 5 | D1 | 70.11 | 71.00 | 53.99 | 54.81 | 62.50 | 53.92 |
| | | D2 | 64.71 | 61.67 | 45.31 | 45.67 | 55.83 | 46.22 |
| | | D3 | 67.89 | 58.33 | 47.95 | 49.75 | 54.86 | 49.74 |
| | | D4 | 75.99 | 67.91 | 50.75 | 50.47 | 58.21 | 53.45 |
| p03 | 6 | D1 | 206.03 | 209.19 | 148.98 | 148.48 | 169.00 | 153.17 |
| | | D2 | 192.01 | 184.71 | 144.33 | 158.80 | 175.67 | 133.81 |
| | | D3 | 198.74 | 176.61 | 135.08 | 147.73 | 164.14 | 149.32 |
| | | D4 | 221.40 | 199.33 | 155.59 | 156.67 | 168.61 | 157.56 |
| p04 | 7 | D1 | 140.36 | 141.79 | 87.50 | 99.88 | 140.75 | 108.98 |
| | | D2 | 130.39 | 126.75 | 103.99 | 108.92 | 124.00 | 109.42 |
| | | D3 | 138.11 | 116.86 | 94.75 | 99.35 | 114.35 | 105.32 |
| | | D4 | 153.35 | 137.56 | 104.78 | 107.78 | 125.83 | 109.59 |
| p05 | 8 | D1 | 1127.91 | 1239.78 | 918.26 | 921.34 | 1173.00 | 960.58 |
| | | D2 | 1155.83 | 1219.85 | 991.35 | 996.10 | 1111.33 | 1033.15 |
| | | D3 | 1141.79 | 1211.56 | 941.37 | 946.18 | 1133.81 | 979.05 |
| | | D4 | 1176.03 | 1129.89 | 962.53 | 966.43 | 1107.36 | 918.70 |
| p06 | 7 | D1 | 2751.88 | 2380.82 | 1475.97 | 1560.94 | 1922.25 | 1633.93 |
| | | D2 | 2449.14 | 2087.00 | 1772.02 | 1776.13 | 1988.67 | 1852.03 |
| | | D3 | 2544.08 | 1987.17 | 1540.79 | 1618.81 | 1881.90 | 1809.05 |
| | | D4 | 3007.52 | 2306.09 | 1637.78 | 1796.06 | 1935.71 | 1820.33 |
| p07 | 15 | D1 | 2137.13 | 1681.26 | 1490.73 | 1546.55 | 1680.75 | 1607.05 |
| | | D2 | 1933.20 | 1570.45 | 1461.14 | 1529.57 | 1570.45 | 1498.95 |
| | | D3 | 2039.99 | 1533.54 | 1389.58 | 1450.84 | 1516.37 | 1461.44 |
| | | D4 | 2304.04 | 1676.91 | 1439.27 | 1486.18 | 1554.73 | 1479.13 |
| p08 | 24 | D1 | 19515886.11 | 15394878.96 | 13258193.23 | 13743118.4 | | |
| | | D2 | 17610353.34 | 14477273.59 | 12898295.79 | 13541522.45 | | |
| | | D3 | 18882530.81 | 14177463.69 | 12996088.67 | 13455440.77 | | |
| | | D4 | 21458854.89 | 15281861 | 13429274.26 | 13937656.95 | | |

Table 4: 100 items, continuous distributions.

| Instance | Distribution | PIR | MCK | Greedy | Adapt. |
|---|---|---|---|---|---|
| 100cor1 | E | 46802.50 | 30013.29 | 30027.84 | - |
| | U1 | 40845.14 | 30013.29 | 29709.86 | 29709.86 |
| | U2 | 35482.96 | 30013.29 | 29659.10 | 29681.43 |
| | N | 36040.95 | 29969.35 | 29660.14 | 29737.86 |
| 100cor2 | E | 46073.75 | 35516.98 | 35523.63 | - |
| | U1 | 40917.92 | 35516.98 | 35067.72 | 35067.72 |
| | U2 | 37030.99 | 35516.98 | 35079.05 | 35343.20 |
| | N | 37466.24 | 35466.19 | 35003.46 | 35283.59 |
| 100cor3 | E | 82190.42 | 40886.27 | 40457.22 | - |
| | U1 | 68049.76 | 40886.27 | 40482.44 | 40482.44 |
| | U2 | 52659.17 | 40886.27 | 40114.07 | 40172.81 |
| | N | 53911.04 | 40826.58 | 40237.92 | 40279.47 |
| 100cor4 | E | 106258.17 | 44168.09 | 43572.86 | - |
| | U1 | 85060.89 | 44168.09 | 43708.70 | 43708.70 |
| | U2 | 58990.90 | 44168.09 | 43492.66 | 43533.25 |
| | N | 61759.77 | 44103.54 | 43352.43 | 43413.47 |
| 100cor5 | E | 60004.28 | 27106.08 | 27231.81 | - |
| | U1 | 48705.49 | 27106.08 | 26657.77 | 26657.77 |
| | U2 | 35705.48 | 27106.08 | 26655.80 | 26632.95 |
| | N | 36991.38 | 27066.39 | 26783.05 | 26793.98 |
| 100cor6 | E | 8286.74 | 4480.44 | 4511.49 | - |
| | U1 | 6950.51 | 4480.44 | 4437.71 | 4437.42 |
| | U2 | 5584.03 | 4480.44 | 4423.21 | 4424.94 |
| | N | 5688.12 | 4473.89 | 4427.14 | 4431.23 |
| 100cor7 | E | 52356.63 | 32128.94 | 32370.66 | - |
| | U1 | 44970.77 | 32128.94 | 31593.64 | 31593.64 |
| | U2 | 38720.08 | 32128.94 | 31748.74 | 31767.77 |
| | N | 39044.63 | 32081.98 | 31742.67 | 31752.78 |
| 100cor8 | E | 65439.41 | 31608.91 | 31946.24 | - |
| | U1 | 53157.53 | 31608.91 | 31073.76 | 31073.76 |
| | U2 | 40607.08 | 31608.91 | 31068.39 | 31091.21 |
| | N | 41831.26 | 31562.71 | 31173.94 | 31230.02 |
| 100cor9 | E | 31530.91 | 16491.00 | 16307.28 | - |
| | U1 | 26472.68 | 16491.00 | 16361.04 | 16361.04 |
| | U2 | 20837.16 | 16491.00 | 16225.89 | 16232.88 |
| | N | 21330.34 | 16466.92 | 16300.77 | 16294.04 |
| 100cor10 | E | 123118.83 | 73558.26 | 73383.78 | - |
| | U1 | 105636.61 | 73558.26 | 72754.22 | 72754.22 |
| | U2 | 89404.56 | 73558.26 | 72482.59 | 72528.14 |
| | N | 90931.11 | 73450.59 | 72719.15 | 72628.02 |
| 100uncor1 | E | 42769.68 | 38457.56 | 38229.38 | - |
| | U1 | 40682.51 | 38457.56 | 38215.75 | 38215.75 |
| | U2 | 39083.20 | 38457.56 | 38169.62 | 38265.87 |
| | N | 39292.73 | 38403.27 | 38188.09 | 38264.49 |
| 100uncor2 | E | 18129.49 | 15212.48 | 15190.47 | - |
| | U1 | 16632.99 | 15212.48 | 15044.51 | 15044.51 |
| | U2 | 15659.71 | 15212.48 | 15065.95 | 15141.41 |
| | N | 15775.43 | 15190.87 | 15021.03 | 15096.71 |
| 100uncor3 | E | 80130.74 | 65503.10 | 64733.69 | - |
| | U1 | 73939.67 | 65503.10 | 65055.60 | 65055.60 |
| | U2 | 67715.12 | 65503.10 | 64584.03 | 64862.53 |
| | N | 68481.73 | 65410.69 | 64863.15 | 65078.84 |
| 100uncor4 | E | 111922.24 | 85139.17 | 83845.66 | - |
| | U1 | 99104.63 | 85139.17 | 84369.32 | 84369.32 |
| | U2 | 87870.48 | 85139.17 | 83987.41 | 84559.16 |
| | N | 89266.56 | 85018.36 | 83877.86 | 84406.85 |
| 100uncor5 | E | 57965.38 | 44361.56 | 44315.31 | - |
| | U1 | 51796.03 | 44361.56 | 43716.36 | 43716.36 |
| | U2 | 46325.52 | 44361.56 | 43694.02 | 43905.13 |
| | N | 47050.67 | 44298.81 | 43843.21 | 43954.99 |
| 100uncor6 | E | 8308.34 | 6950.75 | 6962.80 | - |
| | U1 | 7687.99 | 6950.75 | 6880.37 | 6880.37 |
| | U2 | 7178.39 | 6950.75 | 6888.15 | 6922.83 |
| | N | 7227.76 | 6940.92 | 6896.62 | 6919.74 |
| 100uncor7 | E | 48193.33 | 42746.90 | 42627.88 | - |
| | U1 | 45494.95 | 42746.90 | 42424.58 | 42424.58 |
| | U2 | 43814.00 | 42746.90 | 42468.59 | 42618.12 |
| | N | 43900.48 | 42686.66 | 42393.66 | 42532.98 |
| 100uncor8 | E | 63238.76 | 49921.48 | 50080.69 | - |
| | U1 | 57161.84 | 49921.48 | 49358.21 | 49358.21 |
| | U2 | 51907.49 | 49921.48 | 49312.42 | 49470.25 |
| | N | 52639.12 | 49851.04 | 49375.43 | 49533.99 |
| 100uncor9 | E | 32346.23 | 25826.46 | 25474.05 | - |
| | U1 | 29550.41 | 25826.46 | 25574.67 | 25574.67 |
| | U2 | 26956.78 | 25826.46 | 25524.10 | 25636.02 |
| | N | 27341.30 | 25789.76 | 25554.41 | 25629.55 |
| 100uncor10 | E | 112928.12 | 100349.64 | 99560.75 | - |
| | U1 | 106740.87 | 100349.64 | 99709.53 | 99709.53 |
| | U2 | 102073.99 | 100349.64 | 99555.19 | 99906.87 |
| | N | 102725.17 | 100208.51 | 99653.15 | 99945.11 |

Table 5: 100 items, discrete distributions.

| Instance | Distribution | PIR | MCK | Greedy | Adapt. | PP | PP Time | PP Dual Policy | PP Policy Time |
|---|---|---|---|---|---|---|---|---|---|
| 100cor1 | D1 | 44078.58 | 31111.79 | 29798.00 | 30087.17 | | | | |
| | D2 | 39734.36 | 30562.79 | 29844.35 | 30126.53 | | | | |
| | D3 | 42432.33 | 30379.70 | 29681.42 | 29938.09 | | | | |
| | D4 | 48581.92 | 30995.20 | 29860.49 | 30225.65 | | | | |
| 100cor2 | D1 | 46747.43 | 36615.48 | 35463.69 | 35761.03 | 36615.48 | 4151.59 | 35190.17 | 3549.92 |
| | D2 | 42845.17 | 36066.48 | 35075.34 | 35383.89 | 36044.08 | 16575.69 | 34503.27 | 7204.37 |
| | D3 | 42835.04 | 35883.40 | 35327.66 | 35553.73 | 35880.92 | 7822.71 | 35464.01 | 4142.33 |
| | D4 | 47111.17 | 36462.66 | 34935.83 | 35019.59 | 36036.74 | 45026.36 | 34800.01 | 8278.49 |
| 100cor3 | D1 | 82283.49 | 43071.27 | 40846.87 | 41608.34 | | | | |
| | D2 | 68298.72 | 41981.27 | 40383.90 | 40443.97 | | | | |
| | D3 | 69105.73 | 41617.11 | 40587.21 | 41198.20 | | | | |
| | D4 | 86687.28 | 42707.30 | 41218.08 | 42104.89 | | | | |
| 100cor4 | D1 | 117391.61 | 47466.59 | 44318.90 | 45797.99 | | | | |
| | D2 | 93455.42 | 45817.59 | 43831.99 | 44884.43 | | | | |
| | D3 | 89015.11 | 45267.84 | 43611.86 | 44516.14 | | | | |
| | D4 | 111984.10 | 46307.87 | 43515.92 | 44274.00 | | | | |
| 100cor5 | D1 | 62714.78 | 28754.58 | 27524.52 | 28089.91 | | | | |
| | D2 | 50514.96 | 27930.58 | 26841.43 | 27175.15 | | | | |
| | D3 | 49465.22 | 27655.83 | 26871.22 | 27313.78 | | | | |
| | D4 | 63768.18 | 28303.23 | 27625.25 | 28137.76 | | | | |
| 100cor6 | D1 | 8111.63 | 4699.44 | 4558.40 | 4674.98 | 4699.44 | 5958.01 | 4601.82 | 4599.36 |
| | D2 | 6913.01 | 4590.11 | 4472.36 | 4517.99 | 4586.17 | 17836.77 | 4519.63 | 8730.71 |
| | D3 | 7107.33 | 4553.61 | 4458.65 | 4522.50 | 4553.00 | 8430.32 | 4509.02 | 4298.07 |
| | D4 | 8712.30 | 4648.74 | 4406.69 | 4477.31 | 4576.45 | 55665.90 | 4534.99 | 9986.10 |
| 100cor7 | D1 | 49546.25 | 33436.44 | 32141.17 | 32363.01 | | | | |
| | D2 | 43752.40 | 32784.77 | 31609.47 | 31799.30 | | | | |
| | D3 | 46749.00 | 32566.85 | 32075.44 | 32424.55 | | | | |
| | D4 | 54808.02 | 33217.02 | 32228.47 | 32478.68 | | | | |
| 100cor8 | D1 | 66226.71 | 33454.41 | 31511.43 | 32404.03 | | | | |
| | D2 | 54942.13 | 32531.91 | 31544.68 | 32034.34 | | | | |
| | D3 | 54220.66 | 32224.33 | 31233.43 | 31740.84 | | | | |
| | D4 | 68773.82 | 32946.54 | 31735.01 | 32129.03 | | | | |
| 100cor9 | D1 | 31232.40 | 17358.50 | 16382.66 | 16817.33 | 17358.50 | 5850.62 | 16788.91 | 4575.07 |
| | D2 | 26594.32 | 16926.83 | 16348.04 | 16568.27 | 16912.50 | 19652.41 | 16597.11 | 10150.56 |
| | D3 | 26885.93 | 16782.25 | 16148.36 | 16358.36 | 16780.97 | 9340.53 | 16552.22 | 4361.77 |
| | D4 | 33311.77 | 17173.90 | 16304.31 | 16506.94 | 16870.03 | 54863.39 | 16309.74 | 10862.59 |
| 100cor10 | D1 | 117896.42 | 76476.76 | 72575.76 | 73691.05 | 76476.76 | 7638.23 | 75054.18 | 4311.85 |
| | D2 | 103974.56 | 75021.92 | 73437.07 | 74136.90 | 74956.18 | 21293.51 | 74189.98 | 6974.64 |
| | D3 | 109504.05 | 74535.51 | 73005.68 | 73711.17 | 74529.54 | 8805.79 | 74141.82 | 5041.93 |
| | D4 | 130641.57 | 76078.30 | 73419.19 | 74356.02 | 74893.66 | 60621.26 | 74349.03 | 12521.45 |
| 100uncor1 | D1 | 43115.77 | 39050.18 | 38114.19 | 38283.15 | 39050.18 | 4907.27 | 37556.94 | 4432.39 |
| | D2 | 41689.79 | 38758.85 | 38322.44 | 38544.31 | 38740.96 | 23577.82 | 38063.65 | 8963.41 |
| | D3 | 41423.69 | 38661.78 | 38133.94 | 38303.53 | 38660.16 | 8333.15 | 37402.40 | 4266.99 |
| | D4 | 43230.58 | 39021.74 | 38229.45 | 38327.39 | 38758.69 | 42589.05 | 37639.14 | 11467.26 |
| 100uncor2 | D1 | 18305.73 | 15608.14 | 15128.88 | 15271.97 | | | | |
| | D2 | 17234.28 | 15412.16 | 15037.50 | 15179.70 | | | | |
| | D3 | 17291.71 | 15346.20 | 15077.58 | 15154.24 | | | | |
| | D4 | 18427.78 | 15539.47 | 14963.46 | 15052.44 | | | | |
| 100uncor3 | D1 | 83194.39 | 67360.00 | 64770.18 | 65599.72 | | | | |
| | D2 | 77194.25 | 66434.19 | 64582.40 | 65345.69 | | | | |
| | D3 | 76370.51 | 66125.42 | 65102.31 | 65632.90 | | | | |
| | D4 | 82585.46 | 67221.26 | 65343.83 | 65601.12 | | | | |
| 100uncor4 | D1 | 123509.36 | 88088.17 | 84788.51 | 85715.22 | 88088.17 | 4986.33 | 83504.80 | 2777.83 |
| | D2 | 111075.84 | 86618.00 | 84761.74 | 85598.76 | 86571.83 | 18304.86 | 84222.73 | 6596.08 |
| | D3 | 107340.42 | 86126.50 | 84122.96 | 84848.73 | 86119.50 | 7340.91 | 84475.59 | 3149.88 |
| | D4 | 116383.28 | 87154.15 | 84122.40 | 84715.31 | 86505.92 | 53716.99 | 83185.47 | 7839.50 |
| 100uncor5 | D1 | 61386.43 | 45854.42 | 44556.04 | 45051.44 | | | | |
| | D2 | 55698.81 | 45108.73 | 43922.13 | 44418.12 | | | | |
| | D3 | 54356.52 | 44859.89 | 43913.47 | 44198.48 | | | | |
| | D4 | 60245.42 | 45503.07 | 44623.99 | 44901.59 | | | | |
| 100uncor6 | D1 | 8480.72 | 7102.63 | 6970.49 | 7033.13 | 7102.63 | 4392.03 | 6619.18 | 3341.61 |
| | D2 | 8013.10 | 7025.64 | 6939.90 | 6989.11 | 7022.13 | 15157.58 | 6833.16 | 6890.15 |
| | D3 | 7950.84 | 7000.58 | 6869.04 | 6912.14 | 7000.25 | 7619.60 | 6825.17 | 3696.61 |
| | D4 | 8471.81 | 7093.54 | 6862.53 | 6889.23 | 7025.20 | 46301.52 | 6805.25 | 9100.62 |
| 100uncor7 | D1 | 48576.04 | 43527.70 | 42340.84 | 42708.69 | | | | |
| | D2 | 46607.41 | 43144.45 | 42394.98 | 42700.89 | | | | |
| | D3 | 46716.27 | 43014.32 | 42643.76 | 42797.71 | | | | |
| | D4 | 48788.27 | 43459.26 | 42720.55 | 42862.96 | | | | |
| 100uncor8 | D1 | 65960.71 | 51435.98 | 49780.27 | 50388.94 | | | | |
| | D2 | 60674.89 | 50681.48 | 49548.72 | 50231.46 | | | | |
| | D3 | 59922.36 | 50429.06 | 49508.31 | 49961.63 | | | | |
| | D4 | 65312.69 | 51143.14 | 49988.34 | 50285.08 | | | | |
| 100uncor9 | D1 | 33455.01 | 26586.96 | 25502.40 | 25833.05 | | | | |
| | D2 | 31068.63 | 26209.79 | 25697.86 | 25913.22 | | | | |
| | D3 | 30813.52 | 26083.04 | 25441.93 | 25611.47 | | | | |
| | D4 | 33430.11 | 26438.54 | 25421.77 | 25612.58 | | | | |
| 100uncor10 | D1 | 114587.82 | 102187.14 | 99022.08 | 99565.20 | 102187.14 | 4442.64 | 98938.86 | 3804.11 |
| | D2 | 109941.05 | 101282.97 | 99847.93 | 100505.43 | 101238.42 | 16119.72 | 99264.64 | 8712.19 |
| | D3 | 109400.87 | 100976.72 | 99540.08 | 99943.73 | 100970.68 | 7832.96 | 99153.42 | 4192.16 |
| | D4 | 114394.12 | 101825.76 | 99687.04 | 100010.95 | 101181.44 | 52132.86 | 98602.11 | 9252.24 |

Table 6: 200 items, continuous distributions.

| Instance | Distribution | PIR | MCK | Greedy | Adapt. |
|---|---|---|---|---|---|
| 200cor1 | E | 96338.10 | 60298.25 | 60166.15 | - |
| | U1 | 83447.70 | 60298.25 | 59870.89 | 59883.76 |
| | U2 | 72345.57 | 60298.25 | 59880.24 | 59982.77 |
| | N | 73305.69 | 60209.82 | 59989.26 | 60092.64 |
| 200cor2 | E | 42804.74 | 24268.53 | 24315.10 | - |
| | U1 | 36386.10 | 24268.53 | 24231.19 | 24227.91 |
| | U2 | 29961.30 | 24268.53 | 24105.67 | 24113.98 |
| | N | 30382.63 | 24233.01 | 24061.61 | 24086.15 |
| 200cor3 | E | 160487.58 | 79385.20 | 79848.79 | - |
| | U1 | 133231.17 | 79385.20 | 79010.52 | 79010.52 |
| | U2 | 102274.99 | 79385.20 | 78698.19 | 78727.61 |
| | N | 104766.79 | 79269.21 | 78849.13 | 78924.35 |
| 200cor4 | E | 242915.02 | 119112.85 | 119238.33 | - |
| | U1 | 199950.22 | 119112.85 | 118310.85 | 118310.85 |
| | U2 | 153931.66 | 119112.85 | 118062.12 | 118110.00 |
| | N | 157688.69 | 118938.55 | 118463.94 | 118742.97 |
| 200cor5 | E | 111488.65 | 50763.78 | 51076.94 | - |
| | U1 | 91264.23 | 50763.78 | 50779.10 | 50779.10 |
| | U2 | 66731.70 | 50763.78 | 50318.23 | 50351.14 |
| | N | 68941.15 | 50689.58 | 50432.74 | 50456.02 |
| 200cor6 | E | 17065.10 | 8985.94 | 9015.47 | - |
| | U1 | 14304.83 | 8985.94 | 8969.54 | 8970.44 |
| | U2 | 11411.90 | 8985.94 | 8921.87 | 8928.95 |
| | N | 11600.03 | 8972.78 | 8917.48 | 8922.45 |
| 200cor7 | E | 108614.77 | 65325.84 | 65922.25 | - |
| | U1 | 93490.58 | 65325.84 | 65118.22 | 65118.22 |
| | U2 | 79269.30 | 65325.84 | 64920.35 | 65033.72 |
| | N | 80272.94 | 65230.19 | 64790.51 | 64889.80 |
| 200cor8 | E | 130055.50 | 62201.63 | 61390.91 | - |
| | U1 | 107213.96 | 62201.63 | 62183.34 | 62183.34 |
| | U2 | 80820.28 | 62201.63 | 61582.15 | 61695.99 |
| | N | 83030.03 | 62110.58 | 61730.68 | 61774.10 |
| 200cor9 | E | 66615.12 | 33899.19 | 33749.25 | - |
| | U1 | 55341.71 | 33899.19 | 33601.75 | 33602.05 |
| | U2 | 43378.55 | 33899.19 | 33643.96 | 33662.58 |
| | N | 44269.81 | 33849.60 | 33606.65 | 33643.12 |
| 200cor10 | E | 242014.96 | 140855.75 | 140610.79 | - |
| | U1 | 205686.65 | 140855.75 | 139359.83 | 139359.83 |
| | U2 | 173062.12 | 140855.75 | 139596.58 | 139810.62 |
| | N | 175291.42 | 140649.33 | 139763.01 | 139782.13 |
| 200uncor1 | E | 83826.87 | 74467.00 | 74034.84 | - |
| | U1 | 79249.11 | 74467.00 | 74121.03 | 74121.03 |
| | U2 | 75919.43 | 74467.00 | 74199.79 | 74365.84 |
| | N | 76324.76 | 74361.68 | 74229.32 | 74400.45 |
| 200uncor2 | E | 38095.74 | 32017.00 | 31838.94 | - |
| | U1 | 35427.90 | 32017.00 | 32013.82 | 32013.20 |
| | U2 | 32959.84 | 32017.00 | 31838.69 | 31929.72 |
| | N | 33285.84 | 31971.64 | 31895.73 | 31954.88 |
| 200uncor3 | E | 151605.79 | 121926.52 | 121991.71 | - |
| | U1 | 139020.13 | 121926.52 | 121820.38 | 121820.38 |
| | U2 | 126906.94 | 121926.52 | 121218.31 | 121640.45 |
| | N | 128281.47 | 121754.37 | 121467.90 | 121860.33 |
| 200uncor4 | E | 216014.95 | 163613.50 | 163269.93 | - |
| | U1 | 192158.61 | 163613.50 | 163316.27 | 163316.27 |
| | U2 | 168956.10 | 163613.50 | 162434.29 | 162972.92 |
| | N | 171875.64 | 163381.50 | 162524.27 | 163109.14 |
| 200uncor5 | E | 116500.76 | 89506.64 | 89658.43 | - |
| | U1 | 104709.27 | 89506.64 | 89233.56 | 89233.56 |
| | U2 | 93160.65 | 89506.64 | 88888.94 | 89256.67 |
| | N | 94495.16 | 89379.97 | 89040.21 | 89285.07 |
| 200uncor6 | E | 16061.66 | 13183.89 | 13124.95 | - |
| | U1 | 14799.72 | 13183.89 | 13154.57 | 13154.57 |
| | U2 | 13638.73 | 13183.89 | 13118.35 | 13171.13 |
| | N | 13775.38 | 13165.22 | 13103.32 | 13136.81 |
| 200uncor7 | E | 98162.02 | 85512.00 | 85730.66 | - |
| | U1 | 92160.36 | 85512.00 | 85344.42 | 85342.96 |
| | U2 | 87635.96 | 85512.00 | 85047.56 | 85279.20 |
| | N | 88124.33 | 85391.03 | 85111.43 | 85337.96 |
| 200uncor8 | E | 126731.98 | 99398.79 | 98822.78 | - |
| | U1 | 114892.19 | 99398.79 | 99214.48 | 99214.48 |
| | U2 | 103848.29 | 99398.79 | 98714.26 | 99114.09 |
| | N | 105151.92 | 99257.96 | 98803.43 | 99135.01 |
| 200uncor9 | E | 63113.62 | 52263.71 | 52039.60 | - |
| | U1 | 58289.71 | 52263.71 | 51949.95 | 51949.95 |
| | U2 | 54032.26 | 52263.71 | 51990.51 | 52107.29 |
| | N | 54479.66 | 52189.96 | 51920.80 | 52035.21 |
| 200uncor10 | E | 228662.27 | 200005.23 | 199130.12 | - |
| | U1 | 215045.49 | 200005.23 | 198547.30 | 198547.30 |
| | U2 | 204372.46 | 200005.23 | 199040.80 | 199532.21 |
| | N | 205686.78 | 199722.65 | 199202.04 | 199547.32 |

Table 7: 200 items, discrete distributions.

| Instance | Distribution | PIR | MCK | Greedy | Adapt. |
|---|---|---|---|---|---|
| 200cor1 | D1 | 90688.56 | 61396.75 | 60471.88 | 61118.87 |
| | D2 | 80392.29 | 60847.75 | 59973.48 | 60394.04 |
| | D3 | 87585.08 | 60664.67 | 60080.24 | 60431.39 |
| | D4 | 100236.02 | 61343.83 | 60229.68 | 60489.84 |
| 200cor2 | D1 | 41009.55 | 24817.03 | 24410.57 | 24597.46 |
| | D2 | 35477.81 | 24543.03 | 24236.69 | 24360.78 |
| | D3 | 37442.18 | 24451.62 | 24311.92 | 24428.50 |
| | D4 | 45364.72 | 24744.38 | 24288.52 | 24522.92 |
| 200cor3 | D1 | 162086.31 | 81570.20 | 80423.03 | 81219.60 |
| | D2 | 133748.93 | 80480.20 | 79151.98 | 79762.27 |
| | D3 | 134781.28 | 80116.03 | 79464.88 | 80077.16 |
| | D4 | 169279.67 | 81258.86 | 79164.22 | 79743.83 |
| 200cor4 | D1 | 242362.08 | 122411.35 | 118933.94 | 120813.60 |
| | D2 | 202178.61 | 120762.35 | 119490.55 | 120327.60 |
| | D3 | 202501.13 | 120212.60 | 118489.64 | 119214.81 |
| | D4 | 257291.96 | 121803.45 | 120456.40 | 121202.45 |
| 200cor5 | D1 | 116224.49 | 52412.28 | 50387.32 | 50994.89 |
| | D2 | 93695.69 | 51588.28 | 50121.57 | 50583.33 |
| | D3 | 92933.34 | 51313.53 | 50891.46 | 51311.03 |
| | D4 | 118255.65 | 52009.95 | 50748.44 | 51528.33 |
| 200cor6 | D1 | 16667.82 | 9205.94 | 9003.34 | 9106.95 |
| | D2 | 14088.38 | 9095.94 | 8907.81 | 8968.55 |
| | D3 | 14567.33 | 9059.27 | 8941.13 | 9011.57 |
| | D4 | 18095.95 | 9178.19 | 8952.46 | 8994.52 |
| 200cor7 | D1 | 102607.10 | 66642.84 | 64916.83 | 65453.53 |
| | D2 | 90080.82 | 65984.84 | 64946.43 | 65266.42 |
| | D3 | 96574.97 | 65765.34 | 64822.58 | 65276.06 |
| | D4 | 114182.03 | 66524.37 | 65017.83 | 65403.32 |
| 200cor8 | D1 | 132461.57 | 64047.13 | 62223.70 | 63217.74 |
| | D2 | 108756.60 | 63124.63 | 61399.19 | 61918.21 |
| | D3 | 108834.42 | 62817.05 | 61893.55 | 62310.26 |
| | D4 | 137941.23 | 63603.00 | 61671.69 | 62269.95 |
| 200cor9 | D1 | 66481.90 | 34778.19 | 33987.45 | 34397.40 |
| | D2 | 55414.74 | 34338.86 | 33860.73 | 34054.50 |
| | D3 | 56553.93 | 34192.36 | 33772.89 | 34057.41 |
| | D4 | 70323.25 | 34657.10 | 33988.19 | 34193.42 |
| 200cor10 | D1 | 229580.17 | 143794.25 | 141221.53 | 142346.43 |
| | D2 | 201176.03 | 142326.08 | 140948.56 | 141848.39 |
| | D3 | 212590.51 | 141836.33 | 139834.33 | 140638.72 |
| | D4 | 255690.49 | 143493.23 | 140397.94 | 141323.06 |
| 200uncor1 | D1 | 84447.87 | 75162.20 | 74102.30 | 74307.92 |
| | D2 | 81096.68 | 74813.83 | 74230.71 | 74522.35 |
| | D3 | 81238.31 | 74700.25 | 74330.17 | 74590.06 |
| | D4 | 84706.40 | 75093.47 | 74297.73 | 74582.78 |
| 200uncor2 | D1 | 38811.55 | 32417.83 | 32048.10 | 32157.17 |
| | D2 | 36597.59 | 32219.29 | 31917.73 | 32063.47 |
| | D3 | 36613.97 | 32152.18 | 31985.14 | 32093.07 |
| | D4 | 38989.23 | 32374.63 | 31867.19 | 31942.49 |
| 200uncor3 | D1 | 158094.32 | 123827.63 | 122889.66 | 123655.48 |
| | D2 | 145651.06 | 122875.33 | 121582.47 | 122542.66 |
| | D3 | 144406.92 | 122558.61 | 121787.37 | 122154.81 |
| | D4 | 156247.13 | 123656.16 | 121416.58 | 121944.74 |
| 200uncor4 | D1 | 238200.06 | 166562.50 | 163103.08 | 163965.87 |
| | D2 | 213766.66 | 165092.33 | 163313.96 | 164468.05 |
| | D3 | 205287.22 | 164600.83 | 162319.57 | 162955.55 |
| | D4 | 225416.49 | 166012.44 | 163373.89 | 164113.67 |
| 200uncor5 | D1 | 123852.35 | 90999.64 | 89043.21 | 89447.13 |
| | D2 | 112132.05 | 90253.81 | 88750.38 | 89442.92 |
| | D3 | 110653.75 | 90004.98 | 89358.15 | 89756.49 |
| | D4 | 120673.15 | 90768.88 | 89096.84 | 89682.88 |
| 200uncor6 | D1 | 16517.84 | 13364.33 | 13183.64 | 13256.00 |
| | D2 | 15352.69 | 13274.02 | 13092.94 | 13171.47 |
| | D3 | 15329.02 | 13244.44 | 13167.72 | 13218.32 |
| | D4 | 16503.13 | 13349.34 | 13146.05 | 13179.44 |
| 200uncor7 | D1 | 98688.98 | 86349.50 | 85006.35 | 85282.14 |
| | D2 | 94372.49 | 85930.83 | 84892.45 | 85265.85 |
| | D3 | 94534.69 | 85791.25 | 85087.12 | 85288.38 |
| | D4 | 99266.65 | 86283.90 | 84831.98 | 84951.58 |
| 200uncor8 | D1 | 132038.75 | 101076.29 | 99315.64 | 100116.67 |
| | D2 | 121102.25 | 100237.79 | 98209.04 | 98988.87 |
| | D3 | 120143.78 | 99958.20 | 98840.34 | 99262.39 |
| | D4 | 131384.51 | 100801.51 | 98603.26 | 98916.01 |
| 200uncor9 | D1 | 65358.21 | 52999.90 | 52366.24 | 52726.75 |
| | D2 | 61017.13 | 52632.55 | 52057.64 | 52412.13 |
| | D3 | 60666.27 | 52509.96 | 52029.22 | 52215.54 |
| | D4 | 64814.11 | 52888.59 | 52202.51 | 52374.32 |
| 200uncor10 | D1 | 231720.24 | 202053.73 | 199875.06 | 200492.32 |
| | D2 | 221326.69 | 201050.06 | 199941.10 | 200897.58 |
| | D3 | 220687.45 | 200708.65 | 198916.38 | 199506.77 |
| | D4 | 232013.10 | 201812.81 | 199357.95 | 199830.64 |