Various forms of dependent rounding are useful when handling a mixture of “hard” (e.g., matroid) constraints and “soft” (packing) constraints. We consider a few classes of such problems that arise in facility location, where one aims for small additive violations of the packing constraints, and where we require substantial “near-independence” properties among the variables being rounded. While the classical works in discrepancy theory (Beck-Fiala (1981) and Karp et al. (1987)) as well as more-modern works such as those of Bansal-Nagarajan (2016) – which have many further properties – yield such small additive violations, they do not yield the forms of near-independence that we require.

We develop a new rounding technique, block-selection rounding, which generalizes dependent rounding to allow multiple linear constraints and multiple choices for each item. We show that it satisfies near-independence properties, and use this to develop approximation algorithms for knapsack median and knapsack center problems. We anticipate that this technique will have more-general applicability.
Contents

1 Introduction
   1.1 Overview of our techniques ........................................... 2

2 Our results
   2.1 The BSL guarantee and high-level proof idea .......................... 4
   2.2 Dependent rounding and BSR .......................................... 4
   2.3 Comparisons to related work ....................................... 6
   2.4 Applications ......................................................... 7
      2.4.1 The knapsack median problem .................................. 7
      2.4.2 The knapsack center problem .................................. 8

3 Block-selection with linear constraints .................................. 9
   3.1 BSR followed by independent selection ............................... 12
   3.2 The change in $Q_\lambda(W,y)$ in a single round ....................... 13
   3.3 The total change in $Q_\lambda(w,y)$ .................................. 15
   3.4 A BSR rounding result for a small number of high-volume groups .. 16

4 The multi-knapsack center problem ...................................... 19
   4.1 Proof of Theorem 2.5 ................................................. 20
   4.2 Proof of Theorem 2.6 ................................................. 21

5 Bi-factor approximation algorithm for knapsack median ............... 23

6 Approximation algorithm for multi knapsack median .................... 27
   6.1 The Charikar-Li algorithm ........................................... 27
   6.2 Facility selection as a block-selection process ....................... 28
   6.3 BSR selection strategy .............................................. 29

7 Further correlation bounds for BSR ..................................... 33
   7.1 Analyzing property (E2) for $\lambda < 0$ and small $Q_\lambda(w,y)$ .... 33
   7.2 Analyzing property (E2) for $\lambda > 0$ ................................ 38
   7.3 Concentration bounds for FullBSR .................................. 41

8 Acknowledgment ..................................................................... 42

A Some technical lemmas ....................................................... 44
1 Introduction

Dependent rounding is a popular technique in designing approximation algorithms. It allows us to randomly round a fractional solution $x$ to an integral vector $X$ such that $E[X] = x$, all $X_i$’s are (“cylindrically”) negatively correlated, and the cardinality constraint $\sum_i X_i = \sum_i x_i$ holds with probability one. Over the last two decades, increasingly-sophisticated dependent-rounding techniques have been used to obtain good solutions to optimization problems on matroid- and matroid-intersection- polytopes, subject in some cases to additional constraints; see, e.g., [1][2][10][13][23]. For example, in the context of facility-location problems, Charikar and Li [8] applied dependent rounding in a clever way to obtain a 3.25-approximation for the classical $k$-median problem. More recently, Byrka et. al. [5] showed the “near-independence” property for small subsets of variables when running dependent rounding on a random permutation of the input vector. Dependent rounding is often used to round a fractional vector subject to (almost) satisfying a single linear constraint with non-negative coefficients $[1][23]$. It is not difficult to replace the cardinality constraint by a general linear constraint $\sum_i a_i X_i = \sum_i a_i x_i$ if the $a_i$’s are non-negative. Our overarching question motivated by applications, is: how well can we approximate independence while preserving a set of hard “clustering” constraints and (almost-)%preserving a set of soft “knapsack” constraints?

Facility location with knapsack constraints. We will consider a number of facility-location problems, subject to one or multiple linear constraints. Problem instances consist of a set $\mathcal{F}$ of facilities, a set $\mathcal{C}$ of clients, a symmetric distance metric $d$ on $\mathcal{F} \cup \mathcal{C}$, and an $m \times |\mathcal{F}|$ weight matrix $M$, which corresponds to $m$ non-negative weight functions. Our goal is to choose a set $S \subseteq \mathcal{F}$ of facilities (“centers”) so that all $m$ knapsack constraints are satisfied – i.e., with the usual notation $[u] \triangleq \{1, 2, \ldots, u\}$, we want (exactly or up to some small violation)

$$\forall \ell \in [m], \sum_{i \in S} M_{\ell,i} \leq B_\ell,$$

and such that the distances $d(j, S)$ are “small”. (As usual, $d(j, S)$ refers to $\min_{s \in S} d(j, s)$.) The specific metric used to boil down the values $d(j, S)$ into an objective function is problem-specific. We consider “center-type” problems, in which we minimize $\max_{j \in \mathcal{C}} d(j, S)$, as well as “median-type” problems, in which we minimize $\sum_{j \in \mathcal{C}} d(j, S)$.

Bounded knapsacks. We will be particularly interested in a class of linear constraints we refer to as bounded knapsacks, in which all the entries of the constraint matrix $M$ are bounded in the range $[0, 1]$. Previous rounding algorithms for knapsacks have focused on a multiplicative pseudo-approximation guarantee, namely finding a solution $Y_i$ such that $\sum_{i \in S} M_{\ell,i} Y_i \leq (1 + \epsilon) B_\ell$. If $\epsilon$ is constant and $B_\ell$ is large, this can be a significant deviation from the optimal solution. We examine how to obtain additive pseudo-approximation guarantees, of the form $\sum_{i \in S} M_{\ell,i} Y_i \leq B_\ell + a$, where $a$ is constant (for fixed values of $m$ and the desired approximation ratio). These types of additive guarantees do not appear to be possible from prior methods, which are based on independent rounding. These additive guarantees can also be used to obtain new multiplicative guarantees with improved run-times.

Our results for facility location are summarized by Theorems 2.2, 2.3, 2.4, 2.5, and 2.6.

Block selection with linear constraints (BSL): This scenario of Section 3 which leads to our most technical challenges, generalizes the usual dependent rounding framework in two distinct ways: it allows multiple linear constraints, and it allows multiple possible choices per item. (In dependent rounding, each item has a binary choice, to either set $X_i = 0$ or $X_i = 1$.)

This can be summarized as follows. We are given $y \in [0, 1]^n$, a non-negative $m \times n$ “knapsack constraint” matrix $M$ (where $m$ will generally be viewed as a small parameter such as a constant), and a partition of $[n]$ as $[n] = G_1 \sqcup G_2 \sqcup \cdots \sqcup G_r$, satisfying $y(G_i) = 1$ for all $i$, and $M y = 1$. (As usual,
\(y(A)\) denotes \(\sum_{a \in A} y_a.\) We want to select exactly one item from each group \(G_i,\) subject to preserving the knapsack constraints with respect to the original solution. That is, we have a fractional solution to a partition-matroid constraint that also satisfies \(m\) weight (knapsack) constraints. We aim to round \(y\) to a random \(Y \in [0, 1]^n\) in which: (a) for some ("small") parameter \(t,\) at most \(t\) coordinates \(Y_j\) lie in \((0, 1),\) (b) with probability one, \(Y(G_i) = 1\) for all \(i\) and \(MY = 1,\) and (c) such that we well-mimic (in senses made more precise below) independent rounding – wherein we pick exactly one index for each \(G_i\) using \(y\) as the distribution, and independently for different \(G_i.\) Note that independent rounding is far from satisfying \(MY = 1,\) as there are instances where it will lead to \(\max_i(MY_i) = \Theta(\log m / \log \log m)\) with high probability.

The precise sense in which we aim to mimic independent rounding is somewhat technical, but we can give a simplified example. Under independent rounding, for any \(W \subseteq [n],\) we would have

\[
\Pr[Y(W) = 0] = \prod_{i=1}^r (1 - y(W \cap G_i)).
\]

The simplest form of our rounding procedure that for all sets \(W,\) we have

\[
\Pr[Y(W) = 0] \leq \prod_{i=1}^r (1 - y(W \cap G_i)) + O(m^2/t),
\]

where again \(t\) is the number of variables left unrounded; this error of \(O(m^2/t)\) will be small in our applications for a suitably-small \(t,\) since we view \(m\) as a small parameter. (Please also read the following footnote where we state that our error-bounds are actually more general: e.g., in some of our settings where \(\prod_{i=1}^r (1 - y(W \cap G_i))\) is "small''.)

Why is BSL relevant, and in particular, why might one be interested in upper-bounding terms of the form \(\Pr[Y(W) = 0]?\) This upper-bounding is critical in several facility-location-type algorithms; see, e.g., [6,12]. In such algorithms, one clusters the facilities in some (greedy) manner (these are the \(G_i\) in our notation), and open one – or sometimes more – facilities suitably at random from each cluster. For any given client \(j,\) we first check if some "nearby" facility gets opened and if so, connect \(j\) to one such facility; if not, the clustering guarantees a backup facility, which, however, is farther away. The bad event of no "nearby" facility getting opened is precisely modeled by events of the form \(Y(W) = 0.\) These combined with additional ideas lead to our results for Knapsack Median and Knapsack Center: Theorems 2.2, 2.3, 2.4, 2.5 and 2.6. We anticipate that our tools will be useful more broadly in such location-theoretic problems.

We remark here that (3) is only a simplified form of our rounding results. In particular, equation (3) has an additive gap between the “ideal” distribution (independent selection) and our BSR. If the quantity \(\prod_{i=1}^r (1 - y(W \cap G_i))\) is very small, this type of gap may not suitable. We will also develop improved bounds for two important scenarios: (i) there are a small (constant) number of groups \(G_i\) for which \(y(W \cap G_i)\) is close to one; (ii) we have a lower-bound on the relevant size of \(\prod_{i=1}^r (1 - y(W \cap G_i))\). The first case is critical to our approximation algorithm for knapsack median. These two improved bounds are somewhat technical to state here and are deferred to later. Please also see the Remark following Theorem 2.1.

### 1.1 Overview of our techniques

In Section 3 we will provide a detailed technical description of our rounding algorithm, and its approximation guarantees. In summary, our aim is to develop a randomized rounding scheme preserving near-independence among the random variables involved. These algorithms are based on a series of modification steps, similar to the dependent rounding algorithm of [1,23]: we select a small number of indices, and modify the fractional values \(x_i\) in such a way that the expected values and the linear constraints are preserved, and such that the values become closer to integral. For BSL, where there are \(m\) linear constraints, we must modify \(2m\) indices at a time in order to preserve all the relevant linear constraints.
The BSR rounding algorithm is similar a dependent rounding procedure of \([1,23]\) (which, however, only allows a single knapsack constraint). In order to provide some intuition, we will compare the behavior of our BSR rounding algorithm to dependent rounding, restricted to this special case.

The latter algorithm satisfies a limited but important form of negative correlation, namely the negative cylinder property:

**Definition 1.1** (Negative cylinder property). Random variables \(X_1, X_2, \ldots, X_u \in [0, 1]\) have the “negative cylinder property” if

\[
\forall S \subseteq [u] : \mathbb{E} \left[ \prod_{i \in S} X_i \right] \leq \prod_{i \in S} x_i, \quad \text{and} \quad \mathbb{E} \left[ \prod_{i \in S} (1 - X_i) \right] \leq \prod_{i \in S} (1 - x_i). \tag{4}
\]

To see why this property is preserved, consider co-rounding two variables \(x_1, x_2\) such that the sum \(a_1x_1 + a_2x_2\) is preserved and the expected values of \(x_1, x_2\) do not change. If \(a_1a_2 > 0\), then an increase in \(x_1\) will lead to a decrease in \(x_2\) for the sum to remain the same, and vice versa. Thus, we have negative correlation between \(x_1\) and \(x_2\) and so \(\mathbb{E}[x_1'x_2'] \leq x_1x_2\), where \(x_1'\) and \(x_2'\) are the respective (random) updated values of \(x_1\) and \(x_2\); this explains why we obtain negative correlation for all techniques in \([5, 9, 13, 23]\).

One can hope that for arbitrary sets \(S, T \subseteq n\) we satisfy a similar cylindrical constraint

\[
\mathbb{E} \left[ \prod_{i \in S} X_i \prod_{i \in T} (1 - X_i) \right] \approx \prod_{i \in S} x_i \prod_{i \in T} (1 - x_i)
\]

Due the shape of the solution space, such a constraint cannot be preserved exactly; for example, if \(x_1 = x_2 = 1/2\), then any integral solution must either satisfy \(E[X_1(1 - X_2)] \geq 1/2\) or \(E[(1 - X_1)X_2] \geq 1/2\). This makes (approximately) satisfying general cylinder constraints much more difficult than satisfying negative cylinder constraints.

Let us rephrase this cylinder property in terms of a potential function; for fixed \(S, T\), let us define

\[
\Phi_{S,T}(x_1, \ldots, x_n) = \prod_{i \in S} x_i \prod_{i \in T} (1 - x_i)
\]

As the rounding algorithm proceeds through its rounding steps, we will ensure each rounding step transforms a vector \(x\) into a new, slightly more integral, vector \(x'\) so that the potential function \(\Phi_{S,T}\) does not change too much, i.e.

\[
\mathbb{E}[\Phi_{S,T}(x') \mid x] \approx \Phi_{S,T}(x)
\]

Our rounding algorithms use two major techniques to ensure this. First, every rounding step is symmetric: at each stage, we generate a modification vector \(\delta\), and either \(x' = x + \delta\) or \(x' = x - \delta\), each occurring with probability 1/2. By doing so, we ensure that the change in each coordinate \(x_i\) during a rounding step is bounded by the current value of \(x_i\). This in turn ensures that the typical change in the value of \(\Phi\) is bounded by (and small compared to) the current value of \(\Phi\); in particular, \(\Phi\) is unlikely to change by a large multiplicative factor in any single timestep. This is very different from the situation with the usual dependent rounding, where it is possible for an entry \(x_i\) to jump from a very small fractional value to become fully integral in a single step.

The second major technique is to randomly select which indices of \(x\) to modify, at each stage. This spreads out the (inevitable) correlation among the entries of \(x\). This also lessens the change in \(\mathbb{E}[\Phi_{S,T}]\) at each stage; for example, if we select zero or one indices in \(S \cup T\) to round, then \(\mathbb{E}[\Phi_{S,T}(x')] = \Phi_{S,T}(x)\). Even if we select multiple entries of \(S, T\), we are likely to select entries which have a relatively small contribution to \(\Phi\).
Due to these modifications, we can no longer ensure that each rounding step actually causes a new entry of \( x \) to become integral at each step. Such rounding algorithms can require significantly longer than \( n \) iterations to fully round the vector. Consequently, we expect every entry of \( x \) to have some, small, interaction with every other entry of \( x \). For this reason, the proof-strategy used in [5], which was based on showing that many entries \( x_i, x_j \) for \( i, j \in S \cup T \) do not interact, cannot be used for our analysis.

To analyze BSR, we consider the more general potential function \( Q(W, x) = \prod_{i=1}^{n} (1 - y(W \cap G_i)) \). Our strategy is to derive a bound

\[
E[Q(W, x') \mid x] \leq Q(W, x) + F(x)
\]

for some small quantity \( F(x) \). We can then view this as a type of recurrence relation on \( E[Q(W, x)] \) namely

\[
E[Q(W, x')] \leq E[Q(W, x)] + E[F(x)]
\]

If the function \( F \) were deterministic, this would be a pure recurrence relation, which we could solve using standard techniques. The fact that \( x \) is itself a random variable makes this much more difficult to solve. Our technical analysis is devoted to arguing around the randomness in \( x \) and \( F(x) \), and thereby reducing the stochastic recurrence relation for \( E[Q(W, x)] \) into something which is analogous to a pure recurrence relation.

**Organization:** We formally describe our results in Section 2. Section 3 develops our main rounding theorems; the applications are presented in Sections 4, 5, 6. We present further analysis of rounding algorithm in Section 7.

## 2 Our results

### 2.1 The BSL guarantee and high-level proof idea

The block-selection with linear constraints (BSL) problem is a generalization of dependent rounding. In this setting, we are given a partition \( G_1, \ldots, G_r \) over a ground-set \([n]\) (we refer to the sets \( G_i \) as groups) as well as an \( m \times n \) matrix \( M \). (We can view \( m \) as “small”, e.g., as a constant.) Finally, we are given a fractional vector \( y \in [0, 1]^n \), for which \( y(G_i) = 1 \) for each \( i \). Our goal is to select exactly one item from each group \( G_i \). For each \( j \in [n] \), let \( Y_j \) be the indicator variable for item \( j \) being selected. In an ideal scenario, we would like to generate the vector \( Y \in \{0, 1\}^n \) to satisfy the following desiderata (which are, to a certain extent, mutually incompatible and unattainable – indeed, our Theorem 2.1 instead guarantees certain related conditions (E1)–(E5)):

- (D1) For all \( j \in [n] \) we have \( E[Y_j] = y_j \);
- (D2) The random variables \( Y_j \) have negative cylinder correlation;
- (D3) \( Y(G_i) = 1 \) for \( i = 1, \ldots, r \); and
- (D4) \( MY = My \).

This can be interpreted as randomized rounding for a partition matroid, as the constraint that we select one item per group is a partition matroid constraint, and the vector \( y \) is a fractional vector in the polytope defined by that matroid. Constraint (D3) simply says that we select a single item from each group.

To further understand these constraints, let us consider independent selection, which is the obvious selection strategy: independently for each group \( G_i \) with \( i = 1, 2, \ldots, r \), we select exactly one item \( Z_i \in G_i \) with probability \( \Pr(Z_i = j) = y_j \). We then set \( Y_j = 1 \) if \( j = Z_i \) for some group \( G_i \) and \( Y_j = 0 \)
otherwise. This is a valid probability distribution as \( y(G_i) = 1 \) and the entries of \( y \) are in the range \([0, 1]\). It is easily seen to satisfy desiderata (D1), (D2), (D3) perfectly. More generally, we write \( Y = \text{INDEPENDENTSELECTION}(G_j, y) \) to indicate that the (fully integral) vector \( Y \) is obtained by applying independent selection with respect to \( y, G_j \).

However, independent selection does not satisfy (D4) very well: specifically, for any \( k = 1, \ldots, m \), the value of \( M_k Y \) – where \( M_k \) denotes the \( k \)th row of \( M \) – will be a sum of negatively-correlated random variables with mean \( M_k y \). Depending on the size of \( M_k y \) and the size of the entries of \( M \), this may deviate significantly from \( M_k y \); for instance, if \( M_k y = 1 \) and the entries of \( M \) are in the range \([0, 1]\), then we can expect that \( M_k Y \) can become as large as \( \Theta(\log m / \log \log m) \).

The key to our analysis is defining an appropriate potential function, which in a sense measures how close our algorithm is to matching independent selection. For any \( w \in [0, 1]^n, y \in [0, 1]^n \) and \( \lambda \in [-1, \infty) \), we define

\[
Q_\lambda(w, y) = \prod_{i=1}^{r} (1 + \lambda \sum_{j \in G_i} w_j y_j).
\]

For any set \( W \subseteq [n] \), we extend this to define \( Q_\lambda(W, y) := Q_\lambda(\chi_W, y) \), where \( \chi_W \) is the indicator function for \( W \), i.e.

\[
Q_\lambda(W, y) = \prod_{i=1}^{r} (1 + \lambda y(W \cap G_i)).
\]

The potential functions \( Q_\lambda \) play a number of different roles in our analysis, but we can summarize two of the most important interpretations. Consider some set \( W \subseteq [n] \). If \( Y = \text{INDEPENDENTSELECTION}(G_j, y) \), then note that \( Y(W) = 0 \) iff \( Q^{-1}(W, Y) = 1 \). So

\[
\Pr[Y(W) = 0] = \mathbb{E}[Q^{-1}(W, Y)] = Q^{-1}(W, y).
\]

Because of this connection between independent selection and \( Q^{-1} \), the case \( \lambda = -1 \) is especially important. Therefore, to simplify notation, we write \( Q(W, y) \) (without a subscript), as shorthand for \( Q^{-1}(W, y) \).

To get intuition about BSR, suppose we run \( y' = \text{BSR}(G_j, y) \) and achieved a fully integral vector \( y' \). (There remain a few fractional entries in \( y' \), but it is much more difficult to get intuition about their behavior.) In this, the quantity \( \mathbb{E}[Q^{-1}(W, y')] \) measures the extent to which the probability of having \( y'(W) = 0 \) is similar to the “gold standard” of independent selection.

For another interpretation of \( Q \), suppose we have some weighting vector \( w \in [0, 1]^n \), and we want to give concentration bounds on \( y' \cdot w \) (the usual inner product of \( y' \) and \( w \)), where \( y' = \text{BSR}(G_j, y, t) \). Since \( \mathbb{E}[y'_j] = y_j \) for every \( j \), we would expect \( y' \cdot w \) to be similar to \( y \cdot w \), and we can hope to show that \( y' \cdot w \) is concentrated around \( y \cdot w \); note that if \( Y = \text{INDEPENDENTSELECTION}(G_j, y) \), then values of \( Y \) would be negatively correlated and the Chernoff bound would hold with respect to \( Y \cdot w \). In order to show this type of Chernoff-like concentration, we will make use of a standard proof technique involving a Laplace transform of \( y' \cdot w \). For this purpose, \( Q_\lambda(w, y') \) will serve as the kernel of a type of Laplace transform with exponential parameter \( \lambda \). As we will later see, when \( y' \cdot w \geq s \) then \( Q_\lambda(w, y') \geq (1 + \lambda)^s \), so that

\[
\Pr(y' \cdot w \geq s) \leq \mathbb{E}[Q_\lambda(w, y')]/(1 + \lambda)^s.
\]

Again, estimating the expectation \( \mathbb{E}[Q_\lambda(w, y')] \) during the evolution of BSR, will be the key to showing appropriate concentration bounds on \( y' \cdot w \).

The desiderata (D1)–(D4) cannot be exactly satisfied. Instead, the BSR algorithm we develop in Section 3 will output a vector \( y' \in [0, 1]^n \) satisfying the weaker constraints:

(E1) For all \( j \in [n] \) we have \( \mathbb{E}[y'_j] = y_j \);
(E2) For all $w \in [0,1]^n$ and all $\lambda \in [-1,\infty)$, $\mathbf{E}[Q_\lambda(w,y')]$ is “not much more than” $Q_\lambda(w,y)$;

(E3) $y'(G_i) = 1$ for $i = 1,\ldots,r$;

(E4) $My' = My$;

(E5) At most $t$ entries of $y'$ are fractional.

Theorem 2.1, which is the basic theorem for our BSL-rounding algorithm, summarizes this dependence; please note the important remark following this theorem, which notes that we sometimes need, and obtain, much stronger error bounds than the easy-to-state additive bound of $O(m^2/t)$.

**Theorem 2.1.** BSR guarantees properties (E1)–(E5) above. Specifically for (E2), for all $w \in [0,1]^n$ and $\lambda \in [-1,\infty)$, BSR yields $\mathbf{E}[Q_\lambda(w,y')] \leq Q_\lambda(w,y) + O(m^2/t)$.

**Remark about Theorem 2.1.** Note that Theorem 2.1 provides an additive error term compared to independent selection. If $Q(w,y)$ is already very close to zero, this may be too crude. In Section 7, we show further bounds on the behavior of BSR to better handle cases in which $Q_\lambda(w,y)$ is small. These include concentration bounds on $y \cdot w$; in such cases, $Q_\lambda(w,y)$ estimates the probability of a deviation of $y' \cdot w$ from its mean value $y \cdot w$. Such deviations should have exponentially small probability, and thus the simple estimate of Theorem 2.1 is not suitable.

### 2.2 Dependent rounding and BSR

Dependent rounding can be interpreted as a special case of BSL. For, consider a vector $x_1,\ldots,x_v \in [0,1]^v$. We convert this into a BSL instance with $n = 2v$; for each item $i \in [v]$, we define $y_i = x_i, y_{i+v} = 1 - x_i$, and we define a group $G_i = \{i, v + i\}$. Finally, given an $m$-dimensional weighting function $a_1,\ldots,a_v$, we extend this to $[n]$ by setting $M_i = a_i, M_{i+v} = 0$ for $i \in [v]$. Now observe that if we run BSL on this problem instance, the fractional vector defined by $X_i = Y_i$ for $i \in [v]$ will satisfy $\sum_{i=1}^v a_i X_i = \sum_{i=1}^{2v} M_1 Y_1 = \sum_{i=1}^v x_i$. This technique of creating “dummy elements” (in this case, the elements $i + v$), as indicators for not selecting items, will appear in a number of constructions with BSL.

To simplify the notation, for any problem instance with items 1,\ldots,v, and associated vector $x_1,\ldots,x_v \in [0,1]^v$, let us define

$$X = \text{BSR-DepRound}(x, M, t)$$

to the result of applying BSR to the dummy elements. The resulting values $X_i$ can be interpreted as the fractional value to which item $i$ is selected.

Likewise, we define

$$X = \text{FULL-BSR-DepRound}(x, M, t)$$

to be the result of applying BSR-DepRound, followed by independent selection; the resulting vector $X$ in this case is fully integral.

### 2.3 Comparisons to related work

The basic setting of randomized rounding to respect partition matroid constraints has been considered as column-sparse minimax integer programs in works including [14, 21, 24]. These do not focus on near-approximate-correlation properties such as (E2). On the other hand, our Theorem 2.1 leaves a few variables unrounded. In addition to the settings being somewhat different, the Lovász Local Lemma–like approaches of [14, 21, 24] fundamentally do not carry over to our setting, since our additive error – added on top of a target r.h.s. bound such as $B_x$ – is fundamentally much smaller than the “standard deviation” value of $\sqrt{B_x}$. 
The classical discrepancy-based works of \cite{3, 17} do allow small additive error in packing (knapsack) constraints; more-sophisticated rounding theorems such as those of \cite{2} are also known. However, we are not aware of any approaches that deliver the “near-independence” property (E2) that is critical in our facility-location applications.

2.4 Applications

A typical application of dependent rounding is to convert a fractional solution, for example the solution to some LP relaxation, into an integral solution. For BSR, this is typically a two-part process. First, we round the fractional solution into a nearly integral solution, which preserves the weight function $\sum a_i x_i$, while approximately preserving the moments. Next, one must apply a problem-specific “end-game” to convert the partially integral solution to a fully-integral solution. There are a number of possible strategies for this second step. One attractive option is to apply independent rounding or independent selection. This may cause a violation of the knapsack constraint(s). However, since only the $t$ fractional values of $x_i$ are affected by this second rounding step, this will cause a much smaller change than if one applied independent rounding to the original fractional values. Another possible option, which we will use in Section 4, is to force all fractional values of $x$ to be zero or one in order to minimize $\sum a_i x_i$. This ensures that the final solution has smaller weight than the original fractional solution; again, since there are only $t$ fractional values, which are the only entries of $x$ affected by this rounding down, this process only slightly changes the solution quality.

At this point, the reader may wish to review the introductory sections on “Facility location with knapsack constraints” and “Bounded knapsacks,” before we present our concrete applications next.

2.4.1 The knapsack median problem

In the knapsack median problem, we seek to minimize the total connection cost $\sum_j d(j, S)$ subject to $m = 1$ knapsack constraint. This problem was first studied by Krishnaswamy et. al. \cite{19}. The current best approximation knapsack median is 7.08 due to \cite{20}. The special case when all facilities in $F$ have unit weight, known as the classical $k$-median problem, can be approximated to within a factor of $2.675 + \epsilon$ \cite{5}.

We consider the bounded multi-knapsack median problem; recall that in this context “multi” refers to presence of $m > 1$ knapsack constraints, and “bounded” refers to the property that the weights of all items are in the range $[0, 1]$. For this problem, we obtain a 3.25-pseudo-approximation, in which we slightly violate the knapsack constraints. Specifically, we get a solution satisfying $\sum_{j \in C} d(j, C) \leq (3.25 + \gamma)OPT$ and $\sum_{i \in S} M_{t,i} \leq B_t + \tilde{O}(m/\sqrt{\gamma})$. (Note that it is NP-hard to obtain a true approximation for multi-knapsack median. Chen et. al. \cite{11} have proved this hardness result for the multi-knapsack center problem and it is not hard to see that the same result also holds for multi-knapsack median.) This type of additive approximation guarantee for bounded knapsack can be leveraged into a faster multiplicative approximation guarantee for arbitrary knapsack (where the entries of $M$ can be arbitrarily large).

Our main results here will be the following.

**Theorem 2.2.** There is a polynomial-time algorithm for bounded knapsack median with budget $B$ which generates, for any $\gamma > 0$, a solution $S$ such that $M(S) \leq B + O(1/\gamma)$ and such that the cost of $S$ is at most $(1 + \sqrt{3} + \gamma)$ times the optimal cost.

We can adapt this to get good approximation guarantees for arbitrary knapsacks, at the cost of getting a multiplicative violation of the budget:

**Theorem 2.3.** There is a randomized algorithm running in expected time $n^{O(\epsilon^{-1} \gamma^{-1})}$ for arbitrary knapsack (where the entries of the knapsack matrix $M$ can be arbitrarily large) generating a solution $S$ such that $M(S) \leq B(1 + \epsilon)$, and such that the cost of $S$ is at most $(1 + \sqrt{3} + \gamma)$ times the optimal cost.
For multi knapsack median, we show how to apply BSL rounding for a key step in a rounding algorithm of Charikar & Li [8]. This gives the following guarantee:

**Theorem 2.4.** There is a polynomial-time algorithm for bounded multi knapsack median on $m$ knapsack constraints which generates, for any $\gamma > 0$, a solution $S$ such that $M(S) \leq \vec{B} + \tilde{O}(m/\sqrt{\gamma})$, and whose cost is at most $3.25 + \gamma$ times the optimal cost.

Note in particular here that, for a bounded knapsack, the budget is violated by only a small additive factor which is independent of $B$. By contrast, methods based on independent rounding give a qualitatively different type of approximation guarantee, in which $M(S)$ could become as large as $B + O(\sqrt{B})$.

### 2.4.2 The knapsack center problem

This problem is very similar to knapsack median, except that the cost function is slightly different. We seek to minimize $\max_{j \in C} d(j, S)$ instead of the sum. The Knapsack Center (KC) problem (in the case $m = 1$) was first studied by Hochbaum & Shmoys in [15], under the name “weighted $k$-center”. They gave a 3-approximation algorithm and proved that this is best possible unless $P = NP$; see also [18]. More recently, Chen et. al. [11] considered the case $m > 1$. They showed that this problem is not approximable to within any constant factor, and gave a pseudo 3-approximation algorithm which may violate all but one knapsack constraint by a factor of $(1 + \epsilon)$.

Suppose that the optimal solution is $R$. Our result here is, for any fixed $\epsilon > 0$, a randomized polynomial-time algorithm with small budget violation that guarantees every client $j$ to have $d(j, S) \leq 3R$ with probability one and $E[d(j, S)] \leq (1 + 2/e + \gamma)R$, as well as guaranteeing $\sum_{i \in S} M_{\ell,i} \leq B_\ell + \tilde{O}(m/\sqrt{\gamma})$. (The budget violation is likely unavoidable because it is NP-hard to approximate this problem to within any constant factor.) For the standard Knapsack Center problem with one constraint, we give a related polynomial-time algorithm which returns a feasible solution such that (1) all vertices are within distance $3R$ from some chosen center and (2) almost all vertices have expected connection cost at most about $(1 + 2/e)R \approx 1.74R$.

We are thus able to obtain a finer version of the optimality result of [15]: we ensure that every client gets service that is at most 3-approximate with probability 1, but which is also much better on average. This can be quite helpful in flexible facility location, where the problem is solved repeatedly (such as a streaming-service provider periodically reshuffling its service locations using an algorithm such as ours), thus guaranteeing improved average service for every client even after relatively-few repeats.

We summarize these results as follows:

**Theorem 2.5.** For any $\delta, \gamma > 0$, there exists an algorithm which runs in $O\left(\frac{1}{\delta^2} \right)$ time and returns a feasible solution for the Knapsack Center problem (i.e., with $m = 1$ knapsack constraint) such that $T_j \leq 3R$ for all $i \in V$. Moreover, there is a client set $U \subseteq C$ such that:

1. $|U| \geq (1 - \delta)|C|$,
2. $\forall i \in U, \ E[T_j] \leq (1 + 2/e + \gamma)R$,
3. $\forall i \in C, \ T_j \leq 3R$ with probability one.

**Theorem 2.6.** For any $\gamma, \epsilon \in (0, 1/2)$, there is an algorithm for the bounded multi knapsack center problem which runs in expected time poly$(n)$ and returns a solution $S$ such that:

1. $M(S) \leq \vec{B} + \tilde{O}(m/\sqrt{\gamma})$
2. $\forall j \in C, \ E[T_j] \leq (1 + 2/e + \gamma)R$, 

8
3. ∀i ∈ C, T_j ≤ 3R with probability one.

Although our results for knapsack median are more important, they are also technically complex (both in terms of the dependent rounding they involve, as well as the overall facility selection algorithm). The results on Knapsack Center are much more elementary, so we present them first to serve as a warm-up.

3 Block-selection with linear constraints

We now consider the BSL problem, as defined in Section 2.1, and introduce our main algorithm which we call BSR, for block selection. The following notation will be useful throughout: for any set \( X \subseteq [n] \), and any vector \( z \in [0, 1]^n \), we define \( \text{frac}(X, z) \) to be the set of entries \( i \in X \) with \( z_i / \in \{0, 1\} \). Also, for any vector \( z \in [0, 1]^n \) and any \( i \in [r] \), we define

\[
T(z, i) = \max(0, |\text{frac}(G_i, z)| - 1).
\]

Likewise, for any vector \( z \in [0, 1]^n \), we define

\[
T(z) = \sum_{i=1}^{r} T(z, i).
\]

The algorithm \( \text{BSR}(G_j, M, y, t) \) takes as input a set of groups \( G_j \), a set of linear constraints \( M \), a fractional vector \( y \), and integer parameter \( t \), and returns a mostly rounded vector \( y' = \text{BSR}(G_j, M, y, t) \). The overall algorithm has two phases. The first is a relatively-straightforward dependent rounding within each group \( G_i \); we refer to this step as \( \text{INTRAGROUPREDUCE} \). The second step involves repeated applications of a more-complicated rounding process \( \text{BSR-ITERATION} \) which modifies multiple groups simultaneously.

**Algorithm 1** \( \text{BSR}(G_j, M, y, t) \)

1: \( y \leftarrow \text{INTRAGROUPREDUCE}(G_j, M, y) \)
2: \( \text{while } T(y) > t \text{ do} \)
3: \( \text{Update } y \leftarrow \text{BSR-ITERATION}(y) \).
4: \( \text{Return } y \)

**Algorithm 2** \( \text{INTRAGROUPREDUCE}(G_j, M, y) \)

1: \( \text{for } i = 1, \ldots, r \text{ do} \)
2: \( \text{while } |\text{frac}(G_i, y)| > m + 1 \text{ do} \)
3: \( \text{Let } \delta \in \mathbb{R}^n, \delta \neq 0 \text{ be such that } M\delta = 0, \delta(G_i) = 0, \text{ and } \delta_i = 0 \forall i \notin \text{frac}(G_i) \)
4: \( \text{Choose scaling factors } \gamma_0, \gamma_1 > 0 \text{ such that} \)
   - \( y + \gamma_0 \delta \in [0, 1]^n \text{ and } y - \gamma_1 \delta \in [0, 1]^n \)
   - There is at least one index \( i \in \text{frac}(G_i, y) \) such that \( y_i + \gamma_0 \delta_i \in \{0, 1\} \).
   - There is at least one index \( i \in \text{frac}(G_i, y) \) such that \( y_i - \gamma_1 \delta_i \in \{0, 1\} \).
5: \( \text{With probability } \frac{\gamma_0}{\gamma_0 + \gamma_1}, \text{update } y \leftarrow y + \gamma_0 \delta; \text{ with the complementary probability of } \frac{\gamma_1}{\gamma_0 + \gamma_1}, \text{ update } y \leftarrow y - \gamma_1 \delta. \)

The key rounding step of this algorithm is the \( \text{BSR-ITERATION} \):
Algorithm 3 BSR-ITERATION($y, t$)

1: Form a set $J \subseteq [r]$, wherein each $i \in [r]$ goes into $J$ independently with probability

$$p_i = 3m \times \frac{T(y, i)}{T(y)}.$$

2: For each $i \in J$, select two distinct elements $x_{i1}, x_{i2} \in \text{frac}(G_i, y)$, uniformly at random among all $\binom{|\text{frac}(G_i, y)|}{2}$ pairs.

3: if $|J| \geq m + 1$ then

4: Choose $\delta \in \mathbb{R}^n, \delta \neq 0$ such that

   - $M\delta = 0, y + \delta \in [0, 1]^n$, and $y - \delta \in [0, 1]^n$
   - There is at least one index $i \in \text{frac}([n], y)$ such that $y_i + \delta_i \in \{0, 1\}$ or $y_i - \delta_i \in \{0, 1\}$.
   - For each $i \in J$ we have $\delta_{x_{i1}} = -\delta_{x_{i2}}$
   - $\delta_j = 0$ if $j \in G_i$ and $i \notin J$.

5: With probability $1/2$, update $y \leftarrow y + \delta$; else, update $y \leftarrow y - \delta$

6: Return $y$

We will require throughout that $t > 12m^2$; this assumption will not be stated explicitly again.

By far the most technically-difficult part of the analysis will be to show that this rounding algorithm approximately satisfies criterion (E2). Before we do so, we analyze some of the easier properties of this algorithm; we show that it is well-defined and terminates in polynomial time.

**Proposition 3.1.** The value $p_i$ in Line (1) of BSR-ITERATION is a valid probability.

**Proof.** We clearly have $p_i \geq 0$. After lines (1) – (5) of BSR, we ensure that $T(y, i) \leq m + 1$ for each $i \in [n]$. Since the integral entries cannot later become fractional, this implies that $T(y, i) \leq m + 1$ for each application of BSR-ITERATION. Finally, we only execute BSR-ITERATION if $\sum_i T(y, i) > t$. So $p_i = 3m \times \frac{T(y, i)}{T(y)} \leq 3m \times \frac{m+1}{t}$. By our hypothesis $t > 12m^2$, this is at most 1. \qed

**Proposition 3.2.** One can (efficiently) find a vector $\delta \in \mathbb{R}^n$ as claimed in line 3 of INTRAGROUPREDUCE.

**Proof.** This line is only executed when $G_i$ has at least $m + 2$ fractional variables. On the other hand, we have only $m$ knapsack constraints ($M\delta = 0$) and the additional linear constraint $\delta(G_i) = 0$. This system is under-determined and the claim follows. \qed

**Proposition 3.3.** If $y' = \text{BSR-ITERATION}(G_j, M, y, t)$ for any vector $y$ with $T(y) \geq t$, then there is a probability of at least 0.24 that $y'$ has at least one more integral coordinate than $y$.

**Proof.** If $|J| \geq m + 1$, there is at least a 1/2 probability of producing at least one new rounded entry in $y'$. Also, $|J|$ is a sum of independent random variables with mean $\sum_i p_i \geq 3m$: hence by the Chernoff-Hoeffding bound, $\Pr[|J| \leq m] \leq e^{-3m(2/3)^2/2} \leq 0.52$. Thus, there is a probability of at least $(1 - 0.52) \times (1/2) = 0.24$ that there is at least one new rounded variable in going from $y$ to $y'$. \qed

**Proposition 3.4.** Let $j \geq t$, and let $Z_j$ denote the number of iterations during the execution of BSR in which $T(y) = j$. We have $\mathbb{E}[Z_j] \leq 10$.

**Proof.** By Proposition 3.3, there is a probability of at 1/10 that an entry of $y$ becomes integral in any given round, in which case $T(y') < T(y)$. Now, suppose we condition on that $T(y^k) = j$ for the first time at some iteration $k$. Then, in each subsequent round, there is a probability of at least 1/10 that $T(y)$ is no longer equal to $j$. So the expected number of rounds for which $T(y) = j$ is dominated by a geometric random variable with probability 1/10, and hence has mean $\leq 10$. \qed
**Proposition 3.5.** BSR runs in expected polynomial time.

**Proof.** In each iteration of INTRAGROUPREDUCE, the number of fractional entries in frac\((G_i, y)\) decreases by at least one. Thus, there are at most \(n\) iterations. By Proposition 3.3, each iteration of BSR-ITERATION has a probability of at least \(\Omega(1)\) that an entry of \(y\) becomes integral, causing \(T(y)\) to decrease by at least one. This implies that the expected number of iterations of BSR-ITERATION is \(O(n)\). Each of these iterations requires finding a vector in an appropriate null-space, which takes \(poly(n)\) time. \(\square\)

**Proposition 3.6.** Let \(y'\) be the random variable which is the output of \(y' = \text{INTRAGROUPREDUCE}(G_j, M, y)\); Then the following conditions hold:

1. \(y'(G_i) = 1\) for all \(i = 1, \ldots, r\);
2. \(My' = My\);
3. \(E[y'_j] = y_j\) for all \(j \in [n]\);
4. For any \(w \in [0, 1]^n\) and any \(\lambda \in [-1, \infty)\) we have \(E[Q_\lambda(w, y')] = Q_\lambda(w, y)\).

**Proof.** Let \(y^1\) be the vector before an iteration (i.e. before line (3)) and \(y^2\) be the vector afterwards (after line (5)). We update by either setting \(y^2 \leftarrow y^1 \gamma_0 \delta\) or \(y^2 \leftarrow y^1 \gamma_1 \delta\). Since \(\delta(G_i) = 0\) we have \(y^2(G_i) = y^1(G_i)\). Similarly, \(M \delta = 0\) so \(M y^2 = M y^1\). Also, for any \(j \in [n]\) we have

\[
E[\lambda^j] = y_j + \frac{\gamma_0}{\gamma_0 + \gamma_1}(\gamma_0 \delta_j) - \frac{\gamma_1}{\gamma_0 + \gamma_1}(\gamma_1 \delta_j) = y^1_j.
\]

Finally, for any set \(W\), we have

\[
E[Q_\lambda(w, y^2)] = E[\prod_{u=1}^r (1 + \lambda \sum_{j \in G_u} w_j y^2_j)] = \prod_{u \neq i} (1 + \lambda \sum_{j \in G_u} w_j y^1_j) (1 + \lambda \sum_{j \in G_i} w_j E[y^2_j])
\]

\[
= \prod_{u \neq i} (1 + \lambda \sum_{j \in G_u} w_j y^1_j) (1 + \lambda \sum_{j \in G_i} w_j y^1_j) = Q_\lambda(W, y^1).
\]

The claim follows by induction on all iterations. \(\square\)

**Proposition 3.7.** One can find a vector \(\delta \in \mathbb{R}^n\) as claimed in line (4) of BSR-ITERATION.

**Proof.** Consider forming a vector \(v \in [0, 1]^n\) as follows. For each \(i \in J\), we set \(v_{x_{i1}} = U_i\) and \(v_{x_{i2}} = -U_i\) for some variable \(U_i\) to be determined. All other entries of \(v\) are set to zero. The linear system \(M v = 0\) has \(|J| \geq m + 1\) variables \(U_i\) and only \(m\) constraints, hence is under-determined. Let \(v\) be any non-zero solution vector. Let \(\alpha \in \mathbb{R}\) be maximal such that \(y + \alpha v \in [0, 1]^n\) and \(y - \alpha v \in [0, 1]^n\). One may verify that \(\alpha < \infty\) and setting \(\alpha = \alpha \gamma\) achieves the claimed result. \(\square\)

**Remark: Iverson Notation.** In some places (e.g., in the last line of the proof of Proposition 3.8) and in the statements of Propositions 7.3 and 7.4, we will use the standard Iverson notation: if \(\phi\) is a proposition, then \([\phi]\) equals 1 if \(\phi\) is true, and equals 0 otherwise.

**Proposition 3.8.** Suppose \(y' = \text{BSR}(G_j, M, y, t)\). Then:

1. For all \(i = 1, \ldots, r\) we have \(y'(G_i) = 1\) and \(|\text{frac}(G_i, y)| \leq m + 1\).
2. For all \(j \in [n]\) we have \(E[y'_j] = y_j\).
3. \( M y' = M y. \)

4. There are at most \( t \) indices \( i \in [r] \) such that \( \frac{\lambda}{\delta} (G_i, y') \neq \emptyset. \)

Proof. Proposition 3.6 ensures that conditions (1), (2), (3) are all satisfied after \textsc{IntragroupReduce}. In each application of \textsc{BSR-Iteration}, we update \( y \) as either \( y' := y + \delta \) or \( y' := y - \delta. \) Note that \( \delta \) is chosen so that \( \delta(G_i) = \delta_{x_{i1}} + \delta_{x_{i2}} = 0. \) Thus, \( y'(G_i) = y(G_i). \) Similarly, \( \delta \) is chosen so that \( M \delta = 0, \) so that \( M y = M y'. \) For any \( j \in [n], \) we have \( E[y'_j] = y_j + (1/2)\delta_j - (1/2)\delta_j = y_j. \) Thus, properties (1), (2), (3) are also preserved in each of those steps. Also, we only terminate iterations Lines (1) – (5) if \( |\frac{\lambda}{\delta}(G_i, y)| \leq m + 1. \)

The condition \( y(G_i) = 1 \) ensures that \( |\frac{\lambda}{\delta}(G_i, y)| \) is either zero or is at least two. Therefore, if \( \frac{\lambda}{\delta}(G_i, y') \neq \emptyset, \) then \( T(y') \geq 1. \) The algorithm \textsc{BSR} can only terminate if \( T(y) \leq t, \) in which case \( \sum_{i=1}^r |\frac{\lambda}{\delta}(G_i, y')| \leq T(y') \leq t. \) \( \square \)

Finally, we note a useful bound on \( Q \) that appears in a number of proofs:

**Proposition 3.9.** We have \( Q_{\lambda}(w, y) \leq e^{\lambda y \cdot w}. \)

Proof. Since \( G_1, \ldots, G_r \) partitions \([n], \) we have

\[
Q_{\lambda}(w, y) = \prod_{i=1}^r (1 + \lambda \sum_{j \in G_i} y_j w_j) \leq \prod_{i=1}^r e^{\lambda \sum_{j \in G_i} y_j w_j} = e^{\lambda y \cdot w}.
\]

\( \square \)

### 3.1 BSR followed by independent selection

One natural rounding strategy for a BSL problem is to execute \textsc{BSR} up to some stage \( t > 12m^2, \) and then finish by independent rounding. Since this process comes up so frequently, let us define it formally as an algorithm \textsc{FullBSR}.

**Algorithm 4** \textsc{FullBSR}(\( G_j, M, y, t \))

1: \( y' \leftarrow \textsc{BSR}(G_j, M, y, t) \)
2: \( Y \leftarrow \textsc{IndependentSelection}(G_j, y') \)
3: Return \( Y \)

When we execute this process, we obtain a fully integral vector \( Y \in \{0, 1\}^n. \) This vector \( Y \) will not exactly satisfy the knapsack constraints, but it will be relatively close (depending on the value of \( t). \) Note that independent selection does not change the expectation of \( Q_{\lambda}(w, y); \) thus, all of our analysis of \( Q \) for \textsc{BSR} will carry over immediately to \textsc{FullBSR}.

**Theorem 3.10.** Let \( Y = \textsc{FullBSR}(y, t) \) with \( t > 12m^2 \) and \( y \in [0, 1]^n. \) Suppose \( M_{ki} \in [0, 1] \) for all \( k, i. \) Then, with probability at least \( 1 - \delta, \) it holds that

\[
MY \leq \bar{B} + O(\sqrt{t \log(m/\delta)}/\bar{t}).
\]

Proof. Let \( y' \) denote the vector after executing \( \textsc{BSR}(G_j, M, y, t). \) By Proposition 3.8, there are at most \( t \) groups \( G_{\ell} \) for which the value of \( y' \) within \( G_{\ell} \) is not completely integral. If we condition on the vector \( y', \) then \( M_k Y \) can be viewed as a sum of independent random variables. However, only \( t \) of these variables – corresponding to fractional groups – are random, while the rest are deterministic. Thus \( M_k Y \) is a weighted
sum of \( \ell \leq t \) independent random variables, each variable bounded in the range \([0,1]\). By Hoeffding’s bound, the probability that such a sum exceeds its mean by a factor of \( z \) is at most \( \exp(-2z^2/t) \).

By the union bound, the total probability of violating any constraint is at most \( m \exp(-2z^2/t) \). For \( z = 10\sqrt{\ln(m/\delta)} \), this probability is below \( \delta \).

### 3.2 The change in \( Q_\lambda(W, y) \) in a single round

In this section, we come to the heart of the analysis: bounding the overall change in \( Q_\lambda(W, y) \) caused by the BSR rounding steps. Let us consider a single BSR iteration, applied to a vector \( y \), and let \( y' \) denote the output of this iteration. We want to show that \( E(Q_\lambda(W, y')) \) is close to \( Q_\lambda(W, y) \). We define \( S = Q_\lambda(W, y) \) and \( S' = Q_\lambda(W, y') \). We will assume through that \( T(y) > t \) (as otherwise \( S' = S \) with probability one).

For each \( i \in [r] \), define \( c_i = \sum_{j \in G_i} w_j y_j \) and \( \mu_i = \sum_{j \in G_i} w_j \delta_j \). We also define the random variable \( U_i \) by:

\[
U_i = \begin{cases} 
0 & \text{if } i \notin J \\
\min(y_{x_{i1}}, y_{x_{i2}})|w_{x_{i1}} - w_{x_{i2}}| & \text{if } i \in J.
\end{cases}
\]

Observe that the random variables \( U_i \) are independent (as each only depends on the choices made within \( G_i \)), although the random variables \( \mu_i \) may be highly correlated.

**Proposition 3.11.** For any \( i \in [r] \) we have \( |\mu_i| \leq U_i \).

**Proof.** If \( i \notin J \), then necessarily \( \mu_i = 0 = U_i \). So suppose \( i \in J \). By construction and our choice of \( \delta \), if \( x_{i1}, x_{i2} \in W \) or \( x_{i1}, x_{i2} \notin W \) then \( \mu_i = 0 = U_i \).

Finally, observe that \( |\delta_{x_{i1}}| \leq y_{x_{i1}} \) as \( y \in [0,1]^n \). We have \( \delta_{x_{i1}} = -\delta_{x_{i2}} \) and so \( |\mu_i| = |\delta_{x_{i1}} w_{i1} - \delta_{x_{i1}} w_{i2}| \leq y_{x_{i1}} |w_{x_{i1}} - w_{x_{i2}}| \). Similarly \( |\mu_i| \leq y_{x_{i2}} |w_{x_{i1}} - w_{x_{i2}}| \). \( \square \)

**Proposition 3.12.** For any \( i \in [r] \), we have \( E[U_i] \leq p_i \frac{4c_i(1-c_i)}{T(y, i)} \).

**Proof.** First, observe that \( i \in J \) with probability \( p_i \), and this is independent of \( x_{i1}, x_{i2} \). Suppose we enumerate the fractional entries of \( G_i \) as \( \text{frac}(G_i, y) = \{1, \ldots, q\} \) where \( q = T(y, i) + 1 \), sorted so that \( w_1 \leq w_2 \leq \cdots \leq w_q \). Let \( s_j = w_1 + \cdots + w_j \) for \( j = 1, \ldots, q \). We also assume that none of the entries in \( y(G_i) \) are equal to one (as otherwise \( U_i = 0 \)), and thus \( s_1 + \cdots + s_q = y(G_i) = 1 \). We also have \( c_i = w_{1y_1} + \cdots + w_{qy_q} \).

As \( x_{i1}, x_{i2} \) are chosen uniformly at random from \( \text{frac}(G_i, y) \), we have

\[
E[U_i] = p_i \frac{\sum_{u \neq v} \min(y_u, y_v)|w_u - w_v|}{q \choose 2}
\]

So in order to show the result it suffices to show that

\[
\sum_{u \neq v} \min(y_u, y_v)|w_u - w_v| \leq 2c_i(1-c_i)q.
\]

13
Suppose $c_i \geq 1/2$. We can then bound this quantity by

$$\sum_{u \neq v} \min(y_u, y_v)(w_u - w_v) = \sum_{u=1}^{q} \sum_{v=u+1}^{q} \min(y_u, y_v)(w_v - w_u)$$

$$\leq \sum_{u=1}^{q} y_u \sum_{v=u+1}^{q} (w_v - w_u) \leq \sum_{u=1}^{q} y_u (q - u - 1)(1 - w_u)$$

$$\leq \sum_{u=1}^{q} y_u (q - 1)(1 - w_u) - \sum_{u=1}^{q} uy_u(1 - w_u)$$

$$\leq (q - 1)(1 - c_i) \leq 2(q - 1)c_i(1 - c_i).$$

Otherwise, if $c_i \leq 1/2$, we bound it by

$$\sum_{u \neq v} \min(y_u, y_v)(w_u - w_v) = \sum_{u=1}^{q-1} \sum_{v=u+1}^{q} \min(y_u, y_v)(w_u - w_v)$$

$$\leq \sum_{u=1}^{q} y_u \sum_{v=u+1}^{q-1} (w_u - w_v) \leq \sum_{u=1}^{q} y_u (u - 1)w_u$$

$$\leq \sum_{u=1}^{q} y_u (q - 1)w_u = (q - 1)c_i \leq 2(q - 1)c_i(1 - c_i).$$

\[\square\]

**Proposition 3.13.** For a fixed input $y$ and $y' = \text{BSR-ITERATION}(y, t)$, we have

$$\mathbb{E}[S' \mid \delta] \leq S \cosh \left(\frac{12m \sum_i |\lambda c_i(1 - c_i)/(1 + \lambda c_i)|}{T(y)}\right)$$

where $c_i = \sum_{j \in G_i} w_{ij} y_j$.

**Proof.** Let us first condition on the random variable $\delta$ (which includes all the random choices up to but not including line (4) of Algorithm [1]). With probability 1/2 we change $y$ to $y + \delta$ and with probability 1/2 we change $y$ to $y - \delta$. Thus,

$$\mathbb{E}[S' \mid \delta] = 1/2 \prod_i (1 + \lambda c_i + \lambda \mu_i) + 1/2 \prod_i (1 + \lambda c_i - \lambda \mu_i)$$

$$= 1/2 \sum_{X \subseteq [r]} \prod_{i \in X} (\lambda \mu_i) \prod_{i \in [r] - X} (1 + \lambda c_i) + 1/2 \sum_{X \subseteq [r]} \prod_{i \in X} (-\lambda \mu_i) \prod_{i \in [r] - X} (1 + \lambda c_i)$$

$$= \sum_{X \subseteq W} \prod_{i \in X} (\lambda \mu_i) \prod_{i \in [r] - X} (1 + \lambda c_i) = S \sum_{X \subseteq [r]} \prod_{i \in X} \frac{\lambda \mu_i}{1 + \lambda c_i} \leq S \sum_{X \subseteq [r]} \prod_{i \in X} \frac{|X| \mu_i}{1 + \lambda c_i}$$

$$\leq S \sum_{X \subseteq [r]} \prod_{i \in X} \frac{|X| \mu_i}{1 + \lambda c_i}. \quad \text{(by Proposition 3.11)}$$

14
Now integrate over the remaining randomness, observing that variable $U_i$ are independent:

$$
E[S'] \leq S \sum_{X \subseteq [r]} \mathbb{E} \left[ \prod_{i \in X} \frac{|\lambda| U_i}{1 + \lambda c_i} \right] = S \sum_{X \subseteq [r]} \prod_{i \in X} \frac{|\lambda| \mathbb{E}[U_i]}{1 + \lambda c_i} 
$$

$$
= S \sum_{v=0}^{\infty} \sum_{X \subseteq W} \prod_{i \in X} \frac{|\lambda| \mathbb{E}[U_i]}{1 + \lambda c_i} \leq S \sum_{v=0}^{\infty} \left(\sum_{i \in [r]} \frac{|\lambda| \mathbb{E}[U_i]}{1 + \lambda c_i}\right)^{2v} = S \cosh \left(\sum_{i} \frac{|\lambda| \mathbb{E}[U_i]}{1 + \lambda c_i}\right)
$$

$$
\leq S \cosh \left(\frac{\sum_{i} 3mT(y, i)}{T(y)} \times \frac{4|\lambda|c_i(1 - c_i)}{T(y, i)(1 + \lambda c_i)}\right) \leq S \cosh \left(\frac{12m \sum_{i} |\lambda|c_i(1 - c_i)/ (1 + \lambda c_i)}{T(y)}\right)
$$

Proposition 3.14. For a fixed input $y$ and $y' = \text{BSR-ITERATION}(y, t)$, we have

$$
E[S'] \leq S \cosh \left(\frac{12m \ln S}{T(y)}\right).
$$

Proof. Apply Proposition 3.13 and continue to estimate:

$$
E[S'] \leq S \cosh \left(\frac{12m \sum_{i} |\lambda|c_i(1 - c_i)/ (1 + \lambda c_i)}{T(y)}\right) \leq S \cosh \left(\frac{12m \sum_{i} \ln(1 + \lambda c_i)}{T(y)}\right) \leq S \cosh \left(\frac{12m \ln S}{T(y)}\right)
$$

as $S = \prod_{i} (1 + \lambda c_i)$ and the cosh function is even.

3.3 The total change in $Q_{\lambda}(w, y)$

Now that we can bound the expected change in $Q_{\lambda}(w, y)$ in one rounding step, we can extend this to analyze multiple iterations. Let us define $y^i$ to be the vector after $i$ applications of BSR-ITERATION. Thus, $y^0$ is the output of INTRAGROUPREDUCE, and $y^P$ is the output of BSR, where $P$ is the total number of iterations executed. We have seen that $P < \infty$ with probability one. We correspondingly define $S_p = Q_{\lambda}(w, y^p)$ and $T_p = T(y^p)$.

Theorem 2.1. Let $w \in [0, 1]^n$ and $\lambda \in [-1, 0)$. For a fixed input $y$ and $y' = \text{BSR}(y, t)$, we have

$$
E[Q_{\lambda}(w, y')] \leq Q_{\lambda}(w, y) + O(m^2/t).
$$

Proof. By Proposition 3.14, we get

$$
E[S_{p+1} | y^p] \leq S_p + \left( S_p \cosh \left(\frac{12m \ln S_p}{T_p}\right) - S_p \right).
$$
Since $\lambda \in [-1, 0]$, we have $S_p \in [0, 1]$. Our condition $t \geq 12m$ ensures that $12m/T_p \leq 12m/t \leq 1$. Therefore, by Proposition A.3 (with $u = 0$), we have $E[S_{p+1} \mid y^p] \leq S_p + (12m/T_p)^2$. When $T_p \leq t$, then trivially $S_{p+1} = S_p$ with probability one. Combining these two cases gives

$$E[S_{p+1} \mid y^p] \leq S_p + O\left(\frac{m^2 [T_p > t]}{T_p^2}\right).$$

Integrating over $y^p$ and applying iterated expectations, we therefore get for any $p \geq 0$:

$$E[S_p] \leq E[S_0] + O\left(m^2 \sum_{j=0}^{p-1} \frac{[T_j > t]}{T_j^2}\right).$$

By Proposition 3.6, it holds that $E[S_0] = Q_\lambda(w, y)$. We can rearrange the sum on $j$ as

$$\sum_{j=0}^{p-1} \frac{[T_j > t]}{T_j^2} = \sum_{k=t+1}^{\infty} \frac{1}{k^2} \sum_{j=0}^{p-1} [T_j = k] \leq \sum_{k=t+1}^{\infty} \frac{1}{k^2} \sum_{j=0}^{\infty} [T_j = k] = \sum_{k=t+1}^{\infty} Z_k/k^2.$$

Using Proposition 3.4, we therefore get:

$$E\left[\sum_{j=0}^{p-1} \frac{[T_j > t]}{T_j^2}\right] \leq \sum_{k=t+1}^{\infty} \frac{E[Z_k]}{k^2} \leq \sum_{k=t+1}^{\infty} 10/k^2 \leq 10/t.$$

Therefore, we have shown that for any fixed $p \geq 0$ we have

$$E[S_p] \leq Q_\lambda(w, y) + O(m^2/t).$$

Since $S_p \in [0, 1]$ both hold with probability one, this in turn implies that

$$E[S_p] \leq E[S_p] + Pr[P > p] \leq Q_\lambda(w, y) + O(m^2/t) + Pr[P > p]$$

Now take the limit as $p \to \infty$, and observe that $Pr[P > p] \to 0$. \qed

### 3.4 A BSR rounding result for a small number of high-volume groups

To analyze our knapsack median rounding algorithm, we will need to analyze $E[Q(W, y)]$ for special case of the BSR algorithm: one in which there is a small (constant) number of groups $j$ with $y(G_j \cap W)$ very close to one. This will cause $Q(W, y)$ to be very small; many of our estimate for the change in $E[Q(W, Y)]$ at the termination of BSR will not suitable, since they typically involve an additive term with respect to the original value $Q(W, y)$.

We will consider the following situation: we want to compute $Q(W, y)$ for a given set $W \subseteq [n]$. In addition, we are provided a set $U \subseteq [r]$, of size $|U| = u$. Let us define $W_0 = W \cap (\bigcup_{j \in U} G_j), W_1 = W - \bigcup_{j \in U} G_j$. Observe that

$$Q(W, y) = Q(W_0, y)Q(W_1, y)$$

for any vector $y$.

We will show the following rounding result:

**Theorem 3.15.** Let $y'$ denote the output of BSR$(G_j, M, y, t)$ with $t \geq 5000m^2(u + 1)^2$. Then

$$E[Q(W, y')] \leq Q(W, y) + Q(W_0, y)O\left(\frac{(u + 1)^2 m^2}{t}\right)$$

16
Note in particular that when $Q(W_0, y)$ is small, then this may be much stronger than Theorem 2.1. It is critical however in the proof that $W_0$ spans only a bounded number of groups. This situation is precisely what arises during our analysis of knapsack median. Also, let us observe that Theorem 2.1 is a special case of Theorem 3.15 with $U = \emptyset$. (We could have omitted the proof of Theorem 2.1 entirely, and just derived it as a corollary of Theorem 3.15)

In order to prove Theorem 3.15, let us define the potential function

$$
\Phi(y) = Q(W, y) + aQ(W_0, y)(1/t - 1/\max(t, T(y)))
$$

where $a$ will depend upon $m, u$ (in a manner to be specified later). Our strategy will be to show that if execute a single round of BSR-ITERATION, then for $z' = BSR-ITERATION(z)$ we have

$$
\mathbb{E}[\Phi(z') | z] \leq \Phi(z)
$$

Throughout the remainder of this section, let us therefore consider analyzing a single BSR-iteration, starting from a vector $z$; all the probabilities should be understood as conditional on all the state $z$. We also assume throughout that $T(z) > t$ (as otherwise $z' = z$ with probability one).

**Proposition 3.16.** Let $J$ be the formed at line (1) of BSR-ITERATION, and let $U \subseteq [n]$. If $t \geq 5000m^2(u + 1)^2$ it holds

$$
\Pr[U \cap J = \emptyset \land T(z') < T(z)] \geq 1/10
$$

**Proof.** Let $\mathcal{E}_2$ denote the event that $U \cap J = \emptyset$. Each $i \in U$ goes into $J$ with probability $p_i = 3mT(z, i)/T(z)$. After executing INTRAGROUPREDUCE, we are guaranteed that $T(z, i) < m + 1$, so $p_i \leq 3m(m + 1)/T(z)$. Therefore, overall we have

$$
P(\mathcal{E}_2) \geq (1 - \frac{3m(m + 1)}{T(z)})^u \geq (1 - \frac{6m}{5000m^2u^2})^u \geq (1 - \frac{6}{5000au^2}) \geq 1/2
$$

Next, suppose that we condition on $\mathcal{E}_2$. Observe that if $J \geq m + 1$, then with probability 1/2 there will be at least one rounded variable so that $T(z') < T(z)$. Furthermore, $J$ is a sum of independent random variables, with mean

$$
E[|J| | \mathcal{E}_2] = \sum_{i \in [r] \setminus u} p_i = \frac{3m \sum_{i \in [r] \setminus u} T(z, i)}{T(z)} \geq \frac{3m(T(z) - u(m + 1))}{T(z)} \geq \frac{3m(t - u(m + 1))}{t}
$$

By our hypothesis that $t \geq 5000m^2(u + 1)^2$, we have $u(m + 1) \leq 1/25000$, so overall $E[|J| | \mathcal{E}_2] \geq 2.99m$. By the Chernoff-Hoeffding bound, we have $Pr[|J| \leq m | \mathcal{E}_2] \leq e^{-2(2.99m)^2(0.34)^2/2} \leq 0.60$ and so $Pr[\mathcal{E} | \mathcal{E}_2] \geq 0.20$.

**Proposition 3.17.** When $a = 8000m^2(u + 1)^2$ and $t \geq 5000m^2(u + 1)^2$, we have $\mathbb{E}[\Phi(z')] \leq \Phi(z)$.

**Proof.** Let us define $T = T(z)$. Let $\mathcal{E}_1$ denote the event that $T(z') < T(z)$ and let $\mathcal{E}_2$ denote the event that $U \cap J = \emptyset$, where $J$ is the set formed at step (1) of BSR-ITERATION. We also define $S = Q(W, z), S_0 = Q(W_0, z), S_1 = Q(W_1, z), S' = Q(W, z'), S'_0 = Q(W_0, z'), S'_1 = Q(W_1, z')$. We have $\max(t, T(z')) \leq T - |\mathcal{E}_1|$, so

$$
\Phi(z') = S' + aS'_0(1/t - 1/\max(t, T(z'))) \leq S' + aS'_0\left(\frac{1}{t} - \frac{1}{T - |\mathcal{E}_1|}\right)
$$

$$
\leq S' + aS'_0\left(\frac{1}{t} - \frac{1}{T} - \frac{|\mathcal{E}_1|}{2T^2}\right) \quad \text{for } T > t > 10
$$

17
Taking expectations, we therefore have:

\[
E[\Phi(z') - \Phi(z)] \leq E[S' - S] + a E[S_0' - S_0]\left(\frac{1}{t} - \frac{1}{T}\right) - \frac{a E[S_0'|\mathcal{E}_1] \Pr[\mathcal{E}_1]}{2T^2}
\]

Let us consider these terms in turn. For the last term, we estimate:

\[
E[S_0' | \mathcal{E}_1] \Pr[\mathcal{E}_1] \geq E[S_0' | \mathcal{E}_1 \land \mathcal{E}_2] \Pr[\mathcal{E}_1 | \mathcal{E}_2] \Pr[\mathcal{E}_2] = E[S_0' | \mathcal{E}] \Pr[\mathcal{E}]
\]

Conditional on the event \(\mathcal{E}\), we have \(S_0' = S_0\) with probability one (as none of the entries in \(W_0\) are changed.) By Proposition 3.16 \(\Pr[\mathcal{E}] \geq 1/10\) so overall

\[
E[S_0' | \mathcal{E}_1] \Pr[\mathcal{E}_1] \geq S_0/10
\]

Our next step is to estimate the term \(E[S_0' - S_0]\). By Proposition 3.13 we have

\[
E[S_0'] \leq S_0 \cosh\left(\frac{12m \sum_{i \in U} |\lambda c_i (1 - c_i)/(1 + \lambda c_i)|}{T}\right) = S_0 \cosh\left(\frac{12m \sum_{i \in U} c_i}{T}\right)
\]

where \(c_i = z(W \cap G_i)\) for a group \(i \in [r]\). We then estimate:

\[
E[S_0'] \leq S_0 \cosh\left(\frac{12m \sum_{i \in U} c_i}{T}\right) \leq S_0 \cosh\left(\frac{12mu}{T}\right)
\]

By our assumption that \(T \geq t \geq 5000m^2(u + 1)^2\), this is at most \(S_0(1 + (12mu/T)^2)\), and so

\[
E[S_0'] - S_0 \leq 12S_0(12mu/T)^2
\]

Finally, we turn to estimating \(E[S' - S]\). Proposition 5.13 gives:

\[
E[S'] \leq S \cosh\left(\frac{12m \sum_i |\lambda c_i (1 - c_i)/(1 + \lambda c_i)|}{T}\right) = S \cosh\left(\frac{12m \sum_i c_i}{T}\right)
\]

\[
\leq S \cosh\left(\frac{12m \sum_{i \in U} c_i + 12m \sum_{i \in [r] - U} c_i}{T}\right)
\]

\[
\leq S \cosh\left(\frac{12m \sum_{i \in U} 1 - 12m \sum_{i \in [r] - U} \ln(1 - c_i)}{T}\right)
\]

\[
\leq S \cosh\left(\frac{12m(u - \ln S_1)}{T}\right)
\]

Therefore, as \(S = S_0S_1\), we have

\[
E[S'] - S \leq S_0\left(\frac{12m(u - \ln S_1)}{T}\right) - S_1
\]

Observe that \(12m/T \leq 12m/t \leq 1/(u + 1)\). Therefore, by Proposition A.3, we have

\[
S_1 \cosh\left(\frac{12m(u - \ln S_1)}{T}\right) - S_1 \leq (12m(u + 1)/T)^2
\]

Substituting all these estimates, we see that

\[
E[\Phi(z')] - \Phi(z) \leq 144S_0m^2(u + 1)^2/T^2 + 12aS_0m^2u^2/T^2(1/t - 1/T) - aS_0 \times (20T^2)
\]

\[
= \frac{S_0m^2}{T^2}(144(u + 1)^2 + 12au^2/t - \frac{a}{20m^2})
\]

\[
\leq \frac{S_0m^2}{T^2}(144(u + 1)^2 + \frac{12a(u + 1)^2}{5000m^2(u + 1)^2} - \frac{a}{20m^2}) \leq \frac{S_0m^2}{T^2}(144(u + 1)^2 - 0.04a/m^2)
\]

This quantity is negative for \(a = 8000m^2(u + 1)^2\). \(\blacksquare\)
Proof of Theorem 3.15. Let $y'' = \text{INTRAGROUPREDUCE}$. Since $E[Q(W, y)] = E[Q(W, y'')]$, we can see that

$$E[\Phi(y'')] \leq E[Q(W, y'')] + aE[Q(W_0, y'')] = Q(W, y) + aQ(W_0, y)/t$$

for $a = 8000n^2u^2$.

By induction on all iterations of BSR, Proposition 3.17 shows that we have

$$E[\Phi(y') \mid y''] \leq E[\Phi(y'')]$$

Finally, at the termination of BSR, we have $T(y) \leq t$. Therefore, if $y'$ is the output of BSR, we have

$$\Phi(y') = Q(W, y') + aQ(W_0, y')(1/t - 1/\max(t, T(y))) = Q(W, y')$$

This shows that

$$E[Q(W, y')] \leq Q(W, y) + aQ(W_0, y)/t = Q(w, y) + Q(W_0, y)O(m^2(u + 1)^2/t)$$

as desired.

4 The multi-knapsack center problem

For our first application of BSR, we will consider the multi-knapsack center problem, proving Theorem 2.5 and Theorem 2.6. These are not our most important applications, but they use BSR in a relatively straightforward way and only require the simplest correlation bound. Later sections will require significantly more background on facility location algorithms. Thus, the knapsack center problem serves as a good warm-up.

Suppose $R$ is the optimal radius. Note that there are only $\binom{n}{2}$ possible values for the optimal radius $R$. Thus, we can guess this value in $O(n^2)$ time. For an instance $I$, we define the polytope $P(I, R)$ containing points $(x, y)$ which satisfy the following constraints (C1) – (C6). Given a solution to this LP, our goal is to (randomly) round $y$ to an integral solution.

(C1) $\sum_{i \in F : d(i, j) \leq R} x_{ij} = 1$ for all $j \in C$ (all clients should get connected to some open facility),

(C2) $x_{ij} \leq y_i$ for all $i, j \in C$ (client $j$ can only connect to facility $i$ if it is open),

(C3) $My \leq \vec{B}$ (the $m$ knapsack constraints),

(C4) $0 \leq x_{ij}, y_i \leq 1$ for all $i, j \in C$.

By splitting facilities, and removing facilities with $y_i \in \{0, 1\}$, we may enforce two additional constraints:

(C5) For all $i \in F, j \in C$, we have $x_{ij} \in \{0, y_i\}$,

(C6) Every $i \in F$ has $y_i \in (0, 1)$.

For any client $j \in C$, define $F_j := \{i \in F : x_{ij} > 0\}$; we refer to these sets as clusters. Property (C5) ensures that $y(F_j) = 1$. We may form a subset $C' \subseteq C$, such that the clusters $F_j, F_{j'}$ for $j, j' \in C'$ are pairwise disjoint, and such that $C'$ is maximal with this property. We also define

$$F_0 = \{i \in F \mid i \notin \bigcup_{j \in C'} F_j\}.$$
We turn this into a BSL instance using the following set of groups. First, for each \( j \in C' \), we define the group \( G_j \) to be simply \( F_j \). Next, for each \( i \in F_0 \), we create a group \( G_i \) which consists of two items, namely \( i \) and a dummy item \( \text{Dummy}(i) \), which has \( y(\text{Dummy}(i)) = 1 - y_i \), which has distance \( \infty \) from all \( i \in F \), and which has cost zero according to all \( m \) knapsack constraints. (If we select this dummy item, it simply means that we do not choose to open facility \( i \)). After these dummy items have been added, we see that \( y(G_i) = 1 \) for each group \( \ell \), and the groups form a partition of \( F \). Let us define the set \( L = \{ j \in C' \} \cup \{ i \in F_0 \} \), so that every \( \ell \in L \) corresponds to a distinct group \( G_\ell \).

### 4.1 Proof of Theorem 2.5

Now we show that, when \( m = 1 \) (the standard Knapsack Center problem), one can satisfy the knapsack constraint with no violation while guaranteeing that the expected approximation ratio of at least \((1 - \delta)n\) vertices is at most \(1 + 2/e + \gamma\) for any \( \gamma > 0 \). The overall strategy is to first execute a preprocessing step to generate a large number of fractional solution in which each facility \( i \) with \( y_i > 0 \) will only serve (fractionally) at most \( en \) other clients. At least one of these solutions will also have value at most \( \text{OPT} \).

Next, we use the algorithm BSR to convert the fractional vector \( y \) into a vector \( y' \), which contains \( O(1) \) remaining fractional entries. Finally, in each remaining unrounded group, we open the facility with smaller weight.

We let \( M \) denote the weight function; since we are considering here the case that \( m = 1 \), we view \( M \) as an \( n \)-long vector where \( M_i \) is the weight of facility \( i \in F \).

**Proposition 4.1.** There is an algorithm running in \( n^{O(1/e)} \) time, which returns a fractional solution \((x, y)\) with the property that for every facility \( i \in F \) there are at most \( \epsilon n \) clients \( j \in C \) with \( x_{ij} > 0 \).

*Proof.* For any facility \( i \), define \( U_i = \{ j \in C \mid x_{ij} > 0 \} \). Consider solving the LP to obtain a fractional solution \((x, y)\). Now, suppose that this contains some facility \( i \in F \) such that \( y_i \in (0, 1) \) and \( |U_i| > \epsilon n \). We guess whether to include \( i \) in the optimal solution \( S \); that is, we consider two subproblems in which we force \( y_i = 0 \) and \( y_i = 1 \). In the latter problem instance, we may also set \( x_{ij} = 1 \) for \( j \in U_i \).

Since in the optimal integral solution, the values of \( x, y \) are integral, this branching process will generate at least one subproblem which is feasible. Furthermore, each time we execute a branch of this, the resulting subproblem gains at least \( \epsilon n \) facilities \( j \) with \( x_{ij} \in \{0, 1\} \); so, the depth of the search tree can be at most \( 1/e \). Since each subproblem can be solved in \( \text{poly}(n) \) time, the overall runtime is \( n^{O(1/e)} \).

Finally, at a leaf of this branching process, we must \( |U_i| \leq \epsilon n \) for every fractional facility \( i \in F \).

We may now state our rounding algorithm:

**Proof of Theorem 2.5** Feasibility of \( S \): The fractional solution \((x, y)\) satisfies \( My \leq 1 \). Proposition 3.8 ensures that the vector \( y' \) satisfies \( My' = My \). Lines (3) – (5) can only decrease \( M'y \). Also, \( |\text{frac}(G_i, y')| \leq 2 \) for each group \( i \); thus, at the end of procedure \text{STANDARD-KNAPSACK-DEP-ROUND}, the final vector \( y' \) is fully integral. Consequently, at the termination of the algorithm, we have \( \sum_{i \in S} M_i \leq 1 \) and every group \( j \in L \) has \( |G_j \cap S| = 1 \).

Cost analysis: Let us say that a facility \( i \) with \( y_i \in (0, 1) \) is modified if \( i = s_{j1} \) or \( i = s_{j2} \) for some group \( j \in L \). For each client \( j \), let \( q_j \) denote the probability that there is a modified facility in \( F_j \). Since \( y' \) contains at most \( t \) fractional groups, there are at most \( 2t \) modified facilities. So

\[
\sum_{j \in C} q_j \leq \sum_{j \in C} \sum_{k \in F_j} \Pr[k \text{ is fractional}] = \sum_{k \in C} |F_k| \Pr[k \text{ is fractional}]
\leq (\epsilon n) \sum_{k \in C} \Pr[k \text{ is fractional}] \leq (2t)\epsilon n \quad \text{(by step (1))}
\]
Algorithm 5 STANDARDKNApsackDepRound \((M, V, t)\)

1: Using Proposition 4.1 find a fractional solution \((x, y)\) satisfying (C1)–(C5), for which \(|\{j \in C \mid x_{ij} > 0\}| \leq \epsilon n\) for all \(i \in \mathcal{F}\).
2: Execute \(y' = \text{BSR}(G_j, M, y, t)\)
3: Initialize \(S = \emptyset\)
4: for each \(j \in L\) do
5: \hspace{1cm} if \(\text{frac}(G_j, y') = 2\) then
6: \hspace{2cm} Let \(\text{frac}(G_j, y') = \{s_{j1}, s_{j2}\}\) with \(M s_{j1} < M s_{j2}\)
7: \hspace{2cm} Update \(S \leftarrow S \cup \{s_{j1}\}\)
8: \hspace{1cm} else
9: \hspace{2cm} Let \(s_j \in G_j\) be the unique facility with \(y_i = 1\)
10: \hspace{2cm} Update \(S \leftarrow S \cup \{s_j\}\)
11: return \(S\).

We say that a client \(j\) is good if \(q_j \leq 1/t\) and bad otherwise. There are at most \(2t^2 \epsilon n\) bad clients. Let \(U\) be the set of good clients, and consider some \(k \in U\). For each \(j \in L\) define \(C_j = F_k \cap G_j\). We observe that \(T_k \leq R\) if \(S \cap C_j \neq \emptyset\) for some \(j \in L\), and \(T_k \leq 3R\) otherwise. (By construction, \(F_k \cap F_j \neq \emptyset\) for some \(j \in C'\) and we always open one center inside \(G_j\). Then the distance from \(k\) to this center is at most \(d(k, j) + R \leq 3R\) by triangle inequality.) Therefore,

\[
T_k \leq R \left(1 + 2\left[\bigwedge_{j \in L} S \cap C_j = \emptyset\right]\right)
\]

If \(F_j\) contains no modified facilities, then we have \(\bigwedge_{j \in L} S \cap C_j = \emptyset\) \(= Q(F_k, y')\). By Corollary 2.1 for \(t\) sufficiently large we have \(E[Q(F_k, y')] \leq Q(F_k, y) + O(m^2/t) = Q(F_k, y) + O(1/t)\). Since \(k\) is good, we have \(q_k \leq 1/t\) and so

\[
E[T_k] \leq R \left(1 + 2q_k + 2Q(F_k, y) + O(1/t)\right) \leq R(1 + O(1/t) + 2Q(F_k, y))
\]

Since \(y(F_k) = 1\), Proposition 3.9 shows that \(Q(F_k, y) \leq 1/e\) and therefore

\[
E[T_k] \leq (1 + 2/e + O(1/t))R
\]

Now for any \(\delta, \gamma > 0\), set \(t = K/\gamma\) for some sufficiently large constant \(K\) and \(\epsilon = \frac{\delta}{9\gamma}\). This ensures that there are at most \(\delta n\) bad clients and that every \(k \in U\) has \(E[T_k] \leq (1 + 2/e + \gamma)R\). The running time is \(n^{O(1/e)} = n^{O\left(\frac{1}{\pi^2}\right)}\).

\[\square\]

4.2 Proof of Theorem 2.6

When there are \(m \geq 1\) knapsack constraints, then our algorithm for knapsack center is a straightforward application of FULLBSR, as summarized as Algorithm 6.

Algorithm 6 MKC-ROUNDING \((t)\)

1: Let \(y\) be the solution to the LP. Form groups \(G_j, F_j\) as appropriate.
2: Let \(Y = \text{FULLBSR}(G_j, M, y, t)\)
3: return \(S = \{i \mid Y_i = 1\}\)

21
**Proposition 4.2.** For any client $k \in \mathcal{C}$, we have $P(S \cap F_k = \emptyset) \leq 1/e + O(m^2/t)$.

**Proof.** Note that $S \cap F_k = \emptyset$ iff $Q(F_k, Y) = 0$. By Theorem 7.12 we have $\mathbb{E}[Q(F_k, Y)] \leq Q(F_k, y) + O(m^2/t)$. Since $y$ solves the LP, we have $y(F_k) = 1$ and so Proposition 3.9 shows $Q(F_k, y) \leq 1/e$. □

**Proof of Theorem 2.6** By rescaling, it suffices to assume that $\gamma$ is sufficiently small and $\mathbb{E}[T_j] \leq (1 + 2/e + O(\gamma))R$. Let $z = cm\sqrt{\frac{\log(m/\gamma)}{\gamma}}$ for some constant $c$.

We will independently repeat MKC-ROUNDING with parameter $t = m^2/\gamma$ until $MY \leq \vec{B} + z\vec{1}$; thus it suffices to show that after a single application of MKC-ROUNDING we have

1. $MY \leq \vec{B} + z\vec{1}$ with constant probability
2. $\mathbb{E}[T_j | MY \leq \vec{B} + z\vec{1}] \leq (1 + 2/e + O(\gamma))R$.

If a facility in $F_j$ is opened then $T_j \leq R$, otherwise $T_j \leq 3R$. So, by Proposition 4.2, the expected distance is at most $(1 + 2(1/e + O(m^2/t)))R$. By Theorem 3.10 we have $MY \leq \vec{B} + z\vec{1}$ with probability at least is at least $1 - \gamma \geq \Omega(1)$. So

$$\mathbb{E}[T_j | MY \leq \vec{B} + z\vec{1}] \leq \frac{1 + 2/e + O(m^2/t)R}{1 - \gamma} \leq (1 + 2/e + O(\gamma)).$$

□

By using a strategy of guessing big items, we can leverage this additive approximation guarantee into a faster algorithm for a multiplicative approximation guarantee in arbitrary knapsacks.

**Theorem 4.3.** For any $\gamma, \epsilon \in (0, 1/2)$, there is an algorithm for the multi knapsack center problem which runs in expected time $n^{O(e^{-1} \gamma^{-1/2}m^2)}$ and returns a solution $S$ such that

1. $MY \leq (1 + \epsilon)\vec{B}$, where $Y$ is the indicator vector of $S$,
2. $\forall j \in \mathcal{C}$, $\mathbb{E}[T_j] \leq (1 + 2/e + \gamma)R$.
3. $\forall i \in \mathcal{C}$, $T_j \leq 3R$ with probability one.

**Proof.** By rescaling we may assume that $\vec{B} = \vec{1}$. Let us say that a facility $i$ is big if $M_{ki} \geq \rho$ for any value $k \in [m]$, where $\rho \in [0, 1]$ is a parameter to be determined. We begin by guessing which big facilities are in the solution set $S$; there can be at most $m/\rho$ such facilities, so this step takes $n^{O(m/\rho)}$ time. After this guessing step, we can remove all other big facilities, and solve the LP to obtain a fractional solution satisfying (C1) – (C5).

Let $B'$ denote the residual budget and $M'$ denote the residual weight vector (remaining after removing guessed big items). Every remaining facility has $M'_{ki} \leq \rho B$. Applying Theorem 2.6 to the matrix $M'/(\rho B)$ to obtain a solution $\tilde{S}$ on small facilities such that

1. $(M'/\rho)Y \leq (\vec{B}'/\rho) + z\vec{1}$ with constant probability
2. $\mathbb{E}[T_j | (M'/\rho)Y \leq (\vec{B}'/\rho) + z\vec{1}] \leq (1 + 2/e + O(\gamma))R$.

where $z = O(m\sqrt{\log(m/\gamma)})/\gamma$.

Combining this solution with the guessed big facilities, we get

$$MY \leq \vec{1} + O(\rho\sqrt{\log(m/\gamma)})/\gamma$$

In other words, we achieve our objective with $\epsilon = O(\rho\sqrt{\log(m/\gamma)})$. Setting $\rho = \frac{\epsilon\sqrt{\gamma}}{m\sqrt{\ln(m/\gamma)}}$, we thus get an overall run-time of $n^{O(m/\rho)} \leq n^{O(e^{-1} \gamma^{-1/2}m^2)}$. □
In this section, we show that one can obtain a bi-factor \((1 + \sqrt{3} + \gamma)\)-approximation algorithm for the knapsack median problem if allowed to slightly violate the budget constraint by an additive factor of \(O(1/\gamma)\). This will also yield a pseudo-approximation algorithm for general knapsack median, with a multiplicative budget violation.

This algorithm is inspired by Li & Svensson [22], who gave a \((1 + \sqrt{3} + \gamma)\)-approximation algorithm for the \(k\)-median problem. The idea is to compute a “bi-point” solution which is a convex combination of a feasible solution and another pseudo solution. Then rounding this solution will result in a solution whose cost can be bounded by \((1 + \sqrt{3} + \gamma)\) times the optimal cost but would slightly violate the knapsack constraint. The rounding step can be done using BSR and some \(O(1)\) left-over fractional variables will have to be rounded up to 1, resulting in a small violation in the total weight. In [22], the authors use a postprocessing step to correct the solution, preserving the cardinality constraint exactly. However, this step does not seem to work for the knapsack median problem.

We also note that there have been a number of true approximation algorithms for knapsack median [4, 8, 20]; the current best approximation ratio is 7.08 due to Krishnaswamy et al. [20].

For consistency with the rest of our paper, we denote the weight of a facility \(i \in F\) by \(M_i\). We define the cost of a facility set \(S \subseteq F\) by

\[
\text{cost}(S) = \sum_{j \in C} d(j, S)
\]

We begin by computing the so-called bi-point solution and stars as in [22].

**Theorem 5.1.** There is a polynomial-time algorithm to compute two sets \(F_1, F_2 \subseteq F\) and constants \(a, b \geq 0\) and \(a + b = 1\) such that \(M(F_1) \leq B \leq M(F_2), a \cdot M(F_1) + b \cdot M(F_2) \leq B\) and \(a \cdot \text{cost}(F_1) + b \cdot \text{cost}(F_2) \leq 2OPT_T\). (The pair \((F_1, F_2)\) is called a bi-point solution of \(T\).)

**Proof.** The algorithm of [16, 25] can be easily adapted to the weighted case. \(\square\)

For each client \(j \in C\), let \(i_1(j), i_2(j)\) denote the closest facilities to \(j\) in \(F_1\) and \(F_2\), respectively. Let \(d_1(j) = d(j, i_1(j))\) and \(d_2(j) = d(j, i_2(j))\). Finally, define \(d_1 := \sum_{j \in C} d_1(j) = \text{cost}(F_1)\) and \(d_2 := \sum_{j \in C} d_2(j) = \text{cost}(F_2)\). Note that \(F_1\) is a feasible solution. If \(d_1 \leq d_2\), then \(F_1\) is a feasible solution of value of at most 2OPT and we would be done. Thus, we assume that for the remainder that \(d_1 \geq d_2\).

For \(i \in F_2\) we define \(\sigma(i)\) to be the closest facility of \(F_1\). For each facility \(i \in F_1\) we define the star \(S_i\) as the set of facilities \(k \in F_2\) with \(\sigma(k) = i\).

We will apply dependent rounding such that (i) each facility in \(F_1\) is open with probability \(\approx a\), (ii) each facility in \(F_2\) is open with probability \(\approx b\), and (iii) the budget is not violated by too much. The intent is that for each \(i \in F_1\), with probability \(a\) we open \(i\) and with the complementary probability \(b\) we open all the facilities \(k \in S_i\). (For a few exceptional stars, we may open \(i\) as well as some subset of the facilities \(k \in S_i\).) The full details are spelled out in Algorithm 7.

By contrast, if we used independent rounding, then in order to ensure \(MY \leq (1 + \epsilon)\Bar{B}\) we would need \(n^{O(m \log (m/\gamma) \epsilon^{-2})}\) time, which has a significantly worse dependence upon \(\epsilon\).
Algorithm 7 ROUNDSTARS(\(t\))

1: Initialize the vectors \(x, c\) by \(x_i = b\) and \(c_i = M(S_i) - M_i\) for all \(i \in \mathcal{F}_1\)
2: \(y \leftarrow \text{BSR-DepRound}(x, c, t)\)
3: Define the vector \(z \in [0,1]^{\mathcal{F}_2}\), by setting 
   \[z_i = y_{\sigma(i)}\]
4: Set \(Z \leftarrow \text{FULL-BSR-DepRound}(z, M, 1)\)
5: return 
   \[S = \{i \in \mathcal{F}_1 \mid y_i < 1\} \cup \{i \in \mathcal{F}_2 \mid Z_i = 1\}\]

Proposition 5.2. For a bounded knapsack, we have \(M_S \leq B + t + 1\) with probability one.

Proof. Let us first compute \(M(\mathcal{F}_1 \cap S)\); we have 
\[M(\mathcal{F}_1 \cap S) = \sum_{k \in \mathcal{F}_1} M_k + \sum_{k \in \mathcal{F}_1, y_k \in (0,1)} M_k\]
The second term contributes at most \(t\), since \(y\) has at most \(t\) fractional entries.
Since \(Z\) has at most 1 rounded entry, \(M(\mathcal{F}_2 \cap S) = MZ \leq Mz + 1\). We compute \(Mz\) by
\[Mz = \sum_{k \in \mathcal{F}_1} \sum_{i \in S_k} M_i z_i = \sum_{k \in \mathcal{F}_1} y_k \sum_{i \in S_k} M_i = \sum_{k \in \mathcal{F}_1} y_k M(S_k)\]
Combining all these contributions gives 
\[M(S) \leq (Mz + \sum_{k \in \mathcal{F}_1, y_k = 0} M_k) + \sum_{k \in \mathcal{F}_1, y_k \in (0,1)} M_k + 1\]
\[= \sum_{k \in \mathcal{F}_1} (y_k M(S_k) + [y_k = 0] M_k) + t + 1\]
\[\leq \sum_{k \in \mathcal{F}_1} (M(S_k)y_k + M_k(1 - y_k)) + t + 1\]
\[\leq \sum_{k \in \mathcal{F}_1} (x_k M(S_k) - M_k) + M(\mathcal{F}_1) + t + 1\] since BSR preserves \(\sum_i y_i c_i = \sum_i x_i c_i\)
\[= bM(\mathcal{F}_2) - bM(\mathcal{F}_1) + M(\mathcal{F}_1) + t + 1 \leq B + t + 1\]

Now let us analyze the connection cost of some client \(j\).

Proposition 5.3. For client \(j\), Algorithm 7 yields
\[\Pr[i_1(j) \notin S] \leq 1 - b\]
\[\Pr[i_2(j) \notin S] \leq b\]
\[\Pr[i_1(j) \notin S \land i_2(j) \notin S] \leq b(1 - b)(1 + O(1/t))\]

Proof. Let us define \(E_1, E_2\) to be the events \(i_1(j) \in S, i_2(j) \in S\), respectively. Event \(E_1\) occurs iff \(y_i \neq 1\). Since \(E[y_i] = x_i = b\), we have
\[\Pr[E_1] = \Pr[y_i = 1] \leq E[y_i] = x_i = b\]
Next, note that if we condition on the vector $y$, we have
\[
\Pr[\tilde{E}_2 \mid y] = \Pr[Z_{i_2(j)} = 0 \mid \sigma] = (1 - z_{i_2(j)}) = (1 - y_k)
\]
where $k = \sigma(i_1(j))$.

Integrating over $z$ therefore gives
\[
\Pr[\tilde{E}_2] = \mathbb{E}[1 - y_k] = 1 - x_k = 1 - b
\]

Finally, suppose we condition on the vector $y$; then, in order for $\tilde{E}_1$ and $\tilde{E}_2$ to occur, it must be that $y_i = 1$ and $Z_{i_2(j)} = 0$. Since $\mathbb{E}[Z_{i_2(j)} \mid y] = y_{i_2(j)}$ and $Z$ is completely integral, we have
\[
\Pr[\tilde{E}_1 \wedge \tilde{E}_2 \mid y] \leq [y_i = 1](1 - z_{i_2(j)}) = [y_i = 1](1 - y_k)
\]
If $i = k$, this event will be impossible. Otherwise, we have
\[
\Pr[\tilde{E}_1 \wedge \tilde{E}_2] \leq \mathbb{E}[y_i(1 - y_k)]
\]
This event depends only on two groups, and thus Theorem 3.15 shows that we get
\[
\mathbb{E}[y_i(1 - y_k)] \leq x_i(1 - x_k) + x_i(1 - x_k) \times O(m(2 + 1)^2/t) = x_i(1 - x_k)(1 + O(1/t))
\]

\[\square\]

**Proposition 5.4.** For client $j$ we have
\[
\mathbb{E}[d(j, S)] \leq d_2(j) + (d_1(j) - d_2(j))(1 - b) + 2d_2(j)b(1 - b)(1 + O(1/t))
\]

**Proof.** If $i_2(j) \in S$ then we have $\text{cost}(j) \leq d_2(j)$. Likewise if $i_1(j) \in S$ then $\text{cost}(j) \leq d_1(j)$. If neither holds, then necessarily facility $\sigma(i_2(j))$ is open. Then $\text{cost}(j) \leq d(j, i_2(j)) + d(i_2(j), \sigma(i_2(j)))$. Since $i_2(j)$ connected to the nearest facility in $F_1$, we must have $d(i_2(j), \sigma(i_2(j))) \leq d(i_2(j), i_1(j))$. In the last case, the triangle inequality gives
\[
d(j, S) \leq d(j, i_2(j)) + d(i_2(j), i_1(j)) \leq d(j, i_2(j)) + d(i_2(j), j) + d(j, i_1(j)) = 2d_2(j) + d_1(j)
\]
Putting these together gives
\[
\mathbb{E}[d(j, S)] \leq \Pr[i_2(j) \in S]d_2(j) + \Pr[i_1(j) \in S \wedge i_2(j) \notin S]d_1(j) + \Pr[i_1(j) \notin S \wedge i_2(j) \notin S](2d_2(j) + d_1(j))
\]
\[
= d_2(j) + (d_1(j) - d_2(j)) \Pr[i_2(j) \notin S] + 2 \Pr[i_1(j) \notin S \wedge i_2(j) \notin S]d_2(j)
\]
By Proposition 5.3 we have $\Pr[i_1(j) \notin S \wedge i_2(j) \notin S] \leq b(1 - b)(1 + O(1/t))$ concluding the proof. \[\square\]

We are now ready to combine our algorithms to get an overall pseudo-approximation approximation.

**Theorem 5.5.** For a bounded knapsack, there is a randomized algorithm in expected time $\text{poly}(n)/\gamma$ generating a solution $S$ such that
\[
\text{cost}(S) \leq (1 + \sqrt{3} + \gamma) \cdot \text{OPT}
\]
\[
M(S) \leq B + O(1/\gamma)
\]
Proof. Our algorithm will be to use Algorithm\cite{7} and output either the solution \( S \) returned by Algorithm\cite{7} or the feasible solution \( F_i \) (whichever has least cost). By Proposition\cite{5.2} and using the fact that \( M_i \leq 1 \) for every facility \( i \), the solution \( S \) satisfies \( M(S) \leq B + t + 1 \).

By Proposition\cite{5.4} the solution \( S \) of Algorithm\cite{7} satisfies
\[
\mathbb{E}[\text{cost}(S)] \leq \sum_{j \in C} d_2(j) + (d_1(j) - d_2(j))(1 - b) + 2d_2(j) \min(b, 1 - b, b(1 - b) + O(1/t))
\]
\[
= d_2 + (d_1 - d_2)(1 - b) + 2d_2b(1 - b)(1 + O(1/t))
\]
So overall we have
\[
\mathbb{E}[\text{cost}(A)] \leq \min\{d_1, d_2 + (d_1 - d_2)(1 - b) + 2b(1 - b)(1 + O(1/t))d_2\}
\leq (1 + O(1/t)) \min\{d_1, d_2 + (d_1 - d_2)(1 - b) + 2b(1 - b)d_2\}
\]
\[
= (1 + O(1/t))(ad_1 + bd_2) \min \left\{ \frac{d_1}{ad_1 + bd_2}, \frac{d_2 + (d_1 - d_2)(1 - b) + 2b(1 - b)d_2}{ad_1 + bd_2} \right\}
\]
\[
\leq (1 + O(1/t)) \cdot (2OPT_\gamma) \cdot C_0,
\]
where
\[
C_0 = \frac{\min\{d_1, d_2 + (d_1 - d_2)(1 - b) + 2b(1 - b)d_2\}}{(1 - b)d_1 + bd_2}
\leq \max_{r \geq 0} \frac{\min\{1, r + (1 - r)(1 - b) + 2b(1 - b)r\}}{1 - b + br} \leq \frac{1 + \sqrt{3}}{2}.
\]
Therefore, we get
\[
\mathbb{E}[\text{cost}(S)] \leq (1 + O(1/t)) \cdot (1 + \sqrt{3}) \cdot OPT \leq (1 + O(1/t)) \cdot OPT_\gamma.
\]

Set \( t = \Omega(1/\gamma) \), and run this process until it generates a solution which gives a cost of at most \((1 + 10/\gamma)\). Markov’s inequality now shows this succeeds after an expected \( O(1/\gamma) \) repetitions.

Theorem\cite{5.5} provides an approximation ratio with an additive loss for bounded knapsacks; we can leverage this to obtain a multiplicative loss for arbitrary knapsacks.

**Theorem 5.6.*** There is a randomized algorithm running in expected time \( n^{O(1/\gamma - 1)} \) for arbitrary knapsack generating a solution \( S \) such that
\[
\mathbb{E}[\text{cost}(S)] \leq (1 + \sqrt{3} + \gamma) \cdot OPT
\]
\[
M(S) \leq B(1 + \epsilon)
\]

Proof. By rescaling assume without loss of generality that \( B = 1 \). Call a facility big if \( M_i \geq \rho \), otherwise it is small; here \( \rho \) is a parameter to be determined. The solution may have at most \( 1/\rho \) big facilities; these may be guessed in \( n^{O(1/\rho)} \) time.

Let \( S_{\text{big}} \) denote the (guessed) set of big facilities. Construct a residual instance in which (i) all big facilities outside \( S_{\text{big}} \) are removed and (ii) the weights of facilities in \( S_{\text{big}} \) are set to zero.

We now divide all the remaining weights (which are in the range \([0, \rho]\)) by \( \rho \), and we also remove from the budget the weights of \( S_{\text{big}} \). Apply Theorem\cite{5.5} to this residual instance; after rescaling, we can see that the resulting solution \( S \) satisfies
\[
\mathbb{E}[\text{cost}(S)] \leq (1 + \sqrt{3} + \gamma) \cdot OPT,
\]
\[
M(S) \leq 1 + O(\rho/\gamma)
\]
Setting \( \rho = \epsilon^{-1}\gamma^{-1} \) therefore ensures that \( M(S) \leq 1 + \epsilon \). The overall runtime is \( n^{O(1/\rho)} \times n^{O(1)/\gamma} = n^{O(\epsilon^{-1}\gamma^{-1})} \).

\[\square\]
6 Approximation algorithm for multi knapsack median

In this section, we give a 3.25-pseudo-approximation algorithm for Multi-knapsack-median problem, with a small violation in the budget constraint. This algorithm is inspired by the work of Charikar & Li [8], which gave a 3.25-approximation algorithm for k-median. Although the approximation ratio of that algorithm is worse than other approximation algorithms, its main benefit is that it gives a good approximation ratio as a function of the optimal LP solution. This makes it much more flexible, including allowing the use of BSR for certain key rounding steps.

6.1 The Charikar-Li algorithm

Let us begin by stating the LP relaxation, which is the “obvious” one:

\[
\begin{align*}
\text{minimize} & \quad \sum_{j \in C} \sum_{i \in F} x_{i,j}d(i,j) \\
\text{subject to} & \quad \sum_{i \in F} x_{i,j} = 1 \quad \forall j \in C \\
& \quad 0 \leq x_{i,j} \leq y_i \leq 1 \quad \forall i \in C, j \in F \\
& \quad \sum_{i \in F} M_{k,i}y_i \leq 1 \quad \forall i \in F, k \in [m]
\end{align*}
\]

Here, \(x_{i,j}\) represents fractionally how client \(j\) is matched to facility \(i\), and \(y_i\) is an indicator that facility \(i\) is open. The final constraint specifies that each of the \(m\) knapsack constraints is satisfied.

The Charikar-Li algorithm has two phases, which we briefly summarize here; please see [8] for further details. The first, or bundling phase, can be divided into three stages:

1. A client set \(C' \subseteq C\) is chosen, such that the clients \(j, j' \in C'\) are relatively far apart from each other.
2. For each \(j \in C'\), a facility set \(U_j\) is formed, which is the set of facilities “claimed by” \(j\). The sets \(U_j\) are chosen to be disjoint, and they have the property that \(1/2 \leq y(U_j) \leq 1\).
3. A matching \(M\) is constructed over \(C'\); the intent is that if \((j, j')\) is an edge of the matching, then we will be guaranteed to open at least one facility in \(U_j \cup U_{j'}\).

Charikar-Li also uses a standard facility-splitting step, so that \(x_{i,j} \in \{0, y_i\}\), and facilities with \(y_i = 0\) are removed. Furthermore, it ensures that for \(j \in C'\) and \(i \in U_j\) we have \(x_{i,j} = y_i\).

At the end of this process, there can be at most one client \(j \in C'\) which is not matched in \(M\). This client will be dealt with in a manner similar to the other, matched, clients. To streamline our discussion, we will ignore this one extraneous client. For the remainder of this section, unless stated otherwise, we assume that every client of \(C'\) appears in the matching \(M\); equivalently, we will assume that \(|C'|\) is even.

In the second, or selection phase of the Charikar-Li algorithm, we select for each edge \((j, j') \in M'\), either one or two facilities to open from \(U_j \cup U_{j'}\). The simplest strategy for the selection phase, which will be a baseline for our more advanced algorithms, is referred to as independent selection. This selects a set \(C'' \subseteq C'\); then, for each \(j \in C''\), we will open exactly one facility in \(U_j\). More specifically:

1. For each edge \((j, j') \in M:\)
   (a) Place \(j\) into \(C''\) with probability \(1 - y(U_{j'})\)
   (b) Place \(j'\) into \(C''\) with probability \(1 - y(U_j)\);
   (c) Place both \(j, j'\) into \(C''\) with probability \(y(U_j) + y(U_{j'}) - 1\).
2. For each \( j \in C'' \): select one facility \( X_j \in U_j \), wherein we set \( X_j = i \) with probability \( y_i / y(U_j) \).

3. Set \( S = \{ X_j \mid j \in C'' \} \)

For each facility \( i \), let \( Y_i \) be the event that \( i \in S \). For each client \( j \), let \( r_j := \sum_i x_{ij}d(i, j) \) denote the fractional connection cost of \( j \). Charikar & Li shows that independent selection has a number of useful probabilistic properties:

**Proposition 6.1 (8).** Under independent selection, the following bounds hold:

1. \( \Pr(Y_i) \leq y_i \) for any facility \( i \).

2. The events \( Y_i \), for \( i \in \mathcal{F} \), are all negatively correlated.

3. For any client \( j \in C \) we have \( \mathbb{E}[d(j, S)] \leq 3.25r_j \).

Independent selection would thus give a simple bi-factor approximation algorithm:

**Theorem 6.2.** Let \( \gamma, \epsilon \in (0, 1) \). There is an algorithm running in expected time \( nO(m \log(m/\gamma)/\epsilon^2) \), with the following guarantee. The algorithm returns a facility set \( S \) such that

\[
\sum_{j \in C} d(j, S) \leq (3.25 + \gamma)OPT
\]

\[
\sum_{i \in S} M_{k,i} \leq (1 + \epsilon)B_k \quad \text{for } k = 1, \ldots, m
\]

**Proof (Sketch).** We say a facility \( i \in \mathcal{F} \) is big if \( M_{k,i} \geq \rho \) for any \( k \in [m] \). We begin by guessing the set of opened big facilities; this takes time \( nO(m/\rho) \). Next, solve the LP and run the Charikar-Li rounding on the resulting residual instance. Because we have guessed the big facilities, the remaining facilities have \( M_{k,i} \leq \rho \). By Proposition 6.1, each term \( \sum_i M_{k,i}Y_i \) is a sum of negatively-correlated random variables with mean at most \( \sum_i M_{k,i}y_i \leq 1 \). By Chernoff’s bound, the probability it exceeds \( 1 + \epsilon \) is at most \( e^{-\epsilon^2/(3\rho)} \). Thus, the overall probability that weight constraint is violated, is at most \( me^{-\epsilon^2/(3\rho)} \). Setting \( \rho = \frac{\epsilon^2}{10 \log(m/\gamma)} \) ensures that this is at most \( O(\gamma) \).

\[ \square \]

### 6.2 Facility selection as a block-selection process

There is an alternate way of viewing independent selection, which will make clear the connection to BSL. Consider adding an additional dummy facility to \( \mathcal{F} \), which we denote \( 0 = 0 \), with \( w_0 = 0 \), and setting \( \mathcal{H}' = \mathcal{F} \times \mathcal{F} \). For each edge \( e = (j, j') \in \mathcal{M} \), we create a set \( G_e \subseteq \mathcal{H}' \); we refer to this as the **pair-group for** \( e \).

This is defined by

\[
G_e = \{ (i, 0) \mid i \in U_j \} \cup \{ (0, i') \mid i' \in U_{j'} \} \cup \{ (i, i') \mid i \in U_j, i' \in U_{j'} \}
\]

For each edge \( e = (j, j') \in \mathcal{M} \), we now define a vector \( z_{e,v} \), where \( v \) ranges over \( G_e \), as follows:

\[
z_{e,(i,0)} = (1 - y(U_j))\frac{y_i}{y(U_j)}
\]

\[
z_{e,(0,i')} = (1 - y(U_j))\frac{y_{i'}}{y(U_{j'})}
\]

\[
z_{e,(i,i')} = (y(U_j) + y(U_{j'}) - 1)\frac{y_iy_{i'}}{y(U_j)y(U_{j'})}
\]
Let us define \( \mathcal{H} = \bigcup_{e \in \mathcal{M}} G_e \). We can now re-cast the independent selection phase of Charikar-Li as a block-selection process over the ground-set \( \mathcal{H} \). We need to select exactly one pair \( V_e \in G_e \) for each \( e \in \mathcal{M} \). We do this by setting \( \Pr(V_e = v) = z_{e,v} \); this is a valid probability distribution as \( z_{e,v} \in [0,1] \) and \( \sum_v z_{e,v} = 1 \).

Next, for each edge \( e \in \mathcal{M} \), if \( V_e = (i,i') \) then we will add \( i,i' \) to \( S \). (Except, of course, if \( i = 0 \) or \( i' = 0 \).)

Thus, the selection phase of Charikar-Li is nothing more than applying independent selection to the corresponding block-selection process (drawing exactly one item from each set \( G_e \)).

Let us define \( \mathcal{V} = \{ V_e \mid e \in \mathcal{M} \} \). We say that \( V_e \) is opened, to mean that \( i,i' \) are added to \( S \) if \( V_e = (i,i') \). We also extend the weighting function \( M \) to \( \mathcal{H} \) by setting

\[
M_{k,(i,i')} = M_{k,i} + M_{k,i'}
\]

bearing in mind that \( M_{k,0} = 0 \) for all \( k \). With this notation, one may easily verify the following:

**Observation 6.3.** For each \( k \in [m] \), we have \( \sum_e \sum_v M_{k,v} z_{e,v} \leq \sum_i M_{k,i} y_i \leq 1 \).

Also, with probability one, we have \( \sum_{i\in \mathcal{S}} M_{k,i} = \sum_{v\in \mathcal{V}} M_{k,v} \).

We next show how to connect the behavior of \( \mathcal{V} \) to that of \( \mathcal{S} \). For any set \( W \subseteq \mathcal{F} \), define \( \overline{W} \subseteq \mathcal{H} \) by

\[
\overline{W} = \{(i,i') \in \mathcal{H} \mid i \in W \text{ or } i' \in W\}
\]

The following bound is useful in connecting the behavior of the \( \mathcal{V} \) to that of \( \mathcal{S} \).

**Proposition 6.4.** Let \( W \subseteq \mathcal{F} \). Then \( \mathcal{S} \cap \overline{W} = \emptyset \iff \mathcal{V} \cap \overline{W} = \emptyset \).

**Proof.** Suppose that \( i \in \mathcal{S} \cap \overline{W} \). Then we must have \( V_e = (i,i') \) or \( V_e = (i',i) \) for some \( e \in \mathcal{M} \); say for simplicity the first one. Then \( (i,i') \in \overline{W} \) and so \( \mathcal{V} \cap \overline{W} \ni (i,i') \).

Suppose that \( v \in \mathcal{V} \cap \overline{W} \). Then \( v = (i,i') \) for \( i \in W \) or \( i' \in W \); say for simplicity the first. Then \( i \) will be placed into \( \mathcal{S} \) and so \( \mathcal{S} \cap \overline{W} \ni i \).

### 6.3 BSR selection strategy

We now discuss how to use BSR instead of independent selection in the Charikar-Li algorithm. We summarize the algorithm here:

**Algorithm 8 KnapsackMedianRounding \((t,\rho)\)**

1. Let \( x, y \) be the solution to the LP.
2. Run the Charikar-Li bundling phase, resulting in fractional vector \( z \in [0,1]^{\mathcal{H}} \) and groups \( G_e \subseteq \mathcal{H} \).
3. Let \( Z = \text{FULLBSR}(G_j, M, z, t) \).
4. \textbf{return} \( S = \{ i \in \mathcal{F} \mid \bigvee_{i' \in \mathcal{F}} Z(i,i') = 1 \} \).

We mention a few notations for this section. For any \( j \in \mathcal{C} \) and value \( x \geq 0 \), let us define the \textit{(open) facility-ball} for \( j \) by

\[
B(j, x) = \{ i \in \mathcal{F} \mid d(i, j) < x \}
\]

Since we fix the vector \( z \) throughout, we will use the short-hand

\[
Q(W) = Q_{-1}(W, z)
\]
for any \( W \subseteq \mathcal{H} \). If \( W \subseteq \mathcal{F} \), then we likewise define

\[
Q(W) = Q^{-1}(\mathcal{W}, z)
\]

Likewise, for \( W \subseteq \mathcal{H} \) or \( W \subseteq \mathcal{F} \) we define

\[
Q'(W) = \mathbb{E}[Q_{\lambda}(W, Z)], Q'(W) = \mathbb{E}[Q_{\lambda}(\mathcal{W}, Z)]
\]

For any \( j \in \mathcal{C}' \), we define \( R_j = \frac{1}{2}d(j, \mathcal{C}' - j) \). This quantity plays an important role in the behavior of the Charikar-Li algorithm. We quote the following critical results from [8]:

**Proposition 6.5** ([8]). Let \( j \in \mathcal{C} \), and let \( j_1 \) be its closest facility in \( \mathcal{C}' \). Then:

1. \( r_{j_1} \leq r_j \).
2. \( d(j, j_1) \leq 4r_j \).
3. With probability one, there is some \( i \in \mathcal{S} \) with \( d(i, j) \leq \beta R_j \), where \( \beta \) is some universal constant.

**Proposition 6.6.** For \( j \in \mathcal{C}' \), we have \( y(U_j) \geq 1 - \frac{r_j}{R_j} \).

**Proof.** All the facilities within distance \( R_j \) will be claimed by \( U_j \). So

\[
y(U_j) \geq \sum_{i \in \mathcal{B}(j, R_j)} y_i \geq \sum_{i \in \mathcal{B}(j, R_j)} x_{i,j}
\]

Since \( \sum_i x_{i,j} = 1 \), we therefore have

\[
1 - y(U_j) \leq \sum_{d(i,j) > R_j} x_{i,j} \leq \sum_{d(i,j) > R_j} \frac{d_{i,j}}{R_j} x_{i,j} \leq \frac{r_j}{R_j}
\]

**Proposition 6.7.** For \( j \in \mathcal{C}' \) and \( X \subseteq \mathcal{F} \) we have \( Q(U_j \cap X) \leq 1 - y(U_j \cap X) \).

**Proof.** Suppose \( j \) is matched to \( j' \) (the case in which \( j \) is not matched is similar). we have

\[
Q(U_j \cap X) = 1 - \sum_{i \in U_j \cap X} z_{i,0} - \sum_{i' \in U_j \cap X} z_{0,i'} - \sum_{i \in U_j \cap X, i' \in U_j} z_{i,i'} - \sum_{i, i' \in U_j \cap X} z_{i,i'}
\]

\[
\leq 1 - \sum_{i \in U_j \cap X} z_{i,0} - \sum_{i \in U_j \cap X} z_{i,i'}
\]

\[
= 1 - \sum_{i \in U_j \cap X} (1 - y(U_j')) \frac{y_i}{y(U_j)} - \sum_{i \in U_j \cap X, i' \in U_j} (y(U_j) + y(U_j') - 1) \frac{y_i y_{i'}}{y(U_j) y(U_j')}
\]

\[
= 1 - y(U_j \cap X)
\]

**Proposition 6.8.** Let \( u_0, u_1, u_2, \ldots, u_s \) be any real numbers with \( 0 = u_0 \leq u_1 \leq u_2 \leq \cdots \leq u_s \). For \( j \in \mathcal{C}' \) we have

\[
\sum_{k=1}^{s} (u_k - u_{k-1}) \left( 1 - y(B(j, u_k) \cap U_j) \right) \leq u_s (1 - y(U_j)) + r_j
\]
Proof. We compute:
\[
\sum_{k=1}^{s} (u_k - u_{k-1})(1 - y(B(j, u_k) \cap U_j)) = \sum_{k=1}^{s} (u_k - u_{k-1})(1 - y(U_j) + y(U_j - B(j, u_k))
\]
\[
= u_s(1 - y(U_j)) + \sum_{k=1}^{s} (u_k - u_{k-1})y(U_j - B(j, u_k))
\]
For the second term, we get:
\[
\sum_{k=1}^{s} (u_k - u_{k-1})y(U_j - B(j, u_k)) = \sum_{k=1}^{s} (u_k - u_{k-1}) \sum_{i \in U_j} y_i = \sum_{i \in U_j} y_i \sum_{k=1}^{s} [d(i, j) \geq u_k] (u_k - u_{k-1})
\]
For any \(i \in U_j\), the sum \(\sum_{k=1}^{s} [d(i, j) \geq u_k] (u_k - u_{k-1})\) is equal to \(u_{\ell}\), where \(\ell\) is maximal such that \(u_{\ell} \leq d(i, j)\). In particular, it is at most \(d(i, j)\). We therefore get
\[
\sum_{k=1}^{s} (u_k - u_{k-1})y(U_j - B(j, u_k)) \leq \sum_{i \in U_j} y_i d(i, j) = \sum_{i \in U_j} x_{i,j} d(i, j) \leq \sum_{i \in F} x_{i,j} d(i, j) = r_j
\]
as \(x_{i,j} = y_i\) for \(i \in U_j\).

We now get our main rounding result, which shows that BSR closely mimics independent selection for any client \(j\).

**Theorem 6.9.** Let \(j\) be any client. Suppose that under independent selection, it holds that \(E[d(j, S)] = A\). Suppose that \(t \geq cm^2\) for some sufficiently large constant \(c\). Then the set \(S\) returned by Algorithm 8 satisfies
\[
E[d(j, S)] \leq A + O\left(\frac{r_j m^2}{t}\right)
\]
Proof. Let \(j_1 \in C'\) be the closest facility to \(j\), and let \(R = R_{j_1} = \frac{1}{2} d(j_1, C' - j_1)\). By Proposition 6.5 we have \(d(j, j_1) \leq 4r_j\) and \(r_{j_1} \leq r_j\). Let \(w = d(j, j_1)\).

Let us sort the facilities in increasing distance to \(j\) as \(f_1, \ldots, f_s\), at distances \(u_1, \ldots, u_s\) respectively up to \(u_s \leq \beta R\), for some constant \(\beta\). We also define \(u_0 = 0\). For \(k = 1, \ldots, s\) define \(C_k = B(j, u_k)\).

By Proposition 6.5 it is guaranteed that \(d(j, S) \leq \beta R\). Therefore, we can compute the expected distance of \(j\) by
\[
E[d(j, S)] = \sum_{k=1}^{s} (u_k - u_{k-1}) Pr[S \cap B(j, u_k) = \emptyset]
\]
\[
= \sum_{k=1}^{s} (u_k - u_{k-1}) Pr[V \cap B(j, u_k) = \emptyset] = \sum_{k=1}^{s} (u_k - u_{k-1}) Q'(C_k) \quad \text{Proposition 6.4}
\]

Let \(\ell\) be minimal such that \(u_{\ell} \geq w\). We will separately analyze the sum from \(k = 1, \ldots, \ell - 1\) and the sum from \(k = \ell, \ldots, s\). For the first group, we have:
\[
\sum_{k=1}^{\ell-1} (u_k - u_{k-1}) Q'(C_k) \leq \sum_{k=1}^{\ell-1} (u_k - u_{k-1})(Q(C_k) + O(m^2/t)) \quad \text{Theorem 2.1}
\]
\[
\leq O(u_{\ell-1} m^2/t) + \sum_{k=1}^{\ell-1} (u_k - u_{k-1}) Q(C_k)
\]
\[
\leq O(r_j m^2/t) + \sum_{k=1}^{\ell-1} (u_k - u_{k-1}) Q(C_k)
\]
where the last holds since \( u_{\ell - 1} \leq w \leq O(r_j) \).

We next analyze the sum from \( k = \ell, \ldots, s \). Here, in order to estimate \( \mathbb{E}[Q(C_k, Z)] \), we will use Theorem 3.15 with \( W_0 = U_{j_1} \). Since \( W_0 \) spans a single group, then for \( t \geq cm^2 \) this gives

\[
\sum_{k=\ell}^{s} (u_k - u_{k-1})Q(C_k) \leq \sum_{k=\ell}^{s} (u_k - u_{k-1})(Q(C_k) + O(Q(W_0 \cap C_k)m^2/t))
\]

\[
\leq \sum_{k=\ell}^{s} (u_k - u_{k-1})Q(C_k) + O(m^2/t) \sum_{k=\ell}^{s} (u_k - u_{k-1})Q(W_0 \cap C_k)
\]

\[
\leq \sum_{k=\ell}^{s} (u_k - u_{k-1})Q(C_k) + O(m^2/t) \sum_{k=\ell}^{s} (u_k - u_{k-1})(1 - y(B(j, u_k) \cap U_{j_1}))
\]

Proposition 6.7

Note that \( B(j_1, a - w) \subseteq B(j, w) \) for any \( a \geq w \), so we can estimate

\[
\sum_{k=\ell}^{s} (u_k - u_{k-1})(1 - y(B(j, u_k) \cap U_{j_1})) \leq \sum_{k=\ell}^{s} (u_k - u_{k-1})(1 - y(B(j_1, u_k - w) \cap U_{j_1}))
\]

Now define the sequence \( u_1', \ldots, u_{s'} \) by setting \( u_0' = 0 \) and for \( p = 1, \ldots, s' = s - \ell \) setting \( u_p' = u_{p+\ell} - w \). Finally set \( u_0' = 0 \). With this definition, we observe that

\[
\sum_{k=\ell}^{s} (u_k - u_{k-1})(1 - y(B(j, u_k - w) \cap U_{j_1})) = \sum_{p=1}^{s'} (u_p' - u_{p-1}')(1 - y(B(j_1, u_p') \cap U_{j_1}))
\]

Applying Proposition 6.8 to the sequence \( u_0', \ldots, u_{s'} \), shows that this is at most

\[
u_{s'}(1 - y(U_{j_1})) + r_{j_1} = (u_s - w)(1 - y(U_{j_1})) + r_{j_1} \tag{5}
\]

By Proposition 6.6 we have \( (1 - y(U_{j_1})) \leq \frac{r_{j_1}}{R_{j_1}} \). Since \( u_s \leq \beta R \), expression (5) is at most

\[
(\beta R - w)\frac{r_{j_1}}{R} + r_{j_1} \leq O(r_{j_1}) \leq O(r_j)
\]

Putting all these terms together, we have shown that:

\[
\mathbb{E}[d(j, S)] \leq \sum_{k=1}^{s} (u_k - u_{k-1})Q(C_k) + O(r_j m^2/t)
\]

To finish, observe that \( \sum_{k=1}^{s} (u_k - u_{k-1})Q(C_k) \) is precisely the expected distance of \( j \) to \( S \) under independent selection.

**Corollary 6.10.** Suppose that \( t \geq cm^2 \) for an appropriate constant \( c \). Then Algorithm 8 ensures that for any client \( j \in C \) we have

\[
\mathbb{E}[d(j, S)] \leq (3.25 + O(m^2/t))r_j
\]

Furthermore, we have

\[
\mathbb{E} \left[ \sum_j d(j, S) \right] \leq (3.25 + O(m^2/t))\text{OPT}
\]

32
Theorem 6.11. Let $\gamma, \epsilon \in (0, 1)$. Suppose that $M_{ki} \leq 1$ for all $k, i$. With appropriate choices of $t$, Algorithm \text{Algorithm } \[ \text{Algorithm } \] runs in expected time $\text{poly}(n)/\gamma$ and produces a solution $S$ such that

$$\sum_{j \in C} d(j, S) \leq (3.25 + \gamma)OPT$$

$$\sum_{i \in S} M_{k,i} \leq B_k + O(m \sqrt{\log(m/\gamma) \gamma}) \quad \text{for } k = 1, \ldots, m$$

Proof. Let us rescale so that $OPT = 1$ and assume that $\gamma, \epsilon$ are sufficiently small.

Let $V = \sum_{j \in C} d(j, S)$; Proposition 6.1 shows that $E[V] \leq (3.25 + O(m^2/t))$ for $t \geq cm^2$.

Note that $\sum_{i \in S} M_{k,i} y_i = \sum_{v} M_{k,v} Z_v$, so $MZ = My \leq B$. We set $t = \epsilon m^2 / \gamma$ for some sufficiently large constant $c$. Theorem 3.10 shows that we have $MZ \leq Mz + O(m \sqrt{\log(m/\gamma) \gamma})$ with probability at least $1 - \gamma \geq 1/2$.

Let $E$ denote the event that $MZ \leq Mz + O(m \sqrt{\log(m/\gamma) \gamma})$. Thus, $E[V | E] \leq (3.25 + O(\gamma))$, and we can sample quickly from the conditional distribution on $E$. Furthermore, we have $P(V \leq 3.25 + 2\gamma | E) \geq \Omega(\gamma)$, which implies that after an expected $O(1/\gamma)$ repetitions we find a set $S$ such that $V \leq 3.25 + 2\gamma$ and such that $E$ occurs.

Overall, we get a runtime of $O(1/\gamma) \times n^{O(1)}$; the result follows by rescaling $\gamma$ and simplifying. \hfill \square

Theorem 6.11 can give a multiplicative approximation ratio, with a better dependence on $\epsilon$ compared to independent rounding.

Theorem 6.12. Let $\gamma, \epsilon \in (0, 1)$. There is an algorithm to obtain a solution $S$ in time $n^{O(m^2 \epsilon^{-1} \gamma^{-1} \gamma^{-1/2})}$ such that

$$\sum_{j \in C} d(j, S) \leq (3.25 + \gamma)OPT$$

$$\sum_{i \in S} M_{k,i} \leq (1 + \epsilon)B_k \quad \text{for } k = 1, \ldots, m$$

Proof. Say a facility $i$ is big if $M_{k,i} > \rho$ for any $k \in [m]$. The solution can contain at most $m/\rho$ big facilities. We guess the set $S_{\text{big}}$ of big facilities, incurring a cost of $n^{O(m/\rho)}$. We then construct a residual instance, in which all big facilities outside $S_{\text{big}}$ are discarded, and all those within $S_{\text{big}}$ have their weight set to zero. We will set $\rho = \Theta(\epsilon \sqrt{m^2 log(m/\gamma) \gamma})$. Now apply Theorem 6.11 to the residual instance. Overall, we get a runtime of $O(1/\gamma) \times n^{O(1)} \times n^{O(m/\rho)} \leq n^{O(m/\rho)}$. \hfill \square

7 Further correlation bounds for BSR

In this section, we will show a number of further near-negative-correlation properties for BSR. Although these are not directly used by the Knapsack center and Knapsack median problems, they may be useful for other rounding procedures. We also show that, as a consequence of negative correlation, there are strong concentration inequalities for sums of the form $y \cdot w$, where $w \in [0, 1]^n$ is an arbitrary weight vector.

7.1 Analyzing property (E2) for $\lambda < 0$ and small $Q_L(w, y)$

In this section, we will analyze $E[Q_L(w, y')]$ in cases in which $Q_L(w, y)$ is very small. In such cases, the estimate of Theorem 2.1 is not very useful, since it gives an additive gap of $O(m^2/t)$ compared to
$Q_{\lambda}(w, y)$. We would like a multiplicative gap instead. We cannot quite achieve something this strong, but we can nevertheless achieve an estimate which is significantly stronger than Theorem 2.1. Our main result in this section will be the following.

**Theorem 7.1.** Let $a \geq 0$, $b \in (0, 1/2]$, $\lambda \in [-1, 0]$, and $w \in [0, 1]^n$. Then when $t \geq m^2 \log^3(1/b)/a$ the output $y' = BSR(G_j, M, y, t)$ satisfies

$$E[Q_{\lambda}(w, y')] \leq e^{O(a)}(Q_{\lambda}(w, y) + ab)$$

Let us fix $w, \lambda < 0$ for the remainder of this section, and our overall strategy will be to find a recurrence relation on $Q_{\lambda}(w, y)$ and then solve this recurrence. One significant complication is that the intermediate values of $Q_{\lambda}(w, y)$ are random variables; it will require much technical delicacy to analyze this type of stochastic recurrence. For this reason, we will need to look at $BSR$ as a kind of two-dimensional system (as opposed to a stochastic process evolving in a single time dimension). We will view the algorithm as going through a series of *epochs*, each of which in turn executes a large number of $BSR$ iterations.

We will select a parameter $\beta \in (0, 2^{-20}]$ to be determined, and we define $k = \lceil \log_2(1/\beta) \rceil - 10 \geq 0$.

The process goes through $k$ epochs. We also define a parameter $\alpha_u$ for $u = -1, \ldots, k$ by

$$\alpha_u = \begin{cases} 1 & \text{if } u = k \\ 0 & \text{if } u = -1 \\ 2^u \beta & \text{if } -1 < u < k \end{cases}$$

In each epoch $u$, there are a number of rounds; we let $y_{u,v}$ denote the vector at round $v$ of epoch $u$. We define $T_{u,v} = T(y_{u,v})$. We also define $S_{u,v}$ to be our potential function

$$S_{u,v} = Q_{\lambda}(W, y_{u,v})$$

Here, $y_{0,0}$ will be the vector $y$ after Line (5) of $BSR$, i.e. just before executing the $BSR$-ITERATIONS. We then consider the following random process:

**Algorithm 9** Multi-epoch application of $BSR$-ITERATION

1. for $u = 0, \ldots, k$ do
2. if $S_{0,0} > \alpha_{u-1}$ then
3. for $v = 0, 1, 2, \ldots$ while $T_{u,v} \geq t$ and $S_{u,v} \leq \alpha_u$ do
4. Update $y_{u,v+1} \leftarrow BSR$-ITERATION($y_{u,v}, t$).
5. Set $y_{u+1,0} = y_{u,v}$
6. Return $y = y_{k+1,0}$

Let $V_{\text{final}}(u)$ denote the maximum round number achieved in epoch $u$. Note that by Proposition 3.3, this random variable is well-defined (i.e., it is finite) for all $u$ with probability one. Line (6) of this algorithm should be interpreted as setting $y_{u+1,0} = y_{u,V_{\text{final}}(u)}$. We will write $y_{u,\infty}$ as shorthand for $y_{u,V_{\text{final}}(u)}$, and similarly for $S_{u,\infty}, T_{u,\infty}$. It will also be convenient for us to set

$$y_{u,v} = y_{u,\infty} = y_{u+1,0}$$

$$S_{u,v} = S_{u,\infty} = S_{u+1,0}$$

for $v > V_{\text{final}}$.

We first note that this two-dimensional procedure is equivalent to $BSR$. 

34
Proposition 7.2. The resulting random variable $y^{k+1,0}$ from Algorithm 9 has the same probability distribution as the vector $y$ at the termination of BSR.

**Proof.** We claim that Algorithm 9 is equivalent to simply applying BSR-ITERATION to vector $y$ until $T(y) < t$. So the base case of the induction holds.

First, observe that if the condition at line (2) does not hold, then it must be that $T_{u-1,\infty} < t$. For, when $u = 0$, the condition $S_{u-1,v} > \alpha_{u-1}$ is vacuous since $\alpha_{u-1} = 0$. When $u > 0$, then the only other way to terminate the loop at epoch $u - 1$ would be to have $S_{u-1,v} > \alpha_{u-1}$. So when the condition of line (2) fails, then epochs $u, u + 1, \ldots, k$ do nothing which is the same as BSR.

When $u = k$, note that $\alpha_u = 1$. This means that $S_{u,v} \leq \alpha_u$ always, and so the loop in lines (4), (5) is simply applying BSR-ITERATION until $T(y) < t$. So the base case of the induction holds.

When $u < k$, observe that lines (4), (5) continue to execute BSR-ITERATION until $T(y) < t$ or $S_{u,v} \geq \alpha_u$. By induction hypothesis, the subsequent epochs continue to apply BSR-ITERATION until $T(y) < t$. So epochs $u, u + 1, \ldots, k$ are equivalent to simply applying BSR-ITERATION until $T(y) < t$.  

We will now analyze the evolution of $\mathbf{E}[S_{u,v}]$ during Algorithm 9. We do so in two stages. First, we analyze the change over an epoch (going from $S_{u,0}$ to $S_{u+1,0}$); next, we analyze the total change going from $S_{0,0}$ to $S_{k+1,0}$.

Proposition 7.3. Suppose that we condition on the state at the beginning of epoch $u$ and round $v$, and that $t \geq m \log(1/\beta)$. Then:

$$\mathbf{E}[S_{u,v+1}] \leq S_{u,v} + [T_{u,v} > t]O\left(\frac{m^2(2^u\beta)\log^2(1/\beta)}{T_{u,v}^2}\right)$$

(NB: The term $[T_{u,v} > t]$ here is the Iverson notation)

**Proof.** We may assume that $T_{u,v} > t$ and $S_{u,v} \leq \alpha_u$, as otherwise $S_{u,v+1} = S_{u,v}$ with probability one and the result holds trivially. By Proposition 3.14, we thus have:

$$\mathbf{E}[S_{u,v+1}] \leq S_{u,v} \cosh\left(\frac{12m \ln S_{u,v}}{T_{u,v}}\right) = S_{u,v} + (S_{u,v} \cosh\left(\frac{12m \ln S_{u,v}}{T_{u,v}}\right) - S_{u,v})$$

Define $z = 12m/T_{u,v}$ and consider the function $f(s) = s \cosh(z \ln s) - s$; simple analysis shows that $f(s)$ is an increasing function for $s \leq \left(\frac{1-z}{1+z}\right)^{1/z}$.

Let us first suppose that $u < k$. In this case, as epoch $u$ has not yet terminated, we must have $S_{u,v} \leq \alpha_u = 2^u \beta$; by our choice of $k = \lceil \log_2(1/\beta) \rceil - 10$, this is at most $2^{-9}$. On the other hand, $z \leq 12m/t \leq 12/\log(1/\beta) \leq 12/\log(2^{20}) \leq 0.88$, and so $((1-z)/(1+z))^{1/z} \geq 0.04$. These facts imply that $f(s)$ is increasing up to $s = 2^u \beta$ and accordingly we have

$$f(S_{u,v}) \leq f(2^u \beta)$$

By taking a second-order Taylor series around 0, simple analysis shows that as long as $|z \ln s| \leq O(1)$ we have $f(s) \leq O(s z^2 \log^2 s)$. The condition $T_{u,v} > t$ ensures that $|z \log(2^u \beta)| \leq O(\log(1/\beta)) \leq O(\frac{m^2}{T_{u,v}} \log(1/\beta))$; by hypothesis this is bounded by $O(1)$. So we have

$$f(2^u \beta) \leq O((2^u \beta) z^2 \log^2(2^u \beta)) \leq O\left(\frac{(2^u \beta) \log^2(1/\beta) m^2}{T_{u,v}^2}\right)$$

Next, let us consider the case $u = k$. One may verify in this case that

$$f(s) \leq f\left(\left(\frac{1-z}{1+z}\right)^{1/z}\right) \leq O(z^2)$$

35
So in this case again we have \( f(S_{u,v}) \leq O(z^2) \). As \( u = k \), we have \( 2^u \geq \beta/2^{10} \), and accordingly \( 2^u \beta = \Theta(1) \). So we have shown that \( f(S_{u,v}) \leq O((2^u \beta)^m^2 \geq O(2^m \beta \log^2 (1/\beta)^m^2) \) in this case as well. \( \square \)

**Proposition 7.4.** Suppose that \( t \geq m \log(1/\beta) \). Then:

\[
\mathbb{E}[S_{u+1,0} | S_{u,0}] \leq S_{u,0} + [S_{u,0} \geq \alpha_{u-1}] \times O\left(\frac{m^2 (2^u \beta \log^2 (1/\beta)}{t}\right)
\]

Proof. We will show that this bound holds even after conditioning on the full state at the beginning of epoch \( u \). If \( S_{u,0} < \alpha_{u-1} \), this is trivial, since in this case epoch \( u \) does nothing. So let us assume that \( S_{u,0} \geq \alpha_{u-1} \) and we want to show that

\[
\mathbb{E}[S_{u+1,0}] \leq S_{u,0} + O\left(\frac{m^2 (2^u \beta \log^2 (1/\beta)}{t}\right)
\]

By Proposition 7.3 conditional on the state at each round \( v \) of epoch \( u \), we have

\[
\mathbb{E}[S_{u,v+1} | \text{state at round } v] \leq S_{u,v} + [T_{u,v} > t] O\left(\frac{m^2 (2^u \beta \log^2 (1/\beta)}{T_{u,v}^2}\right)
\]

By iterated expectations, we therefore have for any \( v \geq 0 \):

\[
\mathbb{E}[S_{u,v}] \leq S_{u,0} + K m^2 (2^u \beta \log^2 (1/\beta) \sum_{j=0}^{v-1} \mathbb{E}\left[\frac{T_{u,j} > t}{T_{u,j}}\right]
\]

for some constant \( K \geq 0 \). We can write this equivalently as

\[
\mathbb{E}[S_{u,v}] = S_{u,0} + K m^2 (2^u \beta \log^2 (1/\beta) \sum_{\ell=t+1}^{\infty} 1/\ell^2 \mathbb{E}[\# j : T_{u,j} = \ell]
\]

\[
\leq S_{u,0} + K m^2 (2^u \beta \log^2 (1/\beta) \sum_{\ell=t+1}^{\infty} 10/\ell^2 \quad \text{Proposition 3.4}
\]

\[
\leq S_{u,0} + O\left( m^2 (2^u \beta \log^2 (1/\beta)/t\right)
\]

With probability one, the dependent rounding process terminates at some finite (random) number of rounds \( V_{\text{final}}(u) \). Also, note that \( S_{u,v} \leq 1 \) and for \( v > V_{\text{final}}(u) \) we have \( S_{u,\infty} = S_{u,v} \). Thus, for any \( v \geq 0 \), we have \( S_{u,\infty} \leq S_{u,v} + [V_{\text{final}}(u) > v]1 \). Taking expectations, this implies that

\[
\mathbb{E}[S_{u,\infty}] \leq \mathbb{E}[S_{u,v}] + \mathbb{P}[V_{\text{final}}(u) > v] \leq S_{u,0} + O\left( m^2 (2^u \beta \log^2 (1/\beta)/t\right) + \mathbb{P}[V_{\text{final}}(u) > v]
\]

Now take the limit of both sides as \( v \to \infty \). Observe that \( V_{\text{final}}(u) \) is finite with probability one so that \( P[V_{\text{final}}(u) > v] \to 0 \), and hence

\[
\mathbb{E}[S_{u+1,0}] = S_{u,0} + O\left( m^2 (2^u \beta \log^2 (1/\beta)/t\right)
\]

\( \square \)

**Proposition 7.5.** For \( \beta \in [0, 2^{-20}] \) and \( z = m^2/t \), we have

\[
\mathbb{E}[S_{k+1,0}] \leq (S_{0,0} + O(z \log^2 (1/\beta))) \exp(0(z \log^2 (1/\beta)))
\]
Proof. Suppose first that \( t < m \log(1/\beta) \). In this case, \( z > m/\log(1/\beta) \), and so

\[
z \beta \log^2(1/\beta) \exp(z \log^3(1/\beta)) \geq \beta e^{\log^4(1/\beta)} \log(1/\beta) \geq \Omega(1)
\]

Since \( S_{k+1,0} \leq 1 \), this bound will therefore hold vacuously. So we assume that \( t \geq m \log(1/\beta) \).

By Proposition 7.4 we have for each \( u \geq 0 \):

\[
E[S_{u+1,0} | S_u] \leq S_{u,0} + [S_{u_0} \geq \alpha_{u-1}] \times O(z^2u \beta \log^2(1/\beta))
\]

By iterated expectations, this in turn implies

\[
E[S_{u+1,0}] \leq E[S_{u,0}] + P(S_{u_0} \geq \alpha_{u-1}) \times O(z^2u \beta \log^2(1/\beta))
\]

For each \( u \geq 1 \), we use Markov’s inequality to obtain

\[
E[S_{u+1,0}] \leq E[S_{u,0}] + \frac{E[S_{u,0}]}{2^u \beta} \times O(z^2u \beta \log^2(1/\beta)) = E[S_{u,0}] + O(E[S_{u,0}]z \log^2(1/\beta))
\]

\[
\leq E[S_{u,0}] \exp(O(z \log^2(1/\beta)))
\]

Combining these bounds for \( u = 0, \ldots, k \) gives

\[
E[S_{k+1,0}] \leq (E[S_{0,0}] + O(z \beta \log^2(1/\beta))) \exp(O(kz \log^2(1/\beta)))
\]

Finally, note that \( k = \lceil \log_2 1/\beta \rceil - 10 \leq O(\log(1/\beta)) \). \( \square \)

We are finally ready to show a useful bound \( E[S_{k+1,0}] \).

Proposition 7.6. Let \( a \geq 0 \) and \( b \in [0, 1/2] \). Then for \( t \geq m^2 \log^3(1/b)/a \) we have \( E[S_{k+1,0}] \leq e^{O(a)}(S_{0,0} + ab) \).

Proof. We may assume that \( b \leq e^{-100a}/a \), as otherwise the the \( e^{100a}ab \geq 1 \) and the statement is vacuous. Together with the condition \( b \leq 1/2 \), this ensures that \( t \geq 12m^2 \) and so we can execute BSR.

Our hypothesis that \( t \geq m^2 \log^3(1/b)/a \) ensures that \( z \leq \frac{a}{\log(1/b)^3} \). Let us set \( \beta = b^{20} \) and apply Proposition 7.5. Observe that \( b^{20} \leq 2^{-20} \) as needed. So we get

\[
E[S_{k+1,0}] \leq (S_{0,0} + O(zb^2 \log^2(1/b))) \exp(O(z \log^3(1/\beta)))
\]

\[
\leq (S_{0,0} + O(ab^2 \log^{-1}(1/b)))e^{O(a)} = e^{O(a)}(S_{0,0} + ab)
\]

\( \square \)

Theorem 7.1. Let \( a \geq 0, b \in (0, 1/2], \lambda \in [-1, 0], \) and \( w \in [0, 1]^n \). Then when \( t \geq m^2 \log^3(1/b)/a \) the output \( y' = BSR(G_j, M, y, t) \) satisfies

\[
E[Q_{\lambda}(w, y')] \leq e^{O(a)}(Q_{\lambda}(w, y) + ab)
\]

Proof. By Proposition 7.6 Lines (1) – (5) of BSR do not change \( E[Q(W, y')] \). For the subsequent steps, apply Proposition 7.6 and Proposition 7.2. \( \square \)
7.2 Analyzing property (E2) for $\lambda > 0$

In this section, we will analyze $\mathbb{E}[Q_\lambda(w, y)]$ for $\lambda \geq 0$. In such cases, $\mathbb{E}[Q_\lambda(w, y)]$ can become exponentially large; most of our previous estimates, such as Theorem 7.1, had required for technical reasons that $Q_\lambda(w, y) \in (0, 1]$. Our main result in this section will as follows.

**Proposition 7.7.** Let $t \geq mk$ for some integer $k$. If $\kappa = 3$ and we execute $y' = \text{BSR}(G_j, M, y, t)$ then

$$\mathbb{E}[\min(2^k, Q_\lambda(w, y'))] \leq Q_\lambda(w, y) \exp(O(k^3m^2/t))$$

We fix $w \in [0, 1]^n, \lambda > 0$ for the remainder of this section. We will select an integer parameter $k \geq 1$. The process goes through $k$ epochs. In each epoch $u$, there are a number of rounds; we let $y^{u,v}$ denote the vector at round $v$ of epoch $u$. We define $T_{u,v} = T(y^{u,v})$. We also define $S_{u,v}$ to be our potential function

$$S_{u,v} = Q_\lambda(w, y^{u,v})$$

Here, $y^{0,0}$ will be the vector $y$ after Line (5) of BSR, i.e. just before executing the BSR-ITERATIONS. We then consider the following random process:

**Algorithm 10** Multi-epoch application of BSR-ITERATION

1: for $u = 0, \ldots, k$ do
2: \hspace{1em} if $S_{0,0} > 2^u$ then
3: \hspace{2em} for $v = 0, 1, 2, \ldots$ while $T_{u,v} \geq t$ and $S_{u,v} \leq 2^{u+1}$ do
4: \hspace{3em} Update $y^{u,v+1} \leftarrow \text{BSR-ITERATION}(y^{u,v}, t)$.
5: \hspace{2em} Set $y^{u+1,0} = y^{u,v}$
6: \hspace{1em} Return $y = y^{k+1,0}$

Let $V_{\text{final}}(u)$ denote the maximum round number achieved in epoch $u$. Note that by Proposition 3.3, this random variable is well-defined (i.e., it is finite) for all $u$ with probability one. Line (6) of this algorithm should be interpreted as setting $y^{u+1,0} = y^{u,V_{\text{final}}(u)}$. We will write $y^{u,\infty}$ as shorthand for $y^{u,V_{\text{final}}(u)}$, and similarly for $S_{u,\infty}, T_{u,\infty}$. It will also be convenient for us to set

$$y^{u,v} = y^{u,\infty} = y^{u+1,0}$$

$$S_{u,v} = S_{u,\infty} = S_{u+1,0}$$

for $v > V_{\text{final}}$.

We first note that this two-dimensional procedure is nearly equivalent to BSR.

**Proposition 7.8.** Let $y$ be the vector at the termination of BSR and let $y^{k+1,0}$ be the random variable from Algorithm 10. Then

$$\mathbb{E}[\min(2^k, Q_\lambda(w, y))] \leq \mathbb{E}[Q_\lambda(w, y^{k+1,0})]$$

**Proof.** Consider running Algorithm 3 until the first iteration that $Q_\lambda(w, y) \geq 2^{k+1}$, or otherwise until completion. Let $y''$ denote the resulting vector $y$ at the end of this process. Along the same lines of Proposition 7.2, we see that the distribution of $y^{k+1,0}$ is precisely the same as that of $y''$. So

$$\mathbb{E}[Q_\lambda(w, y'')] = \mathbb{E}[Q_\lambda(w, y^{k+1,0})]$$

Now, suppose we condition on a fixed value of $y''$, and continue running Algorithm 9 until completion. We claim that

$$\mathbb{E}[\min(2^k, Q_\lambda(w, y)) \mid y''] \leq Q_\lambda(w, y'')$$
For, if $Q_\lambda(w, y'') \leq 2^{k+1}$, then Algorithm has already terminated, so $y = y''$. Thus, $E[\min(2^k, Q_\lambda(w, y))] \leq E[Q_\lambda(w, y)] = Q_\lambda(w, y'')$. Otherwise, if $Q_\lambda(w, y'') > 2^{k+1}$, then this is vacuous, as $E[\min(2^k, Q_\lambda(w, y)) | y''] \leq 2^k \leq Q_\lambda(w, y)''. \hfill \Box$

We will now analyze the evolution of $E[S_{u,v}]$ during Algorithm. We do so in two stages. First, we analyze the change over an epoch (going from $S_{u,0}$ to $S_{u+1,0}$); next, we analyze the total change going from $S_{0,0}$ to $S_{k+1,0}$.

**Proposition 7.9.** Suppose that we condition on the state at the beginning of epoch $u$ and round $v$, and that $t \geq mk$. Then:

$$E[S_{u,v+1}] \leq S_{u,v} + [T_{u,v} > t]O\left(\frac{2^u k^2 m^2}{T_{u,v}}\right)$$

(NB: The term “$[T_{u,v} > t]$” here is the Iverson notation.)

**Proof.** We may assume that $T_{u,v} > t$ and $S_{u,v} \leq 2^{u+1}$, as otherwise $S_{u,v+1} = S_{u,v}$ with probability one and the result holds trivially. By Proposition 3.14, we thus have:

$$E[S_{u,v+1}] \leq S_{u,v} \cosh\left(\frac{12m \ln S_{u,v}}{T_{u,v}}\right) = S_{u,v} + \left(S_{u,v} \cosh\left(\frac{12m \ln S_{u,v}}{T_{u,v}}\right) - S_{u,v}\right)$$

Define $z = 12m/T_{u,v}$ and consider the function $f(s) = s \cosh(z \ln s) - s$; simple analysis shows that $f(s)$ is an increasing function for $s \geq 1$. So

$$f(S_{u,v}) \leq f(2^{u+1})$$

By taking a second-order Taylor series around 0, simple analysis shows that as long as $|z \ln s| \leq O(1)$ we have $f(s) \leq O(sz^2 \log^2 s)$. The condition $T_{u,v} > t \geq mk$ ensures that $|z \log(2^u)| \leq O(zu) \leq O(zk) \leq 1$, so

$$f(2^{u+1}) \leq O((2^{u+1}) z^2 \log^2 (2^{u+1})) \leq O\left(\frac{2^u u^2 m^2}{T_{u,v}^2}\right) \leq O\left(\frac{2^u k^2 m^2}{T_{u,v}^2}\right)$$

\hfill \Box

**Proposition 7.10.** Suppose that $t \geq mk$. Then:

$$E[S_{u+1,0} | S_{u,0}] \leq S_{u,0} + [S_{u,0} \geq 2^u] \times O\left(\frac{2^u m^2 k^2}{t}\right)$$

**Proof.** We will show that this bound holds even after conditioning on the full state at the beginning of epoch $u$. If $S_{u,0} < 2^u$, this is trivial, since in this case epoch $u$ does nothing. So let us assume that $S_{u,0} \geq 2^u$ and we want to show that

$$E[S_{u+1,0}] \leq S_{u,0} + O\left(\frac{2^u m^2 k^2}{t}\right)$$

By Proposition 7.9 conditional on the state at each round $v$ of epoch $u$, we have

$$E[S_{u,v+1} | \text{state at round } v] \leq S_{u,v} + [T_{u,v} > t]O\left(\frac{2^u m^2 k^2}{T_{u,v}^2}\right)$$

By iterated expectations, we therefore have for any $v \geq 0$:

$$E[S_{u,v}] \leq S_{u,0} + km^2 2^u k^2 \sum_{j=0}^{v-1} E[\frac{T_{u,j} > t}{T_{u,j}^2}]$$

39
for some constant $K \geq 0$. We can write this equivalently as

$$E[S_{u,v}] = S_{u,0} + Km^22^uk^2 \sum_{\ell=t+1}^{\infty} 1/\ell^2 \ E[\#j : T_{u,j} = \ell]$$

$$\leq S_{u,0} + Km^22^uk^2 \sum_{\ell=t+1}^{\infty} 10/\ell^2 \quad \text{Proposition 3.4}$$

$$\leq S_{u,0} + O(m^22^uk^2/t)$$

With probability one, the dependent rounding process terminates at some finite (random) number of rounds $V_{\text{final}}(u)$. Also, note that $S_{u,v} \leq |\lambda|^n$ and for $v > V_{\text{final}}(u)$ we have $S_{u,\infty} = S_{u,v}$. Thus, for any $v \geq 0$, we have $S_{u,\infty} \leq S_{u,v} + [V_{\text{final}}(u) > v]|\lambda|^n$. Taking expectations, this implies that

$$E[S_{u,\infty}] \leq E[S_{u,v}] + \Pr[V_{\text{final}}(u) > v]|\lambda|^n \leq S_{u,0} + O(m^22^uk^2/t) + P(V_{\text{final}}(u) > v)|\lambda|^n$$

Now take the limit of both sides as $v \to \infty$. Observe that $V_{\text{final}}(u)$ is finite with probability one so that $P(V_{\text{final}}(u) > v) \to 0$, and hence

$$E[S_{u+1,0}] = S_{u,0} + O(m^22^uk^2/t)$$

Proposition 7.11. For $t \geq mk$ and $z = m^2/t$, we have $E[S_{k+1,0}] \leq S_{0,0}e^{O(zk^3)}$.

Proof. By Proposition 7.10 we have for each $u \geq 0$:

$$E[S_{u+1,0} | S_u] \leq S_{u,0} + [S_{u0} \geq 2^u] \times O(z2^uk^2)$$

By iterated expectations, this in turn implies

$$E[S_{u+1,0}] \leq E[S_{u,0}] + P(S_{u0} \geq 2^u) \times O(z2^uk^2)$$

For each $u \geq 1$, we use Markov’s inequality to obtain

$$E[S_{u+1,0}] \leq E[S_{u,0}] + \frac{E[S_{u,0}]}{2^u} \times O(z2^uk^2) = E[S_{u,0}] + O(E[S_{u,0}]zk^2) \leq E[S_{u,0}] \exp(O(zk^2))$$

Combining these bounds for $u = 0, \ldots, k$ gives $E[S_{k+1,0}] \leq S_{0,0} \exp(O(zk^3))$. \hfill \Box

Proposition 7.7. Let $t \geq mk$ for some integer $k$. If $\kappa = 3$ and we execute $y' = BSR(G_j, M, y, t)$ then

$$E[\min(2^k, Q_\lambda(w, y'))] \leq Q_\lambda(w, y) \exp(O(k^3m^2/t))$$

Proof. By Proposition 3.6 Lines (1) – (5) of BSR do not change $E[Q(W, y')]$. For the latter steps, combine Proposition 7.11 with Proposition 7.8. \hfill \Box

40
7.3 Concentration bounds for FullBSR

As we have mentioned, independent selection does not affect the expected value of $Q_\lambda$. Thus, all the bounds on BSR apply immediately to FULLBSR. We record them here for completeness.

**Theorem 7.12.** Let $Y = \text{FULLBSR}(y, t)$ with $t > 12m^2$ and $y \in [0, 1]^n$. Let $\lambda \in [-1, \infty)$ and let $w \in [0, 1]^n$. Then we have the following bounds:

1. For $a \geq 0, b \in (0, 1/2], \lambda \in [-1, 0)$, we have
   \[
   \mathbb{E}[Q_\lambda(w, Y)] \leq e^{O(a)}(Q_\lambda(w, y) + ab)
   \]

2. For $\lambda \in [-1, 0]$, we have
   \[
   \mathbb{E}[Q_\lambda(w, Y)] \leq Q_\lambda(w, y) + O(m^2/t)
   \]

3. For $\lambda \in [0, \infty)$ and any integer $k \geq 0$ we have
   \[
   \mathbb{E}[\min(2^k, Q_\lambda(w, Y))] \leq Q_\lambda(w, y) \exp(O(k^3m^2/t))
   \]

*Proof.* The first two statements follow immediately from Theorem 7.1 and Theorem 2.1.

For the third statement, let $y'$ be the result of executing BSR($G_j, M, y, t$). Conditioned on the random variable $y'$, we claim that $\mathbb{E}[\min(2^k, Q_\lambda(w, Y))] \leq \min(2^k, Q_\lambda(y'))$. For, if $Q_\lambda(w, y) < 2^k$, then $\mathbb{E}[Q_\lambda(Y)] = Q_\lambda(y') = \min(2^k, Q_\lambda(w, y'))$; otherwise, if $Q_\lambda(y') \geq 2^k$, then we vacuously have $\mathbb{E}[\min(2^k, Q_\lambda(w, Y))] \leq 2^k = \min(2^k, Q_\lambda(y'))$.

Proposition 7.7 and Proposition 3.9 therefore give

\[
\mathbb{E}[\min(2^k, Q_\lambda(w, Y))] \leq \mathbb{E}[\min(2^k, Q_\lambda(w, y'))] \leq Q_\lambda(w, y) e^{O(m^2k^2/t)}
\]

We can use these results to show robust concentration bounds on the resulting random variable $Y$.

**Theorem 7.13.** Suppose we execute $Y = \text{FULLBSR}(y, t)$ for $t > 12m^2$. If $y \cdot w \geq \theta$ for some parameter $\theta \geq 0$ and $w \in [0, 1]^n$, then

\[
P(Y \cdot w \leq \theta(1 - \delta)) \leq e^{O(m^2(1+\delta)\theta^3/t)} \times \left(\frac{e^{-\delta}}{(1 - \delta)^{1-\delta}}\right)^\theta
\]

*Proof.* By Theorem 7.12 and Proposition 3.9 we have

\[
\mathbb{E}[Q_\lambda(w, Y)] \leq e^{O(a)}(Q_\lambda(w, y) + ab) \leq e^{O(a)}(e^{\lambda y \cdot w} + ab) \leq e^{O(a)}(e^{\lambda \theta} + ab)
\]

for any parameters $a \geq 0, b \in (0, 1/2], t \geq m^2 \log^3(1/b)/a$, and where $\lambda = -\delta$.

Whenever $Y \leq \theta(1 - \delta)$, then Proposition A.2 shows that $Q_\lambda(w, Y) \geq (1 + \lambda)^\theta(1-\delta)$. So we apply Markov’s inequality to the random variable $Q_\lambda(w, Y)$ to get

\[
P(Y \cdot w \leq \theta(1 - \delta)) \leq \frac{\mathbb{E}[Q_\lambda(w, Y)]}{(1 + \lambda)^\theta(1-\delta)} \leq \frac{e^{O(a)}(e^{\lambda \theta} + ab)}{(1 + \lambda)^\theta(1-\delta)}
\]

We now set $b = e^{-1-\delta \theta} \leq 1/2$ and $a = m^2 \log^3(1/b)/t$, giving

\[
P(Y \cdot w \leq \theta(1 - \delta)) \leq e^{O(m^2(1+\delta)\theta^3/t)}(e^{-\delta \theta} + O(m^2(1+\delta)\theta^3/t) \times e^{-1-\delta \theta}) \leq e^{O(m^2(1+\delta)\theta^3/t)} \left(\frac{e^{-\delta}}{(1 - \delta)^{1-\delta}}\right)^\theta
\]

\[\square\]
Theorem 7.14. Suppose we execute $Y = \text{FULLBSR}(y, t)$ for $t > 12m^2$. If $y \cdot w \leq \theta$ for some parameter $\theta \geq 0$ and $w \in [0, 1]^n$ then

$$P(Y \cdot w \geq \theta(1 + \delta)) \leq e^{O((\theta(1+\delta) \log(1+\delta))^2 m^2 / t)} \times \left(\frac{e^\delta}{(1 + \delta)^{1+\delta}}\right)^\theta$$

Proof. Define the function $H(x) = \min(2^k, Q_\lambda(W, x))$, where $\lambda = \delta$ and $k = \lceil \log_2((1 + \delta)\theta(1+\delta)) \rceil = \Theta((\theta(1+\delta) \log(1+\delta)))$

Theorem 7.12 and Proposition 3.9 shows that

$$E[H(Y)] \leq Q_\lambda(w, y) e^{O(k^3 m^2 / t)} \leq e^{O((\theta(1+\delta) \log(1+\delta))^2 m^2 / t) e^{\lambda y \cdot w}} \leq e^{O((\theta(1+\delta) \log(1+\delta))^2 m^2 / t) e^{\lambda \theta}}$$

If $Y \cdot w \geq \theta(1 + \delta)$, then Proposition A.2 would show that $Q_\lambda(w, Y) \geq (1 + \lambda)^{\theta(1+\delta)}$. Because of our choice of $k$, this also would imply that $H(Y) \geq (1 + \lambda)^{\theta(1+\delta)}$.

So we apply Markov’s inequality to the random variable $H(Y)$ to get

$$P(Y \cdot w \geq \theta(1 + \delta)) \leq \frac{E[H(Y)]}{(1 + \delta)^{\theta(1+\delta)}} \leq \frac{e^{O((\theta(1+\delta) \log(1+\delta))^2 m^2 / t)}}{(1 + \delta)^{\theta(1+\delta)}}$$

$$\leq e^{O((\theta(1+\delta) \log(1+\delta))^2 m^2 / t)} \times \left(\frac{e^\delta}{(1 + \delta)^{1+\delta}}\right)^\theta$$

8 Acknowledgment

We thank Chandra Chekuri as well as the referees of an earlier version for their helpful suggestions.

References

[1] A. Ageev and M. Sviridenko. Pipage rounding: a new method of constructing algorithms with proven performance guarantee. Journal of Combinatorial Optimization, 8(3):307–328, 2004.

[2] N. Bansal and V. Nagarajan. Approximation-friendly discrepancy rounding. In Integer Programming and Combinatorial Optimization - 18th International Conference, IPCO 2016, Liège, Belgium, June 1-3, 2016, Proceedings, pages 375–386, 2016.

[3] J. Beck and T. Fiala. “Integer-making” theorems. Discrete Applied Mathematics, 3:1–8, 1981.

[4] J. Byrka, T. Pensyl, B. Rybicki, J. Spoerhase, A. Srinivasan, and K. Trinh. An improved approximation algorithm for knapsack median using sparsification. In Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings, pages 275–287, 2015.

[5] J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh. An improved approximation for $k$-median, and positive correlation in budgeted optimization. In Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA 2015), pages 737–756, 2015.

[6] J. Byrka, A. Srinivasan, and C. Swamy. Fault-Tolerant Facility Location: A Randomized Dependent LP-Rounding Algorithm. In Integer Programming and Combinatorial Optimization, 14th International Conference, IPCO 2010, Lausanne, Switzerland, June 9-11, 2010. Proceedings, pages 244–257, 2010.
[7] G. Călinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.

[8] M. Charikar and S. Li. A dependent LP-rounding approach for the $k$-median problem. *Automata, Languages, and Programming (ICALP)*, pages 194–205, 2012.

[9] C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *51th Annual IEEE Symposium on Foundations of Computer Science*, pages 575–584, 2010.

[10] C. Chekuri, J. Vondrák, and R. Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1080–1097, 2011.

[11] D. Z. Chen, J. Li, H. Liang, and H. Wang. Matroid and Knapsack Center Problems. In *Integer Programming and Combinatorial Optimization: 16th International Conference, IPCO 2013, Valparaíso, Chile, March 18-20, 2013. Proceedings*, pages 110–122. Springer: Berlin, Heidelberg, 2013.

[12] F. A. Chudak and D. B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM J. Comput.*, 33(1):1–25, 2003.

[13] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53:324–360, 2006.

[14] D. G. Harris and A. Srinivasan. The Moser-Tardos Framework with Partial Resampling. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 469–478, 2013. A more-detailed version is available at the arXiv, https://arxiv.org/abs/1406.5943.

[15] D. S. Hochbaum and D. B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, 1986.

[16] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In STOC, pages 731–740, 2002.

[17] R. M. Karp, F. T. Leighton, R. L. Rivest, C. D. Thompson, U. V. Vazirani, and V. V. Vazirani. Global wire routing in two-dimensional arrays. *Algorithmica*, 2:113–129, 1987.

[18] S. Khuller, R. Pless, and Y. J. Sussmann. Fault tolerant $k$-center problems. *Theor. Comput. Sci.*, 242(1-2):237–245, 2000.

[19] R. Krishnaswamy, A. Kumar, V. Nagarajan, Y. Sabharwal, and B. Saha. The matroid median problem. In *Proceedings of the 22nd annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1117–1130. SIAM, 2011.

[20] R. Krishnaswamy, S. Li, and S. Sandeep. Constant approximation for $k$-median and $k$-means with outliers via iterative rounding. *CoRR*, abs/1711.01323, 2017.

[21] F. T. Leighton, C. Lu, S. Rao, and A. Srinivasan. New Algorithmic Aspects of the Local Lemma with Applications to Routing and Partitioning. *SIAM J. Comput.*, 31(2):626–641, 2001.

[22] S. Li and O. Svensson. Approximating $k$-median via pseudo-approximation. In *Symposium on Theory of Computing Conference, STOC’13, Palo Alto, CA, USA, June 1-4, 2013*, pages 901–910, 2013.
A. Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *IEEE Symposium on Foundations of Computer Science*, pages 588–597, 2001.

A. Srinivasan. An Extension of the Lovász Local Lemma, and its Applications to Integer Programming. *SIAM J. Comput.*, 36(3):609–634, 2006.

D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

### A Some technical lemmas

**Proposition A.1.** Let $\lambda \in [-1, \infty)$ and let $x \in [0, 1]$. Then

$$\frac{\lambda |x(1-x)|}{1+\lambda x} \leq |\ln(1+\lambda x)|$$

**Proof.** When $\lambda = 0$ this holds vacuously. Let us first suppose $\lambda > 0$. In this case want to show that

$$\frac{\lambda x(1-x)}{1+\lambda x} - \ln(1+\lambda x) \leq 0$$

When $x = 0$, the LHS is equal to zero. The derivative of the LHS with respect to $x$ is given by $-\lambda x(2+\lambda + \lambda x)/(1+\lambda x)^2$, which is clearly negative. This shows that the LHS is negative for all $x \geq 0$.

Next, suppose $\lambda < 0$. In this case, we want to show that

$$\frac{-\lambda x(1-x)}{1+\lambda x} + \ln(1+\lambda x) \leq 0$$

Again, at $x = 0$ the LHS is equal to zero. The derivative of the LHS is given by $\lambda x(2+\lambda + \lambda x)/(1+\lambda x)^2$. Since $x \in [0, 1]$ and $\lambda \geq -1$, this is also clearly negative, so the LHS is again negative.

**Proposition A.2.** Suppose $Y \in \{0, 1\}^n$ and $w \in [0, 1]^n$. Then for $\lambda \in [-1, \infty]$ we have the bound:

$$Q_\lambda(w, Y) \geq (1+\lambda)^{Y^w}$$

**Proof.** We have

$$Q_\lambda(w, Y) \geq \prod_{i=1}^r (1 + \lambda \sum_{j \in G_i} w_j Y_j)$$

We claim that $(1 + \lambda \sum_{j \in G_i} w_j Y_j) = \prod_{j \in G_i} (1 + \lambda w_j Y_j)$ for any group $i$. For, since $Y(G_i) \leq 1$, there is at most one value $j \in G_i$ with $Y(G_j) = 0$. If $Y_j = 0$ for all $j \in G_i$, then

$$(1 + \lambda \sum_{j \in G_i} w_j Y_j) = 1 = \prod_{j \in G_i} (1 + \lambda w_j Y_j)$$

Otherwise, if $Y_\ell = 1$ for $\ell \in G_i$, then

$$(1 + \lambda \sum_{j \in G_i} w_j Y_j) = (1 + \lambda w_\ell Y_\ell) = \prod_{j \in G_i} (1 + \lambda w_j Y_j)$$
Therefore, we get

\[ Q_{\lambda}(w, Y) = \prod_{i=1}^{r} (1 + \lambda \sum_{j \in G_i} w_j Y_j) = \prod_{i=1}^{r} \prod_{j \in G_i} (1 + \lambda w_j Y_j) = \prod_{i=1}^{n} (1 + \lambda w_j Y_j) \]

\[ \geq \prod_{i=1}^{n} (1 + \lambda)^{w_j Y_j} \quad \text{as} \quad (1 + ab) \geq (1 + a)^b \quad \text{for} \quad a \in [-1, \infty), b \in [0, 1] \]

\[ = (1 + \lambda)^{Y \cdot w} \]

\[ \square \]

**Proposition A.3.** Let \( a \in [0, 1/(u + 1)] \), \( s \in [0, 1] \), and let \( u \in \mathbb{Z}_+ \). We have

\[ s \cosh(a(u - \ln s)) - s \leq a^2(u + 1)^2 \]

**Proof.** Let \( f(s) = s \cosh(a(u - \ln s)) - s \). Simple calculus shows that the critical points of function \( f(s) \) occur at \( s_0 = e^u \) and \( s_1 = e^{u - \ln(1 + a)/(1 - a)} \). Since \( s_0 \) is outside the allowed parameter range, this means that maximum value of \( f(s) \) in the interval \([0, 1]\) must occur at either \( s = 0, s = s_1 \), or \( s = 1 \).

At \( s = 0 \), we have \( f(s) = 0 \). So let us estimate \( f(1) \) and \( f(s_1) \).

For the former, we have \( f(1) = \cosh(au) - 1 \). Since \( au \leq 1 \), this is at most \((au)^2\).

For the latter, simple calculus gives:

\[ f(s_1) = \left( \frac{1 + a}{1 - a} \right)^{-1/a} e^u(\cosh(\ln(1 + a) - \ln(1 - a)) - 1) \leq e^u a^2 / 2 \]

However, we claim that whenever \( u \geq 3 \) we have \( s_1 > 1 \), and so \( s_1 \) is outside the range of interest. For, we have

\[ s_1 = e^{u - \ln(1+u)/(1-u)} \geq e^{u - \ln(1+u+1)/(u+1)} = e^u(u/(u+2))^{u+1} \]

and this is larger than 1 for \( u \geq 3 \). Thus, since \( u \in \mathbb{Z} \), we only need to show that \( f(s_1) \leq a^2(u + 1)^2 \) for \( u = 0, 1, 2 \). These both easily follow from \( f(s_1) \leq e^u a^2 / 2 \). \( \square \)