Cost Preserving Dependent Rounding for Allocation Problems

Lars Rohwedder
University of Southern Denmark, Odense, Denmark

Arman Rouhani
Maastricht University, The Netherlands

Leo Wennmann
University of Southern Denmark, Odense, Denmark

Abstract

We present a dependent randomized rounding scheme, which rounds fractional solutions to integral solutions satisfying certain hard constraints on the output while preserving Chernoff-like concentration properties. In contrast to previous dependent rounding schemes, our algorithm guarantees that the cost of the rounded integral solution does not exceed that of the fractional solution. Our algorithm works for a class of assignment problems with restrictions similar to those of prior works.

In a non-trivial combination of our general result with a classical approach from Shmoys and Tardos [Math. Programm.'93] and more recent linear programming techniques developed for the restricted assignment variant by Bansal, Sviridenko [STOC'06] and Davies, Rothvoss, Zhang [SODA'20], we derive a $O(\log n)$-approximation algorithm for the *Budgeted Santa Claus Problem*. In this new variant, the goal is to allocate resources with different values to players, maximizing the minimum value a player receives, and satisfying a budget constraint on player-resource allocation costs.

2012 ACM Subject Classification Theory of computation → Rounding techniques

Keywords and phrases Matching, Randomized Rounding, Santa Claus, Approximation Algorithms

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.122

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2502.08267`

Funding *Lars Rohwedder and Leo Wennmann*: Supported by Dutch Research Council (NWO) project "The Twilight Zone of Efficiency: Optimality of Quasi-Polynomial Time Algorithms" [grant number OCEN.W.21.268].

1 Introduction

A successful paradigm in the design of approximation algorithms is to first solve a continuous relaxation, which can typically be done efficiently using linear programming, and then to round the fractional solution $x \in [0,1]^d$ to an integer solution $X \in \{0,1\}^d$. Careful choices need to be made in the rounding step so that the error introduced is low. Independent randomized rounding is one of the most natural rounding schemes. In the simplest variant, we independently set each variable X_i to 1 with probability x_i and to 0 with probability $1 - x_i$. The advantage is that the value of every linear function (over the d variables) is maintained in expectation. Moreover, for linear functions with small coefficients, a Chernoff bound yields strong concentration guarantees for the value. Hence, if the initial solution x satisfies some linear constraints from the continuous relaxation, we can often argue with several Chernoff bounds combined by a union bound that they are still satisfied up to a small error by the rounded solution X.

In some cases, however, the problem dictates structures or hard constraints on the solution.

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).
Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joel Ouaknine, and Gabriele Puppis; Article No. 122; pp. 122:1–122:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

For example, we might require X to be (the incidence vector of) a perfect matching in a given graph or the basis of a given matroid. Perfect matchings or bases are objects that are quite simple in many computational aspects, but it is typically very unlikely that independent randomized rounding on a fractional object, that is, a point in the convex hull of the objects we want, results in one of these objects. This motivates so-called dependent randomized rounding. Here, the goal is to achieve similar guarantees as independent randomized rounding, but with a distribution over a restricted set of objects, which necessarily introduces some dependency between variables. These rounding schemes are typically tailored to specific object structures and achieving comparable goals is already challenging for very simple structures.

For bipartite perfect matchings, a fundamental structure in combinatorial optimization, one cannot hope to achieve similar concentration guarantees to independent randomized rounding, due to the following well-known example. Given a cycle of length $n \in 2\mathbb{N}$, there are only two perfect matchings, the two alternating sets of edges. If the values of a linear function over the edges alternate between 0 and 1, then the fractional matching, which takes every edge with $1/2$ will have a function value of $n/2$, but each of the integral matchings incurs an additive distortion of $\Omega(n)$, much higher than the bound of $O(\sqrt{n})$ that holds with high probability if each edge is picked independently with probability $1/2$. If one considers b-matchings or other more general assignment problems, however, then there are non-trivial guarantees that can be achieved with dependent rounding. Gandhi, Khuller, Parthasarathy, and Srinivasan [11] show that between any two edges incident to the same vertex, they can establish *negative correlation*. Furthermore, their algorithm has the natural property of *marginal preservation*, which means that the probability of $X_e = 1$ is equal to the fractional value x_e for each variable X_e. Together this implies strong concentration guarantees at least for linear functions on the incident edges of each vertex. The following proposition is a consequence of their result.

▶ **Proposition 1.** *Let $G = (A \cup B, E)$ be a bipartite graph and $x \in [0,1]^E$ represent a fractional many-to-many assignment. Furthermore, let $c \in \mathbb{R}^E_{\geq 0}$, and $a_1, \dots, a_k \in [0,1]^E$ such that for each $j \in \{1, \dots, k\}$ there is some $v \in A \cup B$ with $\operatorname{supp}(a_j) \subseteq \delta(v)$. Then, in randomized polynomial time, one can compute $X \in \{0,1\}^E$ satisfying with constant probability*

Cost Approximation. $c^\intercal X \leq (1+\epsilon) \cdot c^\intercal x$

Concentration. $|a_j^\intercal x - a_j^\intercal X| \leq O(\max\{\log k, \sqrt{a_j^\intercal x \cdot \log k}\})$ *for all* $j \in \{1, \dots, k\}$

Degree Preservation. $X(\delta(v)) \in \{\lfloor x(\delta(v)) \rfloor, \lceil x(\delta(v)) \rceil\}$

Here, $\epsilon > 0$ is an arbitrarily small constant that influences the success probability.

A generalization to matroid intersection with a similar restriction was shown by Chekuri, Vondrack, and Zenklusen [7]. The same work also presents a dependent rounding scheme for a single matroid that outputs a basis satisfying similar concentration bounds on linear functions without a restriction on the support. In this study, we ask the following question:

Can we avoid an error in the cost for dependent rounding while maintaining comparable other guarantees?

We call a rounding algorithm *cost preserving* if it does not exceed the cost of the fractional solution we start with. Here, we focus on the stronger variant where distributions are only over objects that are cost preserving, although one might be satisfied with a sufficiently high probability of cost preservation in some cases. We have no evidence that such a relaxation would make the task significantly easier.

There are several situations where even the seemingly small cost approximation of $(1+\epsilon)$, as derived from marginal preservation and Markov's inequality in the previously mentioned result, is unacceptable. For example, the cost of the fractional solution might come from a hard budget constraint $c^\mathsf{T} x \leq C$ in the problem. Another situation is an extension of the objective function to potentially negative values, representing for example the task of maximizing profit = revenue - cost. Here, Markov's inequality cannot be applied at all. Finally, an algorithm that preserves the cost provides polyhedral insights: every fractional object is in the convex hull of integer objects that marginally deviate in the considered linear functions. And similarly, the (non-integral) polytope of a relaxation is contained in an approximate integral polytope. It is easy to see that cost preservation is incompatible with marginal preservation and hence cannot be satisfied by the dependent rounding schemes above: consider $d+1$ variables $x_0, x_1, \ldots, x_d$ of which exactly one is selected, then this can be modeled by bases of a uniform matroid or a degree constraint in the assignment problem above. Suppose that $c_1 = \cdots = c_d = 1/(1-2^{-d}) > 1$ and $c_0 = 0$ where the fractional solution is given by $x_1 = \cdots = x_n = (1-2^{-d})/d$ and $x_0 = 2^{-d}$, leading to a cost of 1. For a marginal preserving distribution, the probability that the integral solution X has a cost lower than 1 (i.e., $X_0 = 1$) is exponentially small. Note, however, that this is not an immediate counter-example to our stated goal: in this example, deterministically choosing $X_0 = 1$ (and $X_1 = \cdots = X_d = 0$) still maintains $|a^\mathsf{T} x - a^\mathsf{T} X| \leq 1$ for every $a \in [0,1]^{d+1}$.

Our contributions

Our results are twofold. First, we show that one can obtain comparable guarantees to Proposition 1 while preserving costs.

▶ **Theorem 2.** *Let $G = (A \cup B, E)$ be a bipartite graph and $x \in [0,1]^E$ represent a many-to-many assignment. Furthermore, let $c \in \mathbb{R}^E$ and $a_1, \ldots, a_k \in [0,1]^E$ such that for each $j \in \{1, \ldots, k\}$ there is some $v \in A \cup B$ with $\operatorname{supp}(a_j) \subseteq \delta(v)$. Then, in randomized polynomial time, one can compute $X \in \{0,1\}^E$ satisfying with constant probability*

Cost Preservation. $c^\mathsf{T} X \leq c^\mathsf{T} x$,

Concentration. $|a_j^\mathsf{T} x - a_j^\mathsf{T} X| \leq O(\max\{\log k, \sqrt{a_j^\mathsf{T} x \cdot \log k}\})$ *for all* $j \in \{1, \ldots, k\}$,

Degree Preservation. $X\big(\delta(v)\big) \in \{\lfloor x\big(\delta(v)\big) \rfloor, \lceil x\big(\delta(v)\big) \rceil\}$.

Note that in contrast to the previous result, we allow for negative components in the cost function c.

Second, we present a non-trivial application of our theorem to an allocation problem we call the *Budgeted Santa Claus Problem (with identical valuations)*. Colloquially, it is often described as Santa Claus distributing gifts to children on Christmas. Formally, there are n resources $\mathcal{R}$ (gifts) to be distributed among m players $\mathcal{P}$ (children). Each resource j has a specific value $v_j \geq 0$. Additionally, there is a total budget of $C \geq 0$, and assigning a resource j to a player i incurs a cost denoted by $c_{ij} \geq 0$. The goal is a distribution of resources among the players where the least happy player is as happy as possible and the total cost does not exceed the budget C. Formally, we aim to find disjoint sets $R_i \subseteq \mathcal{R}$, $i \in \mathcal{P}$, maximizing $\min_{i \in \mathcal{P}} \sum_{j \in R_i} v_j$ while ensuring that $\sum_{i \in \mathcal{P}} \sum_{j \in R_i} c_{ij} \leq C$. Note that not all resources need to be assigned. However, the variant, where all resources must be assigned can be shown to be not more difficult than our problem, see Appendix A.1.

It is possible to consider an even more general variant where each value v_{ij} depends on both player i and resource j, which we call the *unrelated valuations*. Mainly, we restrict ourselves to identical valuations because the understanding of unrelated valuations in literature is rather poor—even without considering costs. In fact, much of the recent literature is focused on the so-called restricted assignment case of unrelated valuations (without costs), where $v_{ij} \in \{0, v_j\}$, meaning each resource is either desired with a value of v_j or worthless to a player. Among players who desire a particular resource, its value is the same. Our budgeted variant generalizes the restricted assignment case: observe that by setting costs $c_{ij} \in \{0, 1\}$ and $C = 0$, we can restrict the set of players to which a resource can be assigned. In a non-trivial framework, we apply our dependent rounding theorem to obtain the following approximation guarantee.

▶ **Theorem 3.** *There is a randomized polynomial time $O(\log n)$-approximation algorithm for the Budgeted Santa Claus problem.*

Other related work for dependent rounding

Saha and Srinivasan [15] also provide a dependent rounding scheme for allocation problems, focusing on combinations of dependent and iterative rounding. Bansal and Nagarajan [5] combine dependent rounding with techniques from discrepancy theory, known as the Lovett-Meka algorithm [14]. They prove that one can round a fractional independent set (or basis) of a matroid to an integral one, while maintaining comparable concentration guarantees to both Lovett-Meka and Chernoff-type bounds. We note that Bansal and Nagarajan also integrate costs in their framework, but they make the assumption that the costs are polynomially bounded, which is inherently different from our setting (apart from the fact that they consider matroids).

Another well-known dependent rounding scheme is the maximum entropy rounding developed by Asadpour, Goemans, Mądry, Gharan and Saberi [3]. This is used to sample a spanning tree, i.e., a basis of a particular matroid, while guaranteeing negative correlation properties and therefore Chernoff-type concentration. This result led to the first improvement over the longstanding approximation rate of $\Theta(\log n)$ for the asymmetric traveling salesman problem (ATSP). However, all algorithms above guarantee marginal preservation, which means they cannot guarantee cost preservation.

At least superficially related to our work is the literature on multi-budgeted independence systems [8, 12]. Here, the goal is to find objects of certain structures, e.g., matchings or independent sets of matroids, subject to several (potentially hard) packing constraints of the form $a^\mathsf{T} x \leq b$ for some $a \in \mathbb{R}^n_{\geq 0}$, $b \in \mathbb{R}_{\geq 0}$. This can also be used to model cost preservation alike to our results. Chekuri, Vondrak, and Zenklusen [8] and Gradoni and Zenklusen [12] show various positive results in a similar spirit to ours. These results, however, are restricted to downward-closed structures where for a given solution, formed by a set of elements, all subsets are valid solutions as well. For example, Chekuri, Vondrak, and Zenklusen achieve strong concentration results for randomized rounding on matchings, but this relies on dropping edges in long augmenting paths or cycles in order to reduce dependencies. Gradoni and Zenklusen [12] give a rounding algorithm for a constant number of hard budgets, but this requires rounding down all components of a fractional solution. Hence, these results are unable to handle instances like matroid basis constraints, perfect matching constraints, or degree preservation as in Theorem 2.

Other related work for the Santa Claus problem

Omitting the costs in the variant we study, the problem becomes significantly easier and admits an EPTAS, see e.g. [13], which relies on techniques that contrast with the ones that are relevant to us. As mentioned before, the problem with costs generalizes the restricted assignment variant and therefore inherits the approximation hardness of $2 - \epsilon$ due to [6]. Here and in the following, we use restricted assignment synonymous with the variant without costs, but $v_{ij} \in \{0, v_j\}$. Bansal and Srividenko [6] developed a randomized rounding algorithm for the restricted assignment. Normally, this would lead to a similar logarithmic approximation rate as ours (for the problem without costs), but they show that combining it with the Lovász Local Lemma yields an even better rate of $O(\log\log m/\log\log\log m)$. Using similar techniques, the rate was improved to a constant by Feige [10]. Note that this randomized rounding uses intricate preprocessing that violates the marginal preserving property and thus cannot even maintain the cost of a solution in expectation. Based on local search, there is also a combinatorial approach, see e.g., [4, 2, 1], which yields a (better) constant approximation for restricted assignment. However, it is not at all clear how costs could be integrated in this framework.

Finally, a classical algorithm by Shmoys and Tardos [17] gives an additive guarantee, where the rounded integral solution is only worse by the maximum value $v_{\max} = \max_{ij} v_{ij}$. Therefore, it even works in the unrelated case without increasing the cost. Notably, they state this result for the dual of minimizing the maximum value, namely the Generalized Assignment Problem. The mentioned guarantee for Santa Claus is followed by a trivial adaption, see Lemma 10. Although very influential, this is the only technique we are aware of which considers the problem with costs. Unfortunately, this additive guarantee does not lead to a multiplicative guarantee, since the optimum may be lower than $v_{\max}$. In fact, it is well known that the linear programming relaxation used in [17] has an unbounded integrality gap even for restricted assignment [6]. Hence, one cannot hope to improve this by a simple modification. Nevertheless, this algorithm forms an important subprocedure in our result.

Notation

First, we introduce some necessary notation. Let $S, T \in \{0,1\}^E$ be edge sets in a bipartite graph $G = (A \cup B, E)$. For all $T \subseteq S$, define $S(T) = \sum_{e \in T} S_e$. Let P be the convex hull of degree preserving edge sets $S \in [0,1]^E$. Moreover, for any $v \in A \cup B$ define $\delta(v) = \{e \in E \mid v$ is incident to $e\}$. For the sake of simplicity, we use the shorthand notation $[q] = \{1, \dots, q\}$ for any $q \in \mathbb{N}$. Furthermore, for any vector $a \in [0,1]^E$, the support of a is denoted by $\mathrm{supp}(a) = \{e \in E \mid a_e \neq 0\}$.

2 Budgeted Dependent Rounding

This section will introduce a dependent randomized rounding procedure, which produces an integral solution satisfying certain concentration guarantees, while preserving the cost and the degree of the fractional solution. The formal properties are summarized in the following theorem.

▶ **Theorem 2.** *Let $G = (A \cup B, E)$ be a bipartite graph and $x \in [0,1]^E$ represent a many-to-many assignment. Furthermore, let $c \in \mathbb{R}^E$ and $a_1, \dots, a_k \in [0,1]^E$ such that for each $j \in \{1, \dots, k\}$ there is some $v \in A \cup B$ with $\mathrm{supp}(a_j) \subseteq \delta(v)$. Then, in randomized polynomial time, one can compute $X \in \{0,1\}^E$ satisfying with constant probability*

Cost Preservation. $c^\mathsf{T} X \leq c^\mathsf{T} x$,

Concentration. $|a_j^\mathsf{T} x - a_j^\mathsf{T} X| \leq O(\max\{\log k, \sqrt{a_j^\mathsf{T} x \cdot \log k}\})$ *for all* $j \in \{1, \dots, k\}$,

Degree Preservation. $X(\delta(v)) \in \{\lfloor x(\delta(v)) \rfloor, \lceil x(\delta(v)) \rceil\}$.

Throughout this section, the proofs of the technical lemmas are deferred to Section 2.1. An oversimplified outline of our algorithm is as follows: imagine x is the average of two integral edge sets, then the result can be shown by decomposing the symmetric difference of both edge sets into cycles and paths. We reduce to this case by starting with many edge sets and iteratively merging pairs of them in a tree-like manner.

In order to find the initial integral edge sets, we compute a representation of x (or rather another similar assignment y') that is a convex combination of degree preserving edge sets such that its scalars satisfy a certain level of discreteness. Let P be the convex hull of degree preserving edge sets $S \in \{0,1\}^E$, that is, those S that satisfy for all $v \in A \cup B$

$$S(\delta(v)) \in \{\lfloor x(\delta(v))\rfloor, \lceil x(\delta(v)\rceil\}.$$

It can be shown that x is contained in P and, in particular, x is a convex combination of degree preserving sets. In the following lemma, we show something even stronger: there exists a fractional assignment y at least as good as x, which is the convex combination of only few edge sets and has few fractional variables in the support of each constraint.

▶ **Lemma 4.** *There exists a convex combination $y = \sum_{i\in[k]} \lambda_i S_i$ where $\lambda_i \in [0,1]$ and $\sum_{i\in[k]} \lambda_i = 1$ and $S_i \in P \cap \{0,1\}^E$ with*

$$\begin{aligned} c^\mathsf{T} y &\le c^\mathsf{T} x && && (1)\\ a_j^\mathsf{T} y &= a_j^\mathsf{T} x && \forall j \in \{1,\dots,k\} && (2)\\ |\{e \in \delta(v) \mid y_e \notin \{0,1\}\}| &\le 2k && \forall v \in A \cup B && (3) \end{aligned}$$

Considering y as a vertex solution of a linear program, the proof follows from analyzing the structure of polytope P. For our algorithm, however, the scalars λ_i are not discrete enough. Hence, we use the following lemma to round y to a more discrete assignment y'.

▶ **Lemma 5.** *Let $\ell \in \mathbb{N}_{\ge 0}$ and $y = \sum_{i\in[k]} \lambda_i S_i$ where $\lambda_i \in [0,1]$, $\sum_{i\in[k]} \lambda_i = 1$, and $S_i \in \{0,1\}^E$. In polynomial time, we can compute $y' = \sum_{i\in[k]} \lambda'_i S_i$ where $\sum_{i\in[k]} \lambda'_i = 1$ and*

$$\begin{aligned} \lambda'_i &\in \tfrac{1}{2^\ell} \cdot \mathbb{Z}_{\ge 0}, && \forall i \in \{1,\dots,k\} && (4)\\ \lambda'_i &= \lambda_i, && \forall i \in \{1,\dots,k\}, \lambda_i \in \{0,1\} && (5)\\ |y_e - y'_e| &\le k \cdot \tfrac{1}{2^\ell}, && \forall e \in E && (6)\\ c^\mathsf{T} y' &\le c^\mathsf{T} y. && && (7) \end{aligned}$$

We prove this lemma by constructing a flow network and the standard argument that integral capacities imply existence of an integral min-cost circulation.

Notably, this is the first time we incur a small error for the linear functions a_j while the cost is preserved. More precisely, we use the lemma with $\ell := 2\log(2k)$. From Equation (6) follows that for all $e \in E$

$$|y_e - y'_e| \le k \cdot \tfrac{1}{2^\ell} \le \tfrac{1}{2k}. \qquad (8)$$

Therefore, the linear functions also slightly change. Using Equations (3) and (8), it holds that for all $j \in \{1,\dots,k\}$

$$|a_j^\mathsf{T} x - a_j^\mathsf{T} y'| = |a_j^\mathsf{T} y - a_j^\mathsf{T} y'| = \sum_{e\in E} (a_j)_e |y'_e - y_e| \le 1. \qquad (9)$$

Since y' is a convex combination of (integral) degree preserving sets in P, we have $y' \in P$. In other words, the scalar rounding in Lemma 5 does in fact preserve the degree of y.

Next, we construct a complete binary tree $\mathcal{T}$ with levels $0, 1, \dots, \ell$, where each node will be labeled with an edge set. When the algorithm finishes, the label of the root will be $X \in \{0,1\}^E$ and satisfy the properties stated in Theorem 2. In the following, we describe how the algorithm TreeMerge creates the labels on $\mathcal{T}$. The lowest level ℓ represents the fractional assignment $y' = \sum_{i\in[k]} \lambda'_i S_i$ where $S_i \in P \cap \{0,1\}^E$ are degree preserving edge sets. As we can write $\lambda'_i = h_i/2^\ell$ for some $h_i \in \mathbb{Z}_{\geq 0}$ and all h_i sum to 2^ℓ, we can naturally label h_i leaves of level ℓ with S_i for all $i \in \{1,\ldots,k\}$. Thus, y' is the average of all labels of level ℓ. For all $j \in \{0,\ldots,\ell-1\}$, the labels of level j are derived from those in level $j+1$ such that each node's label is closely related to those of its two children. Similar to the last level, each level j represents a (fractional) edge set y'_j by taking the average of all labels in this level.

One of the central goals in the construction of y'_j is to guarantee $c^\mathsf{T} y'_j \leq c^\mathsf{T} y'_{j+1}$. We achieve this by creating two complementary labelings of level j and selecting the better of the two. Denote the labels of level $j+1$ by S_{2i-1}, S_{2i} for all $i \in \{1,\ldots,2^j\}$. Here, each pair S_{2i-1}, S_{2i} represents the children of the i-th node in level j. For node i, we construct two potential labels $T_i, T'_i \in \{0,1\}^E$ using a random procedure with the following guarantees.

▶ **Lemma 6.** *Let $S_1, S_2 \in \{0,1\}^E$. There exists a random polynomial time procedure that constructs two random edge sets $T, T' \in \{0,1\}^E$ with the following properties.*

- *It holds that $T + T' = S_1 + S_2$ and $\mathbb{E}(T) = \mathbb{E}(T') = (S_1 + S_2)/2$.*
- *For all $v \in A \cup B$ it holds that $T(\delta(v)), T'(\delta(v)) \in \{\lfloor (S_1(\delta(v)) + S_2(\delta(v)))/2 \rfloor, \lceil (S_1(\delta(v)) + S_2(\delta(v)))/2 \rceil\}$.*
- *For all $v \in A \cup B$ and all $e \in \delta(v)$, there is at most one edge $e' \in \delta(v) \setminus \{e\}$ such that T_e depends on $T_{e'}$. Likewise, there is at most one edge $e' \in \delta(v) \setminus \{e\}$ such that T'_e depends on $T'_{e'}$.*

The lemma can be derived in two steps. First, decompose the symmetric difference of S_1 and S_2 into cycles and paths. Second, for each of cycle and path, randomly select one of the alternating edge sets for T and the other for T'. This random process is similar to other dependent rounding approaches, e.g. [7, 11], except that we also store T' that contains the "opposite" to every decision in T.

We create one fractional assignment from the random edge sets T_i, $i \in \{1,\ldots,2^j\}$, and one from T'_i, $i \in \{1,\ldots,2^j\}$, and pick the lower cost assignment for y'_j. Formally, let

$$z_j = \tfrac{1}{2^j} \sum_{i\in[2^j]} T_i \qquad \text{and} \qquad z'_j = \tfrac{1}{2^j} \sum_{i\in[2^j]} T'_i$$

From the fact that $T_i + T'_i = S_{2i-1} + S_{2i}$, it immediately follows that $(z_j + z'_j)/2 = y'_{j+1}$. If $c^\mathsf{T} z_j \leq c^\mathsf{T} z'_j$, set $y'_j = z_j$. Otherwise, $y'_j = z'_j$. Consequently, we have that

$$c^\mathsf{T} y'_j \leq (c^\mathsf{T} z_j + c^\mathsf{T} z'_j)/2 = c^\mathsf{T} y'_{j+1}. \tag{10}$$

We determine the labels of level j by picking either T_i (if z_j was chosen) or T'_i (if z'_j was chosen). Repeating the procedure for all $j \in \{\ell-1,\ldots,0\}$ results in a label for the root node that is identical to y'_0. We conclude by setting $X = y'_0$. As a last step before proving the main theorem, we bound how much the linear functions a_j can change in each level.

▶ **Lemma 7.** *Let $v \in A \cup B$ and $a \in [0,1]^E$ with $\operatorname{supp}(a) \subseteq \delta(v)$. Let $y'_{j+1} \in [0,1]^E$ be the fractional solution of the $(j+1)$-th level of TreeMerge and $y'_j \in [0,1]^E$ that of the j-th level. Let $t = 132 \ln k$. Then with probability at least $1 - 1/k^{10}$, it holds that*

$$|a^\mathsf{T} y'_j - a^\mathsf{T} y'_{j+1}| \leq 2^{-j/2} \left(t + \sqrt{a^\mathsf{T} y'_{j+1} \cdot t} \right). \tag{11}$$

This lemma follows from a standard Chernoff bound. We are now in the position to prove Theorem 2 using the lemmas above.

Proof of Theorem 2. Let $\ell = 2\log(2k)$. As explained throughout Section 2, we use Lemmas 4 and 5 to obtain a fractional degree preserving assignment $y' \in P$ with $c^\mathsf{T} y' \leq c^\mathsf{T} x$. Note that the rounding of x to y' marginally changes the linear function values, but we are able to maintain $|a_j^\mathsf{T} y' - a_j^\mathsf{T} x| \leq 1$, see Equation (9). Afterwards, we use the algorithm TreeMerge to construct a complete binary tree $\mathcal{T}$ with $\ell + 1$ levels and corresponding fractional solutions $y'_{\ell+1} = y', y'_\ell, \ldots, y'_0 = X$. It remains to show that X satisfies all three properties from the theorem. By construction, more precisely Equation (10), it holds that

$$c^\mathsf{T} X = c^\mathsf{T} y'_0 \leq \cdots \leq y'_{\ell+1} = c^\mathsf{T} y' \leq c^\mathsf{T} x.$$

Thus, the cost is preserved. Next, we will show that the rounding of x to X also preserves the degree. Due to Lemmas 4, 5, and 8, we have that $y' = \sum_{i \in [k]} \lambda'_i S_i$, where $S_i \in P \cap \{0,1\}^E$ form the labels of level ℓ of $\mathcal{T}$. For all $i \in \{1, \ldots, k\}$, the fact that $S_i \in P$ implies

$$S_i\big(\delta(v)\big) \in \{\lfloor x\big(\delta(v)\big) \rfloor, \lceil x\big(\delta(v)\big) \rceil\}. \tag{12}$$

By induction over the tree $\mathcal{T}$, we show that all labels and, in particular, X are indeed degree preserving. Equation (12) proves the base case. Let S_1, S_2 be the labels for two children of some node in $\mathcal{T}$ and T, T' be the two potential labels for the said node (derived using Lemma 6). From the third property of Lemma 6 directly follows that for all $v \in A \cup B$

$$T(\delta(v)), T'(\delta(v)) \in \Big\{\lfloor \tfrac{1}{2} S_1\big(\delta(v)\big) + \tfrac{1}{2} S_2\big(\delta(v)\big) \rfloor, \lceil \tfrac{1}{2} S_1\big(\delta(v)\big) + \tfrac{1}{2} S_2\big(\delta(v)\big) \rceil\Big\}. \tag{13}$$

By induction hypothesis, S_1 and S_2 are degree preserving, so

$\lfloor \tfrac{1}{2} S_1\big(\delta(v)\big) + \tfrac{1}{2} S_2\big(\delta(v)\big) \rfloor \geq \lfloor x(\delta(v)) \rfloor$ and similarly
$\lceil \tfrac{1}{2} S_1\big(\delta(v)\big) + \tfrac{1}{2} S_2\big(\delta(v)\big) \rceil \leq \lceil x(\delta(v)) \rceil$.

This concludes the induction step.

It remains to prove the concentration, i.e., that X marginally deviates from x in each of the given linear functions a_j. We apply Lemma 7 together with a union bound over all k linear functions and all $\ell = 2\log(2k) \leq k$ levels of $\mathcal{T}$. Let $t = 30\log k$. As a consequence, with probability at least $1 - 1/k^8$, it holds for all levels $i \in \{1, \ldots, \ell\}$ and linear functions $a_j, j \in \{1, \ldots, k\}$ that

$$|a_i^\mathsf{T} y'_{i+1} - a_i^\mathsf{T} y'_i| \leq 2^{-i/2}\Big(t + \sqrt{a^\mathsf{T} y'_{i+1} \cdot t}\Big). \tag{14}$$

For some universal constant d, we prove that for all $j \in \{1, \ldots, k\}$ and all $i \in \{0, \ldots, \ell\}$

$$a_j^\mathsf{T} y'_i \leq d(1 + t + a^\mathsf{T} x). \tag{15}$$

Let us first argue that this in fact implies the last part of the theorem. Using triangle inequality and geometric series, it holds that

$$\begin{aligned}
|a_j^\mathsf{T} X - a_j^\mathsf{T} x| = |a_j^\mathsf{T} y'_{\ell+1} - a_j^\mathsf{T} y'_\ell| &\leq \sum_{i \in \{0,\ldots,\ell-1\}} |a_j^\mathsf{T} y'_{i+1} - a_j^\mathsf{T} y'_i| \\
&\leq \sum_{i \in \{0,\ldots,\ell-1\}} \frac{1}{(\sqrt{2})^i}\Big(t + \sqrt{a^\mathsf{T} y'_{i+1} \cdot t}\Big) \\
&\leq \sum_{i \in \{0,\ldots,\ell-1\}} \frac{1}{(\sqrt{2})^i}\Big(t + \sqrt{d(1 + t + a^\mathsf{T} x) \cdot t}\Big) \\
&= O(\max\{\log k, \sqrt{a^\mathsf{T} x \cdot \log k}\}).
\end{aligned}$$

Finally, we prove Equation (15). Let $j \in \{1, \ldots, k\}$ and $i \in \{0, \ldots, \ell - 1\}$. Let $i' \geq i$ be the minimal index such that $a_j^\intercal y'_{i'} \leq 1 + t + a_j^\intercal x$. As $a_j^\intercal y'_\ell = a_j^\intercal y' \leq 1 + a_j^\intercal x$, such i' must exist. If $i' = i$, we are done. If $i' \neq i$, then it follows from Equation (14) that

$$a_j^\intercal y'_{i'-1} \leq (1 + t + a_j^\intercal x) + 2^{-(i'-1)/2}\big(t + (1 + t + a_j^\intercal x)\big) \leq 3(1 + t + a_j^\intercal x) \ .$$

For all $i'' \in \{i' - 1, \ldots, i\}$, we have $a_j^\intercal y'_{i''} > t$. Thus, the same equation implies

$$a_j^\intercal y'_{i''-1} \leq a_j^\intercal y'_{i''} + \frac{2}{2^{(i''-1)/2}} a_j^\intercal y'_{i''} = \left(1 + \frac{1}{2^{(i''-3)/2}}\right) a_j^\intercal y'_{i''} \ .$$

Using the inequality $1 + z \leq \exp(z)$ for all $z \in \mathbb{R}$, we have

$$\begin{aligned}
a_j^\intercal y'_i &\leq a_j^\intercal y'_{i'-1} \cdot \prod_{i''=i'-1}^{i+1} \left(1 + \frac{1}{2^{(i''-3)/2}}\right) \\
&\leq a_j^\intercal y'_{i'-1} \cdot \exp\left(\sum_{i''=i'-1}^{i+1} \frac{1}{2^{(i''-3)/2}}\right) \\
&\leq 3e^{O(1)} \cdot (1 + t + a_j^\intercal x).
\end{aligned}$$

This shows Equation (15) and thereby concludes the proof. ◀

2.1 Omitted Proofs of Dependent Rounding

In this section, we provide the proofs of the previously stated lemmas. Recall that P is the convex hull of integral degree preserving edge sets for $x \in [0, 1]^E$. Let the operator $\oplus$ denote the symmetric difference (i.e., XOR) of two edge sets. We start by showing that P indeed contains x.

▶ **Lemma 8.** *Let $q = |E|$. There exists a convex representation $x = \sum_{i \in [q]} \lambda_i S_i \in P$ where $S_i \in P \cap \{0, 1\}^E$ and $\lambda_i \in [0, 1]$ and $\sum_{i \in [q]} \lambda_i = 1$ such that for all $v \in A \cup B$ and $i \in \{1, \ldots, q\}$ holds that $S_i\big(\delta(v)\big) \in \{\lfloor x\big(\delta(v)\big) \rfloor, \lceil x\big(\delta(v)\big) \rceil\}$.*

Proof. We rely on a standard flow argument. To this end, construct a digraph $D_f = (V_f, A_f)$ as follows. Let $V_f = \{s, t\} \cup V$ be the set of vertices, A_s be a set of arcs directed from s to each $a \in A$, A_t be the set of arcs directed from each $b \in B$ to t, and A_E be a directed variant of E from A to B. Let $A_f = A_s \cup A_E \cup A_t \cup \{(t, s)\}$ be the set of arcs in D_f. Set the capacity interval for arc $(s, a) \in A_s$ as $[\lfloor x(\delta(a)) \rfloor, \lceil x(\delta(a)) \rceil]$ and similarly for each arc $(b, t) \in A_t$ as $[\lfloor x(\delta(b)) \rfloor, \lceil x(\delta(b)) \rceil]$. Moreover, set the capacity interval for each arc $e \in E$ to $[0, 1]$ and for the arc (t, s) to $[0, \infty]$. A function $f : A_f \to \mathbb{R}$ is called a circulation if $f(\delta^{\text{in}}(v)) = f(\delta^{\text{out}}(v))$ for each vertex $v \in V_f$.

One can naturally derive a feasible fractional circulation from the vector $x \in [0, 1]^E$: the flow from s to $a \in A$ is $x(\delta(a))$, which lies in $[\lfloor x(\delta(a)) \rfloor, \lceil x(\delta(a)) \rceil]$, the circulation from $b \in B$ to t is $x(\delta(b))$, which lies in $[\lfloor x(\delta(b)) \rfloor, \lceil x(\delta(b)) \rceil]$, the circulation on each arc $(a, b) \in A_E$ is x_{ab}, and the circulation on arc (t, s) is $x(E)$. Hence, x satisfies the capacity and flow conservation constraints. It is well known, see e.g. [16, Corollary 13.10b], that for integral capacities the set of all feasible circulations (including x) forms an integral polytope. Thus, we can rewrite the circulation above as a convex combination of integral circulations. Each of these integral circulations corresponds to a degree preserving edge set. ◀

Further, there always exists a comparable convex representation which has only few fractional variables in the support of each constraint.

▶ **Lemma 4.** *There exists a convex combination* $y = \sum_{i\in[k]} \lambda_i S_i$ *where* $\lambda_i \in [0,1]$ *and* $\sum_{i\in[k]} \lambda_i = 1$ *and* $S_i \in P \cap \{0,1\}^E$ *with*

$$c^\mathsf{T} y \leq c^\mathsf{T} x \tag{1}$$
$$a_j^\mathsf{T} y = a_j^\mathsf{T} x \qquad \forall j \in \{1,\dots,k\} \tag{2}$$
$$|\{e \in \delta(v) \mid y_e \notin \{0,1\}\}| \leq 2k \qquad \forall v \in A \cup B \tag{3}$$

Proof. First, we argue about the structure of the edges of polytope P. Let $S, T \in \{0,1\}^E$ be the two vertices at the two ends of some edge of P. We claim that $S \oplus T$ is a simple cycle or a simple path. Suppose not. If $S \oplus T$ is acyclic, let D be a maximal path in $S \oplus T$; otherwise let D be a simple cycle contained in $S \oplus T$. Note that $S \oplus D \in P$ and $T \oplus D \in P$ and both points do not lie on the edge between S and T. However, $(S+T)/2 = (T \oplus D + S \oplus D)/2$ contradicts that S and T are connected by an edge.

Let y be an optimal vertex solution of the linear program

$$\begin{aligned} &\min c^\mathsf{T} y \\ &\quad a_j^\mathsf{T} y = a_j^\mathsf{T} x \qquad \forall j \in \{1,\dots,k\} \\ &\quad y \in P \end{aligned}$$

Due to Lemma 8, the linear program is feasible. The solution y must lie on a face F of P with dimension at most k. Consider an arbitrary vertex S of F. Furthermore, let $T_1,\dots,T_h$, $h \leq k$, be the vertices of F such that there is an edge between each T_i and S. Thus, we can write $y = S + \sum_{i\in[h]} \lambda_i (T_i - S)$ where $\lambda_1,\dots,\lambda_h \geq 0$. By our previous argument, $T_i \oplus S$ is a simple cycle or path for each $i \in \{1,\dots,h\}$. In particular, $|(T_i \oplus S) \cap \delta(v)| \leq 2$ for each $v \in A \cup B$. This implies that the last property holds for y. ◀

Allowing a small rounding error, there always exists a convex representation where the scalars are integer multiples of a power of two.

▶ **Lemma 5.** *Let* $\ell \in \mathbb{N}_{\geq 0}$ *and* $y = \sum_{i\in[k]} \lambda_i S_i$ *where* $\lambda_i \in [0,1]$, $\sum_{i\in[k]} \lambda_i = 1$, *and* $S_i \in \{0,1\}^E$. *In polynomial time, we can compute* $y' = \sum_{i\in[k]} \lambda'_i S_i$ *where* $\sum_{i\in[k]} \lambda'_i = 1$ *and*

$$\lambda'_i \in \tfrac{1}{2^\ell} \cdot \mathbb{Z}_{\geq 0}, \qquad \forall i \in \{1,\dots,k\} \tag{4}$$
$$\lambda'_i = \lambda_i, \qquad \forall i \in \{1,\dots,k\}, \lambda_i \in \{0,1\} \tag{5}$$
$$|y_e - y'_e| \leq k \cdot \tfrac{1}{2^\ell}, \qquad \forall e \in E \tag{6}$$
$$c^\mathsf{T} y' \leq c^\mathsf{T} y. \tag{7}$$

Proof. Using a standard flow argument, we construct a digraph $D_f = (V_f, A_f)$ that represents a circulation network. For each arc in $a \in A_f$, we adjust a capacity interval such that every feasible circulation corresponds to a solution that is "close" to x and preserves the cost. Let $V_f = \{s,t\} \cup \{v_i : i \in [k]\}$ where nodes v_i correspond to each S_i. The set of arcs A_f contains arcs (t,s), (s,v_i), and (v_i,t) for $i \in [k]$. A function $f : A_f \to \mathbb{R}$ is called a circulation if $f(\delta^{\text{in}}(v)) = f(\delta^{\text{out}}(v))$ for each vertex $v \in V_f$. Set the capacity interval of arc (t,s) to $[2^\ell, 2^\ell]$ and the one of arcs (s,v_i) and (v_i,t) to $[\lfloor \lambda_i 2^\ell \rfloor, \lceil \lambda_i 2^\ell \rceil]$ for $i \in [k]$. Moreover, define a linear cost function $\text{cost}(s,v_i) = \sum_{e\in E} c_e (S_i)_e$, the contribution of S_i to the total cost. Set $\text{cost}(t,s) = \text{cost}(v_i,t) = 0$. The scalars λ_i induce a natural circulation f: on arcs (s,v_i) and (v_i,t), we send a flow of $\lambda_i 2^\ell$ and on arc (t,s), we send a flow of 2^ℓ.

For integral capacities the set of feasible circulations forms an integral polytope, see e.g. [16, Corollary 13.10b]. Thus, there exists a minimum cost integral circulation. Let $\bar{f}$ be the this integral circulation. We define $\lambda'_i = \bar{f}(s, v_i)/2^\ell$ for each $i \in \{1, \dots, k\}$, then

$$\sum_{i\in[k]} \lambda'_i = \sum_{i\in[k]} \frac{\bar{f}(s, v_i)}{2^\ell} = \frac{1}{2^\ell} \cdot \bar{f}(t, s) = \frac{1}{2^\ell} \cdot 2^\ell = 1.$$

Consequently, λ'_i creates a valid convex combination for y' where each λ'_i is a multiple of $1/2^\ell$. For each $e \in E$

$$|y_e - y'_e| = \left|\sum_{i\in[k]} \lambda_i (S_i)_e - \sum_{i\in[k]} \lambda'_i (S_i)_e\right| = \sum_{i\in[k]} |\lambda_i - \lambda'_i| \cdot (S_i)_e \leq \sum_{i\in[k]} \frac{1}{2^\ell} = k \cdot \frac{1}{2^\ell}.$$

Since the integer circulation minimizes the total cost, we have

$$c^\mathsf{T} y' = \sum_{e\in E} c_e \sum_{i\in[k]} \lambda'_i (S_i)_e = \frac{1}{2^\ell} \sum_{e\in E} \bar{f}(e) \cdot \mathrm{cost}(e) \leq \frac{1}{2^\ell} \sum_{e\in E} f(e) \cdot \mathrm{cost}(e) = c^\mathsf{T} x. \tag{16}$$

Moreover, constructing D_f and solving minimum cost flow can be done in polynomial time [16]. ◀

The TreeMerge algorithm described in Section 2 uses the following lemma to construct new random edge sets while preserving the degree of the former sets.

▶ **Lemma 6.** *Let $S_1, S_2 \in \{0,1\}^E$. There exists a random polynomial time procedure that constructs two random edge sets $T, T' \in \{0,1\}^E$ with the following properties.*

- *It holds that $T + T' = S_1 + S_2$ and $\mathbb{E}(T) = \mathbb{E}(T') = (S_1 + S_2)/2$.*
- *For all $v \in A \cup B$ it holds that $T(\delta(v)), T'(\delta(v)) \in \{\lfloor (S_1(\delta(v)) + S_2(\delta(v)))/2 \rfloor, \lceil (S_1(\delta(v)) + S_2(\delta(v)))/2 \rceil\}$.*
- *For all $v \in A \cup B$ and all $e \in \delta(v)$, there is at most one edge $e' \in \delta(v) \setminus \{e\}$ such that T_e depends on $T_{e'}$. Likewise, there is at most one edge $e' \in \delta(v) \setminus \{e\}$ such that T'_e depends on $T'_{e'}$.*

Proof. For the randomized construction of T and T', we distinguish whether an edge is present in the symmetric difference $S_1 \oplus S_2$ or not. For any edge $e \in E$ where $(S_1)_e = (S_2)_e$, we set $T_e = \big((S_1)_e + (S_2)_e\big)/2$ and $T'_e = \big((S_1)_e + (S_2)_e\big)/2$. Next, partition the edges in $S_1 \oplus S_2$ into simple paths and cycles with $E_1 \dot\cup \cdots \dot\cup E_k = S_1 \oplus S_2$ with the following property: each odd-degree vertex is the endpoint of exactly one path and no path ends in an even-degree vertex. This easily follows from iteratively removing cycles and maximal paths.

Let $i \in \{1, \dots, k\}$. Choose $C_i \dot\cup D_i = E_i$ such that no two edges in C_i or in D_i are adjacent. This is possible by alternatingly assigning edges to C_i and D_i, as E_i is an even length cycle or a path. We make a uniform binary random decision $R_i \in \{0, 1\}$, which is independent of all $R_{i'}$, $i' \neq i$. This random variable indicates whether C_i is assigned to T or T' (D_i is then assigned to the other edge set). More precisely, for each $e \in E_i$ we set

$$T_e = \begin{cases} (C_i)_e & \text{if } R_i = 0, \\ (D_i)_e & \text{if } R_i = 1, \end{cases} \quad \text{and} \quad T'_e = \begin{cases} (D_i)_e & \text{if } R_i = 0, \\ (C_i)_e & \text{if } R_i = 1. \end{cases} \tag{17}$$

In particular, $T_e + T'_e = (C_i)_e + (D_i)_e = 1 = (S_1)_e + (S_2)_e$. Since both outcomes $R_i = 1$ and $R_i = 0$ have probability $1/2$, it holds that $\mathbb{E}(T_e) = \mathbb{E}(T_e) = \big((C_i)_e + (D_i)_e\big)/2 = \big((S_1)_e + (S_2)_e\big)/2$ for all $e \in S \oplus T$. Thus, it follows that $\mathbb{E}(T) = \mathbb{E}(T') = (S_1 + S_2)/2$. Let $\mathcal{F} = \{\delta(v) \mid v \in A \cup B\}$. Considering each vertex in a simple path or cycle is incident to at most two edges, we have $|F \cap E_i| \leq 2$ for all $F \in \mathcal{F}$ and $i \in \{1, \dots, k\}$. As the random choice for each path and cycle is independent, it holds that for all $F \in \mathcal{F}$ and $e \in F$, there is at most one edge $e' \in F$ with $e \neq e'$ such that T_e depends on $T_{e'}$ (and the same for T').

It remains to show that $T(F), T'(F) \in \{\lfloor (S_1(F)+S_2(F))/2 \rfloor, \lceil (S_1(F)+S_2(F))(2) \rceil\}$. To this end, we distinguish whether a vertex $v \in A \cup B$ has even or odd degree. If $|\delta(v)|$ is even, then v is always incident to exactly one edge in C_i and one in D_i, for each $i \in \{1, \ldots, k\}$. Hence, $T(F) = T'(F) = |\delta(v)|/2 = (S_1(F) + S_2(F))/2$. If $|\delta(v)|$ is odd, then there is exactly one path that ends in v due to the choice of decomposition into cycles and paths. Consequently, the arguments above apply to all but one edge in $\delta(v)$ and therefore we have $|T(F) - T'(F)| = 1$. This implies the claim. ◀

The last lemma bounds the change of a linear function between two consecutive levels of TreeMerge.

▶ **Lemma 7.** *Let $v \in A \cup B$ and $a \in [0,1]^E$ with $\operatorname{supp}(a) \subseteq \delta(v)$. Let $y'_{j+1} \in [0,1]^E$ be the fractional solution of the $(j+1)$-th level of TreeMerge and $y'_j \in [0,1]^E$ that of the j-th level. Let $t = 132 \ln k$. Then with probability at least $1 - 1/k^{10}$, it holds that*

$$|a^\mathsf{T} y'_j - a^\mathsf{T} y'_{j+1}| \leq 2^{-j/2} \left(t + \sqrt{a^\mathsf{T} y'_{j+1} \cdot t} \right). \tag{11}$$

Proof. Let T_i, $i \in [2^j]$, be the random edge set created on the j-th level from the edge sets S_{2i-1},S_{2i}, $i \in [2^j]$ of the $(j+1)$-th level. Recall that the procedure from Lemma 6 actually creates two alternative edge sets T_i, T'_i of which TreeMerge selects just one. However, the solution derived from T_i satisfies Equation (11) if and only if the one from T'_i does. Hence, it suffices to show it for one of the two. Further, for the sake of convenience, we analyze the scaled expression $|2^j \cdot a^\mathsf{T} y'_j - 2^j \cdot a^\mathsf{T} y'_{j+1}|$ instead of $|a^\mathsf{T} y'_j - a^\mathsf{T} y'_{j+1}|$. Due to Lemma 6, it holds that $\mathbb{E}(T_i) = (S_{2i-1} + S_{2i})/2$. Thus,

$$\mathbb{E}[2^j \cdot a^\mathsf{T} y'_j] = a^\mathsf{T} \Big(\sum_{i \in [2^j]} \mathbb{E}[T_i] \Big) = \tfrac{1}{2} a^\mathsf{T} \Big(\sum_{i \in [2^{j+1}]} S_i \Big) = 2^j \cdot a^\mathsf{T} y'_{j+1}.$$

Since TreeMerge independently constructs the T_i on the j-th level, any two random edge sets T_i and T_{i+1} are independent. Moreover, it follows from Lemma 6 that for all T_i and $e \in \delta(v)$, there exists at most one edge $e' \in \delta(v)$ such that $(T_i)_e$ depends on $(T_i)_{e'}$. Consequently, there exists a partition of $\delta(v)$ into P_1 and P_2 such that all variables $(T_i)_e$, $e \in P_1$, as well as $(T_i)_e$, $e \in P_2$ are independent. Let $u = \sum_{i \in [2^j]} U_i \in [0,1]^E$ and $w = \sum_{i \in [2^j]} W_i \in [0,1]^E$ where $U_i, W_i \in \{0,1\}^E$ are defined for all $e \in E$ as

$$(U_i)_e = \begin{cases} 1, & \text{if } e \in \delta(v) \text{ and } (T_i)_e \in P_1 \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad (W_i)_e = \begin{cases} 1, & \text{if } e \in \delta(v) \text{ and } (T_i)_e \in P_2 \\ 0, & \text{otherwise.} \end{cases}$$

Hence, we have $2^j \cdot a^\mathsf{T} y'_j = a^\mathsf{T}(u + w)$. Next, we show that

$$\mathbb{P}\left[|a^\mathsf{T} u - \mathbb{E}[a^\mathsf{T} u]| > \tfrac{1}{2} t + \tfrac{1}{2} \sqrt{\mathbb{E}[a^\mathsf{T} u] \cdot t} \right] < \tfrac{1}{2k^{10}}. \tag{18}$$

For brevity, define $\mu = \mathbb{E}[a^\mathsf{T} u]$. We distinguish two cases. If $t > 4\mu$, then set $\delta = t/(4\mu) > 1$. Notice that $a^\mathsf{T} u$ is the sum of independent variables, each contained in $[0,1]$. We have

$$\mathbb{P}\left[a^\mathsf{T} u < \mu - t/4 \right] = 0$$

It follows from a standard Chernoff bound that

$$\begin{aligned}\mathbb{P}\left[a^\mathsf{T}u > \mu + t/4\right] &= \mathbb{P}\left[a^\mathsf{T}u > (1+\delta)\mu\right] \\ &\le \exp(-\tfrac{1}{3}\delta\mu) \\ &= \exp(-11\ln k) \\ &\le \tfrac{1}{2k^{10}}.\end{aligned}$$

If $t \le 4\mu$, then set $\delta = 1/2 \cdot \sqrt{t/\mu} \in (0,1]$. Again by a Chernoff bound, we have

$$\begin{aligned}\mathbb{P}\left[|a^\mathsf{T}u - \mu| > \tfrac{1}{2}\sqrt{\mu t}\,\right] &= \mathbb{P}\left[|a^\mathsf{T}u - \mu| > \delta\mu\right] \\ &\le 2\exp(-\tfrac{1}{3}\delta^2\mu) \\ &= 2\exp(-11\ln k) \le \tfrac{1}{2k^{10}}.\end{aligned}$$

By symmetry, Equation (18) holds for w as well. We conclude that with probability at least $1 - 1/k^{10}$, we have

$$\begin{aligned}|2^j a^\mathsf{T} y'_j - 2^j a^\mathsf{T} y'_{j+1}| &\le |a^\mathsf{T}u - \mathbb{E}[a^\mathsf{T}u]| + |a^\mathsf{T}w - \mathbb{E}[a^\mathsf{T}w]| \\ &\le \tfrac{1}{2}t + \tfrac{1}{2}\sqrt{\mathbb{E}[a^\mathsf{T}u]\cdot t} + \tfrac{1}{2}t + \tfrac{1}{2}\sqrt{\mathbb{E}[a^\mathsf{T}w]\cdot t} \\ &\le t + \sqrt{\mathbb{E}[2^j \cdot a^\mathsf{T}y'_j]\cdot t} \\ &\le 2^{j/2}\big(t + \sqrt{a^\mathsf{T}y'_{j+1}\cdot t}\big).\end{aligned}$$ ◀

3 Application to Budgeted Santa Claus Problem

In this section, we present our approximation algorithm the Budgeted Santa Claus Problem based on the dependent rounding scheme described in Section 2.

3.1 Linear Programming Formulation

Introducing an LP relaxation for the Budgeted Santa Claus Problem, we first reduce the problem to its decision variant. For a given threshold $T \ge 0$, the goal is to either find a solution of value T/α or determine that OPT $< T$, where OPT is the optimal value of the original optimization problem. This variant is equivalent to an α-approximation algorithm by a standard binary search framework.

Based on T, intuitively thought of as the optimal value, we partition the resources into two sets by size. Set $\mathcal{B}$ consists of the *big resources* with $\mathcal{B} := \{j \in \mathcal{R} : v_j \ge T/\alpha\}$ and set $\mathcal{S}$ consists of the *small resources* $\mathcal{S} := \{j \in \mathcal{R} : v_j < T/\alpha\}$. We use assignment variables that indicate whether a particular resource is assigned to a particular player. For clarity, we use different symbols for big and small resources. Let $x_{ib} \in [0,1]$ denote the portion of big resource $b \in \mathcal{B}$ that player $i \in \mathcal{P}$ receives. Similarly, denote by $z_{is} \in [0,1]$ the portion of the small resource $s \in \mathcal{S}$ that player i receives. Unlike the original problem, we allow these assignments to be fractional in the relaxation. Since naive constraints on these variables lead to an unbounded integrality gap, see e.g. [6], we use non-trivial constraints inspired by an LP formulation of Davies, Rothvoss and Zhang [9]. Here, we make the structural assumption that in any solution, a player either receives exactly one big resource (and nothing else) or only small resources. Towards the goal of obtaining a solution of value T/α, any big resource is sufficient for any player and receiving more would be wasteful. If there is a solution of value T, then there is also a pseudo-solution such that each player either gets exactly one big resource (and nothing else) or a value of at least T from small resources only. Note that it might be that the former type of player only has a value of T/α. Thus, if $T \leq \mathrm{OPT}$, then the following relaxation called economical(T) is feasible and has a value at most C.

$$\min \sum_{i\in\mathcal{P}} \left[\sum_{b\in\mathcal{B}} c_{ib} \cdot x_{ib} + \sum_{s\in\mathcal{S}} c_{is} \cdot z_{is}\right] \tag{19}$$

$$\sum_{s\in\mathcal{S}} v_s \cdot z_{is} \geq T \cdot \left(1 - \sum_{b\in\mathcal{B}} x_{ib}\right) \qquad \forall i \in \mathcal{P} \tag{20}$$

$$z_{is} \leq 1 - \sum_{b\in\mathcal{B}} x_{ib} \qquad \forall s \in \mathcal{S}, i \in \mathcal{P} \tag{21}$$

$$\sum_{i\in\mathcal{P}} x_{ib} \leq 1 \qquad \forall b \in \mathcal{B} \tag{22}$$

$$\sum_{i\in\mathcal{P}} z_{is} \leq 1 \qquad \forall s \in \mathcal{S} \tag{23}$$

$$\sum_{b\in\mathcal{B}} x_{ib} \leq 1 \qquad \forall i \in \mathcal{P} \tag{24}$$

$$z_{is}, x_{ib} \geq 0 \qquad \forall s \in \mathcal{S}, b \in \mathcal{B}, i \in \mathcal{P} \tag{25}$$

The constraints (22) and (23) describe that each big or small resource is only assigned once. Justified by earlier arguments, constraint (24) ensures that each player receives at most one big resource. Considering constraint (21), we only need to verify that the constraint is valid for integer solutions. By our assumption, if player i receives one big resource, then it should not get any small resources, which is exactly what the constraint expresses. Conversely, if the player does not receive any big resources, the constraint is trivially satisfied. Similarly, there are two cases for Constraints (20). If player i receives one big resource, the constraint is trivially satisfied. Otherwise, it must receive a value of at least T in small resources. During our rounding procedure in Section 3.4, we essentially lose some value from small resources and can only guarantee a value of T/α for players without a big resource.

3.2 Technical Goals

Let (x, z) be a feasible solution to economical(T), where x and z represent the vectors of assignment variables corresponding to big and small resources, respectively. Formally, we have $x_{ib}, z_{is} \in [0, 1]$ for $i \in \mathcal{P}, b \in \mathcal{B}, s \in \mathcal{S}$. We will define a randomized rounding procedure that constructs a distribution over the binary variables $X_{ib} \in \{0, 1\}$ and $Z_{is} \in \{0, 1\}$ describing whether a big resource $b \in \mathcal{B}$ or small resource $s \in \mathcal{S}$ is assigned to player $i \in \mathcal{P}$. For notational convenience, define $Y_i = 1 - \sum_{b\in B} X_{ib}$ as the indicator variable whether a player i *does not* get a big resource (and needs small resource). Similarly, let $y_i = 1 - \sum_{b\in\mathcal{B}} x_{ib}$ be the corresponding value to Y_i from the corresponding value from the LP variables.

Our goal is that the total cost of assignments does not exceed the budget C and the integral solution (X, Z) is an α-approximation solution with respect to the minimum value a player receives. In other words, we want to obtain an integral solution (X, Z) that satisfies the following two properties.

1. $\sum_{i\in\mathcal{P}} \left[\sum_{b\in\mathcal{B}} c_{ib} \cdot X_{ib} + \sum_{s\in\mathcal{S}} c_{is} \cdot Z_{is}\right] \leq C$.
2. Every player receives resources of value at least T/α with high probability.

We first apply our dependent rounding scheme to round the assignment of big resources to an integral one. To cover the players that do not receive big resources, i.e., those with $Y_i = 1$, we need to change the assignment of small resources as well. Initially, some small resources will be assigned fractionally and even more than once. In a second step, we transform the solution into one where each small resource is assigned only once—incurring a loss in the value that the players receive.

3.3 Rounding of Big Resources

The following lemma summarizes the properties we derive from the dependent rounding scheme. Note that while the assignment of small resources can change, it remains fractional for now.

▶ **Lemma 9.** *Let (x, z) be a feasible solution to* economical(T, C)*. There is a randomized algorithm that produces an assignment $X_{ib} \in \{0, 1\}$, $i \in \mathcal{P}$, $b \in \mathcal{B}$, and $z'_{is} \in [0, 1]$, $i \in \mathcal{P}$, $s \in \mathcal{S}$ such that with high probability*

1. $\sum_{i\in\mathcal{P}} \sum_{b\in\mathcal{B}} c_{ib} \cdot X_{ib} + \sum_{i\in\mathcal{P}} \sum_{s\in\mathcal{S}} c_{is} z'_{is} \leq C$,
2. *Each big resource is assigned at most once, i.e., $\forall b \in \mathcal{B} : \sum_{i\in\mathcal{P}} X_{ib} \leq 1$,*
3. *Each small resource is assigned at most $O(\log n)$ times, i.e., $\forall s \in \mathcal{S} : \sum_{i\in\mathcal{P}} z'_{is} \leq O(\log n)$,*
4. *Each player receives either one big resource or a value of at least T in small resources.*

Proof. We first describe the instance to which we will apply the dependent rounding procedure from Theorem 2. We build a bipartite graph $G_{x,z} = (\mathcal{P} \cup (\mathcal{B} \cup \{d\}), E)$, where $\mathcal{P}$ is the set of players, $\mathcal{B}$ is the set of big resources, and d is a *dummy node*. The role of d can be summarized as: every player who does not get a big resource selects the edge to d. Let graph $G_{x,z}$ contain an edge $(i, b) \in E$ labeled with cost c_{ib} between each big resource b and player i with $x_{ib} > 0$. Let J denote the set of all these edges. For every player i with $y_i > 0$, add an edge (i, d) to the dummy node of cost $\sum_{s\in\mathcal{S}} c_{is} \cdot (z_{is}/y_i) \in \mathbb{R}^n$. Let K denote the set of these edges and set $E = J \cup K$. Intuitively, x_{ib} is a fractional edge selection of edges in J and y_i a fractional edge selection of K (where y_i corresponds to edge $(i, d) \in K$). This fractional edge set satisfies the following.

(i) For each player i, we have $\sum_{b\in\mathcal{B}} x_{ib} + y_i = 1$ due to the definition of y_i.
(ii) For each big resource b, we have $\sum_{i\in\mathcal{P}} x_{ib} \leq 1$, due to Constraint (22).

Applying the dependent rounding procedure of Theorem 2 with $k = |\mathcal{S}| = n$ linear functions (we defer a precise definition to the end of the proof), we obtain an integral edge selection $X_{ib} \in \{0, 1\}$, $i \in \mathcal{P}$, $b \in \mathcal{B}$ and $Y_i \in \{0, 1\}$, $i \in \mathcal{P}$. Here, similar to before, Y_i defines the edge selection in K, i.e., edge (i, d) is selected if $Y_i = 1$. From (ii) and Theorem 2, it follows that $\sum_{i\in\mathcal{P}} X_{ib} \leq 1$ for all $b \in \mathcal{B}$, which means that Property 2 of the lemma holds. Further,

$$\begin{aligned}
\sum_{i\in\mathcal{P}} \sum_{b\in\mathcal{B}} c_{ib} \cdot X_{ib} + \sum_{i\in\mathcal{P}: y_i>0} Y_i \cdot (\sum_{s\in\mathcal{S}} c_{is} \cdot \frac{z_{is}}{y_i}) &\leq \sum_{i\in\mathcal{P}} \sum_{b\in\mathcal{B}} c_{ib} \cdot x_{ib} + \sum_{i\in\mathcal{P}: y_i>0} (\sum_{s\in\mathcal{S}} c_{is} \cdot \frac{z_{is}}{y_i}) \cdot y_i \\
&\leq \sum_{i\in\mathcal{P}} \sum_{b\in\mathcal{B}} c_{ib} \cdot x_{ib} + \sum_{i\in\mathcal{P}} \sum_{s\in\mathcal{S}} c_{is} \cdot z_{is} \\
&\leq C,
\end{aligned}$$

where the first inequality follows from the cost preservation property of Theorem 2 and the last inequality from the feasibility of (x, z). Notably, if $y_i = 0$, then $Y_i = 0$ as there does not exist an edge (i, d). In particular, $Y_i = 1$ implies that $y_i > 0$. We define the assignment of small resources as

$$z'_{is} = \begin{cases} z_{is}/y_i & \text{if } Y_i = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Due to Constraint (21), we have $z_{is} \leq y_i$ and hence $z'_{is} \in [0, 1]$. Consequently, Property 1 immediately follows from the previous cost calculation and the definition of z'_{is}. From (i) and Theorem 2, we obtain $\sum_{b \in \mathcal{B}} X_{ib} + Y_i = 1$ for each $i \in \mathcal{P}$. In words, player i either gets a big resource or $Y_i = 1$. In the latter case holds that

$$\sum_{s \in \mathcal{S}} v_s z'_{is} = \sum_{s \in \mathcal{S}} v_s z_{is}/y_i \geq T.$$

Here, the inequality follows from the definition of y_i and Constraint (20). Therefore, Property 4 holds.

It remains to show Property 3, i.e, that every small resource is assigned at most $O(\log n)$ times. To this end, we define the linear functions provided to Theorem 2: for each small resource $s \in \mathcal{S}$, there is one linear function $a^{(s)} \in [0, 1]^K$, i.e., specified by the edges incident to d. Implicitly, the coefficient for all other edges is zero. Thus, the linear function satisfies the support restriction of Theorem 2. For $e = (i, d) \in K$, we define $a_e^{(s)} = z_{is}/y_i$. Using Constraint (21), it holds that

$$\sum_{e=(i,d) \in K} y_i \cdot a_e^{(s)} = \sum_{e=(i,d) \in K} y_i \cdot \frac{z_{is}}{y_i} = \sum_{e=(i,d) \in K} z_{is} \leq 1.$$

Finally, from the concentration bound of Theorem 2 follows

$$\sum_{i \in \mathcal{P}} z'_{is} = \sum_{e=(i,d) \in K} Y_i \cdot z'_{is} = \sum_{e=(i,d) \in K} Y_i \cdot z_{is}/y_i = \sum_{e=(i,d) \in K} Y_i \cdot a_e^{(s)} \leq O(\log n). \qquad \blacktriangleleft$$

3.4 Rounding of Small Resources

In the previous subsection, we described how big resources $b \in \mathcal{B}$ were integrally assigned to the players. As some players did not receive any big resources and still need to be covered by small resources, let $\mathcal{Q}$ be the set of those players, i.e., players $i \in \mathcal{P}$ for which $Y_i = 1$. The linear program in the following lemma corresponds to the property of the assignment variables for small resources from the previous section.

▶ **Lemma 10.** *Let $\mathcal{Q} \subseteq \mathcal{P}$ and consider the LP* small(T, β) *defined as*

$$\begin{aligned} \sum_{s \in \mathcal{S}} v_s \cdot z'_{is} &\geq T && \forall i \in \mathcal{P} \\ \sum_{i \in \mathcal{Q}} z'_{is} &\leq \beta && \forall s \in \mathcal{S} \\ z'_{is} &\geq 0 && \forall i \in \mathcal{Q}, s \in \mathcal{S} \end{aligned}$$

If small(T, β) *has a fractional solution z', then* small$(T/\beta - \max_{s \in \mathcal{S}} v_s, 1)$ *has an integral solution Z that can be found in polynomial time with*

$$\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{Q}} c_{is} \cdot Z_{is} \leq \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{Q}} c_{is} \cdot z'_{is},$$

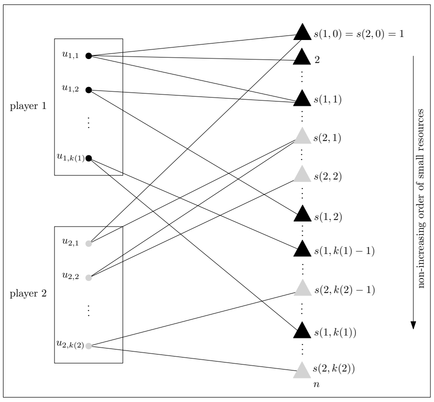

Figure 1 Bipartite graph G where on left side there are $k(i)$ copies of each player i for an instance of two players with edges $(u_{i,\ell}, s)$ of weight c_{is} for each $s = s(i,\ell-1), s(i,\ell-1)+1, \dots, s(i,\ell)$.

Proof. Since small(T,β) is feasible, there exists a solution z'. Further, we obtain a (fractional) solution z'' to small$(T/\beta, 1)$ by dividing all variables of z' by β. Using the approach by Shmoys and Tardos [17], we round z'' to an integral solution. For the sake of completeness, we provide the proof below.

Assume without loss of generality that $\mathcal{S} = \{1, 2, \dots, |\mathcal{S}|\}$ is ordered such that $v_s \geq v_{s-1}$ for all $s = 2, 3, \dots, |\mathcal{S}|$. We construct an auxiliary bipartite graph G (see Figure 1 for an illustration). The elements on the right side of G are elements of $\mathcal{S}$. On the left side, there are $k(i) := \lfloor \sum_{s\in\mathcal{S}} z''_{is} \rfloor$ many vertices $u_{i,1}, \dots, u_{i,k(i)}$ for each $i \in \mathcal{Q}$. In the fractional solution z'', a player Q can get several small resources. Let $k(i)$ denote their (rounded) number. Introducing $u_{i,1}, \dots, u_{i,k(i)}$ as copies of player $i \in Q$ allows us to argue about matchings (where every vertex is only involved in one assignment). Suppose we add one edge between every two vertices of the different sides of the bipartite graph. It is straight-forward that there exists a (fractional) left-perfect matching by distributing the resources assigned in z'' among the copies of each player. Due to the rounding in $k(i)$, this matching gives a slightly lower value to each player, but stays within the bounds we are aiming for.

To round to a good integral matching, however, we require a specific definition of the fractional left-perfect matching. Essentially, we need a monotone assignment where the first copy $u_{i,1}$ of player $i \in Q$ has the highest value resources (the first in the order above) and the last player the lowest value resources. Then G only contains the edges that are actually used in this assignment.

This requires some careful definitions. For each $i \in \mathcal{Q}$, we set $s(i,0) = 1$. This describes the first resource that can be fractionally assigned to player i. For $\ell = 1, 2, \dots, k(i)$, we choose $s(i,\ell)$ as the element in $\mathcal{S}$ that satisfies

$$z''_{i,1} + \dots + z''_{i,s(i,\ell)-1} < \ell \text{ and } z''_{i,1} + \dots + z''_{i,s(i,\ell)} \geq \ell.$$

Note that $s(i,\ell)$ exists, because the sum of all $z''_{i,j}$ is at least $k(i) \geq \ell$. We only assign resources $s(i,\ell-1), \dots, s(i,\ell)$ to copy $u_{i,\ell}$. Intuitively, the resources $1, 2, \dots, s(i,\ell-1)-1$ should be exclusively used for the copies $u_{i,1}, \dots, u_{i,\ell-1}$. Simply because the sum of fractions associated with i (according to z'') is not enough to cover the copies and $u_{i,\ell}$ should not receive any of them in order to maintain the monotonicity goals. Similarly, as the sum of fractions of resources $1, 2, \dots, s(i,\ell)$ belonging to i exceeds ℓ, they are enough to cover all players in $u_{i,1}, \dots, u_{i,\ell}$. Thus, it is not necessary to give any less valuable resources to $u_{i,\ell}$.

Consequently, for all $i \in \mathcal{Q}$ and $\ell = 1, 2, \dots, k(i)$, we introduce an edge $(u_{i,\ell}, s)$ of weight c_{is} for each $s = s(i, \ell - 1), s(i, \ell - 1) + 1, \dots, s(i, \ell)$. We now formally show that there is a left-perfect fractional matching of weight at most $\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{Q}} c_{is} \cdot z''_{is}$. Towards this, consider some $i \in \mathcal{Q}$ and $\ell \in \{1, 2, \dots, k(i)\}$. We select edge $(u_{i,\ell}, s(i, \ell - 1))$ to an extend of

$$z''_{i,1} + \dots + z''_{i,s(i,\ell-1)} - (\ell - 1) \in [0, z''_{i,s(i,\ell-1)}].$$

For $s = s(i, \ell - 1) + 1, \dots, s(i, \ell) - 1$, we pick edge $(u_{i,\ell}, s)$ to an extend of $z''_{i,s}$. Finally, we choose edge $(u_{i,\ell}, s(i, \ell))$ to an extend of

$$\ell - (z''_{i,1} + \dots + z''_{i,s(i,\ell)-1}) \in [0, z''_{i,s(i,\ell)}].$$

This fractional selection of edges satisfies the following properties.

1. For each $i \in \mathcal{Q}$ and $\ell \in \{1, 2, \dots, k(i)\}$, the total fractional amount of selected edges that are incident to $u_{i,\ell}$ is exactly 1.
2. For each $i \in \mathcal{Q}$ and $s \in \mathcal{S}$, the total fractional amount of selected edges that are between $u_{i,1}, \dots, u_{i,k(i)}$ and s is at most $z''_{i,s}$.

Property 2 implies that the total fractional amount that edges incident to s (over all $i \in \mathcal{Q}$) are selected is at most 1. Hence, the total weight is at most the cost of z''. For a bipartite graph, the set of all fractional matchings is precisely the convex hull of integral matchings, see e.g. [16, Chapter 18]. Furthermore, z'' must lie on a facet spanned by only left-perfect matchings. Thus, there must also exist an integral left-perfect matching M of weight at most

$$\sum_{i \in \mathcal{Q}} \sum_{s \in \mathcal{S}} z''_{i,s} c_{i,s} \leq \sum_{i \in \mathcal{Q}} \sum_{s \in \mathcal{S}} z'_{i,s} c_{i,s}.$$

We can find such a matching using standard algorithms for minimum weight bipartite matching. We interpret M as an integral assignment Z where each $i \in \mathcal{Q}$ receives all $s \in \mathcal{S}$, for which there is an edge $(u_{i,\ell}, s)$ in M for some $\ell \in \{1, 2, \dots, k(i)\}$. Finally, we analyze how much value each $i \in \mathcal{Q}$ receives in this assignment. Notice that all resources $s \in \mathcal{S}$ that are connected to $u_{i,\ell}$ have a value of at least $v_{s(i,\ell)}$. Therefore, i receives a total value of at least

$$v_{s(i,1)} + \dots + v_{s(i,k(i))}.$$

On the other hand, since $z''_{i,1} + \dots + z''_{i,s(i,\ell)-1} + z''_{i,s(i,\ell)} < \ell + z''_{i,s(i,\ell)} \leq \ell + 1$ for each ℓ, the total fractional amount of resources that i receives in z'' with value at least $v_{s(i,\ell)}$ is less than $\ell + 1$. As a result, the total value that i received in z'' is at most

$$v_{s(i,0)} + \dots + v_{s(i,k(i))}.$$

This is at least T/β, because of $z'' \in \mathrm{small}(T/\beta, 1)$. As a consequence, in Z player i receives a value of at least $T/\beta - v_{s(i,0)} \geq T/\beta - \max_{s \in \mathcal{S}} v_s$. ◀

3.5 Approximation Factor

Concluding the previous subsections, this section provides an α-approximation for the Budgeted Santa Claus Problem, where $\alpha = O(\log n)$.

▶ **Theorem 3.** *There is a randomized polynomial time $O(\log n)$-approximation algorithm for the Budgeted Santa Claus problem.*

Proof. Let $\beta = O(\log n)$ be the value from Property 3 of Lemma 9. We define $\alpha = 2\beta$ and run our binary search over value T, which is then used in our definition of big and small resources. Then we solve the LP relaxation economical(T). Let (x, z) be the resulting solution. If the cost of the solution is more than C, we fail (and increase the value of T). Assume now that (x, z) has a cost of at most C. In Lemma 9, using the dependent rounding procedure from Theorem 2, we show that the fractional assignments x of big resources to players can be rounded to an integral assignment X and the assignment of small resources z can be changed to z' (which is still fractional), such that with high probability each small resource up to β times and the cost is still below C. Lemma 10 proves that z' can be rounded to an integral assignment Z such that the cost does not increase and each player i that does not receive a big resource gets small resources of total value at least

$$\frac{T}{\beta} - \max_{s \in \mathcal{S}} v_s \geq \frac{2T}{\alpha} - \frac{T}{\alpha} = \frac{T}{\alpha}. \quad \blacktriangleleft$$

4 Conclusion

Based on the finding in this paper there are several interesting questions arising for future research. For the Budgeted Santa Claus Problem, a naturally arising question is whether the approximation factor of $O(\log n)$ can be improved. Notably, the special case of restricted assignment (without a budget constraint) admits a constant approximation due to Feige [10]. We are not aware of any hardness results indicating that such a result cannot hold for our problem. As an intermediate question, one could look at the bi-criteria approximation that approximates both the minimum player value and the cost by a constant. This would still generalize the aforementioned algorithm for restricted assignment.

Another intriguing question is whether a dependent rounding scheme exists for rounding matroid bases that simultaneously guarantees cost preservation and Chernoff-type concentration like SwapRounding [7] does. It seems likely that the techniques from this paper transfer at least to a limited class of matroids, namely strongly base-orderable matroids, because these have very strong decomposition properties for the symmetric difference of two bases. It might, however, require other ideas to generalize to arbitrary matroids.

References

1. Chidambaram Annamalai, Christos Kalaitzis, and Ola Svensson. Combinatorial algorithm for restricted max-min fair allocation. *ACM Trans. Algorithms*, 13(3):37:1–37:28, 2017. doi:10.1145/3070694.
2. Arash Asadpour, Uriel Feige, and Amin Saberi. Santa claus meets hypergraph matchings. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques, 11th International Workshop (APPROX 2008) and 12th International Workshop (RANDOM 2008)*, volume 5171 of *Lecture Notes in Computer Science*, pages 10–20. Springer, 2008. doi:10.1007/978-3-540-85363-3_2.
3. Arash Asadpour, Michel X. Goemans, Aleksander Madry, Shayan Oveis Gharan, and Amin Saberi. An $O(\log n/\log\log n)$-approximation algorithm for the asymmetric traveling salesman problem. *Oper. Res.*, 65(4):1043–1061, 2017. URL: https://doi.org/10.1287/opre.2017.1603, doi:10.1287/OPRE.2017.1603.
4. Étienne Bamas, Alexander Lindermayr, Nicole Megow, Lars Rohwedder, and Jens Schlöter. Santa claus meets makespan and matroids: Algorithms and reductions. In *35th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA 2024)*, pages 2829–2860. SIAM, 2024. doi:10.1137/1.9781611977912.100.
5. Nikhil Bansal and Viswanath Nagarajan. Approximation-friendly discrepancy rounding. In *18th Integer Programming and Combinatorial Optimization Conference (IPCO 2016)*, volume 9682 of *Lecture Notes in Computer Science*, pages 375–386. Springer, 2016. doi:10.1007/978-3-319-33461-5_31.

6 Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *38th Annual ACM Symposium on Theory of Computing, (STOC 2006)*, pages 31–40. ACM, 2006. doi:10.1145/1132516.1132522.

7 Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*, pages 575–584. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.60.

8 Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011)*, pages 1080–1097. SIAM, 2011. doi:10.1137/1.9781611973082.82.

9 Sami Davies, Thomas Rothvoss, and Yihao Zhang. A tale of santa claus, hypergraphs and matroids. In *31st Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA 2020)*, pages 2748–2757. SIAM, 2020. doi:10.1137/1.9781611975994.167.

10 Uriel Feige. On allocations that maximize fairness. In *19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2008)*, pages 287–293. SIAM, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347114.

11 Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *J. ACM*, 53(3):324–360, 2006. doi:10.1145/1147954.1147956.

12 Fabrizio Grandoni and Rico Zenklusen. Approximation schemes for multi-budgeted independence systems. In *18th Annual European Symposium (ESA 2010)*, volume 6346 of *Lecture Notes in Computer Science*, pages 536–548. Springer, 2010. doi:10.1007/978-3-642-15775-2_46.

13 Klaus Jansen, Kim-Manuel Klein, and José Verschae. Closing the gap for makespan scheduling via sparsification techniques. *Math. Oper. Res.*, 45(4):1371–1392, 2020. URL: https://doi.org/10.1287/moor.2019.1036, doi:10.1287/MOOR.2019.1036.

14 Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. *SIAM J. Comput.*, 44(5):1573–1582, 2015. doi:10.1137/130929400.

15 Barna Saha and Aravind Srinivasan. A new approximation technique for resource-allocation problems. *Random Struct. Algorithms*, 52(4):680–715, 2018. URL: https://doi.org/10.1002/rsa.20756, doi:10.1002/RSA.20756.

16 Alexander Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

17 David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993. doi:10.1007/BF01585178.

A Appendix

A.1 Santa Claus Problem where all resources need to be assigned

In the following, we show that the Budgeted Santa Claus Problem with the requirement that all items need to be assigned is not harder than the variant we study. For each resource j, adjust the cost of assigning j to any player i to $c'_{ij} = c_{ij} - \min_{i\in\mathcal{P}} c_{ij}$. This reduces the total budget by the sum of these minimum values across all resources, $C' = C - \sum_{j\in\mathcal{R}} \min_{i\in\mathcal{P}} c_{ij}$. Then we solve the reduced problem without enforcing that all the resources have to be assigned. If some resources remain unassigned, we allocate them to the players with zero cost (i.e., players $i \in \mathcal{P}$ with $c'_{ij} = 0$). This ensures that all resources are assigned without exceeding the original budget, as the reduced budget C' already accounted for these assignments. This reduction maintains the approximation ratio of the solution, as values remain the same and costs are not approximated.