The Submodular Santa Claus Problem in the Restricted Assignment Case

Etienne Bamas
EPFL, Lausanne, Switzerland

Paritosh Garg
EPFL, Lausanne, Switzerland

Lars Rohwedder
EPFL, Lausanne, Switzerland

Abstract

The submodular Santa Claus problem was introduced in a seminal work by Goemans, Harvey, Iwata, and Mirrokni (SODA'09) as an application of their structural result. In the mentioned problem n unsplittable resources have to be assigned to m players, each with a monotone submodular utility function f_i. The goal is to maximize $\min_i f_i(S_i)$ where $S_1, \dots, S_m$ is a partition of the resources. The result by Goemans et al. implies a polynomial time $O(n^{1/2+\varepsilon})$-approximation algorithm.

Since then progress on this problem was limited to the linear case, that is, all f_i are linear functions. In particular, a line of research has shown that there is a polynomial time constant approximation algorithm for linear valuation functions in the restricted assignment case. This is the special case where each player is given a set of desired resources Γ_i and the individual valuation functions are defined as $f_i(S) = f(S \cap \Gamma_i)$ for a global linear function f. This can also be interpreted as maximizing $\min_i f(S_i)$ with additional assignment restrictions, i.e., resources can only be assigned to certain players.

In this paper we make comparable progress for the submodular variant: If f is a monotone submodular function, we can in polynomial time compute an $O(\log\log(n))$-approximate solution.

2012 ACM Subject Classification Theory of computation → Scheduling algorithms

Keywords and phrases Scheduling, submodularity, approximation algorithm, hypergraph matching

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.22

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2011.06939

Funding This research was supported in part by the Swiss National Science Foundation project 200021-184656 "Randomness in Problem Instances and Randomized Algorithms."

Acknowledgements The authors wish to thank Ola Svensson for helpful discussions on the problem.

1 Introduction

In the Santa Claus problem (sometimes referred to as Max-Min Fair Allocation) we are given a set of n players P and a set of m indivisible resources R. In its full generality, each player $i \in P$ has a utility function $f_i : 2^R \mapsto \mathbb{R}_{\geq 0}$, where $f_i(S)$ measures the happiness of player i if he is assigned the resource set S. The goal is to find a partition of the resources that maximizes the happiness of the least happy player. Formally, we want to find a partition $\{S_i\}_{i \in P}$ of the resources that maximizes $\min_{i \in P} f_i(S_i)$.

With such an objective function one seeks to find the fairest solution as opposed to for example the best average happiness. Most of the recent literature on this problem focuses on cases where f_i is a linear function for all players i. If we assume all valuation functions are linear, the best approximation algorithm known for this problem, designed by Chakrabarty, Chuzhoy, and Khanna [4], has an approximation rate of n^ϵ and runs in time $n^{O(1/\epsilon)}$ for $\epsilon \in \Omega(\log\log(n)/\log(n))$. On the negative side, it is only known that computing

48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).
Editors: Nikhil Bansal, Emanuela Merelli, and James Worrell; Article No. 22; pp. 22:1–22:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

a $(2-\delta)$-approximation is NP-hard [13]. Apart from this there has been significant attention on the so-called *restricted assignment case*. Here the utility functions are defined by one linear function f and a set of resources Γ_i for each player i. Intuitively, player i is interested in the resources Γ_i, whereas the other resources are worthless for him. The individual utility functions are then implicitly defined by $f_i(S) = f(S \cap \Gamma_i)$. In a seminal work, Bansal and Srividenko [3] provide a $O(\log\log(m)/\log\log\log(m))$-approximation algorithm for this case. This was improved by Feige [8] to an $O(1)$-approximation. Further progress on the constant or the running time was made since then, see e.g. [1, 7, 6, 5, 10, 2, 15].

Let us now move to the non-linear case. Indeed, the problem becomes hopelessly difficult without any restrictions on the utility functions. Consider the following reduction from set packing. There are sets of resources $\{S_1, \dots, S_k\}$ and all utility functions are equal and defined by $f_i(S) = 1$ if $S_j \subseteq S$ for some j and $f_i(S) = 0$ otherwise. Deciding whether there are m disjoint sets in $S_1, \dots, S_k$ (a classical NP-hard problem) is equivalent to deciding whether the optimum of the Santa Claus problem is non-zero. In particular, obtaining any bounded approximation ratio for Santa Claus in this case is NP-hard.

Two naturally arising properties of utility functions are monotonicity and submodularity, see for example the related submodular welfare problem [12, 16] where the goal is to maximize $\sum_i f_i(S_i)$. A function f is monotone, if $f(S) \leq f(T)$ for all $S \subseteq T$. It is submodular, if $f(S \cup \{a\}) - f(S) \geq f(T \cup \{a\}) - f(T)$ for all $S \subseteq T$ and $a \notin T$. The latter is also known as the *diminishing returns* property in economics. A standard assumption on monotone submodular functions (used throughout this work) is that the value on the empty set is zero, i.e., $f(\emptyset) = 0$. Goemans, Harvey, Iwata, and Mirrokni [9] first considered the Santa Claus problem with monotone submodular utility functions as an application of their fundamental result on submodular functions. Together with the algorithm of [4] it implies an $O(n^{1/2+\epsilon})$-approximation in time $n^{O(1/\epsilon)}$. In the case that the valuation functions are all equal, that is, $f_i(S) = f(S)$ for a monotone submodular function f, Krause, Rajagopal, Gupta, and Guestrin gave a constant approximation [11]. We also refer to their work for an application of this problem in sensor placement.

In this paper we investigate the restricted assignment case with a monotone submodular utility function. That is, all utility functions are defined by $f_i(S) = f(S \cap \Gamma_i)$, where f is a monotone submodular function and Γ_i is a subset of resources for each players i. Before our work, the state-of-the-art for this problem was the $O(n^{1/2+\epsilon})$-approximation algorithm mentioned above, since none of the previous results for the restricted assignment case with a linear utility function apply when the utility function becomes monotone submodular.

1.1 Overview of results and techniques

Our main result is an approximation algorithm for the submodular Santa Claus problem in the restricted assignment case.

▶ **Theorem 1.** *There is a randomized polynomial time $O(\log\log(n))$-approximation algorithm for the restricted assignment case with a monotone submodular utility function.*

Our way to this result is organised as follows. In Section 2, we first reduce our problem to a hypergraph matching problem (see next paragraph for a formal definition). We then solve this problem using Lovasz Local Lemma (LLL) in Section 3. In [3] the authors also reduce to a hypergraph matching problem which they then solve using LLL, although both parts are substantially simpler. The higher generality of our utility functions is reflected in the more general hypergraph matching problem. Namely, our problem is precisely the weighted variant of the (unweighted) problem in [3]. We will elaborate later in this section why the previous techniques do not easily extend to the weighted variant.

The hypergraph matching problem. After the reduction in Section 2 we arrive at the following problem. There is a hypergraph $\mathcal{H} = (P \cup R, \mathcal{C})$ with hyperedges $\mathcal{C}$ over the vertices P and R. We write $m = |P|$ and $n = |R|$. We will refer to hyperedges as configurations, the vertices in P as players and R as resources[1]. Moreover, a hypergraph is said to be regular if all vertices in P and R have the same degree, that is, they are contained in the same number of configurations. The hypergraph may contain multiple copies of the same configuration. Each configuration $C \in \mathcal{C}$ contains exactly one vertex in P, that is, $|C \cap P| = 1$. Additionally, for each configuration $C \in \mathcal{C}$ the resources $j \in C$ have weights $w_{j,C} \geq 0$. We emphasize that the same resource j can be given different weights in two different configurations, that is, we may have $w_{j,C} \neq w_{j,C'}$ for two different configurations C, C'.

We require to select for each player $i \in P$ one configuration C that contains i. For each configuration C that was selected we require to assign a subset of the resources in C which has a total weight of at least $(1/\alpha) \cdot \sum_{j \in C} w_{j,C}$ to the player in C. A resource can only be assigned to one player. We call such a solution an α-relaxed perfect matching. One seeks to minimize α.

We show that every regular hypergraph has an α-relaxed perfect matching for some $\alpha = O(\log\log(n))$ assuming that $w_{j,C} \leq (1/\alpha) \cdot \sum_{j' \in C} w_{j',C}$ for all j, C, that is, all weights are small compared to the total weight of the configuration. Moreover, we can find such a matching in randomized polynomial time. In the reduction we use this result to round a certain LP relaxation and α essentially translates to the approximation rate. This result generalizes that of Bansal and Srividenko on hypergraph matching in the following way. They proved the same result for unit weights and uniform hyperedges, that is, $w_{j,C} = 1$ for all j, C and all hyperedges have the same number of resources[2]. In the next paragraph we briefly go over the techniques to prove our result for the hypergraph matching problem.

Our techniques. Already the extension from uniform to non-uniform hypergraphs (assuming unit weights) is highly non-trivial and captures the core difficulty of our result. Indeed, we show with a (perhaps surprising) reduction, that we can reduce our weighted hypergraph matching problem to the unweighted (but non-uniform) version by introducing some bounded dependencies between the choices of the different players. For sake of brevity we therefore focus in this section on the unweighted non-uniform variant, that is, we need to assign to each player a configuration C and at least $|C|/\alpha$ resources in C. We show that for any regular hypergraph there exists such a matching for $\alpha = O(\log\log(n))$ assuming that all configurations contain at least α resources and we can find it in randomized polynomial time. Without the assumption of uniformity the problem becomes significantly more challenging. To see this, we lay out the techniques of Bansal and Srividenko that allowed them to solve the problem in the uniform case. We note that for $\alpha = O(\log(n))$ the statement is easy to prove: We select for each player i one of the configurations containing i uniformly at random. Then by standard concentration bounds each resource is contained in at most $O(\log(n))$ of the selected configurations with high probability. This implies that there is a fractional assignment of resources to configurations such that each of the selected configurations C receives $\lfloor |C|/O(\log(n)) \rfloor$ of the resources in C. By integrality of the bipartite matching polytope, there is also an integral assignment with this property.

To improve to $\alpha = O(\log\log(n))$ in the uniform case, Bansal and Srividenko proceed as follows. Let k be the size of each configuration. First they reduce the degree of each player and resource to $O(\log(n))$ using the argument above, but taking $O(\log(n))$ configurations for

[1] We note that these do not have to be the same players and resources as in the Santa Claus problem we reduced from, but n and m do not increase.

[2] In fact they get a slightly better ratio of $\alpha = O(\log\log(m)/\log\log\log(m))$.

each player. Then they sample uniformly at random $O(n\log(n)/k)$ resources and drop all others. This is sensible, because they manage to prove the (perhaps surprising) fact that an α-relaxed perfect matching with respect to the smaller set of resources is still an $O(\alpha)$-relaxed perfect matching with respect to all resources with high probability (when assigning the dropped resources to the selected configurations appropriately). Indeed, the smaller instance is easier to solve: With high probability all configurations have size $O(\log(n))$ and this greatly reduces the dependencies between the bad events of the random experiment above (the event that a resource is contained in too many selected configurations). This allows them to apply Lovász Local Lemma (LLL) in order to show that with positive probability the experiment succeeds for $\alpha = O(\log\log(n))$.

It is not obvious how to extend this approach to non-uniform hypergraphs: Sampling a fixed fraction of the resources will either make the small configurations empty – which makes it impossible to retain guarantees for the original instance – or it leaves the big configurations big – which fails to reduce the dependencies enough to apply LLL. Hence it requires new sophisticated ideas for non-uniform hypergraphs, which we describe next.

Suppose we are able to find a set $\mathcal{K} \subseteq \mathcal{C}$ of configurations (one for each player) such that for each $K \in \mathcal{K}$ the sum of intersections $|K \cap K'|$ with smaller configurations $K' \in \mathcal{K}$ is very small, say at most $|K|/2$. Then it is easy to derive a 2-relaxed perfect matching: We iterate over all $K \in \mathcal{K}$ from large to small and reassign all resources to K (possibly stealing them from the configuration that previously had them). In this process every configuration gets stolen at most $|K|/2$ of its resources, in particular, it keeps the other half. However, it is non-trivial to obtain a property like the one mentioned above. If we take a random configuration for each player, the dependencies of the intersections are too complex. To avoid this we invoke an advanced variant of the sampling approach where we construct not only one set of resources, but a hierarchy of resource sets $R_0 \supseteq \cdots \supseteq R_d$ by repeatedly dropping a fraction of resources from the previous set. We then formulate bad events based on the intersections of a configuration C with smaller configurations C', but we write it only considering a resource set R_k of convenient granularity (chosen based on the size of C'). In this way we formulate a number of bad events using various sets R_k. This succeeds in reducing the dependencies enough to apply LLL. Unfortunately, even with this new way of defining bad events, the guarantee that for each $K \in \mathcal{K}$ the sum of intersections $|K \cap K'|$ with smaller configurations $K' \in \mathcal{K}$ is at most $|K|/2$ is still too much to ask. We can only prove some weaker property which makes it more difficult to reconstruct a good solution from it. The reconstruction still starts from the biggest configurations and iterates to finish by including the smallest configurations but it requires a delicate induction where at each step, both the resource set expands and some new small configurations that were not considered before come into play.

Additional implications of non-uniform hypergraph matchings to the Santa Claus problem. We believe this hypergraph matching problem is interesting in its own right. Our last contribution is to show that finding good matchings in unweighted hypergraphs with fewer assumptions than ours would have important applications for the Santa Claus problem with linear utility functions. We recall that here, each player i has its own utility function f_i that can be any linear function. In this case, the best approximation algorithm is due to Chakrabarty, Chuzhoy, and Khanna [4] who gave a $O(n^{\epsilon})$-approximation running in time $O(n^{1/\epsilon})$. In particular, no sub-polynomial approximation running in polynomial time is known. Consider as before $\mathcal{H} = (P \cup R, \mathcal{C})$ a non-uniform hypergraph with unit weights ($w_{j,C} = 1$ for all j, C such that $j \in C$). Finding the smallest α (or an approximation of it) such that there exists an α-relaxed perfect matching in $\mathcal{H}$ is already a very non-trivial question to solve in polynomial time.

We show, via a reduction, that a c-approximation for this problem would yield a $O((c \log^*(n))^2)$-approximation for the Santa Claus problem with arbitrary linear utility functions. In particular, any sub-polynomial approximation for this problem would significantly improve the state-of-the-art[3]. Details of this last result can be found in the full version of the paper.

A remark on local search techniques. We focus here on an extension of the LLL technique of Bansal and Srividenko. However, another technique proved itself very successful for the Santa Claus problem in the restricted assignment case with a linear utility function. This is a local search technique discovered by Asadpour, Feige, and Saberi [2] who used it to give a non-constructive proof that the integrality gap of the configuration LP of Bansal and Srividenko is at most 4. One may wonder if this technique could also be extended to the submodular case as we did with LLL. Unfortunately, this seems problematic as the local search arguments heavily rely on amortizing different volumes of configurations (i.e., the sum of their resources' weights or the number of resources in the unweighted case). Amortizing the volumes of configurations works well, if each configuration has the same volume, which is the case for the problem derived from linear valuation functions, but not the one derived from submodular functions. If the volumes differ then the amortization arguments break and the authors of this paper believe this is a fundamental problem for this approach.

2 Reduction to hypergraph matching problem

In this section we give a reduction of the restricted submodular Santa Claus problem to the hypergraph matching problem. As a starting point we solve the configuration LP, a linear programming relaxation of our problem. The LP is constructed using a parameter T which denotes the value of its solution. The goal is to find the maximal T such that the LP is feasible. In the LP we have a variable $x_{i,C}$ for every player $i \in P$ and every configuration $C \in \mathcal{C}(i,T)$. The configurations $\mathcal{C}(i,T)$ are defined as the sets of resources $C \subseteq \Gamma_i$ such that $f(C) \geq T$. We require every player $i \in P$ to have at least one configuration and every resource $j \in R$ to be contained in at most one configuration.

$$\sum_{C \in \mathcal{C}(i,T)} x_{i,C} \geq 1 \quad \text{for all } i \in P$$

$$\sum_{i \in P} \sum_{C \in \mathcal{C}(i,T): j \in C} x_{i,C} \leq 1 \quad \text{for all } j \in R$$

$$x_{i,C} \geq 0 \quad \text{for all } i \in P, C \in \mathcal{C}(i,T)$$

Since this linear program has exponentially many variables, we cannot directly solve it in polynomial time. We will give a polynomial time constant approximation for it via its dual. This is similar to the linear variant in [3], but requires some more work. In their case they can reduce the problem to one where the separation problem of the dual can be solved in polynomial time. In our case even the separation problem can only be approximated. Nevertheless, this is sufficient to approximate the linear program in polynomial time.

▶ **Theorem 2.** *The configuration LP of the restricted submodular Santa Claus problem can be approximated within a factor of $(1 - 1/e)/2$ in polynomial time.*

[3] We mention that our result on relaxed matchings in Section 3 does not imply an $O(\log \log(n))$-approximation for this problem since we make additional assumptions on the regularity of the hypergraph or the size of hyperedges.

We defer the proof of this theorem to the full version of the paper. Given a solution x^* of the configuration LP we want to arrive at the hypergraph matching problem from the introduction such that an α-relaxed perfect matching of that problem corresponds to an $O(\alpha)$-approximate solution of the restricted submodular Santa Claus problem. Let T^* denote the value of the solution x^*. We will define a resource $j \in R$ as *fat* if $f(\{j\}) \geq T^*/(100\alpha)$.

Resources that are not fat are called *thin*. We call a configuration $C \in \mathcal{C}(i, T)$ thin, if it contains only thin resources and denote by $\mathcal{C}_t(i, T) \subseteq \mathcal{C}(i, T)$ the set of thin configurations. Intuitively in order to obtain an $O(\alpha)$-approximate solution, it suffices to give each player i either one fat resource $j \in \Gamma_i$ or a thin configuration $C \in \mathcal{C}_t(i, T^*/O(\alpha))$. For our next step towards the hypergraph problem we use a technique borrowed from Bansal and Srividenko [3]. This technique allows us to simplify the structure of the problem significantly using the solution of the configuration LP. Namely, one can find a partition of the players into clusters such that we only need to cover one player from each cluster with thin resources. All other players can then be covered by fat resources. Informally speaking, the following lemma is proved by sampling configurations randomly according to a distribution derived in a non-trivial way from the configuration LP.

▶ **Lemma 3.** *Let $\ell \geq 12\log(n)$. Given a solution of value T^* for the configuration LP in randomized polynomial time we can find a partition of the players into clusters $K_1 \cup \cdots \cup K_k \cup Q = P$ and multisets of configurations $\mathcal{C}_h \subseteq \bigcup_{i \in K_h} \mathcal{C}_t(i, T^*/5)$, $h = 1, \ldots, k$, such that*

1. *$|\mathcal{C}_h| = \ell$ for all $h = 1, \ldots, k$ and*
2. *Each small resource appears in at most ℓ configurations of $\bigcup_h \mathcal{C}_h$.*
3. *given any $i_1 \in K_1, i_2 \in K_2, \ldots, i_k \in K_k$ there is a matching of fat resources to players $P \setminus \{i_1, \ldots, i_k\}$ such that each of these players i gets a unique fat resource $j \in \Gamma_i$.*

The role of the players Q in the lemma above is that each one of them gets a fat resource for certain. The proof follows closely that in [3]. For completeness we include it in the full version of the paper. We are now ready to define the hypergraph matching instance. The vertices of our hypergraph are the clusters $K_1, \ldots, K_k$ and the thin resources. Let $\mathcal{C}_1, \ldots, \mathcal{C}_k$ be the multisets of configurations as in Lemma 3. For each K_h and $C \in \mathcal{C}_h$ there is a hyperedge containing K_h and all resources in C. Let $\{j_1, \ldots, j_m\} = C$ ordered arbitrarily, but consistently. Then we define the weights as normalized marginal gains of resources if they are taken in this order, that is,

$$w_{j_i,C} = \frac{5}{T^*} f(\{j_i\} \mid \{j_1, \ldots, j_{i-1}\}) = \frac{5}{T^*}(f(\{j_1, \ldots, j_{i-1}, j_i\}) - f(\{j_1, \ldots, j_{i-1}\})).$$

This implies that $\sum_{j \in C} w_{j,C} \geq 5f(C)/T^* \geq 1$ for each $C \in \mathcal{C}_h$, $h = 1, \ldots, k$.

▶ **Lemma 4.** *Given an α-relaxed perfect matching to the instance as described by the reduction, one can find in polynomial time an $O(\alpha)$-approximation to the instance of restricted submodular Santa Claus.*

Proof. The α-relaxed perfect matching implies that each cluster K_h gets some small resources C' where $C' \subseteq C$ for some $C \in \mathcal{C}_h$ and $\sum_{j \in C'} w_{j,C} \geq 1/\alpha$. By submodularity we have that $f(C') \geq T^*/(5\alpha)$. Therefore we can satisfy one player in each cluster using thin resources and by Lemma 3 all others using fat resources. ◀

The proof above is the most critical place in the paper where we make use of the submodularity of the valuation function f. We note that since all resources considered are thin resources we have, by submodularity of f, the assumption that

$$w_{j,C} \leq \frac{5}{T^*} f(\{j\}) \leq \frac{5}{T^*} \frac{T^*}{100\alpha} \leq \frac{5}{100\alpha} \sum_{j \in C} w_{j,C}$$

for all j, C such that $j \in C$. This means that the weights are all small enough, as promised in introduction. From now on, we will assume that $\sum_{j\in C} w_{j,C} = 1$ for all configurations C. This is without loss of generality, since we can just rescale the weights inside each configuration. This does not hurt the property that all weights are small enough.

2.1 Reduction to unweighted hypergraph matching

Before proceeding to the solution of this hypergraph matching problem, we first give a reduction to an unweighted variant of the problem. We will then solve this unweighted variant in the next section. First, we note that we can assume that all the weights $w_{j,C}$ are powers of 2 by standard rounding arguments. This only loses a constant factor in the approximation rate. Second, we can assume that inside each configuration C, each resource has a weight that is at least a $1/(2n)$. Formally, we can assume that $\min_{j\in C} w_{j,C} \geq 1/(2n)$ for all $C \in \mathcal{C}$. If this is not the case for some $C \in \mathcal{C}$, simply delete from C all the resources that have a weight less than $1/(2n)$. By doing this, the total weight of C is only decreased by a factor $1/2$ since it looses in total at most a weight of $n \cdot (1/2n) = 1/2$. (Recall that we rescaled the weights so that $\sum_{j\in C} w_{j,C} = 1$).

Hence after these two operations, an α-relaxed perfect matching in the new hypergraph is still an $O(\alpha)$-relaxed perfect matching in the original hypergraph. From there we reduce to an unweighted variant of the matching problem. Note that each configuration contains resources of at most $\log(n)$ different possible weights (powers of 2 from $1/(2n)$ to $1/\alpha$). We create the following new unweighted hypergraph $\mathcal{H}' = (P' \cup R, \mathcal{C}')$. The resource set R remains unchanged. For each player $i \in P$, we create $\log(n)$ players, which later correspond each to a distinct weight. We will say that the players obtained from duplicating the original player form a *group*. For every configuration C containing player i in the hypergraph $\mathcal{H}$, we add a set $\mathcal{S}_C = \{C_1, \dots, C_s, \dots, C_{\log(n)}\}$ of configurations in $\mathcal{H}'$. C_s contains player i_s and all resources that are given a weight $2^{-(s+1)}$ in C. In this new hypergraph, the resources are not weighted. Note that if the hypergraph $\mathcal{H}$ is regular then $\mathcal{H}'$ is regular as well.

Additionally, for a group of player and a set of $\log(n)$ configurations (one for each player in the group), we say that this set of configurations is *consistent* if all the configurations selected are obtained from the same configuration in the original hypergraph $\mathcal{H}$ (i.e. the selected configurations all belong to $\mathcal{S}_C$ for some C in $\mathcal{H}$).

Formally, we focus of the following problem. Given the regular hypergraph $\mathcal{H}'$, we want to select, for each group of $\log(n)$ players, a consistent set of configurations $C_1, \dots, C_s, \dots, C_{\log(n)}$ and assign to each player i_s a subset of the resources in the corresponding configuration C_s so that i_s is assigned at least $\lfloor |C_s|/\alpha \rfloor$ resources. No resource can be assigned to more than one player. We refer to this assignment as a consistent α-relaxed perfect matching. Note that in the case where $|C_s|$ is small (e.g. of constant size) we are not required to assign any resource to player i_s.

▶ **Lemma 5.** *A consistent α-relaxed matching in $\mathcal{H}'$ induces a $O(\alpha)$-relaxed matching in $\mathcal{H}$.*

Due to space constraint, the proof of this lemma is moved to the full version of the paper.

3 Matchings in regular hypergraphs

In this section we solve the hypergraph matching problem we arrived to in the previous section. For convenience, we give a self contained definition of the problem before formulating and proving our result.

Input. We are given $\mathcal{H} = (P \cup R, \mathcal{C})$ a hypergraph with hyperedges $\mathcal{C}$ over the vertices P (players) and R (resources) with $m = |P|$ and $n = |R|$. As in previous sections, we will refer to hyperedges as configurations. Each configuration $C \in \mathcal{C}$ contains exactly one vertex in P, that is, $|C \cap P| = 1$. The set of players is partitioned into groups of size at most $\log(n)$, we will use A to denote a group. These groups are disjoint and contain all players. Finally there exists an integer ℓ such that for each group A there are ℓ consistent sets of configurations. A consistent set of configurations for a group A is a set of $|A|$ configurations such that all players in the group appear in exactly one of these configurations. We will denote by $\mathcal{S}_A$ such a set and for a player $i \in A$, we will denote by $\mathcal{S}_A^{(i)}$ the unique configuration in $\mathcal{S}_A$ containing i. Finally, no resource appears in more than ℓ configurations. We say that the hypergraph is regular (although some resources may appear in less than ℓ configurations).

Output. We wish to select a matching that covers all players in P. More precisely, for each group A we want to select a consistent set of configurations (denoted by $\{\mathcal{S}_A^{(i)}\}_{i\in A}$). Then for each player $i \in A$, we wish to assign a subset of the resources in $\mathcal{S}_A^{(i)}$ to the player i such that:

1. No resource is assigned to more than one player in total.
2. For any group A and any player $i \in A$, player i is assigned at least $\lfloor |\mathcal{S}_A^{(i)}|/\alpha \rfloor$ resources from $\mathcal{S}_A^{(i)}$.

We call this a consistent α-relaxed perfect matching. Our goal in this section will be to prove the following theorem.

▶ **Theorem 6.** *Let $\mathcal{H} = (P \cup R, \mathcal{C})$ be a regular (non-uniform) hypergraph where the set of players is partitioned into groups of size at most $\log(n)$. Then we can, in randomized polynomial time, compute a consistent α-relaxed perfect matching for $\alpha = O(\log\log(n))$.*

We note that Theorem 6 together with the reduction from the previous section will prove our main result (Theorem 1) stated in introduction.

3.1 Overview and notations

To prove Theorem 6, we introduce the following notations. Let $\ell \in \mathbb{N}$ be the regularity parameter as described in the problem input (i.e. each group has ℓ consistent sets and each resource appears in no more than ℓ configurations). As we proved in Lemma 3 we can assume with standard sampling arguments that $\ell = 300.000\log^3(n)$ at a constant loss. If this is not the case because we might want to solve the hypergraph matching problem by itself (i.e. not obtained by the reduction in Section 2), the proof of Lemma 3 can be repeated in a very similar way here.

For a configuration C, its size will be defined as $|C \cap R|$ (i.e. its cardinality over the resource set). For each player i, we denote by $\mathcal{C}_i$ the set of configurations that contain i. We now group the configurations in $\mathcal{C}_i$ by size: We denote by $\mathcal{C}_i^{(0)}$ the configurations of size in $[0, \ell^4)$ and for $k \geq 1$ we write $\mathcal{C}_i^{(k)}$ for the configurations of size in $[\ell^{k+3}, \ell^{k+4})$. Moreover, define $\mathcal{C}^{(k)} = \bigcup_i \mathcal{C}_i^{(k)}$ and $\mathcal{C}^{(\geq k)} = \bigcup_{h\geq k} \mathcal{C}^{(h)}$. Let d be the smallest number such that $\mathcal{C}^{(\geq d)}$ is empty. Note that $d \leq \log(n)/\log(\ell)$. Now consider the following random process.

▶ **Random Experiment 7.** *We construct a nested sequence of resource sets $R = R_0 \supseteq R_1 \supseteq \ldots \supseteq R_d$ as follows. Each R_k is obtained from R_{k-1} by deleting every resource in R_{k-1} independently with probability $(\ell-1)/\ell$.*

In expectation only a $1/\ell$ fraction of resources in R_{k-1} survives in R_k. Also notice that for $C \in \mathcal{C}^{(k)}$ we have that $\mathbb{E}[|R_k \cap C|] = \text{poly}(\ell)$.

The proof of Theorem 6 is organized as follows. In Section 3.2, we give some properties of the resource sets constructed by Random Experiment 7 that hold with high probability. Then in Section 3.3, we show that we can find a single consistent set of configurations for each group of players such that for each configuration selected, its intersection with smaller selected configurations is bounded if we restrict the resource set to an appropriate R_k. Restricting the resource set is important to bound the dependencies of bad events in order to apply Lovasz Local Lemma. Finally in Section 3.4, we demonstrate how these configurations allow us to reconstruct a consistent α-relaxed perfect matching for an appropriate assignment of resources to configurations.

3.2 Properties of resource sets

In this subsection, we give a precise statement of the key properties that we need from Random Experiment 7. The first two lemmas have a straight-forward proof. The last one is a generalization of an argument used by Bansal and Srividenko [3]. Since the proof is more technical and tedious, we also defer it to the full version of the paper along with the proof of the first two statements.

We start with the first property which bounds the size of the configurations when restricted to some R_k. This property is useful to reduce the dependencies while applying LLL later.

▶ **Lemma 8.** *Consider Random Experiment 7 with $\ell \geq 300.000 \log^3(n)$. For any $k \geq 0$ and any $C \in \mathcal{C}^{(\geq k)}$ we have*

$$\frac{1}{2}\ell^{-k}|C| \leq |R_k \cap C| \leq \frac{3}{2}\ell^{-k}|C|$$

with probability at least $1 - 1/n^{10}$.

The next property expresses that for any configuration the sum of intersections with configurations of a particular size does not deviate much from its expectation. In particular, for any configuration C, the sum of it's intersections with other configurations is at most $|C|\ell$ as each resource is in at most ℓ configurations. By the lemma stated below, we recover this up to a multiplicative constant factor when we consider the appropriately weighted sum of the intersection of C with other configurations C' of smaller sizes where each configuration $C' \in \mathcal{C}^{(k)}$ is restricted to the resource set R_k.

▶ **Lemma 9.** *Consider Random Experiment 7 with $\ell \geq 300.000 \log^3(n)$. For any $k \geq 0$ and any $C \in \mathcal{C}^{(\geq k)}$ we have*

$$\sum_{C' \in \mathcal{C}^{(k)}} |C' \cap C \cap R_k| \leq \frac{10}{\ell^k}\left(|C| + \sum_{C' \in \mathcal{C}^{(k)}} |C' \cap C|\right)$$

with probability at least $1 - 1/n^{10}$.

We now define the notion of *good* solutions which is helpful in stating our last property. Let $\mathcal{F}$ be a set of configurations, $\alpha : \mathcal{F} \to \mathbb{N}$, $\gamma \in \mathbb{N}$, and $R' \subseteq R$. We say that an assignment of R' to $\mathcal{F}$ is (α, γ)-good if every configuration $C \in \mathcal{F}$ receives at least $\alpha(C)$ resources of $C \cap R'$ and if no resource in R' is assigned more than γ times in total.

Below we obtain that given a (α, γ)-good solution with respect to resource set R_{k+1}, one can construct an almost $(\ell \cdot \alpha, \gamma)$-good solution with respect to the bigger resource set R_k. Informally, starting from a good solution with respect to the final resource set and iteratively applying this lemma would give us a good solution with respect to our complete set of resources.

▶ **Lemma 10.** *Consider Random Experiment 7 with $\ell \geq 300.000\log^3(n)$. Fix $k \geq 0$. Conditioned on the event that the bounds in Lemma 8 hold for k, then with probability at least $1 - 1/n^{10}$ the following holds for all $\mathcal{F} \subseteq \mathcal{C}^{(\geq k+1)}$, $\alpha : \mathcal{F} \to \mathbb{N}$, and $\gamma \in \mathbb{N}$ such that $\ell^3/1000 \leq \alpha(C) \leq n$ for all $C \in \mathcal{F}$ and $\gamma \in \{1, \ldots, \ell\}$: If there is a (α, γ)-good assignment of R_{k+1} to $\mathcal{F}$, then there is a (α', γ)-good assignment of R_k to $\mathcal{F}$ where*

$$\alpha'(C) \geq \ell \left(1 - \frac{1}{\log(n)}\right) \alpha(C)$$

for all $C \in \mathcal{F}$. Moreover, this assignment can be found in polynomial time.

Given the lemmata above, by a simple union bound one gets that all the properties of resource sets hold.

3.3 Selection of configurations

In this subsection, we give a random process that selects one consistent set of configurations for each group of players such that the intersection of the selected configurations with smaller configurations is bounded when considered on appropriate sets R_k. We will denote $\mathcal{S}_A$ the selected consistent set for group A and for ease of notation we will denote $K_i = \mathcal{S}_A^{(i)}$ the selected configuration for player $i \in A$. For any integer k, we write $\mathcal{K}_i^{(k)} = \{K_i\}$ if $K_i \in \mathcal{C}_i^{(k)}$ and $\mathcal{K}_i^{(k)} = \emptyset$ otherwise. As for the configuration set, we will also denote $\mathcal{K}^{(k)} = \bigcup_i \mathcal{K}_i^{(k)}$ and $\mathcal{K} = \bigcup_k \mathcal{K}^{(k)}$. The following lemma describes what are the properties we want to have while selecting the configurations. For better clarity we also recall what the properties of the sets $R_0, \ldots, R_d$ that we need are. These hold with high probability by the lemmata of the previous section.

▶ **Lemma 11.** *Let $R = R_0 \supseteq \ldots \supseteq R_d$ be sets of fewer and fewer resources. Assume that for each k and $C \in \mathcal{C}_i^{(k)}$ we have*

$$1/2 \cdot \ell^{k-h} \leq |C \cap R_h| \leq 3/2 \cdot \ell^{-h}|C| < 3/2 \cdot \ell^{k-h+4}$$

for all $h = 0, \ldots, k$. Then there exists a selection of one consistent set $\mathcal{S}_A$ for each group A such for all $k = 0, \ldots, d$, $C \in \mathcal{C}^{(k)}$ and $j = 0, \ldots, k$ then we have

$$\sum_{j \leq h \leq k} \sum_{K \in \mathcal{K}^{(h)}} \ell^h |K \cap C \cap R_h| \leq \frac{1}{\ell} \sum_{j \leq h \leq k} \sum_{C' \in \mathcal{C}^{(h)}} \ell^h |C' \cap C \cap R_h| + 1000\frac{d+\ell}{\ell}\log(\ell)|C|.$$

Moreover, this selection of consistent sets can be found in polynomial time.

Before we prove this lemma, we give an intuition of the statement. Consider the sets $R_1, \ldots, R_d$ constructed as in Random Experiment 7. Then for $C' \in \mathcal{C}^{(h)}$ we have $\mathbb{E}[\ell^h |C' \cap C \cap R_h|] = |C' \cap C|$. Hence

$$\sum_{h \leq k} \sum_{K \in \mathcal{K}^{(h)}} |K \cap C| = \mathbb{E}[\sum_{h \leq k} \sum_{K \in \mathcal{K}^{(h)}} \ell^h |K \cap C \cap R_h|].$$

Similarly for the right-hand side we have

$$\mathbb{E}[\frac{1}{\ell} \sum_{j \leq h \leq k} \sum_{C' \in \mathcal{C}^{(h)}} \ell^h |C' \cap C \cap R_h| + O(\frac{d+\ell}{\ell}\log(\ell)|C|)]$$
$$= \frac{1}{\ell} \underbrace{\sum_{j \leq h \leq k} \sum_{C' \in \mathcal{C}^{(h)}} |C' \cap C|}_{\leq \ell |C|} + O\left(\frac{d+\ell}{\ell}\log(\ell)|C|\right) = O\left(\frac{d+\ell}{\ell}\log(\ell)|C|\right).$$

Hence, the lemma says that each resource in C is roughly covered $O((d+\ell)/\ell \cdot \log(\ell))$ times by smaller configurations.

We now proceed to the proof of Lemma 11.

Proof. We perform the following random experiment and show with LLL that there is a positive probability of success.

▶ **Random Experiment 12.** *For each group A, select one consistent set $\mathcal{S}_A$ uniformly at random. Then for each player $i \in A$ set $K_i = \mathcal{S}_A^{(i)}$.*

Given this experiment we can define the following random variables. For all $h = 0, \dots, d$ and $i \in P$ we define

$$X_{i,C}^{(h)} = \sum_{K \in \mathcal{K}_i^{(h)}} |K \cap C \cap R_h| \leq \min\{3/2 \cdot \ell^4, |C \cap R_h|\}.$$

Let $X_C^{(h)} = \sum_{i=1}^m X_{i,C}^{(h)}$. Then

$$\mathbb{E}[X_C^{(h)}] \leq \frac{1}{\ell} \sum_{C' \in \mathcal{C}^{(h)}} |C' \cap C \cap R_h| \leq |C \cap R_h|.$$

We are now ready to define the bad events on which we will apply the Lovasz Local Lemma. As we will show later, if none of them occur, Lemma 11 will hold. For each k, $C \in \mathcal{C}^{(k)}$, and $h \leq k$ let $B_C^{(h)}$ be the event that

$$X_C^{(h)} \geq \begin{cases} \mathbb{E}[X_C^{(h)}] + 63|C \cap R_h| \log(\ell) & \text{if } k-5 \leq h \leq k, \\ \mathbb{E}[X_C^{(h)}] + 135|C \cap R_h| \log(\ell) \cdot \ell^{-1} & \text{if } h \leq k-6. \end{cases}$$

The intuitive reason as to why we define these two different bad events can be summarized as follows. In the case $h \leq k-6$, we are counting how many times C is intersected by configurations that are much smaller than C. Hence the size of this intersection can be written as a sum of independent random variables of value at most $O(\ell^4)$ which is much smaller than the total size of the configuration $|C \cap R_h|$. Since the random variables are in a much smaller range, Chernoff bounds give much better concentration guarantees and we can afford a very small deviation from the expectation. In the other case, we do not have this property hence we need a bigger deviation to maintain a sufficiently low probability of failure. However, this does not hurt the statement of Lemma 11 since we sum this bigger deviation only a constant number of times. One key idea to be able to apply Lovasz Local Lemma here is also to consider intersection of C with smaller configurations but restricted to a set R_h of convenient granularity. One can notice that $|C' \cap R_h| = \mathrm{poly}(\ell)$ if $C' \in \mathcal{C}^{(h)}$ (by the assumption made in Lemma 11). This allows to reduce significantly the dependencies between bad events which is crucial to make any use of LLL here.

With this in mind, we claim that the probability of each bad event happening is small.

▷ Claim 13. For each k, $C \in \mathcal{C}^{(k)}$, and $h \leq k$ we have

$$\mathbb{P}[B_C^{(h)}] \leq \exp\left(-2\frac{|C \cap R_h|}{\ell^9} - 18\log(\ell)\right).$$

Proof. Consider first the case that $h \geq k-5$. By a Chernoff bound (see full version for the precise formulation) with

$$\delta = 63\frac{|C \cap R_h| \log(\ell)}{\mathbb{E}[X_C^{(h)}]} \geq 1$$

we get

$$\mathbb{P}[B_C^{(h)}] \leq \exp\left(-\frac{\delta \mathbb{E}[X_C^{(h)}]}{3|C \cap R_h|}\right) \leq \exp(-21\log(\ell))) \leq \exp\left(-2\underbrace{\frac{|C \cap R_h|}{\ell^9}}_{\leq 3/2} -18\log(\ell)\right).$$

Now consider $h \leq k - 6$. We apply again a Chernoff bound with

$$\delta = 135\frac{|C \cap R_h|\log(\ell)}{\ell \mathbb{E}[X_C^{(h)}]} \geq \frac{1}{\ell}.$$

This implies

$$\begin{aligned}\mathbb{P}[B_C^{(h)}] \leq \exp\left(-\frac{\min\{\delta, \delta^2\}\mathbb{E}[X_C^{(h)}]}{3 \cdot 3/2 \cdot \ell^4}\right) &\leq \exp\left(-30\frac{|C \cap R_h|\log(\ell)}{\ell^6}\right) \\ &\leq \exp\left(-2\frac{|C \cap R_h|}{\ell^9} - 18\log(\ell)\right). \end{aligned}$$ ◁

We can now state Lovasz Local Lemma and use it in our setting.

▶ **Proposition 14** (Lovasz Local Lemma (LLL)). *Let $B_1, \ldots, B_t$ be bad events, and let $G = (\{B_1, \ldots, B_t\}, E)$ be a dependency graph for them, in which for every i, event B_i is mutually independent of all events B_j for which $(B_i, B_j) \notin E$. Let x_i for $1 \leq i \leq t$ be such that $0 < x(B_i) < 1$ and $\mathbb{P}[B_i] \leq x(B_i)\prod_{(B_i,B_j)\in E}(1 - x(B_j))$. Then with positive probability no event B_i holds.*

Let $k \in \{0, \ldots, d\}$, $C \in \mathcal{C}^{(k)}$ and $h \leq k$. For event $B_C^{(h)}$ we set

$$x(B_C^{(h)}) = \exp(-|C \cap R_h|/\ell^9 - 18\log(\ell)).$$

We now analyze the dependencies of $B_C^{(h)}$. The event depends only on random variables $\mathcal{S}_A$ for groups A that contain at least one player i that has a configuration in $\mathcal{C}_i^{(h)}$ which overlaps with $C \cap R_h$. The number of such configurations (in particular, of such groups) is at most $\ell|C \cap R_h|$ since the hypergraph is regular.

In each of these groups, we count at most $\log(n)$ players, each having ℓ configurations hence in total at most $\ell \cdot \log(n)$ configurations.

Each configuration $C' \in \mathcal{C}^{(h')}$ can only influence those events $B_{C''}^{(h')}$ where $C' \cap C'' \cap R_{h'} \neq \emptyset$. Since $|C' \cap R_{h'}| \leq 3/2 \cdot \ell^4$ and since each resource appears in at most ℓ configurations, we see that each configuration can influence at most $3/2 \cdot \ell^5$ events.

Putting everything together, we see that the bad event $B_C^{(h)}$ is independent of all but at most

$$(\ell|C \cap R_h|) \cdot (\ell \cdot \log(n)) \cdot (3/2 \cdot \ell^5) = 3/2 \cdot \ell^7 \cdot \log(n)|C \cap R_h| \leq |C \cap R_h|\ell^8$$

other bad events.

We can now verify the condition for Proposition 14 by calculating

$$\begin{aligned} x(B_C^{(h)}) &\prod_{(B_C^{(h)}, B_{C'}^{(h')})\in E} (1 - x(B_{C'}^{(h')})) \\ &\geq \exp(-|C \cap R_h|/\ell^9 - 18\log(\ell)) \cdot (1 - \ell^{-18})^{|C \cap R_h|\ell^8} \\ &\geq \exp(-|C \cap R_h|/\ell^9 - 18\log(\ell)) \cdot \exp(-|C \cap R_h|/\ell^9) \\ &\geq \exp(-2|C \cap R_h|/\ell^9 - 18\log(\ell)) \geq \mathbb{P}[B_C^{(h)}]. \end{aligned}$$

By LLL we have that with positive probability none of the bad events happen. Let $k \in \{0, \dots, d\}$ and $C \in \mathcal{C}^{(k)}$. Then for $k - 5 \le h \le k$ we have

$$\ell^h X_C^{(h)} \le \ell^h \mathbb{E}[X_C^{(h)}] + 63\ell^h |C \cap R_h| \log(\ell) \le \ell^h \mathbb{E}[X_C^{(h)}] + 95|C|\log(\ell).$$

Moreover, for $h \le k - 6$ it holds that

$$\ell^h X_C^{(h)} \le \ell^h \mathbb{E}[X_C^{(h)}] + 135\ell^{h-1}|C \cap R_h| \log(\ell) \le \ell^h \mathbb{E}[X_C^{(h)}] + 203|C|\log(\ell) \cdot \ell^{-1}.$$

We conclude that, for any $0 \le j \le k$,

$$\sum_{j \le h \le k} \sum_{K \in \mathcal{K}^{(h)}} \ell^h |K \cap C \cap R_h| \le \sum_{j \le h \le k} \ell^h \mathbb{E}[X_C^{(h)}] + 1000 \frac{(k - j + 1) + \ell}{\ell} |C| \log(\ell)$$
$$\le \frac{1}{\ell} \sum_{j \le h \le k} \ell^h \sum_{C' \in \mathcal{C}^{(h)}} |C' \cap C \cap R_h| + 1000 \frac{d + \ell}{\ell} |C| \log(\ell).$$

This proves Lemma 11. ◀

▶ Remark 15. Since there are at most $\text{poly}(n, m, \ell)$ bad events and each bad event B has $\frac{x(B)}{1-x(B)} \le 1/2$ (because $x(B) \le \ell^{-18}$), the constructive variant of LLL by Moser and Tardos [14] can be applied to find a selection of configurations such that no bad events occur in randomized polynomial time.

3.4 Assignment of resources to configurations

In this subsection, we show how all the previously established properties allow us to find, in polynomial time, a good assignment of resources to the configurations $\mathcal{K}$ chosen as in the previous subsection. We will denote as in the previous subsection $\mathcal{K}_i^{(k)} = \{K_i\}$ if $K_i \in \mathcal{C}_i^{(k)}$ and $\mathcal{K}_i^{(k)} = \emptyset$ otherwise. We also define $\mathcal{K}^{(k)} = \bigcup_i \mathcal{K}_i^{(k)}$ and $\mathcal{K}^{(\ge k)} = \bigcup_{h \ge k} \mathcal{K}^{(k)}$. Finally we define the parameter

$$\gamma = 100.000 \frac{d + \ell}{\ell} \log(\ell),$$

which will define how many times each resource can be assigned to configurations in an intermediate solution. Note that $d \le \log(n)/\log(\ell)$. By our choice of $\ell = 300.000 \log^3(n)$, we have that $\gamma \le 310.000 \log\log(n)$. Lemma 11 implies the following bound.

▷ Claim 16. For any $k \ge 0$, any $0 \le j \le k$, and any $C \in \mathcal{K}^{(k)}$

$$\sum_{j \le h \le k} \sum_{K \in \mathcal{K}^{(h)}} \ell^h |K \cap C \cap R_h| \le 2000 \frac{d + \ell}{\ell} \log(\ell) |C|$$

Proof. By Lemma 11 we have that

$$\sum_{j \le h \le k} \sum_{K \in \mathcal{K}^{(h)}} \ell^h |K \cap C \cap R_h| \le \frac{1}{\ell} \sum_{j \le h \le k} \sum_{C' \in \mathcal{C}^{(h)}} \ell^h |C' \cap C \cap R_h| + 1000 \frac{d + \ell}{\ell} \log(\ell) |C|.$$

Furthermore, by Lemma 9, we get

$$\sum_{C' \in \mathcal{C}^{(h)}} \ell^h |C' \cap C \cap R_h| \le \ell^h \frac{10}{\ell^h} \left(|C| + \sum_{C' \in \mathcal{C}^{(h)}} |C' \cap C| \right).$$

Finally note that each resource appears in at most ℓ configurations, hence

$$\sum_{j\leq h\leq k}\sum_{C'\in\mathcal{C}^{(h)}}|C'\cap C|\leq \ell|C|.$$

Putting everything together we conclude

$$\begin{aligned}\sum_{j\leq h\leq k}\sum_{K\in\mathcal{K}^{(h)}}\ell^h|K\cap C\cap R_h| &\leq \frac{1}{\ell}\sum_{j\leq h\leq k}\sum_{C'\in\mathcal{C}^{(h)}}\ell^h|C'\cap C\cap R_h| + 1000\frac{d+\ell}{\ell}\log(\ell)|C|\\ &\leq \frac{1}{\ell}\sum_{j\leq h\leq k}10\left(|C| + \sum_{C'\in\mathcal{C}^{(h)}}|C'\cap C|\right) + 1000\frac{d+\ell}{\ell}\log(\ell)|C|\\ &\leq \frac{k-j}{\ell}10|C| + 10|C| + 1000\frac{d+\ell}{\ell}\log(\ell)|C|\\ &\leq 20|C| + 1000\frac{d+\ell}{\ell}\log(\ell)|C|\\ &\leq 2000\frac{d+\ell}{\ell}\log(\ell)|C|. \end{aligned}$$ ◁

We can now proceed to the main technical part of this section which is the following lemma proved by induction.

▶ **Lemma 17.** *For any $j\geq 0$, there exists an assignment of resources of R_j to configurations in $\mathcal{K}^{(\geq j)}$ such that no resource is taken more than γ times and each configuration $C\in\mathcal{K}^{(k)}$ $(k\geq j)$ receives at least*

$$\left(1-\frac{1}{\log(n)}\right)^{2(k-j)}\ell^{k-j}|C\cap R_k| - \frac{3}{\gamma}\sum_{j\leq h\leq k}\sum_{K\in\mathcal{K}^{(h)}}\ell^{h-j}|K\cap C\cap R_h|$$

resources from R_k.

Before going through the proof, we give here the intuition of why this is what we want to prove. Note that the term $\ell^{k-j}|C\cap R_k|$ is roughly equal to $\ell^{-j}|C|$ by the properties of the resource sets (precisely Lemma 8). The second term

$$\sum_{j\leq h\leq k}\sum_{K\in\mathcal{K}^{(h)}}\ell^{h-j}|K\cap C\cap R_h|$$

can be shown to be

$$O\left(\ell^{-j}\frac{d+\ell}{\ell}\log(\ell)|C|\right) = O(\ell^{-j}\log\log(n)|C|)$$

by Claim 16. Hence by choosing γ to be $\Theta(\log\log(n))$ we get that the bound in Lemma 17 will be $\Theta(\ell^{-j}|C|)$. At the end of the induction, we have $j=0$ which indeed implies that we have an assignment in which configurations receive

$$\Theta(\ell^{-0}|C|) = \Theta(|C|)$$

resources and such that each resource is assigned to at most $O(\log\log(n))$ configurations. With this in mind, we give the formal proof of Lemma 17.

Proof. We start from the biggest configurations and then iteratively reconstruct a good solution for smaller and smaller configurations. Recall d is the smallest integer such that $\mathcal{K}^{(\geq d)}$ is empty. Our base case for these configurations in $\mathcal{K}^{(\geq d)}$ is vacuously satisfied.

Now assume that we have a solution at level j, i.e. an assignment of resources to configurations in $\mathcal{K}^{(\geq j)}$ such that no resource is taken more than γ times and each configuration $C \in \mathcal{K}^{(k)}$ such that $k \geq j$ receives at least

$$\left(1-\frac{1}{\log(n)}\right)^{2(k-j)} \ell^{k-j}|C\cap R_k| - \frac{3}{\gamma}\sum_{j\leq h\leq k}\sum_{K\in\mathcal{K}^{(h)}} \ell^{h-j}|K\cap C\cap R_h|$$

resources from R_j. We show that this implies a solution at level $j-1$ in the following way. First by Lemma 10, this implies an assignment of resources of R_{j-1} to configurations in $\mathcal{K}^{(\geq j)}$ such that each $C \in \mathcal{K}^{(k)}$ receives at least

$$\begin{aligned}
&\left(1-\frac{1}{\log(n)}\right)\ell\left(\ell^{k-j}\left(1-\frac{1}{\log(n)}\right)^{2(k-j)}|C\cap R_k| - \frac{3}{\gamma}\sum_{j\leq h\leq k}\sum_{K\in\mathcal{K}^{(h)}} \ell^{h-j}|K\cap C\cap R_h|\right)\\
&= \left(1-\frac{1}{\log(n)}\right)^{2(k-(j-1))-1} \ell^{k-(j-1)}|C\cap R_k| - \frac{3}{\gamma}\left(1-\frac{1}{\log(n)}\right)\sum_{j\leq h\leq k}\sum_{K\in\mathcal{K}^{(h)}} \ell^{h-(j-1)}|K\cap C\cap R_h|\\
&\geq \left(1-\frac{1}{\log(n)}\right)^{2(k-(j-1))-1} \ell^{k-(j-1)}|C\cap R_k| - \frac{3}{\gamma}\sum_{j\leq h\leq k}\sum_{K\in\mathcal{K}^{(h)}} \ell^{h-(j-1)}|K\cap C\cap R_h|
\end{aligned}$$

resources and no resource of R_{j-1} is taken more than γ times. Note that we can apply Lemma 10 since we have by Claim 16 and Lemma 8

$$\begin{aligned}
&\left(1-\frac{1}{\log(n)}\right)^{2(k-j)} \ell^{k-j}|C\cap R_k| - \frac{3}{\gamma}\sum_{j\leq h\leq k}\sum_{K\in\mathcal{K}^{(h)}} \ell^{h-j}|K\cap C\cap R_h|\\
&\geq \frac{\ell^{k-j}}{e^2}|C\cap R_k| - \frac{3}{\gamma}2000\ell^{-j}\frac{d+\ell}{\ell}\log(\ell)|C|\\
&\geq \ell^{-j}|C|\left(\frac{1}{2e^2} - \frac{6000}{\gamma}\frac{d+\ell}{\ell}\log(\ell)\right)\\
&\geq \frac{\ell^{-j}|C|}{3e^2} > \frac{\ell^3}{1000}
\end{aligned}$$

Now consider configurations in $\mathcal{K}^{(j-1)}$ and proceed for them as follows. Give to each $C \in \mathcal{K}^{(j-1)}$ all the resources in $C \cap R_{j-1}$ except all the resources that appear in more than γ configurations in $\mathcal{K}^{(j-1)}$. Since each deleted resource is counted at least γ times in the sum $\sum_{K\in\mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1}|$, we have that each configuration C in $\mathcal{K}^{(j-1)}$ receives at least

$$|C\cap R_{j-1}| - \frac{1}{\gamma}\sum_{K\in\mathcal{K}^{(j-1)}}|K\cap C\cap R_{j-1}|$$

resources and no resource is taken more than γ times by configurations in $\mathcal{K}^{(j-1)}$. Notice that now every resource is taken no more than γ times by configurations in $\mathcal{K}^{(\geq j)}$ and no more than γ times by configurations in $\mathcal{K}^{(j-1)}$ which in total can sum up to 2γ times.

Therefore to finish the proof consider an resource $i \in R_{j-1}$. This resource is taken b_i times by configurations in $\mathcal{K}^{(\geq j)}$ and a_i times by configurations in $\mathcal{K}^{(j-1)}$. If $a_i + b_i \leq \gamma$, nothing needs to be done. Otherwise, denote by O the set of problematic resources (i.e. resources i such that $a_i + b_i > \gamma$). For every $i \in O$, select uniformly at random $a_i + b_i - \gamma$ configurations in $\mathcal{K}^{(\geq j)}$ that currently contain resource i and delete the resource from these configurations. When this happens, each configuration in $C \in \mathcal{K}^{(\geq j)}$ that contains i has a probability of $(a_i + b_i - \gamma)/b_i$ to be selected to loose this resource. Hence the expected number of resources that C looses with such a process is

$$\mu = \sum_{i\in O\cap C}\frac{a_i+b_i-\gamma}{b_i}$$

It is not difficult to prove the following claim.

▷ Claim 18. For any $C \in \mathcal{K}^{(\geq j)}$,

$$\frac{1}{\gamma^2} \sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1} \cap O| \leq \mu \leq \frac{2}{\gamma} \sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1} \cap O|$$

Proof. Note that we can write

$$\mu = \sum_{i \in O \cap C} \frac{a_i + b_i - \gamma}{b_i} \leq \max_{i \in O \cap C} \left\{ \frac{a_i + b_i - \gamma}{a_i b_i} \right\} \sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1} \cap O|.$$

The reason for this is that each resource i accounts for an expected loss of $(a_i + b_i - \gamma)/b_i$ while it is counted a_i times in the sum

$$\sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1} \cap O|.$$

Similarly,

$$\mu = \sum_{i \in O \cap C} \frac{a_i + b_i - \gamma}{b_i} \geq \min_{i \in O \cap C} \left\{ \frac{a_i + b_i - \gamma}{a_i b_i} \right\} \sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1} \cap O|.$$

Note that by assumption we have that $a_i + b_i > \gamma$. This implies that either a_i or b_i is greater than $\gamma/2$. Assume w.l.o.g. that $a_i \geq \gamma/2$. Since by assumption $a_i \leq \gamma$ we have that

$$\frac{a_i + b_i - \gamma}{a_i b_i} \leq \frac{b_i}{a_i b_i} = \frac{1}{a_i} \leq \frac{2}{\gamma}.$$

In the same manner, since $a_i + b_i > \gamma$ and that $a_i, b_i \leq \gamma$, we can write

$$\frac{a_i + b_i - \gamma}{a_i b_i} \geq \frac{1}{a_i b_i} \geq \frac{1}{\gamma^2}.$$

We therefore get the following bounds

$$\frac{1}{\gamma^2} \sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1} \cap O| \leq \mu \leq \frac{2}{\gamma} \sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1} \cap O|,$$

which is what we wanted to prove. ◁

Assume then that $\mu \leq \frac{|C \cap R_k|}{10^{12} \log^3(n)}$. Note that C cannot loose more than $\sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1} \cap O|$ resources in any case. Therefore, by assumption on μ, and since

$$\mu \geq \frac{1}{\gamma^2} \sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1} \cap O| \ ,$$

we have that

$$\sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1} \cap O| \leq \frac{\gamma^2}{10^{12} \log^3(n)} |C \cap R_k| \leq \frac{10^{11} \log^2 \log(n)}{10^{12} \log^3(n)} |C \cap R_k| \leq \frac{1}{\log(n)} |C \cap R_k| \ .$$

Therefore C looses at most $|C \cap R_k|/\log(n)$ resources. Otherwise we have that

$$\mu > \frac{|C \cap R_k|}{10^{12} \log^2(n)} \geq \frac{\ell^3}{10^{12} \log^3(n)} \geq 200 \log(n)$$

by Lemma 8. Hence noting X the number of deleted resources in C we have that

$$\mathbb{P}\left(X \geq \frac{3}{2}\mu\right) \leq \exp\left(-\frac{\mu}{12}\right) \leq \frac{1}{n^{10}}.$$

With high probability no configuration looses more than

$$\frac{3}{2}\mu \leq \frac{3}{\gamma} \sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1} \cap O| \leq \frac{3}{\gamma} \sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1}|$$

resources. Hence each configuration $C \in \mathcal{K}^{(\geq j)}$ ends with at least

$$\begin{aligned}
&\left(1 - \frac{1}{\log(n)}\right)^{2(k-(j-1))-1} \ell^{k-(j-1)} |C \cap R_k| - \frac{3}{\gamma} \sum_{j \leq h \leq k} \sum_{K \in \mathcal{K}^{(h)}} \ell^{h-(j-1)} |K \cap C \cap R_h| \\
&- \frac{1}{\log(n)} \left(1 - \frac{1}{\log(n)}\right)^{2(k-(j-1))-1} \ell^{k-(j-1)} |C \cap R_k| - \frac{3}{\gamma} \sum_{K \in \mathcal{K}^{(j-1)}} |K \cap C \cap R_{j-1}| \\
&\geq \left(1 - \frac{1}{\log(n)}\right)^{2(k-(j-1))} \ell^{k-(j-1)} |C \cap R_k| - \frac{3}{\gamma} \sum_{j-1 \leq h \leq k} \sum_{K \in \mathcal{K}^{(h)}} \ell^{h-(j-1)} |K \cap C \cap R_h|
\end{aligned}$$

resources which concludes the proof of Lemma 17. ◀

Given Lemma 17 and the intuition below it, it is straightforward to prove the following corollary which will complete the proof of Theorem 6.

▶ **Corollary 19.** *There exists an assignment of resources R to $\mathcal{K}$ such that each configuration $C \in \mathcal{K}$ receives at least $\lfloor |C|/(100\gamma) \rfloor$ resources. Moreover, this assignment can be found in polynomial time.*

Proof. Lemma 17 with $k = 0$ and Claim 16 together imply that we can assign at least

$$\frac{|C|}{2e^2} - \frac{6000}{100.000}|C| \geq \frac{|C|}{100}$$

resources to every $C \in \mathcal{K}$ such that no resource in R is assigned more than γ times. In particular, we can fractionally assign at least $|C|/(100\gamma)$ resources to each $C \in \mathcal{K}$ such that no resource is assigned more than once. By integrality of the bipartite matching polytope, the corollary follows. ◀

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. *ACM Trans. Algorithms*, 8(3), 2012. doi:10.1145/2229163.2229168.

3 Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, page 31–40, New York, NY, USA, 2006. Association for Computing Machinery. doi:10.1145/1132516.1132522.

4 Deeparnab Chakrabarty, Julia Chuzhoy, and Sanjeev Khanna. On allocating goods to maximize fairness. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 107–116. IEEE Computer Society, 2009. doi:10.1109/FOCS.2009.51.

5 Siu-Wing Cheng and Yuchen Mao. Restricted max-min fair allocation. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 37:1–37:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.37.

6 Siu-Wing Cheng and Yuchen Mao. Restricted max-min allocation: Approximation and integrality gap. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 38:1–38:13, 2019. doi:10.4230/LIPIcs.ICALP.2019.38.

7 Sami Davies, Thomas Rothvoss, and Yihao Zhang. A tale of santa claus, hypergraphs and matroids. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2748–2757, 2020. doi:10.1137/1.9781611975994.167.

8 Uriel Feige. On allocations that maximize fairness. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, page 287–293, USA, 2008. Society for Industrial and Applied Mathematics.

9 Michel X Goemans, Nicholas JA Harvey, Satoru Iwata, and Vahab Mirrokni. Approximating submodular functions everywhere. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 535–544. SIAM, 2009.

10 Klaus Jansen and Lars Rohwedder. A note on the integrality gap of the configuration lp for restricted santa claus. *Information Processing Letters*, 164:106025, 2020. doi:10.1016/j.ipl.2020.106025.

11 Andreas Krause, Ram Rajagopal, Anupam Gupta, and Carlos Guestrin. Simultaneous placement and scheduling of sensors. In *Proceedings of the 8th International Conference on Information Processing in Sensor Networks, IPSN 2009, April 13-16, 2009, San Francisco, California, USA*, pages 181–192. IEEE Computer Society, 2009. URL: http://ieeexplore.ieee.org/document/5211932/.

12 Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games Econ. Behav.*, 55(2):270–296, 2006. doi:10.1016/j.geb.2005.02.006.

13 J. K. Lenstra, D. B. Shmoys, and E. Tardos. Approximation algorithms for scheduling unrelated parallel machines. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 217–224, 1987. doi:10.1109/SFCS.1987.8.

14 Robin A Moser and Gábor Tardos. A constructive proof of the general lovász local lemma. *Journal of the ACM (JACM)*, 57(2):1–15, 2010.

15 Lukas Polacek and Ola Svensson. Quasi-polynomial local search for restricted max-min fair allocation. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming*, pages 726–737, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

16 Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 67–74. ACM, 2008. doi:10.1145/1374376.1374389.