Worst-case Bounds for Spending a Common Budget

Pierre Cardi
Université Paris-Dauphine, Université PSL, CNRS, LAMSADE, 75016
Paris, France
pierrecardi2@gmail.com

Laurent Gourvès
Université Paris-Dauphine, Université PSL, CNRS, LAMSADE, 75016
Paris, France
laurent.gourves@lamsade.dauphine.fr

Julien Lesca
Université Paris-Dauphine, Université PSL, CNRS, LAMSADE, 75016
Paris, France
julien.lesca@lamsade.dauphine.fr

ABSTRACT

We study the problem of spending a budget that is common to n agents. Agents submit demands to a central planner who uses the budget to fund a subset of them. The utility of an agent is the part of the budget spent on her own accepted demands. In a fair solution, the successful demands of each agent would represent a $1/n$ fraction of the budget. However, this is rarely possible because every demand is indivisible, i.e. either accepted in its entirety or rejected. We are interested in worst-case bounds on the largest proportion of the budget that is dedicated to the least funded agent. Our approach is not to solve the corresponding max min problem for every instance, but to tackle the problem from a higher level. The size of the largest demand compared to the budget and the number of agents, are two parameters that significantly influence how much the worst-off agent gets. We propose worst-case bounds on the best utility of the least funded agent for the class of instances where the number of agents and the most expensive demand are fixed to given values. A characterization of this quantity is provided for 1 and 2 agents. For more than 2 agents, we propose lower and upper bounds that constitute a $\frac{14}{15}$-approximation of the optimal value. Every existence result is complemented with a polynomial algorithm that builds a feasible solution satisfying our bounds.

KEYWORDS

Fairness; Computational Social Choice; Worst Case Analysis

ACM Reference Format:

Pierre Cardi, Laurent Gourvès, and Julien Lesca. 2021. Worst-case Bounds for Spending a Common Budget. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

We study the problem of spending a budget that is common to n agents. Agents have demands which correspond to indivisible portions of the budget. A solution is to accept a selection of demands which fits in the budget. The problem occurs when, for example, there is a server with a given memory capacity and its users want to store some of their files [10]. Another application is when an organization has a given amount of money to be spent for funding the projects (or any kind of expense) of its members (see [12] for an example with the EU).

The situation can be illustrated with a small example. There are Alice and Bob whose demand sets are $\{29, 25, 20\}$ and $\{19, 17, 16\}$, respectively. The common budget is 100. Several feasible solutions exist, one of them consists of spending 85 by accepting $\{29, 20\}$ and $\{19, 17\}$. Here, it is assumed that every agent's utility is the part of the budget spent on her own accepted demands. We would like to find a fair solution and proportionality is a well accepted notion of fairness. In a proportional solution, the successful demands of each agent represent a $1/n$ fraction of the budget. However, being proportional is rarely possible with indivisible objects. In the previous example, Alice and Bob have utility 49 and 36, respectively, but no feasible solution gives 50 to both agents.

When proportionality is not achievable, we can strive to be as close as possible to it by maximizing the utility of the worst-off agent (max min problem). Beyond the computational difficulty of this task, it is interesting to know, explicitly, how close to $1/n$ we can be. The intuition is that the smaller the demands, the closer we can get to proportionality. Our approach is not to solve the corresponding max min problem for every instance, but to tackle the problem from a higher level. We are interested in explicit worst-case bounds on the largest proportion of the budget that can be dedicated to the least funded agent. These bounds are based on two parameters: the number n of agents and the proportion α of the largest demand compared with the budget. Our primary objective is to quantify how close to proportionality we can be in the class of instances with parameters n and α, and where the demands of every agent exceed the budget divided by n. Different instances with the same parameters n and α are not equivalent: some may admit a solution very close to proportionality, while some others may not. Therefore, we want to lower bound the proportion of the budget that the maximum minimum utility represents, in the worst case. We call $\rho_n(\alpha)$ this quantity for any given pair (α, n). Having an explicit and compact expression of $\rho_n(\alpha)$ provides a quick estimation of how good the utility of the worst-off agent can be. Namely, if B is the common budget, then a feasible solution guaranteeing a utility of $\rho_n(\alpha)B$ to all agents exists. Furthermore, we are interested in the efficient computation of a feasible solution exhibiting this guarantee.

1.1 Related work

Fairly dividing common resources among agents is one of the most prominent problems in social choice. It has been considered in economics for many years [20], and most recently it has attracted the attention of computer scientists [4]. One of the most studied problems in this area consists in allocating a set of items to n agents. Many concepts of fairness have been considered in the literature, including envy-freeness (each agent prefers her share to the share of another), and proportionality (each agent values her share at least a $1/n$ fraction of the whole manna). The items may be divisible (e.g. money), or indivisible. In the former case, envy-free or proportional allocations are guaranteed to exist and can be computed efficiently

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

[6, 18]. When items are indivisible, there exist instances without envy-free or proportional allocation. In order to circumvent this impossibility, different relaxations have been proposed, such as bounded maximal envy [13] or envy-freeness up to one good [7] for envy-freeness. Regarding proportionality, one can mention maxmin fairness, which stipulates that an allocation is fair whenever the utility of the poorest agent is maximized. Unfortunately, computing a maxmin allocation is a hard problem, even for two agents [1, 3, 5]. Another relaxation of proportionality is to target a value smaller than $1/n$. Concretely, each agent's utility should be at least r times her utility for the whole set of items, for some $r \leq 1/n$. Demko and Hill follow this path for the allocation of indivisible goods [9]. They describe a continuous and monotonic function V_n and prove that every instance, where x is the largest utility for a single good, admits a solution where the utility of the worst-off agent is at least $V_n(x)$. Improvements and further algorithmic and game theoretic results have been provided by Markakis and Psomas [14]. Gourvès et al. [11] made some extensions to matroids and improvements of V_n .

This article deals with the problem of fairly spending a common budget. Its connection with the fair allocation of indivisible items comes from the presence of multiple agents having indivisible demands. Our approach is similar to [9, 11, 14] because worst case bounds on the maximum minimum utility are derived with the help of the proportion α of the largest demand. Nevertheless, our results show a significant singularity: the worst-case guarantee in the common budget problem is, as opposed to the fair allocation of indivisible goods, not a continuous function of α. One of the main achievements of this article is a characterization of the worst-case guarantee for 2 agents. Its counterpart in the fair allocation of indivisible goods remains open.

To conclude, one can mention that our model comprises a budget shared a group of agents but it differs from the concept of participatory budgeting [8]. Indeed agents do not vote for on the demands. It also differs from budget division (see e.g. [16]) because the demands have a fixed size and they are either accepted or rejected. A similar difference exists with the problem studied in [10].

1.2 The Model

We consider the situation where a set N of n agents shares a common budget $B \in \mathbb{R}_{>0}$. Each agent j has a demand set $D^j = \{d_1^j, \dots, d_{m^j}^j\}$ where $d_i^j \in \mathbb{R}_{>0}$.

For any integer k, let $[k]$ denote $\{1, \dots, k\}$. In this article, we suppose that the instances satisfy the following assumptions:

$$0 < d_i^j \quad \leq \quad B, \qquad \forall j \in N \text{ and } \forall i \in [m^j] \tag{1}$$

$$\sum_{i=1}^{m^j} d_i^j \quad > \quad B/n, \qquad \forall j \in N \tag{2}$$

Assumption (1) is usual for SUBSET SUM (see for example [15, chapter 4]) because null demands can be accepted, and demands exceeding the budget must be rejected. The motivation for (2) is that the agents have equal rights [17]. Every agent can enjoy a $1/n$ fraction of the whole budget. If the demands of an agent do not exceed B/n, then all her demands can be accepted immediately. Thus, we can reduce the budget by the total value of these demands and remove the agent from the instance. Once all such agents are removed, and the budget is updated accordingly, the resulting instance satisfies (2). Moreover, we assume without loss of generality that

$$d_1^j \geq d_2^j \geq \cdots \geq d_{m^j}^j, \qquad \forall j \in N \tag{3}$$

The largest demand $\max_{j \in N} d_1^j$ is denoted by d^* and parameter α is defined as d^*/B. For example, in the illustrative instance provided in Introduction, the largest demand is 29, and $\alpha = 29/100$. The *value* of a set of demands X is denoted by $v(X)$ and defined as $\sum_{d \in X} d$. A feasible solution $(S^1, \dots, S^n)$ consists of a subset of accepted demands $S^j \subseteq D^j$ for each agent j , such that $\sum_{j \in N} v(S^j) \leq B$. The utility provided by solution $(S_1, \dots, S_n)$ to agent j corresponds to the sum of her accepted demands $v(S^j)$.

We are interested in the "best" function $\rho_n :]0, 1] \rightarrow [0, 1]$ such that any instance with n agents, budget B, and parameter α admits a feasible solution $(S^1, \dots, S^n)$ satisfying $v(S^j) \geq \rho_n(\alpha)B$, for any agent j. By "best" we mean that $\rho_n(\alpha)$ should take the largest possible value. It is easy to check that in the example from the introduction, the largest share of budget that we can ensure to both agents is 45 (when Alice and Bob receive $\{20, 25\}$ and $\{19, 16, 17\}$, respectively). This means that $\rho_2(29/100)$ cannot be larger than $45/100$. We will see that the value of $\rho_2(29/100)$ is even smaller than $45/100$. More formally, let $\mathsf{Inst}(n, \alpha)$ be the set of all instances with n agents, the largest demand is equal to α times the budget, and (1) and (2) are satisfied. For a given instance I of $\mathsf{Inst}(n, \alpha)$ with a budget B_I, let $Z_I^* B_I$ be the maximum utility of the least funded agent. Then we have

$$\rho_n(\alpha) = \inf_{I \in \mathsf{Inst}(n,\alpha)} Z_I^*$$

The range of α is $]0, 1]$ because of (1) and $\rho_n(\alpha)$ is in $[0, 1/n]$. The analysis of ρ_n is often made with the help of the following sub-intervals of $]0, 1]$. For any n, any positive integer k, and any budget B, let $\mathcal{I}_0^n :=]\frac{1}{n}, 1]$ when $n > 1$, $\mathcal{I}_k^n :=]\frac{1}{n(k+1)}, \frac{1}{nk}]$, and $\mathcal{I}_k^n(B) :=]\frac{B}{n(k+1)}, \frac{B}{nk}]$.

1.3 Results and Organization

We start by studying the case with a single agent. We fully characterize ρ_1 which plays an important role in the analysis of the cases with multiple agents. For two agents or more, we provide a characterization of $\tilde{\rho}_n(\alpha)$, which is the largest non-increasing part of $\rho_n(\alpha)$. Moreover, we provide a characterization of ρ_2 and it is the most technical part of the article. Thought $\rho_n(\alpha)$ is not characterized, we determine lower and upper bounds which approximate it with high precision. The existence result of each section is complemented with a polynomial algorithm which outputs a solution such that the value of the worst-off agent is at least the bounds that we found.

Some proofs are omitted due to space limitation.

2 THE SINGLE AGENT CASE

We are given a single agent ($n = 1$), a set of demands $D^1 = \{d_1^1, \dots, d_{m^1}^1\}$ and a positive budget B. Only instances satisfying (1) and (2) are considered. Due to assumption (3), α is equal to $\max_{i \in [m^1]} d_i^1/B = d_1^1/B = d^*/B$. We shall give the following characterization of $\rho_1(\alpha)$. See Figure 1 for an illustration.

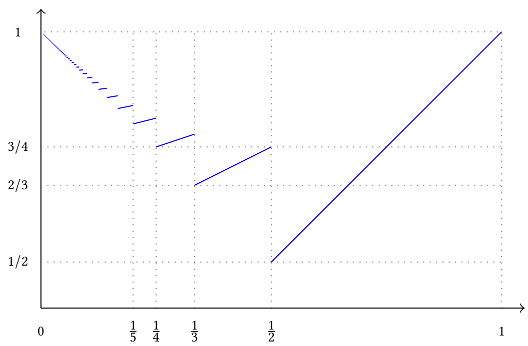

Figure 1: $\rho_1(\alpha)$.

Theorem 1. *For any positive integer k, $\rho_1(\alpha) = \frac{k-1+\alpha}{k}$ when $\alpha \in]\frac{1}{k+1}, \frac{1}{k}] = \mathcal{I}_k^1$.*

Let us begin with lower bounds on $\rho_1(\alpha)$ which derive from the solution returned by Algorithm 1. The algorithm is greedy: start from scratch and keep adding demands by descending value, until inserting an element would exceed the budget. Because of (2), Algorithm 1 cannot select all demands.

Algorithm 1 $(n = 1)$

Require: Budget B, demand set $\{d_1^1, \dots, d_{m^1}^1\}$ satisfying (1) and (2)
If necessary, sort and rename the demands by non-increasing value in order to satisfy (3)
$S \leftarrow \emptyset$
$i \leftarrow 1$
while $d_i^1 + v(S) \leq B$ **do**
  $S \leftarrow S \cup \{d_i^1\}$
  $i \leftarrow i + 1$
end while
return S

Lemma 1. $\rho_1(\alpha) \geq \max(\alpha, 1-\alpha) \geq \frac{1}{2}$ *when* $\alpha \in]0, 1]$.

Lemma 2. *For any integer $k \geq 2$, $\rho_1(\alpha) \geq \frac{k-1+\alpha}{k}$ when $\alpha \in [\frac{1}{k+1}, \frac{1}{k}]$ and $\rho_1(\alpha) \geq \frac{k}{k+1}$ when $\alpha \in]0, \frac{1}{k}]$.*

Proof. Consider the solution S returned by Algorithm 1. The proof is by induction. Let us begin with the base case $k = 2$: $\alpha \in [\frac{1}{3}, \frac{1}{2}]$ and $d_1^1 \in [\frac{B}{3}, \frac{B}{2}]$. The first step of Algorithm 1 is to place d_1^1 in S. The remaining budget is $B - d_1^1$. By Lemma 1, the solution S uses at least half of the remaining budget (conditions (1), (2) and (3) are satisfied). Therefore, the value of S is at least $d_1^1 + \frac{B-d_1^1}{2}$. We have $d_1^1 + \frac{B-d_1^1}{2} \geq \frac{2B}{3}$ because $d_1^1 \geq B/3$. Thus, $\rho_1(\alpha) \geq \frac{2}{3}$ when $\alpha \in [\frac{1}{3}, \frac{1}{2}]$. When $\alpha \in]0, \frac{1}{3}]$, the value of S is at least $(1-\alpha)B$ (see the proof of Lemma 1) which is at least $\frac{2B}{3}$. In all, the value of S is at least $\frac{2B}{3}$ when $\alpha \in]0, \frac{1}{2}]$. In other words, $\rho_1(\alpha) \geq \frac{2}{3}$ when $\alpha \in]0, \frac{1}{2}]$.

Let us prove the lemma for some $k > 2$ provided that it holds for all $t \in \{2, \dots, k-1\}$. Again, Algorithm 1 first places d_1^1 in S. The remaining budget is $B - d_1^1$ and the parameter α' of an auxiliary instance with budget $B' = 1 - d_1^1$ and demand set $D^1 \setminus \{d_1^1\}$ is in $]0, \frac{\alpha}{1-\alpha}]$. When $\alpha \in [\frac{1}{k+1}, \frac{1}{k}]$, α' is upper bounded by $\frac{1}{k-1}$. By induction, we know that $\rho_1(\alpha) \geq \frac{k-1}{k}$ when $\alpha \in]0, \frac{1}{k-1}]$. Thus, Algorithm 1 complements $S = \{d_1^1\}$ with demands whose sum is at least a $\frac{k-1}{k}$ fraction of the remaining budget $B' = B - d_1$. The final value of S is at least $d_1^1 + \frac{k-1}{k}(B - d_1^1) = \frac{k-1+\alpha}{k}B$. Thus, $\rho_1(\alpha) \geq \frac{k-1+\alpha}{k}$ when $\alpha \in [\frac{1}{k+1}, \frac{1}{k}]$.

Finally, let us prove that $\rho_1(\alpha)$ is at least $\frac{k}{k+1}$ when $\alpha \in]0, \frac{1}{k}]$. We have just seen that $\rho_1(\alpha) \geq \frac{k-1+\alpha}{k}$ when $\alpha \in [\frac{1}{k+1}, \frac{1}{k}]$. Therefore, $\rho_1(\alpha)$ is at least $\frac{k-1+\frac{1}{k+1}}{k} = \frac{k}{k+1}$ when $\alpha \in [\frac{1}{k+1}, \frac{1}{k}]$. When $\alpha \in]0, \frac{1}{k+1}]$, we have $\rho_1(\alpha) \geq 1 - \alpha$ by Lemma 1, and this is proved with the solution returned by Algorithm 1. Therefore, $\rho_1(\alpha)$ is at least $1 - \frac{1}{k+1} = \frac{k}{k+1}$. In all, $\rho_1(\alpha)$ is at least $\frac{k}{k+1}$ when $\alpha \in]0, \frac{1}{k}]$. □

The next result is a matching upper bound on $\rho_1(\alpha)$.

Lemma 3. $\rho_1(\alpha) \leq \frac{k-1+\alpha}{k}$ *for any integer $k \geq 1$ and $\alpha \in]\frac{1}{k+1}, \frac{1}{k}] = \mathcal{I}_k^1$.*

Proof. The upper bound derives from the following family of instances.

Instance 1. *Consider a number δ in the interval $[\frac{1}{k+1} + \epsilon, \frac{1}{k}]$ for any integer $k \geq 1$, and $0 < \epsilon < \frac{1}{k} - \frac{1}{k+1}$. The budget B is equal to 1. The agent has $k+1$ demands: one of value δ and k others of value $\frac{1-\delta}{k} + \epsilon$. We have $\delta \geq \frac{1}{k+1} + \epsilon > \frac{1}{k+1} + \frac{k\epsilon}{k+1}$ from which we deduce that $\delta \geq \frac{1-\delta}{k} + \epsilon$. The largest demand being δ, we have $\delta = \alpha$.*

We cannot accept all demands because $\delta + k(\frac{1-\delta}{k} + \epsilon) = 1 + k\epsilon > B$. However, we can accept all demands but the last one whose value is the smallest. By doing so, we get the best feasible solution of value $d_1^1 + \dots + d_k^1 = \left(\alpha + (k-1)\left(\frac{1-\alpha}{k} + \epsilon\right)\right) B = \left(\frac{k-1+\alpha}{k} + \epsilon(k-1)\right) B$. When $\epsilon \rightarrow 0$, the value of this solution tends to $\frac{k-1+\alpha}{k}B$, and the range of α is $]\frac{1}{k+1}, \frac{1}{k}]$. □

Lemmas 1, 2, and 3 give Theorem 1 while Algorithm 1 outputs a feasible solution of value at least $\rho_1(\alpha)B$ in $O(m^1 \log m^1)$ time.

3 DEALING WITH TWO AGENTS OR MORE

This section deals with the case of at least two agents. In addition to (3), we will suppose without loss of generality that $d_1^1 = d^* = \max_{j \in N} d_1^j$. Therefore, parameter α equals d_1^1/B. Before starting we make the following observation.

Observation 1. $\rho_n(\alpha) = 0$ *when* $\alpha > \frac{1}{n}$.

Indeed, if every agent has a unique demand whose value exceeds B/n, then the demand of at least one agent cannot be accepted (otherwise the budget is exceeded). It follows that we can focus on the case $\alpha \in]0, 1/n]$ and strengthen (1) with $0 < d_i^j \leq B/n, \forall j \in N$ and $i \in [m^j]$.

We first propose a characterization of the largest worst-case utility of the least funded agent *with the help of a monotone non-increasing function $\tilde{\rho}_n$* of α (Section 3.1). It is noteworthy that most previous works use non-increasing functions to describe explicit lower bounds on the utility of the worst-off agent in the context of allocating indivisible goods [9, 14]. Function $\tilde{\rho}_n$ constitutes a lower bound on ρ_n, complemented in Section 3.2 with upper bounds for all $\alpha \in]0, 1]$. We conclude in Section 3.3 with a characterization of $\rho_n(\alpha)$ when $\alpha \in \mathcal{I}_1^n$.

3.1 An Optimal *non-increasing* Bound

The following lower bounds derive from the solution of Algorithm 2 which runs in $O(n\mu \log \mu)$ where $\mu = \max_{j\in N} |m^j|$. The algorithm consists in reserving a budget of B/n to every agent. Afterwards Algorithm 1 is run for every agent separately.

Algorithm 2 $(n > 1)$

Require: A budget B, a set N of n agents, and a demand set D^j for all $j \in N$ satisfying (1) and (2)
for all $j \in N$ **do**
  $S^j \leftarrow$ the solution of Algorithm 1 with input B/n and D^j
end for
return $S = (S^1, \dots, S^n)$

Lemma 4. *For any integer $k \geq 1$, $\tilde{\rho}_n(\alpha) \geq \frac{k}{n(k+1)}$ when $\alpha \leq \frac{1}{nk}$.*

Proof. Suppose $\alpha \leq \frac{1}{nk}$ for some positive integer k. The value of every demand is at most $\frac{B}{nk}$. Algorithm 2 reserves a budget of B/n for every agent. A demand represents a fraction α' of at most $\frac{B/(nk)}{B/n} = \frac{1}{k}$ of an agent's reserved budget. Since $\alpha' \leq 1/k$, Lemma 2 indicates that every agent is guaranteed to receive a fraction of $\frac{k}{k+1}$ of her reserved budget, namely $\frac{k}{k+1}\frac{B}{n}$. Thus, $\rho_n(\alpha) \geq \frac{k}{n(k+1)}$. □

Lemma 5. *For any integer $k \geq 1$ and any positive ϵ, $\tilde{\rho}_n(\alpha) \leq \frac{k(1+\epsilon)}{n(k+1)}$ when $\alpha \geq \frac{1+\epsilon}{n(k+1)}$.*

We can derive the following characterization of $\tilde{\rho}_n(\alpha)$ from Observation 1, Lemma 4 and Lemma 5. See Figure 2 for an illustration of $\tilde{\rho}_3(\alpha)$ (in blue).

Corollary 1. *The largest worst-case guarantee with the help of a non-increasing function is $\tilde{\rho}_n(\alpha) := \frac{k}{n(k+1)}$ when $\alpha \in \mathcal{I}_k^n$, for any $k \geq 0$.*

3.2 Upper Bounds for the Multi-agent Case

The upper bounds of ρ_n rely on families of instances. See Figure 2 for an illustration when $n = 3$ (in red).

Proposition 1. *For $n > 1$ and $k \geq 1$ it holds that:*

$$\rho_n(\alpha) \leq \begin{cases} \frac{(k-1)/n+\alpha}{k}, & \text{if } \alpha \in]\frac{1}{n(k+1)}, \frac{k^2+n-1}{n((n-1)(k+1)+k^3)}] \\ \frac{k(1-k\alpha)}{(n-1)(k+1)}, & \text{if } \alpha \in]\frac{k^2+n-1}{n((n-1)(k+1)+k^3)}, \frac{1}{nk}] \end{cases}$$

3.3 A Characterization of $\rho_n(\alpha)$ when $\alpha \in \mathcal{I}_1^n$

The following lemma provides a characterization of $\rho_n(\alpha)$ when $\alpha \in]\frac{1}{2n}, \frac{1}{n}] = \mathcal{I}_1^n$.

Lemma 6. *$\rho_n(\alpha) = \min(\alpha, \frac{1-\alpha}{2(n-1)})$ when $\alpha \in \mathcal{I}_1^n$.*

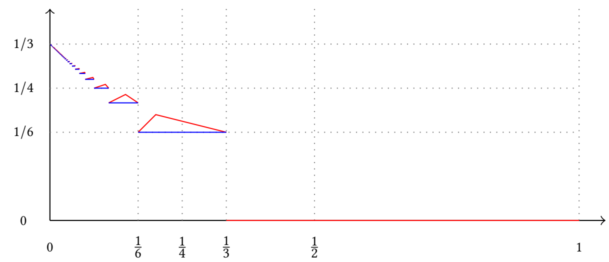

Figure 2: $\tilde{\rho}_3(\alpha)$ in blue and upper bounds on $\rho_3(\alpha)$ in red.

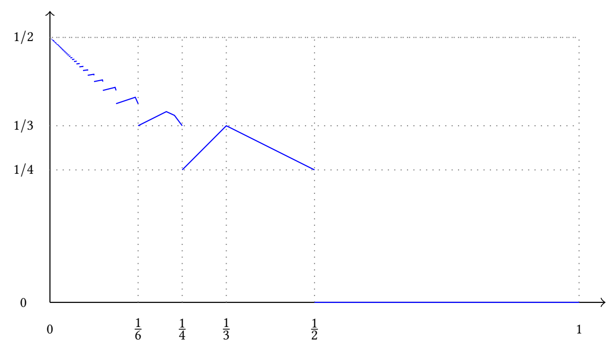

Figure 3: $\rho_2(\alpha)$.

From an algorithmic perspective, when $\alpha \in \mathcal{I}_1^n$, accept d^* (it belongs to agent 1 by hypothesis). Proceed as follows for every $j \neq 1$. Reserve a budget of $\frac{1-\alpha}{n-1}B$. If all the demands of j fit in her reserved budget, then accept them all. Otherwise, run Algorithm 1.

4 A CHARACTERIZATION FOR 2 AGENTS

This section deals with a characterization of $\rho_2(\alpha)$ which is given in (4) and depicted in Figure 3. We also propose a polynomial algorithm which builds a solution (S^1, S^2) such that $\min(v(S^1), v(S^2)) \geq \rho_2(\alpha)B$ for every instance.

$$\rho_2(\alpha) = \begin{cases} \frac{k-1+2\alpha}{2k}, & \text{if } \frac{1}{2(k+1)} < \alpha \leq \frac{k^2+1}{2(k^3+k+1)} \text{ and } k > 2 \\ \frac{k(1-k\alpha)}{k+1}, & \text{if } \frac{k^2+1}{2(k^3+k+1)} < \alpha \leq \frac{1}{2k} \text{ and } k > 2 \\ \frac{1}{4} + \frac{\alpha}{2}, & \text{if } 1/6 < \alpha \leq 11/50 \\ \frac{6(1-\alpha)}{13}, & \text{if } 11/50 < \alpha \leq 4/17 \\ \frac{2-4\alpha}{3}, & \text{if } 4/17 < \alpha \leq 1/4 \\ \alpha, & \text{if } 1/4 < \alpha \leq 1/3 \\ \frac{1-\alpha}{2}, & \text{if } 1/3 < \alpha \leq 1/2 \\ 0, & \text{if } 1/2 < \alpha \leq 1 \end{cases} \quad (4)$$

The strategy is to split the budget between the two agents, and to reduce the problem to two single-agent problems. Hence, Algorithm 1 and the characterization of $\rho_1(\alpha)$ given in (1) can be reused. A direct approach would be to cut the budget evenly (as done in Section 3) and to apply Algorithm 1 separately for each agent. However, this strategy may fail to give the best guarantee.

Let us illustrate this with an instance where $n = 2$, $B = 400$, $D^1 = \{80, 59, 59, 59\}$, and $D^2 = \{75, 64, 64\}$. If we reserve a budget of $B/2 = 200$ to both agents, then the best subsets for the agents are $\{80, 59, 59\}$ and $\{75, 64\}$, respectively. Agents 1 and 2 have utility 198 and 139, respectively. The minimum utility is 139 and no additional demand of agent 2 can be accepted without exceeding the budget. However, (4) indicates that a solution with minimum utility at least $\rho_2(20/100) * 400 = 140$ must exist. Under solution $(\{59, 59, 59\}, \{75, 64, 64\})$, the minimum utility is larger (177 instead of 139) but agent 2's utility exceeds $B/2$. Therefore, reserving a budget of $B/2$ to both agents can be an obstacle to the achievement of the best possible guarantee on the minimum utility.

Our strategy requires to decide how the common budget is split. It also consists of saving a part E of the budget reserved to agent 1, and to transfer this amount to the budget of agent 2. Therefore, an intermediate step is to quantify E (the details are in Section 4.1). To do so, we reduce the set of instances to those which satisfy two hypotheses. We shall see that every instance can be modified in polynomial time to satisfy these hypotheses, and assuming these hypotheses is without loss of generality.

Characterization (4) is proven in Section 4.2. It turns out that $(B/2, B/2 + E)$ is not always the best apportionment of the budget. Sometimes, a better strategy is to give a budget of $\beta^1 = B/2 - \Delta$ to the first agent so that $\rho_1(d^*/\beta^1) * \beta^1$ is guaranteed to her. Then, we can save a certain amount E_Δ, and a budget of $\beta^2 = B/2 + \Delta + E_\Delta$ is given to the second agent. With this budget, the second agent is guaranteed to have a utility of $\rho_1(d_2^*/\beta^2) * \beta^2$ where d_2^* is her largest demand.

The two-agent case differs from the results of Sections 2 and 3 because, sometimes, the best guarantee cannot be reached by a simple greedy algorithm where the demands are taken by non-increasing value until the desired guarantee is obtained. In the previous example, a feasible solution with minimum utility at least 140 exists. Taking the demands by non-increasing value in order to guarantee 140 to both agents would lead to the solution $(\{80, 59, 59\}, \{75, 64, 64\})$ which is infeasible because it violates the budget constraint. Therefore, it is sometimes necessary to depart from the greedy approach to build solutions that meet the guarantee defined in (4).

As a reminder, we suppose that an instance satisfies (1), (2) and (3), and $d^* = \max_{j \in N} d_j^1 = \alpha B$. We also suppose w.l.o.g. that the largest demand belongs to agent 1: $d^* = d_1^1 \geq d_1^2$. Thereafter, we will denote by γ_j the function:

$$\gamma_j(d^*, B) := \rho_j(\alpha)B \tag{5}$$

Therefore, the characterization (1) of ρ_1 becomes

$$\gamma_1(d^*, B) = d^* + \frac{k-1}{k}(B - d^*) \text{ when } d^* \in \mathcal{I}_k^1(B) \tag{6}$$

for any positive integer k. In terms of interpretation, $\gamma_j(d^*, B)$ is the quantity that is guaranteed to the worst-off agent while $\rho_j(\alpha)$ represents the guaranteed proportion of the budget. Characterizing ρ_j is strictly equivalent to characterizing γ_j but we will find convenient to use γ_j instead of ρ_j.

4.1 Bounding the Budget Saving for a Single Agent

The goal of this section is to determine how much can be saved, in the worst case, from the reserved budget of an agent, while a certain quantity is guaranteed to her. Let $\mathsf{Inst}_1(d^*, B)$ be the set of all instances with a single agent ($n = 1$), a budget B, the largest demand is equal to d^*, and (1) and (2) are satisfied. For any instance $I \in \mathsf{Inst}_1(d^*, B)$ with demand set D_I, let R_I be a subset of demands of minimum value within $\{S \subseteq D_I \mid v(S) \geq \gamma_1(d^*, B)\}$ where $v(S) := \sum_{d \in S} d$. If R_I is not unique, then select one arbitrarily. We then define:

$$R(d^*, B) := \sup\{v(R_I) \mid I \in \mathsf{Inst}_1(d^*, B)\} \tag{7}$$

Thus, any instance in $\mathsf{Inst}_1(d^*, B)$ admits a solution S such that $\gamma_1(d^*, B) \leq v(S) \leq R(d^*, B)$. At least $B - R(d^*, B)$ can be saved from the budget, so we are interested in the saving $E(d^*, B)$ which is defined as $B - R(d^*, B)$. Moreover, we aim at constructing S such that $\gamma_1(d^*, B) \leq v(S) \leq R(d^*, B)$ in polynomial time.

Theorem 2. *For every single agent instance $I \in \mathsf{Inst}_1(d^*, B)$ with demand set D_I, one can build in polynomial time a solution $S \subseteq D_I$ such that $\gamma_1(d^*, B) \leq v(S) \leq B - E(d^*, B)$ where $E(d^*, B)$ is greater or equal to*

$$\begin{cases} B - \frac{k+1}{k}(B - d^*), & \text{for } \frac{1}{k+1}B < d^* \leq \frac{k+1}{k^2+k+1}B \text{ and } k > 2 \\ B - kd^*, & \text{for } \frac{k+1}{k^2+k+1}B < d^* \leq \frac{1}{k}B \text{ and } k > 2 \\ \frac{1}{2}(3d^* - B), & \text{for } B/3 < d^* \leq 7B/17 \\ \frac{2}{3}(B - 2d^*), & \text{for } 7B/17 < d^* \leq B/2 \\ B - d^*, & \text{for } B/2 < d^* \leq B \end{cases}$$

On our way to proving Theorem 2, the first step is to observe that, instead of studying the entire set $\mathsf{Inst}_1(d^*, B)$, one can suppose that an instance satisfies the following two hypotheses.

Hypothesis 1. *At most one demand of D_I is strictly smaller than $d^*/2$.*

Hypothesis 2. *$v(D_I) - v(\{d\}) \leq B$ holds for all $d \in D_I$. Said differently, the removal of every single demand would violate (2).*

Every instance can be modified in polynomial time to satisfy these hypotheses. Use Algorithm 3 to do so.

Algorithm 3 Enforcing Hypotheses 1 and 2

Require: An instance $I \in \mathsf{Inst}_1(d^*, B)$ with demand set D_I
  while D_I contains two demands d and d' such that $d \leq d' < d^*/2$ **do**
    $D_I \leftarrow (D_I \setminus \{d, d'\}) \cup \{d + d'\}$
  end while
  Rename the demands of D_I by non increasing value
  Let t be the index such that $\sum_{i=1}^{t-1} d_i^1 \leq B$ and $\sum_{i=1}^{t} d_i^1 > B$
  $D_I \leftarrow \{d_1^1, \ldots, d_1^t\}$
  return D_I

Algorithm 3, which runs in $O(|D_I| \log |D_I|)$, has the following properties: B and d^* remain unchanged, and (1) and (2) are preserved. We can also observe that, though Algorithm 3 modifies the original instance, a feasible solution S of the original instance can be retrieved from any feasible solution S' returned by Algorithm 3, and $v(S) = v(S')$. Indeed, by construction, every demand of S' corresponds to a bundle of demands of the original instance. Our strategy is to prove Theorem 2 for any instance of $\mathsf{Inst}_1(d^*, B)$ satisfying Hypotheses 1 and 2. Then, we proceed as follows. Take $I \in \mathsf{Inst}_1(d^*, B)$ and run Algorithm 3. Find a solution S' that satisfies Theorem 2. Derive from S' a feasible solution S of I which also satisfies Theorem 2. See Algorithm 4 for the construction of S'.

Algorithm 4

Require: A single agent instance $I \in \mathsf{Inst}_1(d^*, B)$
Ensure: A subset of demands S such that $\gamma_1(d^*, B) \leq v(S) \leq B - E(d^*, B)$
  Run Algorithm 3 on I to satisfy Hypotheses 1 and 2
  The new demands, denoted by $D = \{d_1, \ldots, d_m\}$, satisfy $d_1 \geq \ldots \geq d_m$, where $d_1 = d^*$
  $S \leftarrow \emptyset$ and $S' \leftarrow \emptyset$
  if $B/3 < d^* \leq B$ **then**
    Find $A \subseteq D$ such that $\gamma_1(d^*, B) \leq v(A) \leq B - E(d^*, B)$ by exhaustive search
    $S' \leftarrow A$
  else
    $S' \leftarrow$ Algorithm 5 (B, D)
  end if
  Retrieve S from S'
  return S

4.1.1 Budget saving when $d^* \in \mathcal{I}_k^1(B)$ for any $k > 2$. We are going to show that $R(d^*, B) \leq \max(kd^*, \frac{k+1}{k}(B - d^*))$ holds when $k > 2$.

To this end, we will see that no matter the instance, there always exists a solution A such that:

$$\gamma_1(d^*, B) \leq v(A) \leq \max(kd^*, \frac{k+1}{k}(B - d^*)) \tag{8}$$

This solution A is built with Algorithm 5 which is *almost* greedy in the sense that it produces a first solution A_1 in a greedy manner. Then, two other solutions A_2 and A_3, which are close to A_1, are computed. We are going to see that one of them satisfies (8).

Algorithm 5

Require: A budget B and a demand set $\{d_1^1, \ldots, d_{m^1}^1\}$ satisfying (1), (2) and (3) as well as Hypotheses 1 and 2.
  $p \leftarrow \max(kd^*, \frac{k+1}{k}(B - d^*))$
  $A_1 \leftarrow \emptyset, A_2 \leftarrow \emptyset, A_3 \leftarrow \emptyset$
  $S \leftarrow 0$
  $i \leftarrow 1$
  while $S \leq \gamma_1(d^*, B)$ **do**
    $A_1 \leftarrow A_1 \cup \{d_i^1\}$
    $S \leftarrow S + d_i^1$
    $i \leftarrow i + 1$
  end while
  $A_2 \leftarrow A_1 \setminus \{d_{i-1}^1\} \cup \{d_i^1\}$
  $A_3 \leftarrow A_1 \setminus \{d^*\} \cup \{d_i^1\}$ where $d^* = d_1^1$
  if $v(A_1) > p$ **then**
    if $v(A_2) > p$ **then return** A_3
    else return A_2
  else return A_1

Lemma 7. *Algorithm 5 runs in $O(m^1)$ and returns a solution A satisfying (8) when $d^* \in \mathcal{I}_k^1(B)$ and $k > 2$.*

Lemma 7 implies that $R(d^*, B) \leq \max(kd^*, \frac{k+1}{k}(B - d^*))$ when $d^* \in \mathcal{I}_k^1(B)$ and $k > 2$. Combined with $E(d^*, B) = B - R(d^*, B)$, we obtain the following bound:

Corollary 2. *For $d^* \in \mathcal{I}_k^1(B)$ and $k > 2$, it holds that:*

$$E(d^*, B) \geq \begin{cases} B - \frac{k+1}{k}(B - d^*), & \text{for } \frac{1}{k+1}B < d^* \leq \frac{k+1}{k^2+k+1}B \\ B - kd^*, & \text{for } \frac{k+1}{k^2+k+1}B < d^* \leq \frac{1}{k}B \end{cases}$$

4.1.2 Budget saving when $d^* \in \mathcal{I}_2^1(B)$. The case $k = 2$ is considered in this section. The result is slightly different from $k > 2$, since $v(R_I)$ can be greater than $\max(kd^*, \frac{k+1}{k}(B - d^*))$ for some instances I when $k = 2$. We are going to show that $R(d^*, B)$ is smaller or equal to $\max(2d^* + \frac{B-d^*}{3}, \frac{3}{2}(B - d^*))$ for any $d^* \in \mathcal{I}_2^1(B)$. Our proof strategy consists of separating the cases where the largest demand is in or out of R_I.

Lemma 8. *$v(R_I) \leq \max(2d^* + \frac{B-2d^*}{3}, \frac{3}{2}(B - d^*))$ holds for all instance I such that the largest demand belongs to R_I and $d^* \in \mathcal{I}_2^1(B)$.*

Afterwards, we show that $v(R_I) \leq \max(2d^*, \frac{3}{2}(B - d^*))$ when d^* is out of R_I.

Lemma 9. *$v(R_I) \leq \max(2d^*, \frac{3}{2}(B - d^*))$ holds for all instance I such that the largest demand is out of R_I and $d^* \in \mathcal{I}_2^1(B)$.*

Lemmas 8 and 9 indicate that $v(R_I) \leq \max(2d^* + \frac{B-2d^*}{3}, \frac{3}{2}(B - d^*))$. Using the fact that $E(d^*, B) = B - R(d^*, B)$, we obtain the following lower bound:

Corollary 3. *For any $d^* \in \mathcal{I}_2^1(B)$ we have:*

$$E(d^*, B) \geq \begin{cases} \frac{1}{2}(3d^* - B), & \text{for } B/3 < d^* \leq 7B/17 \\ \frac{2}{3}(B - 2d^*), & \text{for } 7B/17 < d^* \leq B/2 \end{cases}$$

4.1.3 Budget saving when $d^* \in \mathcal{I}_1^1(B)$. It turns out that $S' = \{d^*\}$ satisfies $\gamma_1(d^*, B) = v(S') = B - E(d^*, B)$ when $d^* \in \mathcal{I}_1^1(B)$.

Lemma 10. *$E(d^*, B) = B - d^*$ when $d^* \in \mathcal{I}_1^1(B)$.*

4.1.4 Proof of Theorem 2. The bounds on E announced in Theorem 2 follows from Corollaries 2 and 3, and Lemma 10. The solution S such that $\gamma_1(d^*, B) \leq v(S) \leq B - E(d^*, B)$ is built with Algorithm 4. More precisely, Algorithm 5 is run when $0 < d^* \leq B/3$, and an exhaustive search is used when $B/3 < d^* \leq B$.

Let us first argue that Algorithm 4 finds a correct solution. We know from Lemma 7 that the solution S returned by Algorithm 5 satisfies $\gamma_1(d^*, B) \leq v(S) \leq B - E(d^*, B)$ when $0 < d^* \leq B/3$. Lemmas 8 and 9 indicate that $v(R_I) \leq B - E(d^*, B)$, and $\gamma_1(d^*, B) \leq v(R_I)$ holds by the definition of R_I. Therefore, there exists a feasible solution S such that $\gamma_1(d^*, B) \leq v(S) \leq B - E(d^*, B)$ when $B/3 < d^* \leq B$, and the exhaustive search finds it. In all, Algorithm 4 is correct.

It remains to prove that Algorithm 4 runs in polynomial time. Starting with a set of m^1 demands, Algorithm 3 is run. It takes $O(m^1 \log m^1)$ time to get a modified instance with m demands satisfying Hypotheses 1 and 2. If $B/3 < d^* \leq B$ then S' such that $\gamma_1(d^*, B) \leq v(S') \leq B - E(d^*, B)$ is found by exhaustive search. Observe that an instance satisfying Hypotheses 1 and 2, with $d^* \in \mathcal{I}_k^1(B)$, contains at most $2k + 1$ demands: one demand of value d^* with $d^* > B/(k + 1)$ because $d^* \in \mathcal{I}_k^1(B)$, and at most $2k$ other smaller demands because at most one demand is strictly smaller than $d^*/2$. We have $k \leq 2$ when $B/3 < d^* \leq B$. Therefore the instance contains at most 5 demands and a simple exhaustive search gives S' in constant time. If $0 < d^* \leq B/3$ then S' is found with

Algorithm 5 whose running time is $O(m)$ (see Lemma 7). Finally, Algorithm 4 retrieves S, solution of the original instance, from S'. Concretely, every demand of S originally comes from a group of demands which were bundled during the execution of Algorithm 3. Thus, the task is made in $O(m^1)$ operations.

4.2 Characterization for Two Agents

The goal of this section is to characterize $\rho_2(\alpha)$ and to build a solution in which the worst-off agent has utility at least $\rho_2(\alpha)B$. When $d^* \in]B/4, B]$, the characterization of $\rho_2(\alpha)$ is immediate from Observation 1 and Lemma 6 with $n = 2$. It corresponds to $\alpha \in]1/4, 1]$ (lines 6, 7 and 8 of (4)). For $d^* \in]B/2, 0]$ nothing can be guaranteed because $\rho_2(\alpha) = 0$. Regarding the construction of a solution which guarantees $\rho_2(\alpha)B$ to both agents when $d^* \in]B/4, B/2]$, see the polynomial algorithm described right after Lemma 6.

It remains to tackle the case $d^* \in]0, B/4]$. To do so, we characterize $\gamma_2(d^*, B)$ and $\rho_2(\alpha)$ follows because $\gamma_2(d^*, B) = \rho_2(\alpha)B$. The proof is cut in two parts, whether $d^* \in]0, B/6]$ or $d^* \in]B/6, B/4]$. These cases correspond to $\alpha \in]0, 1/6]$ (lines 1 and 2 of (4)), and $\alpha \in]1/6, 1/4]$ (lines 3, 4 and 5 of (4)), respectively.

Thereafter B' denotes the *semi-budget*, namely $B' := B/2$. We suppose w.l.o.g. that agent 1 (also denoted by $A1$) has a demand of value d^*. $A2$ stands for agent 2 and we denote by d_2^* her largest demand, namely $d_2^* := \max(D^2)$. Algorithm 6 describes how to build a solution in which the worst-off agent has utility at least $\rho_2(\alpha)B$ when $d^* \in]0, B/4]$. Note that Algorithm 6 is polynomial because Algorithms 4 and 1 are polynomial. Thus, it remains to prove the correctness of Algorithm 6: see Sections 4.2.1 and 4.2.2 for $d^* \in]0, B/6]$ and $d^* \in]B/6, B/4]$, respectively.

Algorithm 6

Require: Budget B and D^j for all $j \in \{1, 2\}$ satisfying (1) and (2)
$B' \leftarrow B/2$
Let $\Delta := \begin{cases} \frac{25d^*-11B'}{13}, & \text{if } \frac{11}{25}B' < d^* \leq \frac{8}{17}B' \text{ and } \frac{B'}{3} < d_2^* \\ B' - 2d^*, & \text{if } \frac{8}{17}B' < d^* < B'/2 \text{ and } \frac{B'}{3} < d_2^* \\ 0, & \text{otherwise} \end{cases}$
$B^{A1} \leftarrow B' - \Delta$
Use Algorithm 4 to build $S^1 \subseteq D^1$ of value at least $\gamma_1(d^*, B^{A1})$ such that $E(d^*, B^{A1})$ is saved
$B^{A2} \leftarrow B' + \Delta + E(d^*, B^{A1})$
Use Algorithm 1 to build $S^2 \subseteq D^2$ of value at least $\gamma_1(d_2^*, B^{A2})$
return (S^1, S^2)

In the sequel, we use the following result multiple times where $t_{B,k} := \lim_{x \to B/(k+1)^+} \gamma_1(x, B)$ for any $k \geq 2$.

Lemma 11. *For any $k \geq 2$ we have that:*

$$\inf_{x \leq \frac{B}{k}} \gamma_1(x, B) = t_{B,k} = \frac{k}{k+1}B \tag{9}$$

4.2.1 When $d^* \in]0, B/6]$. In this section we suppose that $d^* \in \mathcal{I}_k^1(B') = \mathcal{I}_k^2(B)$ with $k > 2$, which corresponds to $d^* \in]0, B/6]$. Based on the lower bound on $E(d^*, B)$ given in Corollary 2, we will distinguish two cases: $\frac{B'}{k+1} < d^* \leq \frac{k+1}{k^2+k+1}B'$ and $\frac{k+1}{k^2+k+1}B' < d^* \leq \frac{B'}{k}$.

Lemma 12. *It holds that $\gamma_2(d^*, B) \geq \gamma_1(d^*, B')$ when $\frac{B'}{k+1} < d^* \leq \frac{k+1}{k^2+k+1}B'$.*

Proof. Suppose that we assign half of the budget to $A1$, as indicated in Algorithm 6. Then, there exists a solution guaranteeing $\gamma_1(d^*, B')$ to her, and $E(d^*, B')$ can be saved from the budget of $A1$ (use Algorithm 4). Thus, we can assign a budget of $B^{A2} = B' + E(d^*, B')$ to the second agent. Corollary 2 gives that:

$$B^{A2} = B - \frac{k+1}{k}(B' - d^*) \tag{10}$$

Then, we can guarantee $\gamma_1(d_2^*, B^{A2})$ to the second agent (use Algorithm 1). Because $d_2^* \leq \frac{B'}{k} \leq \frac{B^{A2}}{k}$, we get $\gamma_1(d_2^*, B^{A2}) \geq \frac{k}{k+1}B^{A2}$ by Lemma 11. Furthermore, by using (10) and (6), we can verify that $\frac{k}{k+1}B^{A2} \geq \gamma_1(d^*, B')$. This inequality ensures that we can guarantee at least $\gamma_1(d^*, B')$ to both agents, i.e. $\gamma_2(d^*, B) \geq \gamma_1(d^*, B')$, when $\frac{B'}{k+1} < d^* \leq \frac{k+1}{k^2+k+1}B'$. □

Lemma 13. *We have the following bound for $\frac{k+1}{k^2+k+1}B' < d^* \leq \frac{B'}{k}$:*

$$\gamma_2(d^*, B) \geq \begin{cases} \gamma_1(d^*, B'), & \text{if } \frac{(k+1)B'}{k^2+k+1} < d^* \leq \frac{(k^2+1)B'}{k^3+k+1} \\ \frac{k}{k+1}(B - kd^*), & \text{if } \frac{(k^2+1)B'}{k^3+k+1} < d^* \leq \frac{B'}{k} \end{cases}$$

Proof. Let $B^{A1} = B'$, as indicated in Algorithm 6. Use Algorithm 4 to select a subset S^1 of agent 1's demands so that

$$v(S^1) \geq \gamma_1(d^*, B') \tag{11}$$

At least $E(d^*, B') = B' - kd^*$ is saved according to Corollary 2. This quantity is added to the budget of agent 2:

$$B^{A2} = B' + E(d^*, B') = B - kd^* \tag{12}$$

Knowing this, we can select with Algorithm 1 a subset of her demands S^2 such that $v(S^2) \geq \gamma_1(d_2^*, B^{A2})$. Here again, since $B' \leq B^{A2}$ and $d_2^* \leq d^* \leq \frac{B'}{k}$ we get that $d_2^* \leq \frac{B^{A2}}{k}$. Thus, we can apply Lemma 11 and get that $v(S^2) \geq \frac{k}{k+1}B^{A2}$. It follows from (12) that:

$$v(S^2) \geq \frac{k}{k+1}(B - kd^*) \tag{13}$$

In addition, we obtain from (6) that:

$$\frac{k}{k+1}(B - kd^*) < \gamma_1(d^*, B') \iff d^* > \frac{k^2+1}{k^3+k+1}B' \tag{14}$$

Thus, the expected result follows from (11), (13) and (14). □

Lemmas 12 and 13 provide the following lower bounds on the quantity $\gamma_2(d^*, B)$ when $d^* \in \mathcal{I}_k^2(B)$ and $k > 2$. These lower bounds are obtained with the solution returned by Algorithm 6.

$$\gamma_2(d^*, B) \geq \begin{cases} \gamma_1(d^*, B'), & \text{for } \frac{1}{k+1}B' < d^* \leq \frac{k^2+1}{k^3+k+1}B' \\ \frac{k}{k+1}(B - kd^*), & \text{for } \frac{k^2+1}{k^3+k+1}B' < d^* \leq \frac{1}{k}B' \end{cases} \tag{15}$$

Corollary 4. *For $k > 2$ and $\frac{1}{2(k+1)} < \alpha \leq \frac{1}{2k}$ it holds that*
$$\rho_2(\alpha) = \begin{cases} \frac{k-1+2\alpha}{2k}, & \text{if } \frac{1}{2(k+1)} < \alpha \leq \frac{k^2+1}{2(k^3+k+1)} \\ \frac{k(1-k\alpha)}{k+1}, & \text{if } \frac{k^2+1}{2(k^3+k+1)} < \alpha \leq \frac{1}{2k} \end{cases}.$$

Proof. We obtain $\rho_2(\alpha) \geq \frac{k-1+2\alpha}{2k}$ if $\frac{1}{2(k+1)} < \alpha \leq \frac{k^2+1}{2(k^3+k+1)}$, and $\rho_2(\alpha) \geq \frac{k(1-k\alpha)}{k+1}$ if $\frac{k^2+1}{2(k^3+k+1)} < \alpha \leq \frac{1}{2k}$ from (15) with the help of $\rho_2(\alpha)B = \gamma_2(d^*, B)$, $B' = B/2$, $d^* = \alpha B$, and the definition of γ_1 given in (6). Conversely, we know that $\rho_2(\alpha) \leq \frac{k-1+2\alpha}{2k}$ if $\frac{1}{2(k+1)} < \alpha \leq \frac{k^2+1}{2(k^3+k+1)}$, and $\rho_2(\alpha) \leq \frac{k(1-k\alpha)}{k+1}$ if $\frac{k^2+1}{2(k^3+k+1)} < \alpha \leq \frac{1}{2k}$ from Proposition 1 where $n = 2$. □

Thus, Corollary 4 gives a characterization of $\rho_2(\alpha)$ when $\alpha \in]0, B/6]$ as claimed in (4), and Algorithm 6 produces a solution where the utility of the worst-off agent is $\rho_2(\alpha)B$ when $\alpha \in]0, B/6]$.

4.2.2 When $d^* \in]B/6, B/4]$. Now we consider the case where $k = 2$, $d^* \in \mathcal{I}_2^1(B')$, which corresponds to $d^* \in]B/6, B/4]$. The bounds are obtained with the solution built by Algorithm 6.

Lemma 14. *For $d^* \in \mathcal{I}_2^1(B')$, it holds that:*

$$\gamma_2(d^*, B) \geq \begin{cases} \gamma_1(d^*, B'), & \text{for } d^* \leq \frac{11}{25}B' \\ \gamma_1(d^*, B' - \Delta), & \text{for } \frac{11}{25}B' < d^* \leq \frac{8}{17}B' \\ \frac{4}{3}(B' - d^*), & \text{for } \frac{8}{17}B' < d^* \end{cases}$$

and

$$\Delta := \begin{cases} 0, & \text{for } \frac{1}{3}B' < d^* \leq \frac{11}{25}B' \\ \frac{25d^* - 11B'}{13}, & \text{for } \frac{11}{25}B' < d^* \leq \frac{8}{17}B' \\ B' - 2d^*, & \text{for } \frac{8}{17}B' < d^* \end{cases}$$

Corollary 5. *It holds that:*

$$\rho_2(\alpha) = \begin{cases} 1/4 + \alpha/2, & \text{if } 1/6 < \alpha \leq 11/50 \\ \frac{6}{13}(1 - \alpha), & \text{if } 11/50 < \alpha \leq 4/17 \\ 2/3 - 4\alpha/3, & \text{if } 4/17 < \alpha \leq 1/4 \end{cases}$$

Proof. The lower bounds on $\rho_2(\alpha)$ are immediate from Lemma 14 where $B' = B/2$, $d^* = \alpha B$, $\gamma_2(d^*, B) = \rho_2(\alpha)B$, and the definition of γ_1.

These lower bounds can be paired with matching upper bounds as follows. Use Proposition 1 with $n = k = 2$ to get that $\rho_2(\alpha) \leq 1/4 + \alpha/2$ if $1/6 < \alpha \leq 5/22$, and $\rho_2(\alpha) \leq 2/3 - 4\alpha/3$ if $5/22 < \alpha \leq 1/4$. Since $11/50 < 5/22 < 4/17$, it follows that

$$\rho_2(\alpha) \leq \begin{cases} 1/4 + \alpha/2, & \text{if } 1/6 < \alpha \leq 11/50 \\ 2/3 - 4\alpha/3, & \text{if } 4/17 < \alpha \leq 1/4 \end{cases}$$

The remaining part is obtained with the following instance.

Instance 2. *Suppose $B = 17 - \delta$ with $\delta \in [0, 1/3]$. Agent 1 has 4 demands of values $4-\delta, 2+\epsilon, 2+\epsilon$, and 2, where $1 \gg \epsilon > 0$. Agent 2 has 3 demands, each of value 3. Thus, $\alpha = \frac{4-\delta}{17-\delta}$ and $\alpha \in [11/50, 4/17]$.*

If we accept 1, 2 or 3 demands of agent 2 then she gets 3, 6, or 9, respectively. An exhaustive search of all the possible subsets of demands for agent 1 provides the following positive amounts: 2, $2 + \epsilon$, $4 - \delta$, $4 + \epsilon$, $4 + 2\epsilon$, $6 - \delta$, $6 - \delta + \epsilon$, $6 + 2\epsilon$, $8 - \delta + \epsilon$, $8 - \delta + 2\epsilon$, and $10 - \delta + 2\epsilon$. If we accept the 3 demands of agent 2, then the remaining budget is $8 - \delta$. In that case we can give at most $6 + 2\epsilon$ to agent 1. If we accept two demands of agent 2 then she gets 6. The best guarantee for this instance when $\epsilon \to 0$ is $\frac{6}{17-\delta}$. Since $\frac{6}{17-\delta} = \frac{6}{13}(1 - \alpha)$ when $\alpha = \frac{4-\delta}{17-\delta}$, the best guarantee for this instance is $\frac{6}{13}(1 - \alpha)$ when $\alpha \in [\frac{11}{50}, \frac{4}{17}]$. Thus, we can conclude that $\rho_2(\alpha) \leq \frac{6}{13}(1 - \alpha)$ if $\frac{11}{50} < \alpha \leq \frac{4}{17}$. □

Therefore, Corollary 5 gives the characterization of $\rho_2(\alpha)$ when $\alpha \in]1/6, 1/4]$ as claimed in (4), and Algorithm 6 produces a solution where the utility of the worst-off agent is $\rho_2(\alpha)B$ when $\alpha \in]B/6, B/4]$.

5 ACCURACY

For n agents, let $\text{Acc}(n)$ be the *accuracy* of our bounds on ρ_n. It is defined as the largest ratio between the lower and the upper bound.

$$\text{Acc}(n) = \sup_{\alpha \in]0,1/n]} \frac{\text{best known lower bound on } \rho_n(\alpha)}{\text{best known upper bound on } \rho_n(\alpha)}.$$

We exclude $\alpha \in]\frac{1}{n}, 1] = \mathcal{I}_0^n$ in the definition of $\text{Acc}(n)$ because $\rho_n(\alpha) = 0$ on $\mathcal{I}_0^n$ (see Observation 1). The accuracy is between 0 and 1. The closer to 1 the better the accuracy. Having $\text{Acc}(n) = 1$ would mean that we have a characterization of ρ_n.

Proposition 2. *$\text{Acc}(n)$ restricted to $\mathcal{I}_k^n$ is $\frac{(n-1)(k+1)+k^3}{(n-1)(k+1)+k^3+1}$.*

Since a characterization of ρ_n is known for $n = 1, 2$, the worst accuracy is for $n > 2$. According to Proposition 2, the worst case occurs when $k = 2$ and $n = 3$, namely $\text{Acc}(n) \geq 14/15$.

6 FUTURE WORK

An immediate future work would be to characterize ρ_n for any number of agents, or to provide an approximation with a ratio better than 14/15. The present work proposed a common worst-case bound and it depends on α defined as the largest demand (over all agents) divided by B. However, as done by Markakis and Psomas [14], each agent j may have her own worst-case guarantee as a function of α_j where α_j is agent j's largest demand divided by B.

The common budget problem is cast as a multi-agent subset sum problem. The utility for a demand corresponds to its value. However, studying a multi-agent knapsack problem where the utility for an object is not necessarily aligned with its size would be interesting [2]. In that case, one needs significant parameters (analogous to α) before determining worst-case bounds on the value of an agent. In the same vein, one can think of a bi-dimensional version of the problem studied in this article. Suppose there is a common piece of land where multiple agents want to locate private facilities (see for example [19] for a recent work on the fair division of land). The problem can be to place some geometric shapes (the facilities) on a given area (the piece of land) in such a way that the shapes fit in the area and do not overlap. Then, which surface an agent is guaranteed to cover?

Finally, the strategic aspect of the common budget problem deserves attention. If the agents submit their demands to a central authority, then the selection of accepted demands should incentivize the agents to report the true values of their demands. At first glance, the fact that we accept all the demands of an agent if their sum is at most B/n, does not promote truthfulness. Indeed, every agent is tempted to submit her largest subset of demands which is below B/n. However, the agents are not necessarily aware of B and n when they communicate their demands.

REFERENCES

[1] Nikhil Bansal and Maxim Sviridenko. 2006. The Santa Claus problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, May 21-23, 2006.* Association for Computing Machinery, New York, NY, United States, Seattle, WA, USA, 31–40.

[2] Nawal Benabbou and Patrice Perny. 2016. Solving Multi-Agent Knapsack Problems Using Incremental Approval Voting. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016.* IOS Press, The Hague, The Netherlands, 1318–1326.

[3] Ivona Bezáková and Varsha Dani. 2005. Allocating indivisible goods. *SIGecom Exchanges* 5, 3 (2005), 11–18.

[4] Sylvain Bouveret, Yann Chevaleyre, Nicolas Maudet, and Hervé Moulin. 2016. *Fair Allocation of Indivisible Goods.* Cambridge University Press, Cambridge, UK, 284–310.

[5] Sylvain Bouveret, Michel Lemaître, Hélène Fargier, and Jérôme Lang. 2005. Allocation of Indivisible Goods: A General Model and Some Complexity Results. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems.* Association for Computing Machinery, New York, NY, United States, 1309–1310.

[6] Steven J. Brams and Alan D. Taylor. 1996. *Fair Division: From Cake-Cutting to Dispute Resolution.* Cambridge University Press, Cambridge, UK.

[7] Eric Budish. 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy* 119, 6 (2011), 1061–1103.

[8] Yves Cabannes. 2004. Participatory budgeting: a significant contri- bution to participatory democracy. *Environment and Urbanization* 16, 1 (2004), 27–46.

[9] Stephen Demko and Theodore P. Hill. 1988. Equitable distribution of indivisible objects. *Mathematical Social Sciences* 16, 2 (1988), 145–158.

[10] Eric J. Friedman, Vasilis Gkatzelis, Christos-Alexandros Psomas, and Scott Shenker. 2019. Fair and Efficient Memory Sharing: Confronting Free Riders. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.* The AAAI Press, Palo Alto, California USA, Palo Alto, California, USA, 1965–1972.

[11] Laurent Gourvès, Jérôme Monnot, and Lydia Tlilane. 2015. Worst case compromises in matroids with applications to the allocation of indivisible goods. *Theoretical Computer Science* 589 (2015), 121 – 140.

[12] Ewa Kiryluk-Dryjska. 2014. Fair Division Approach for the European Union's Structural Policy Budget Allocation: An Application Study. *Group Decision and Negotiation* 23 (2014), 597–615.

[13] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *Proceedings 5th ACM Conference on Electronic Commerce (EC-2004), May 17-20, 2004.* Association for Computing Machinery, New York, NY, United States, New York, NY, USA, 125–131.

[14] Evangelos Markakis and Christos-Alexandros Psomas. 2011. On Worst-Case Allocations in the Presence of Indivisible Goods. In *Internet and Network Economics - 7th International Workshop, WINE 2011, December 11-14, 2011. Proceedings.* Springer, 2011e édition, Singapore, 278–289.

[15] Silvano Martello and Paolo Toth. 1990. *Knapsack Problems: Algorithms and Computer Implementations.* John Wiley & Sons, Inc., USA.

[16] Marcin Michorzewski, Dominik Peters, and Piotr Skowron. 2020. Price of Fairness in Budget Division and Probabilistic Social Choice. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020.* The AAAI Press, Palo Alto, California USA, Palo Alto, California, USA, 2184–2191.

[17] Hervé Moulin. 2019. Fair Division in the Internet Age. *Annual Review of Economics* 11, 1 (2019), 407–441.

[18] Ariel D. Procaccia and Hervé Moulin. 2016. *Cake Cutting Algorithms.* Cambridge University Press, Cambridge, UK, 311–330.

[19] Erel Segal-Halevi, Shmuel Nitzan, Avinatan Hassidim, and Yonatan Aumann. 2020. Envy-Free Division of Land. *Math. Oper. Res.* 45, 3 (2020), 896–922. https://doi.org/10.1287/moor.2019.1016

[20] William Thomson. 2011. Chapter Twenty-One - Fair Allocation Rules. In *Handbook of Social Choice and Welfare*, Kenneth J. Arrow, Amartya Sen, and Kotaro Suzumura (Eds.). Handbook of Social Choice and Welfare, Vol. 2. Elsevier, Amsterdam, The Netherlands, 393 – 506.