Optimal Payments in Dominant-Strategy Mechanisms for Single-Parameter Domains

Victor Naroditskiy Maria Polukarov Nicholas R. Jennings
School of Electronics and Computer Science,
University of Southampton, United Kingdom
{vn,mp3,nrj}@ecs.soton.ac.uk

Abstract

We study dominant-strategy mechanisms in allocation domains where agents have one-dimensional types and quasi-linear utilities. Taking an allocation function as an input, we present an algorithmic technique for finding optimal payments in a class of mechanism design problems, including utilitarian and egalitarian allocation of homogeneous items with nondecreasing marginal costs. Our results link optimality of payment functions to a geometric condition involving triangulations of polytopes. When this condition is satisfied, we constructively show the existence of an optimal payment function that is piecewise linear in agent types.

1. Introduction

A celebrated positive result in dominant-strategy implementation is the class of Groves mechanism [10] for quasi-linear environments. For unrestricted type spaces, the Groves class describes all deterministic dominant-strategy mechanisms [21]. When types are given by single numbers—in *single-parameter domains*—it characterizes all *efficient* mechanisms. Mechanisms within the Groves class differ from one another in the amounts of payments they collect from the agents. Until recently, most of the attention in the literature was given to a single Groves mechanism called VCG.[1] Our work develops a

[1] Also known as the Vickrey-Clarke-Groves, the pivotal, or the Clarke mechanism.

technique for finding the best mechanism from the Groves class (and a more general class that includes non-efficient mechanisms) for a given objective.

To see the need to optimize over mechanisms in the Groves class, consider allocating free items among a group of participants each having a private value for receiving an item (or, more generally, scenarios with no residual claimant absorbing the surplus or covering the deficit). Applying the standard VCG mechanism in this scenario generates a budget surplus, which must be "burnt" in order to maintain truthfulness. Thus, a natural objective is to choose a Groves mechanism that minimizes the amount burnt, or equivalently maximizes the *social welfare* (i.e., the sum of the agents' utilities). This objective is pursued in [17] and [12], where such a mechanism is independently derived for the allocation of free homogenous items. Another natural objective is to design payments that guarantee each agent a certain *fair* share of the social welfare. To this end, [20] searches the class of Groves mechanisms for the one that optimizes fairness.[2] The same objectives can be considered in a number of important open problems such as allocation of items that have costs (this allows modeling tragedy of the commons scenarios as we discuss in Section 5.2) and public project problems.

In this paper, we propose a general algorithmic method[3] for approaching a class of problems including the ones mentioned above. For previously solved problems (see Section 5.1), our technique provides a common approach (and a unifying geometric interpretation) while existing solutions were derived using different custom-made technique. We also apply our approach to solve some of the open problems from economic theory (see Section 5.2).

Specifically, we consider mechanism design problems in single-parameter domains. All of the examples in this paper come from allocation domains, but we emphasize that the technique is not limited to them: for example, the public project problem (see, e.g., [19] section 9.3.5.5) can also be studied using the technique. We will however, for the ease of exposition, use the terminology of allocation domains by referring to the outcome function as the *allocation function* and saying that the private type of an agent corresponds to his *value* for being "allocated".

For a given allocation function assigning goods to n agents, a dominant-

[2][3] and [4] propose an equivalent mechanism.

[3]We refer to our approach as *algorithmic* as, in general, it requires solving a linear program.

strategy mechanism is defined by a *rebate* function $h : \mathbb{R}^{n-1} \to \mathbb{R}$ that determines the amount of payment to be redistributed back to each agent based on the values of the other agents. Our goal is to apply our algorithmic technique to find an optimal rebate function according to the objective specified by the problem (e.g., welfare-maximization, fairness, or revenue-maximization) and satisfying the provided constraints (e.g., individual rationality, no subsidy, or k-fairness).

Our work takes a geometric view of rebate optimization. The value space can be viewed as an n-dimensional hypercube (or a simplex), while the corresponding domain of rebate functions is a hypercube in $(n-1)$ dimensions. We characterize optimal rebates based on geometric properties of subdivisions of these hypercubes. Given subdivisions satisfying these properties, we consider a *restricted problem* that includes only value profiles that correspond to the extreme points of the subdivision of the n-dimensional cube. The restricted problem can be solved using linear programming. We then show how to obtain an optimal solution to the original, unrestricted problem by interpolating optimal rebates of the restricted problem. By construction, the optimal rebate function is (piecewise) linear, thus proving the existence of optimal linear rebates.

This paper contributes to the literature in the following ways. First, we develop a general approach, which can be used to solve to a class of mechanism design problems. Second, we provide a geometric perspective on optimizing payment functions in dominant-strategy mechanisms: optimal payments can be obtained if one can find a subdivision of n and $n-1$ dimensional polytopes satisfying certain properties. Thus, we reduce the problem of finding optimal payments to the problem of polyhedral subdivisions. While in general, our approach relies on solving a linear program, in some problems it leads to a solution without any computation (see Sections 5.1.2 and 5.2.2).

The paper unfolds as follows. We start with preliminaries in Section 2 and illustrate our approach on a simple example in Section 3. The approach of finding optimal payments based on geometric properties of the space of values and the domain of payment functions is in Section 4. The applicability of our approach is then demonstrated by applying it, first to previously solved problems (Section 5.1) and then to more intricate open problems (Section 5.2). Discussion and directions for future research are presented in Section 6.

2. Model and Preliminaries

We consider anonymous[4] allocation domains where each of the n agents desires one unit of a (homogenous) good, and has an identical single-parameter type space $[0, 1]$. A *value profile* $v \in [0, 1]^n$ represents the agents' values for consuming the good. Without loss of generality for anonymous mechanisms, we assume $v_1 \geq v_2 \geq \ldots \geq v_n$. We denote the space of value profiles by $V = \{v \in \mathbb{R}^n \mid 1 \geq v_1 \geq v_2 \geq \ldots \geq v_n \geq 0\}$. The vector $v_{-i} \in [0, 1]_+^{n-1}$ stands for values of the agents other than i and (v_i, v_{-i}) is equivalent to v. The space of all $(n-1)$-dimensional vectors v_{-i} is the same for each i, and is denoted by $W = \{w \in \mathbb{R}^{n-1} \mid 1 \geq w_1 \geq w_2 \geq \ldots \geq w_{n-1} \geq 0\}$. A *mechanism* is defined by a pair of functions $f : V \to \{0, 1\}^n$ and $t : V \to \mathbb{R}^n$ that determine the allocation and payments for each possible report from the agents regarding their types. Agent i receives a (possibly negative) payment of $t_i(v)$ and is allocated if $f_i(v) = 1$. Utility of agent i is quasi-linear: $u_i(v) = f_i(v)v_i + t_i(v)$.

The following folk theorem describes the class of all dominant-strategy mechanisms in single-parameter domains.[5]

Theorem 1 (see, e.g., [19] p. 229). *A mechanism (f, t) is implementable in dominant strategies if and only if for each agent i:* (i) *$f_i(v)$ is monotone*[6] *in v_i;* (ii) *$t_i(v) = h(v_{-i}) - \tau(v_{-i})$ if $f_i(v) = 1$ (i.e., i is allocated) and $t_i(v) = h(v_{-i})$ otherwise, where $\tau(v_{-i}) = \sup_{v_i \mid f_i(v_i, v_{-i})=0} v_i$ defines the* threshold.[7]

In this work, we take a monotone allocation function as an input (an efficient allocation function means that we are considering Groves mechanisms). The agent's threshold $\tau(v_{-i})$ is determined by the allocation function, and the only remaining degree of freedom is the function $h(v_{-i})$ that adjusts payments to the agents—this is the function that we optimize. In some applications, it is intuitive to view τ as the price for being allocated and h as the rebate distributed back to all agents; henceforth, we refer to h as the *rebate function.*

[4]Allocation and payment do not depend on agent identities.

[5]We emphasize that our focus is on deterministic mechanisms. A more general version of the theorem for randomized mechanisms that are incentive compatible in expectation appears in [18, 2].

[6]Agent i is allocated if and only if his report is above the threshold $\tau(v_{-i})$.

[7]In stating the theorem, we restricted attention to *anonymous* payment functions: payment functions that do not depend on an agent's identity. This is without loss of generality for all applications considered in this paper as we discuss in Section 6.

As mentioned earlier, a mechanism design problem is given by an objective function and constraints. The problem of finding optimal rebates for a given mechanism design problem can be expressed as an optimization problem[8]

$$
\begin{aligned}
&\text{optimize}_{h:W\to\mathbb{R}} \text{ objective value} \quad \text{s.t.} \quad \forall\, v \in V \\
&\quad \text{objective value is achieved} \\
&\quad \text{constraints hold}
\end{aligned}
$$

At the first glance, this problem is hard: optimization is over functions and there is an infinite number of constraints. However, in this paper we propose a technique that makes it possible to tackle such problems effectively. In the next section, we illustrate our approach with a simple example, and then provide formal results in Section 4.

3. Illustrative Example

Our approach exploits the linear structure, which characterizes standard mechanism design problems in quasi-linear domains. Specifically, whenever the allocation of items is fixed, typical constraints (e.g., individual rationality, no subsidy) and objectives (e.g., utility maximization, deficit minimization) are linear in values and payments of the agents. For example, the no-subsidy (or, weak budget balance) constraint requires the *sum* of payments to the agents to be non-positive; the utilitarian objective function maximizes the *sum* of agents' values and payments. Theorem 1 decomposes a payment into threshold and rebate. If the thresholds of allocated agents are linear in values, then these constraints become linear in values and rebates. That is, a constraint at a value profile v can be represented by parameters $\beta \in \mathbb{R}^{n+1}$ and $\beta' \in \mathbb{R}^n$

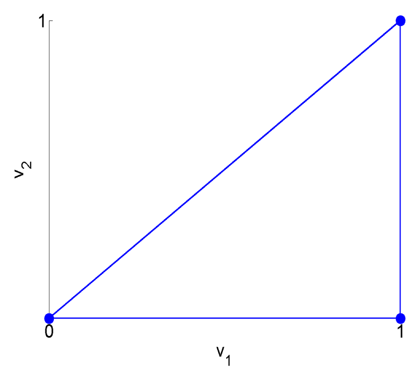
$$\sum_{i=1}^{n} \beta_i v_i + \sum_{i=1}^{n} \beta'_i h(v_{-i}) \geq \beta_{n+1} \tag{1}$$

[8]Some combinations of constraints may be impossible to implement: this is identified by the lack of a feasible solution to the optimization problem. We will see an example of this in Section 5.1.2.

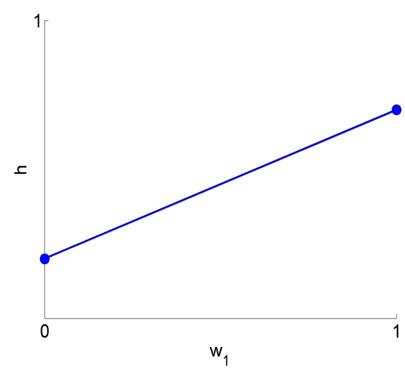
If in addition the rebate function h is itself linear in values, i.e., $h(v_{-i}) = \sum_{j \neq i} \gamma_j v_j$ where $\gamma \in \mathbb{R}^{n-1}$, then, Equation 1 can be fully specified by $\alpha \in \mathbb{R}^{n+1}$

$$\sum_{i=1}^{n} \alpha_i v_i \geq \alpha_{n+1} \tag{2}$$

Our results rely on this linearity and we will define the rebate function h to be (piecewise) linear so that the constraints take the form of Equation 2. This is illustrated in the following 2 examples.



(a) The allocation region.



(b) Optimal rebate function.

Figure 1: Constant allocation throughout the value space.

Consider a simple mechanism design problem where one item needs to be allocated between two agents. The item should be allocated to the agent with the higher value (i.e., the allocation function is efficient), the sum of payments to the agents must be non-positive (i.e., no subsidy), and the objective is to find a rebate function that maximizes the sum of agents' utilities measured as a percentage z of the allocated agent's value.

In this example, the domain of the rebate function $W = \{w \in \mathbb{R} \mid 1 \geq w_1 \geq 0\}$ is the real interval between 0 and 1. The space of values $V = \{v \in \mathbb{R}^2 \mid 1 \geq v_1 \geq v_2 \geq 0\}$ is a *triangle* given by the extreme points $(0, 0)$, $(1, 0)$, and $(1, 1)$ shown in Figure 1(a)—recall that the value vectors are non-decreasing without loss of generality for anonymous mechanisms, and thus only the bottom half of the unit square is relevant. The allocation is fixed for all profiles of values: as the agent with a higher value is called agent 1, agent 1 always gets the item. The threshold for agent 1 is the value of agent 2, $\tau(v_{-1}) = v_2$, which is linear throughout the value space. The optimization problem is

$$\max_{h:W\to\mathbb{R}, z\in\mathbb{R}} z \quad \text{s.t.}$$
$$v_1 + (h(v_2) - v_2) + h(v_1) \geq zv_1 \qquad \forall v \in V$$
$$(h(v_2) - v_2) + h(v_1) \leq 0 \qquad \forall v \in V$$

Notice that the constraints are linear in v and $h(v_{-i})$, and also linear in z (for a constant z, the constraints are of the form given in Equation 1). Thus, we have a linear program. However, the number of constraints and variables (one variable for the objective value z and one rebate variable for each $w \in W$) is infinite. The following observation lets us reduce this linear program to a finite one.

Observation 1. *A linear constraint (see Equation 2) holds at all $v \in p$ of a (convex) polytope $p \subseteq \mathbb{R}^n$ if and only if it holds at the points $v \in$ ExtremePoints(p), where ExtremePoints(p) denotes the set of extreme points of the polytope p.*

The constraints in the linear program above must hold at all profiles $v \in V$. By the observation above, since V is a polytope, we only need to make sure the constraints hold at the extreme points $(0, 0)$, $(1, 0)$ and $(1, 1)$. When we enforce the constraints at these points only, the set of rebate variables becomes finite as well: $h(0)$ and $h(1)$. We refer to the problem that enforces constraints at the extreme points of the value space as a *restricted problem*.

Let $\hat{h}(0)$, $\hat{h}(1)$, and $\hat{z}$ denote the optimal rebates and the objective value for the restricted problem. Since a restricted problem includes only a subset of constraints of the original problem, $\hat{z}$ provides an upper (in the case of maximization) bound on the objective value of the original problem (in problems with no objective function, if the restricted problem has no feasible solution, neither does the original problem).

Our goal now is to define a rebate function $h(w)$ that satisfies the constraints (including the constraint that checks the objective value of $\hat{z}$ is achieved) at all value profiles. Treating $\hat{z}$ as a constant, constraints for all value profiles are of the form given by Equation 1. We would like a rebate function that allows us to represent these constraints as in Equation 2. This can be done by defining $h(w)$ to be the line segment connecting points $(0, \hat{h}(0))$ and $(1, \hat{h}(1))$ (see Figure 1(b)). By Observation 1, such constraints are satisfied throughout the value space if they are satisfied at the extreme value profiles $(0, 0)$, $(1, 0)$ and $(1, 1)$. But by construction of h, the constraints are satisfied at these points. In particular, the objective value of the restricted problem (i.e., an upper bound) is achieved at all value profiles for the solution $h(w)$. Therefore, the upper bound $\hat{z}$ is tight, and the solution $h(w)$ is optimal.[9]

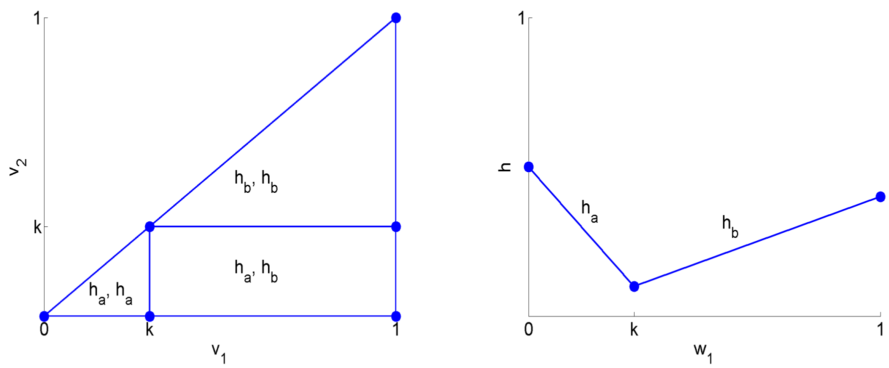
In the last example, the constraints were of the form in Equation 1 throughout the value space and the optimal rebates were linear throughout the domain of the rebate function, which we call the *rebate space*. In general, the value space needs to be subdivided into regions where the constraints take the form of Equation 1. We illustrate this next by modifying the previous example.

Let the item have a cost of $k \in (0, 1)$. Under the efficient allocation agent 2 is never allocated, and agent 1 is allocated only if his value exceeds the cost, in which case his threshold is $\tau(v_2) = \max(k, v_2)$. The constraints of the linear program take the form of Equation 1 (again, treating z as a constant) on the regions where the allocation is fixed and the threshold is linear. There are three such regions shown in Figure 2(a). In more detail, agent 1 is not allocated in the left bottom region, agent 1 is allocated and pays $\tau(v_2) = k$ in the rectangular region, and agent 1 is allocated and pays $\tau(v_2) = v_2$ in the top region. The extreme points of this partition are $(0, 0)$, $(k, 0)$, $(1, 0)$, (k, k), $(1, k)$, and $(1, 1)$ and the corresponding rebates are $\hat{h}(0)$, $\hat{h}(k)$, and $\hat{h}(1)$. Proceeding as we did in the previous example we would have to linearly connect these rebate values, which is impossible when these three values do not fall on a line. In general, as the domain of the rebate function is in $\mathbb{R}^{n-1}$, we can linearly interpolate n rebate values. Thus, we would like to subdivide the rebate space into regions with n (in our example $n = 2$) extreme points (i.e., into simplexes). In this example, such subdivision is natural: into the intervals $(0, k)$ and $(k, 1)$. We can define a linear rebate function on each region: one connecting $\hat{h}(0)$ to $\hat{h}(k)$ and the other connecting $\hat{h}(k)$ to $\hat{h}(1)$ (see Figure 2(b)). We refer to these functions as h_a and h_b:

$$h(w) = \begin{cases} h_a(w) & \text{if } 0 \leq w_1 \leq k \\ h_b(w) & \text{if } k \leq w_1 \leq 1 \end{cases}$$

[9]The approach described above is equivalent to the one proposed by Guo *et al.* [11] in the context of a particular allocation model and objective function. Also see footnote 13.

Now we can apply the argument from the previous example to each of the three value space regions. In more detail, throughout each value region the constrains are of the form in Equation 1. Crucially, the agent's rebate at all points of a region is given by the same linear function (i.e., either h_a or h_b). Thus, the definition of h allows us to represent the constrains in the form of Equation 2 and apply Observation 1 at each region. In Figure 2(a), the rebate function used by first and second agent respectively, is shown in each region. As in the previous example, the no-subsidy constraint holds and the objective value of $\hat{z}$ is achieved for all value profiles guaranteeing optimality of the rebate function h.



(a) Regions where constraints are linear in v.

(b) Optimal rebate function.

Figure 2: Two allocation regions.

In these examples, we partitioned the rebate space into regions each with n (in these examples, $n = 2$) extreme points and defined a linear rebate function on each region by interpolating the optimal rebate values of the restricted problem. The value space is partitioned in such a way that the constraints on each value region are linear (see Equation 2). This can be achieved only if for all points within a value region the agent's rebate is given by the same linear rebate function. A partitioning of the rebate and value spaces satisfying these properties is the main part of our method, which is formalized in the next section.

4. Main Results

Our technique takes a geometric view of the problem. Specifically, we introduce the notion of *consistent partitions* of value and rebate spaces, and then present an algorithm for finding an optimal rebate function given such partitions. Finally, we describe a class of problems that admit consistent partitions.

We start with a series of definitions from polyhedral geometry that we need to define value regions where constraints take a linear form (see Equation 2). Let relative interior *relint*(p) of a polytope p denote the polytope without its facets.

Definition 1. *A set P_X of polytopes is a* subdivision *(equivalently, a* partition*) of the polytope X if the polytopes P_X do not overlap:* $\text{relint}(p) \cap \text{relint}(q) = \varnothing, \ \ \forall\, p, q \in P_X$*, and cover exactly the polytope X:* $\bigcup_{p \in P_X} p = X$.

Definition 2. *A subdivision P_X* refines *a subdivision P'_X if for each $p \in P_X$ there is a $p' \in P'_X \mid p \subseteq p'$.*

We extend this definition to refinements of sets of polytopes that are not necessarily subdivisions.

Definition 3. *A subdivision P_X* refines a polytope *q if for all $p \in P_X$ the intersection with q is either empty or p:* $\text{relint}(p) \cap \text{relint}(q) = \varnothing \vee p \cap q = p$.

Definition 4. *A subdivision P_X* refines a set of polytopes *Q if P_X refines all polytopes $q \in Q$.*

As we discussed in the examples in Section 3, an allocation function induces a subdivision of the value space into regions where the allocation is fixed and the threshold is linear (i.e., the constraints are as in Equation 1). We call this an *initial subdivision* P_V^{init}. Note that for the constraints to be given by the same coefficients α (see Equation 2), they must belong to the same initial subdivision region. Thus, we would like the subdivision P_V of the value space to refine this initial subdivision P_V^{init}.

Building on the definitions above, we describe additional properties of the subdivisions of the value and rebate spaces that guarantee linearity of the constraints on each value space region if the rebate function is linear on each rebate space region. We use the following notation. A d-dimensional polytope p is a finite intersection of halfspaces: $p = \{x \in \mathbb{R}^d \mid Ax \geq b\}$, where $A \in \mathbb{R}^{k \times d}$, $b \in \mathbb{R}^k$, and k is the number of halfspaces. Thus, a pair (A, b) defines the corresponding polytope.

Definition 5. Lifting *of a subdivision* P_W *from* W *to* V *is a set of polytopes in* V

$$\text{lift}(P_W) = \bigcup_{(A,b)\in P_W} \bigcup_{i=1}^{n} (Av_{-i} \geq b) \cap V$$

Each polytope $(A, b) \in P_W$ adds n (possibly overlapping) polytopes to *lift*(P_W). For example, lifting the polytope $k \geq w_1 \geq 0$ yields the polytopes $\{k \geq v_2\} \cap \{1 \geq v_1 \geq v_2 \geq 0\}$ and $k \geq v_1 \geq v_2 \geq 0$ (see Figures 2(a) and 2(b)).

The partition P_W of the rebate space in Figure 2(b) is given by 2 polytopes $k \geq w_1 \geq 0$ and $1 \geq w_1 \geq k$ with a linear rebate function defined on each region (h_a and h_b, respectively). For constraints to be linear throughout a value region $q \in P_V$, the rebate $h(v_{-i})$ of each agent i must be linear for all $v \in q$. This holds if the choice of the rebate function for agent i is the same throughout q (e.g., the rebate of agent 1 is given by h_a throughout the rectangular region in Figure 2(a)). Stating this property formally, we obtain $\forall q \in P_V,\ \forall i \in \{1, \ldots, n\}$ there exists $p \in P_W \mid v_{-i} \in p,\ \forall v \in q$. In words, for a given agent i, and value region q, the vectors v_{-i} must belong to the same rebate region p for all $v \in q$. The next lemma shows that this requirement is equivalent to a geometric property we call *region consistency*.

Definition 6 (Region consistency). *Subdivisions* P_V *and* P_W *are* region-consistent *if* P_V *refines the polytopes* lift(P_W).

To illustrate region consistency, consider the value space subdivision P_V shown in Figure 2(a) and the rebate space subdivision P_W in Figure 2(b). The polytopes defining the partition P_V always fit within a single polytope from *lift*(P_W): indeed, lifting the polytope $k \geq w_1 \geq 0$ yields the polytopes $\{k \geq v_2\} \cap \{1 \geq v_1 \geq v_2 \geq 0\}$ and $\{k \geq v_1\} \cap \{1 \geq v_1 \geq v_2 \geq 0\}$, and lifting the polytope $1 \geq w_1 \geq k$ yields the polytopes $\{1 \geq v_2 \geq k\} \cap \{1 \geq v_1 \geq v_2 \geq 0\}$ and $\{1 \geq v_1\} \cap \{1 \geq v_1 \geq v_2 \geq 0\}$.

Lemma 1. *Let* P_V *and* P_W *be partitions of the value and the rebate spaces, respectively. Then,* $\forall q \in P_V,\ \forall i \in \{1, \ldots, n\}$ *there exists* $p \in P_W \mid v_{-i} \in p,\ \forall v \in q$ *if and only if* P_V *and* P_W *are region-consistent.*

Proof *condition ⇒ region consistency.* Pick an agent i and a polytope $p = (Aw \geq b) \in P_W$. Define $q' = ((Av_{-i} \geq b) \cap V)$. We show that P_V must refine q'. Suppose the contrary: there is a polytope $q \in P_V$ that overlaps with q' and $q \cap q' \neq q$. Pick $v' \in q' \cap q$ and $v \in q \setminus q'$. By construction, $v'_{-i} \in p$ but $v_{-i} \notin p$. This contradicts the property that the rebate for agent i at all points in q must come from the same region. The argument holds for all i and $p \in W$ and the claim follows.

region consistency ⇒ condition. Pick an agent i. Consider the polytopes $p = (Aw \geq b) \in P_W$ and $q' = ((Av_{-i} \geq b) \cap V) \in \mathit{lift}(P_W)$. By refinement, for any polytope $q \in P_V \mid \mathit{relint}(q') \cap \mathit{relint}(q) \neq \varnothing$ it holds that $q' \cap q = q$. But then for such polytope q and agent i, it holds that $\{v_{-i} \mid v \in q\} \subseteq p$. The argument holds for all i and $q' \in \mathit{lift}(P_W)$. The polytopes $\mathit{lift}(P_W)$ cover (with overlap) the polytope V: any point $v \in V$ belongs to the lifted polytopes $\{p \in P_W \mid v_{-i} \in p\}_{i=1}^{n}$. Thus, the argument holds for all $q \in P_V$. □

As in the examples above, we consider the restricted problem where the value space is limited to the extreme points of P_V, denoted by $\hat{P}_V$. The rebates of the restricted problem, can be described as a "projection" of the extreme value profiles. For instance, for $n = 2$, "projecting" the point $(1, 0)$ yields 0 for the projection of the first agent, and 1 for the projection of the second agent. The projection of the extreme point of the value space partition in Figure 2(a) yields 0, k, and 1: i.e., the rebates that appear in Figure 2(b).

Definition 7. *Given a subdivision P_V, the projection of its extreme points $\hat{P}_V$ on W is*

$$\Pi_W(\hat{P}_V) = \bigcup_{v \in \hat{P}_V} \bigcup_{i=1}^{n} v_{-i}$$

When region consistency is satisfied and the rebate function is linear on each rebate region, the constraints are linear on each value region and Observation 1 applies. Thus, our goal is to define a linear rebate function on each rebate region. As in the examples, we want the rebate function to yield the optimal rebates for the restricted problem. A natural case is when the rebates from the restricted problem are the extreme points of the rebate space subdivision. In this case, a rebate function can be defined by linearly interpolating the extreme points (this may be not possible if a region has more than n extreme points as we discuss below). We call this property *vertex consistency.*

Definition 8 (Vertex consistency). *Subdivisions P_V and P_W are* vertex-consistent *if the projection of the extreme points of P_V is the extreme points of P_W*

$$\Pi_W(\hat{P}_V) = \hat{P}_W$$

For instance, the subdivisions shown in Figures 2(a) and 2(b)) are vertex-consistent. Indeed, the set of extreme points of the value space partition is $\hat{P}_V = \{(0,0),(k,0),(1,0),(k,k),(1,k),(1,1)\}$, which projection onto the rebate space gives $\{(0),(k),(1)\} = \hat{P}_W$.

We are going to linearly interpolate the rebate values at the extreme points of each rebate region. One can linearly combine $d+1$ of d-dimensional points, and a linear rebate function (with $n-1$ dimensional domain) is guaranteed to exist if each region has n extreme points. We refer to subdivisions satisfying this property along with the ones described above as *consistent* and state our main result.

Definition 9. *For a given initial partition P_V^{init}, partitions P_V and P_W are* consistent *if:* (i) *P_V refines P_V^{init};* (ii) *P_V and P_W are region- and vertex-consistent; and* (iii) *each polytope in P_W has n extreme points.*

In the above definition, property (i) implies that the allocation is constant on each region of P_V, property (ii) guarantees that within each region of P_V, the rebate of each agent is given by a unique rebate function, and finally, property (iii) allows to define linear rebate functions throughout each region of P_W. Based on this, the next theorem provides a constructive characterization of optimal rebate functions for mechanism design problems that admit consistent partitions.

Theorem 2. *Let P_V and P_W denote consistent subdivisions for a mechanism design problem with an allocation function inducing the initial partition P_V^{init}. Let $\{\hat{h}(w) \mid w \in \hat{P}_W\}$ denote the set of rebates from an optimal solution to the restricted problem, which only considers profiles $\hat{P}_V$. Further, let $\hat{p}$ denote the set of n extreme points of a polytope $p \in P_W$. For each polytope, define a linear rebate function $h_p(w) = \sum_{i=1}^{n-1} a_i^p w_i + b^p$ with coefficients $a^p \in \mathbb{R}^{n-1}, b^p \in \mathbb{R}$ given by a solution to the system of linear equations $\{\hat{h}(w) = \sum_{i=1}^{n-1} a_i^p w_i + b^p \mid w \in \hat{p}\}$. Then, the following rebate function is* optimal *for the mechanism design problem: for $w \in p$, $h(w) = h_p(w)$.*

Proof Property (i) of Definition 9 ensures that the constraints on a value region $q \in P_V$ are of the form given in Equation 1. Further, property (ii) guarantees that if h is linear on P_W, then the constraints can be represented by linear coefficients as shown in Equation 2. By construction (using property (iii) to ensure existence), $h(w)$ is linear on P_W and the constraints hold at the extreme points of each $q \in P_V$. Thus, Observation 1 applies, ensuring $h(v_{-i})$ satisfies the constraints at all points $v \in q$ for each q. As the objective function is represented by a constraint, and the objective value of the restricted problem (i.e., upper bound) is achieved for all $v \in V$, the rebate function $h(w)$ is optimal. □

Theorem 2 shows how to construct an optimal rebate function for any mechanism design problem that admits consistent partitions. Given such partitions P_V and P_W, one just needs to solve the restricted problem and then define a linear rebate function for each rebate region by linearly interpolating optimal rebates at its n extreme points.

Importantly, Theorem 2 provides a constructive proof of the existence of a piecewise linear optimal solution for any mechanism design problem that admits consistent partitions. Now, the question is which problems admit such partitions. More specifically, it is about allocation functions, since the initial partition is determined by the allocation function.

While full characterization remains an avenue for future work, we describe a class of subdivisions of the value space, for which we can find consistent partitions in Theorem 3. Our approach applies to any mechanism design problem with allocation functions[10] that are constant on each region of this subdivision: i.e., we can find a rebate function that is optimal according to the objective function and satisfies the constraints of the given mechanism design problem.

In more detail, the class of subdivisions is parameterized by a set of constants $C = \{c_1, \ldots, c_m\}$. Partitions of V and W obtained by hyperplanes $v_i = c_j \ \forall i, j$ are region- and vertex-consistent (in W space the hyperplanes are $w_i = c_j$). These hyperplanes define a grid. Each cell in the grid is a hyperrectangle. Partitioning each hyperrectangle with $\binom{n}{2}$ (for W space, $\binom{n-1}{2}$) hyperplanes corresponding to the equation of the diagonal from the lower left to the upper right corner of the rectangle obtained by projecting the hyper-

[10]The allocation function still needs to be monotone as required by Theorem 1.

rectangle on each pair of the coordinates, results in consistent subdivisions. This subdivision is defined in Figure 3.

Theorem 3. *The partitions* $P_V = \text{partition}(V, C)$ *and* $P_W = \text{partition}(W, C)$ *are consistent for any allocation function* f *that is constant on each region* $q \in P_V$*.*

Proof We need to show that for any set of constants C, the subdivisions $P_V = \textit{partition}(V, C)$ and $P_W = \textit{partition}(W, C)$ satisfy three properties of Definition 9.

Let C denote the set of m constants between 0 and 1. Consider the grid in $\mathbb{R}^n$ obtained after step 1 of $\textit{partition}(V, C)$ (see Figure 3):

$$v_i = c_j \quad \forall\, i \in \{1, \ldots, n\},\ j \in \{1, \ldots, m\}$$

The grid partitions the value space $\{v \in \mathbb{R}^n \mid 1 \geq v_1 \geq \ldots \geq v_n \geq 0\}$ into hyperrectangles[11]. We call this partition P_V^{grid} and the corresponding partition of the rebate space P_W^{grid}.

Step 2 of the *partition* algorithm, further subdivides each of the hyperrectangles. In fact, each hyperrectangle is subdivided into simplices; i.e., *triangulated*. The triangulation performed in step 2 is a trivial extension of a canonical hypercube triangulation (see, e.g., [8] p.312). A hyperrectangle in $\mathbb{R}^n$ can be triangulated into $n!$ simplices. Each simplex is given by a permutation $\sigma = (\sigma(1), \ldots, \sigma(n))$ of the numbers $1 \ldots n$:

$$0 \leq \frac{v_{\sigma(1)}}{a_{\sigma(1)}} \leq \frac{v_{\sigma(2)}}{a_{\sigma(2)}} \leq \cdots \leq \frac{v_{\sigma(n)}}{a_{\sigma(n)}} \leq 1$$

where $a_i \in \mathbb{R}$ denotes the length of the hyperrectangle in dimension i. Thus, a simplex in the triangulation corresponds to an ordering of (weighted) coordinates. This triangulation is obtained by cutting the hyperrectangle with $\binom{n}{2}$ hyperplanes of the form $\frac{v_i}{a_i} = \frac{v_j}{a_j}$. Each hyperplane corresponds to the diagonal from the lower left to the upper right corner (henceforth, the *main diagonal*) of the rectangle obtained by projecting V on (i, j). Without loss of generality assume, the "lower left" vertex of the hypercube is the origin. The extreme points of a simplex σ are the $n+1$ vertices of the hyperrectangle: v^k for $k \in \{0, 1, \ldots, n\}$, where $v^k_{\sigma(i)} = 0$ for $i \leq k$ and $v^k_{\sigma(i)} = a_{\sigma(i)}$ for $i > k$.

[11] On a boundary $v_i = v_j$ only part of the hyperrectangle is in the polytope.

Property (iii) of Definition 9 follows immediately. Vertex consistency also follows trivially as the extreme points of P_V and P_W are all non-decreasing vectors where each elements takes a value from $C \cup \{0\} \cup \{1\}$. To argue region consistency, we need additional notation.

Definition 10. *For a hyperrectangle $p = (A, b) \in P_W$, a* tunnel *across dimension i in P_V is $T_i(p) = (Av_{-i} \geq b) \cap (0 \leq v_i \leq 1)$.*

In words, a tunnel is a polytope in $\mathbb{R}^n$ obtained by lifting region $p \in P_W$ for agent i. For ease of exposition, we define a tunnel for $0 \leq v_i \leq 1$ instead of $v_i \mid v \in V$ (i.e., $0 \leq v_1 \leq \ldots, v_{i-1} \leq v_i \leq v_{i+1} \leq \ldots \leq v_n \leq 1$).

A tunnel consists of $m+1$ disjoint hyperrectangles $q \in [0,1]^n$ given by the corners $v \in \mathbb{R}^n$ as defined below

$$T_i(p) = \bigcup_{c \in C \cup \{0\}} v \mid (v_i = c,\ v_{-i} = \mathit{corner}(p))$$

A hyperrectangle $q \in P_V$ is given by the intersection of n tunnels $T_i(q_{-i})$ for $1 \leq i \leq n$ implying that the grid subdivisions P_V^{grid} and P_W^{grid} are region-consistent: q refines each of these tunnels, and no other tunnels intersect with q.

It remains to argue that triangulations within each hyperrectangle of P_V and P_W are region-consistent. Consider each tunnel along with the subdivision defined inside of it: a tunnel $T_i(p)$ contains $\binom{n-1}{2}$ diagonal hyperplanes triangulating p. As noted before, the only tunnels that cross $q \in P_V$ are $T_i(q_{-i})$ for $1 \leq i \leq n$. We show that they bring exactly $\binom{n}{2}$ diagonal hyperplanes that triangulate q. Specifically, it can be shown that the intersection of any 3 of these tunnels introduces all triangulating hyperplanes. Diagonal hyperplanes from the other tunnels coincide with the existing ones. In more detail, a tunnel $T_i(q_{-i})$ introduces all of its $\binom{n-1}{2}$ diagonal hyperplanes to q. Consider another tunnel $T_j(q_{-j})$ with $j \neq i$. The diagonals from faces of $q_{-(i,j)}$ were already introduced by $T_i(q_{-i})$. The diagonal hyperplanes from $T_j(q_{-j})$ that are not in $T_i(q_{-i})$, are from the remaining $(n-2)$ comparisons of q_i to $q_{-(i,j)}$. After introducing the hyperplanes from tunnels T_i and T_j, the only pair of coordinates that has not been compared yet is i, j. This is the only other diagonal that the remaining $(n-2)$ tunnels contribute. All the other diagonals coincide with the ones introduced by T_i or T_j. The total number of diagonal hyperplanes is $\binom{n-1}{2}$ from T_i plus $(n-2)$ from T_j plus 1 from the others. The total is exactly the number of pairwise comparisons of

n numbers: $\binom{n}{2}$. Since all of the hyperplanes from $lift(P_W)$ coincide with the hyperplanes defining P_V, consistency of P_V and P_W follows.

The only property of Definition 9 that we have not shown is (i). By a condition of the theorem, f is constant on $q \in P_V$. We only need to argue that the threshold is linear on $q \in P_V$. For a region $q \in P_V$, the threshold for each allocated agent (we do not need to worry about thresholds of unallocated agents as they do not show up in the constraints), is the minimum value he can report to remain in q when other agents report q_{-i}. By construction, this value is given by one of the *hyperplanes* obtained from $lift(p)$ for agent i where $p \mid q_{-i} \in p$. Equation of the hyperplane represents the *linear* threshold. □

Note that for the allocation function that is constant throughout the entire value space, the trivial subdivisions $P_V = V$ and $P_W = W$ are consistent and Theorem 3 applies.[12,13] Two exemplar mechanism design problems with such allocations are the focus of Sections 5.1.1 and 5.1.2.

Any allocation rule that makes decisions based on comparing an agent's value to a constant also admits a consistent partition. Here the set of constants C would include all constants used by the allocation function. We apply the approach to an example mechanism design problem with such allocation function in Section 5.2.

Remark 1. *Any allocation function can be approximated with a "constant-dependent" function. Indeed, the hyperplanes $v_i = c \mid c \in C$ create a grid over the value space V, and any function can be approximated by its piecewise constant components; moreover, this can be done arbitrarily well by taking finer grids.*[14] *Thereby, our technique applies to (an approximation of) any possible mechanism design problem in single-parameter allocation domains.*[15]

---

[12] The value space V is a simplex given by $n+1$ extreme points $(0,0,0\ldots,0)$, $(1,0,0\ldots,0)$, $(1,1,0,\ldots,0)$, $\ldots$, $(1,1,0,\ldots,1)$. Similarly the rebate space W is a simplex given by n extreme points.

[13] For the case of trivial subdivisions, Guo and Conitzer [11] developed the technique of interpolating rebates for the restricted problem, which we generalize in Theorem 2 to arbitrary consistent subdivisions.

[14] We acknowledge that computational feasibility is a concern: the size of the linear program grows exponentially as the grid becomes finer.

[15] We thank Warren Schudy for making this observation.

Algorithm *partition*
Inputs: polytope $X \subset \mathbb{R}^d$, set of constants $C = \{c_1, \ldots, c_m\}$
1. partition X along $x_i = c_j \quad \forall j \in \{1, \ldots, m\},\ i \in \{1, \ldots, d\}$
   /* denote the partition by P_X^{grid} */
2. **for** each hyperrectangle $p \in P_X^{grid}$
      for each pair (i, j) of dimensions $i, j \in \{1, \ldots, d\},\ i \neq j$,
         partition p along $x_i = ax_j + b$
         where $a, b \in \mathbb{R}$ define the diagonal from the lower left
         to the upper right corner of the projection onto the i-j plane

Figure 3: Consistent partitions.

5. Applications

We demonstrate the applicability of our approach on two fundamental allocation models (with free or non-free items) under different objectives. We begin with a previously studied allocation model where free homogeneous items are efficiently allocated to agents with unit demand [17, 12, 20]. In Section 5.1.1, the objective of welfare-maximization is pursued. The objective of fairness is tackled in Section 5.1.2. A more general allocation model where items can be produced for an increasing marginal cost is presented in Section 5.2. The rebate functions provided by our approach for settings with costs are novel. Note that optimal rebates can be obtained for welfare-maximizing as well as fairness-optimizing mechanisms and all of the solutions are derived following the same approach.

5.1. Free Homogeneous Items

5.1.1. Welfare-maximizing Allocation

In this section, we consider welfare-maximizing allocation of free items in single-parameter domains. The problem has been solved analytically in [17, 12]. Nonetheless, we choose to start with this setting as it provides a natural introduction to the following sections.

There are m identical items and n agents with unit demand. The items are (weakly) desirable, so we talk of distribution of “goods”, and there is a rationing problem: each agent claims a unit but all claims cannot be met ($m < n$). Efficiency requires assigning the items to the m agents who value them the most. Agents' valuations for consuming the item, $1 \geq v_1 \geq \ldots \geq v_n \geq 0$, are private information. To ensure truthful reporting (see Theorem 1), payment to each agent i must be given by two functions h and τ as described in Theorem 1. For the efficient allocation function, the price for allocated agents is $\tau(v_{-i}) = v_{m+1} \quad \forall i \leq m$.

The natural constraints that the mechanism must satisfy are described next. The mechanism must be subsidy-free; that is, the sum of payments is non-positive $\sum_{i=1}^{n} t_i(v) \leq 0$. Utility of each agent after participating in the mechanism should be non-negative: $u_i(v) \geq 0$, to provide incentives for agents to participate in the mechanism. This constraint is known as individual rationality or voluntary participation.

Under no subsidy, the goal is to minimize the budget surplus (the amount burnt) of the mechanism or, equivalently, to maximize the welfare it achieves. Note, however, that the absolute welfare realized by the mechanism is not an appropriate measure of its performance since it does not show how far this value is from the first-best solution; i.e., the maximal welfare one could achieve if the agents' values were known. In order to have an index that is unit-free (i.e., homogenous of degree zero), it is natural to use a ratio. Finally, since the agents' values are not known and no prior is available, we consider a worst-case index. Therefore, following [17, 12], we use the following *welfare ratio* to evaluate the performance of a dominant-strategy mechanism:

$$\min_{v \in V} \frac{\sum_{i=1}^{m} v_i - m v_{m+1} + \sum_{i=1}^{n} h(v_{-i})}{\sum_{i=1}^{m} v_i}$$

The numerator is the social welfare achieved by the mechanism, while the denominator is the value of the efficient allocation, which is the highest possible social welfare for subsidy-free mechanisms. Finding a mechanism whose welfare ratio (henceforth, the *ratio*) is z means that a proportion z of the efficient allocation value (i.e., maximum social welfare) is achieved, *independently* of what the agents' values are.

Below we formally state the problem of finding the welfare-maximizing allocation mechanism. In order to remove minimization over all value profiles from the objective function, we introduce additional variable z and require it not to exceed the welfare ratio realized for each profile:

$$\max_{z\in\mathbb{R},\ h:\mathbb{R}^{n-1}\to\mathbb{R}} z \quad \text{s.t.} \quad \forall\, v \in V \tag{3}$$

$$\sum_{i=1}^{m} v_i - m v_{m+1} + \sum_{i=1}^{n} h(v_{-i}) \geq z \sum_{i=1}^{m} v_i \tag{4}$$

$$h(v_{-i}) \geq 0 \quad \forall i \tag{5}$$

$$\sum_{i=1}^{n} h(v_{-i}) \leq m v_{m+1} \tag{6}$$

Equation 4 ensures that the ratio is achieved for all value profiles. Equation 5 guarantees that the utility of each agent is non-negative.[16] Finally, the no-subsidy constraint is enforced in (6): the total amount redistributed cannot exceed the total price paid by the allocated agents.

To apply our approach, note that the allocation is fixed and the threshold of allocated agents (i.e., v_{m+1}) is linear on the entire value space. Therefore, the constraints (4)-(6) are of the form described by Equation 1. The empty partitions $P_V = V$ and $P_W = W$ are consistent and thus we can find an optimal rebate function by solving the restricted problem and then linearly interpolating the rebates throughout the only W space region (see Theorem 2). Restricting the problem to the natural set of extreme points of the value space $V \cap \{0,1\}^n$, we get the system of $2(n+1)$ linear inequalities (constraints (4) and (6) are enforced for each of $n+1$ extreme points; constraint (5) follows from non-negativity of the variables) with exactly n nonnegative variables h_x, $x = 0, 1, \ldots, n-1$, that correspond to rebates $h(w^x)$ where w^x is an $(n-1)$-dimensional non-decreasing binary vector with x ones: $w^x \in W \cap \{0,1\}^{n-1}$, $\sum_{i=1}^{n-1} w_i^x = x$.

While in order to obtain an optimal mechanism for a problem with n agents and m items we need to solve a linear program, Moulin [17] and Guo

[16]One may notice that the individual rationality constraint for allocated agents is $v_i - v_{m+1} + h(v_{-i}) \geq 0$, which seems weaker than Equation 5. In fact, since this seemingly weaker constraint must hold for all $v \in V$ and $i \in N$, it implies $h(v_{-i}) \geq 0 \quad \forall\, i \leq m$ and so can be equivalently replaced by the latter. Indeed, fix any $v \in V$ and $i \in N$, and consider a value profile $v' \in V$ such that $v'_j = (v_{-i})_j$ for all $j \leq m-1$, $v'_m = v'_{m+1} = (v_{-i})_m$ and $v'_{m+k} = (v_{-i})_{m+k-1}$ for $k = 1, \ldots, n-m$. From the individual rationality constraint for allocated agent m at profile v', i.e. $v'_m - v'_{m+1} + h(v'_{-m}) \geq 0$, we get $h(v_{-i}) \geq 0 \quad \forall\, i \leq m$. This has been also noted in [17].

and Conitzer [12] derived the optimal solution analytically for any n and m. Clearly, an analytical solution is preferable; however, it required involved proofs at each step. In contrast, some of the analytical results derived in those papers, follow immediately when approaching the problem using our method. For instance, the existence of an optimal linear rebate function is directly implied by Theorem 2. The uniqueness of this solution, which had to be proven in [17, 12], also follows immediately after checking that the restricted problem has a unique optimal solution.[17] Furthermore, in some problems, our approach yields a complete closed-form mechanism without any computation. We give an example of this in the next section.

5.1.2. Fair Allocation

In this section we apply our approach to fair task imposition. We start by re-deriving the results for a single task by Porter *et al.* [20]. We then proceed to show that an optimal payment function for a setting with multiple tasks, for which no closed-form solution has been previously derived, can be easily obtained using our method. Importantly, our approach provides a new perspective on this problem: instead of using counterexamples to show impossibility theorems and proving optimality and uniqueness of a constructed rebate function, in our case, impossibility, optimality and uniqueness results, all are obtained by solving a simple system of equations. Specifically, impossibility results are obtained when a corresponding system has no feasible solution, while optimality and uniqueness are implied by the uniqueness of a solution to the system of equations.

Suppose one wishes to fairly assign tasks to agents each having private information about the level of effort required from the agent to perform the task. The net payment to the agents is non-positive (that is, there is no budget deficit), and the assignment is efficient (agents with lowest levels of required effort get the tasks). Since the agents are *obligated* to provide the service and make payments to the center, the standard constraint of individual rationality is replaced with *k-fairness*, which requires the agents' disutilities to be "as low as possible"—namely, no greater than $\frac{m}{n}$ of the k^{th} lowest level of effort. Finally, truthfulness has to be satisfied as both efficiency and k-fairness depend on the true agents' types.

[17] Uniqueness can be checked by generating all solutions to the linear program, which can be done with solvers such as Cplex.

For convenience, we consider this problem in the context of "distribution of goods" rather than "imposition of bads"—this can be done equivalently by viewing tasks as items with negative values.[18] We choose this interpretation as it is consistent with the previous allocation example.[19] Notice that this model is exactly the same as the one in Section 5.1.1. The only difference is that in this setting there is no objective function and the k-fairness constraint strengthens the individual rationality constraint.

Stated mathematically, a k-fair mechanism exists if and only if there exists a rebate function $h : W \to \mathbb{R}$ satisfying the constraints (7)-(8) below for each possible value profile v. These constraints are linear in v and h:

$$h(v_{-i}) \geq \frac{mv_k}{n} \quad \forall i \tag{7}$$

$$\sum_{i=1}^{n} h(v_{-i}) \leq mv_{m+1} \tag{8}$$

We now show that the fair imposition problem can be easily solved using our method. We first re-derive the results by Porter *et al.* [20]—the impossibility theorem for $1, 2$-fairness and a 3-fair mechanism for a single task ($m = 1$), and then generalize these results to the case with multiple tasks: we show impossibility for $k \leq m+1$ and find an $(m+2)$-fair mechanism. Although, as proposed in [20], this mechanism could be achieved by applying sequentially the Porter's protocol for a single task, our approach provides a "one-shot" solution in a closed form. Moreover, our technique implies the uniqueness of a linear $(m + 2)$-fair mechanism.

As in the previous example, the allocation is fixed on the whole value space, and the trivial partitions $P_V = V$ and $P_W = W$ are consistent. The extreme points of V define a system of inequalities with n variables h_x, $x = 0, 1, \ldots, n-1$, that correspond to rebates $h(w^x)$ where w^x is an $(n-1)$-dimensional binary vector with x leading ones. By Theorem 2, we only need to find a feasible solution to this system of inequalities and then linearly extend the rebate values obtained from this solution: this way we, in particular, find an $(m + 2)$-fair mechanism. The impossibility result for $k \leq m+1$ follows from the fact that the above system has no feasible solution. These derivations are shown next.

[18] See discussion in [20].

[19] In the extended setting considered in the next section, we could equivalently represent items with costs by tasks with benefits they provide to the society when completed.

Single task

Let $k \leq 2$ and show there is no feasible solution to the system (7)-(8), even when restricted to the extreme points of V. From (7) for $v^0 = (0, 0, \ldots, 0)$ we have $h_0 \geq \frac{v_k^0}{n} = 0$; coupled with (8) for $v^1 = (1, 0, \ldots, 0)$: $h_0 + (n-1)h_1 \leq mv_2^1 = 0$, this implies that $h_1 \leq 0$. This contradicts the k-fairness constraint (7) for $i = 2$ and $v^2 = (1, 1, 0, \ldots, 0)$: $h_1 \geq \frac{v_k^1}{n} = \frac{1}{n} > 0$.

However, for $k = 3$ the system does have a feasible solution. First, we solve the system for vectors in $\hat{P}_V$. From (7) and (8) for $v^n = (1, 1, \ldots, 1)$ we get $h_n = \frac{1}{n}$. Then, (7) and (8) for $v^{n-1} = (1, 1, \ldots, 1, 0)$ we have $h_{n-1} \geq \frac{1}{n}$ and $h_n + (n-1)h_{n-1} = \frac{1}{n} + (n-1)h_{n-1} \leq 1$, implying $h_{n-1} = \frac{1}{n}$. Proceeding this way, for any $x \geq 2$ we obtain $h_x = \frac{1}{n}$. Finally, from (7) and (8) for $v^0 = (0, 0, \ldots, 0)$ and $v^1 = (1, 0, \ldots, 0)$ we have $h_0 = h_1 = 0$. Note that this solution is unique.

Now, for any $w \in W$ its corresponding rebate is defined by the equation $h(w) = \sum_{i=1}^{n-1} a_i w_i + b$, where the coefficients are determined by the extreme points of W:

$$
\begin{aligned}
&h_0 = a_1 0 + a_2 0 + \ldots + a_{n-1} 0 + b \qquad \Rightarrow b = 0 \\
&h_1 = a_1 1 + a_2 0 + \ldots + a_{n-1} 0 + b \qquad \Rightarrow a_1 = 0 \\
&h_2 = a_1 1 + a_2 1 + a_3 0 + \ldots + a_{n-1} 0 + b \qquad \Rightarrow a_2 = h_2 = \frac{1}{n} \\
&h_x = a_1 1 + a_2 1 + \ldots + a_x 1 + a_{x+1} 0 + \ldots + a_{n-1} 0 + b \\
&\qquad \qquad \Rightarrow a_x = 0, \forall 3 \geq x \geq n-1
\end{aligned}
$$

Thus, for any $w \in W$, $h(w) = \frac{1}{n} w_2$, which coincides with the mechanism by Porter *et al.* [20].

We have demonstrated that a 3-fair mechanism can be obtained by solving simple systems of linear equations; moreover, our technique implies that such a linear mechanism is unique. Next, we show that a closed-form solution can also be found for settings with multiple tasks.

Multiple tasks

The following proposition generalizes the results to the case with $m \geq 1$.

Proposition 1. *There is no efficient dominant-strategy mechanism that satisfies no subsidy and k-fairness for $k \leq m + 1$. There exists a unique efficient linear $(m + 2)$-fair such mechanism given by: for $i \leq m$, $f_i(v) = 1$, $t_i(v) = h(v_{-i}) - v_{m+1}$, and for $i > m$, $f_i(v) = 0$, $t_i(v) = h(v_{-i})$, where $h(w) = \frac{m}{n} w_{m+1}$.*

Proof Let $v^x \in \hat{P}_V$ denote an extreme point of V with x ones followed by zeros, and assume $k \leq m+1$. From (7) for v^{m-1} and (8) for v^m it follows that $h_m \leq 0$. However, from (7) for v^{m+1}, $i = m+1$, we have $h_m \geq \frac{m}{n} > 0$, a contradiction.

Now let $k = m+2$. Solving the system (7)-(8) for vectors in $\hat{P}_V$, we obtain the following unique solution: $h_x = 0$ for $x = 0, \ldots, m$, and $h_x = \frac{m}{n}$ for $x = m+1, \ldots, n$. Now, solve $\{h(w) = \sum_{i=1}^{n-1} a_i w_i + b \,|\, w \in \hat{P}_W\}$ and get $a_{m+1} = \frac{m}{n}$, $a_x = b = 0$, where $x = 1, \ldots, m, m+2, \ldots, n$. Thus, we have $h(w) = \frac{m}{n} w_{m+1}$. □

5.2. Allocation with Costs

In this section, we apply our technique to solve a number of open mechanism design problems. Specifically, we consider scenarios where items are not free but have (increasing marginal) costs $c_1 \leq c_2 \leq \ldots \leq c_n$. The goal, as before, may be either to maximize the social welfare or to achieve k-fairness.

The allocation problem with increasing marginal costs is a simple and fundamental example of the *tragedy of the commons* [15], in which multiple participants, acting independently to optimize their own objectives, will ultimately deplete a shared limited resource even when it is clear that it is not in anyone's long-term interest for this to happen. Increasing marginal costs model decreasing returns to every agent as the number of allocated items increases. For instance, consider membership in a free gym. As the gym becomes more crowded, the utility each member derives from exercising there decreases. Membership in the gym corresponds to an item in our model. Cost of item i represents the marginal disutility of the members for sharing the gym with another person.

The question of allocating homogeneous items with costs was previously considered in [6, 16], although for a different purpose—to compare "random priority" and "average cost" mechanisms. We observe that Theorem 2 holds for this model where the set of constants C coincides with the set of marginal costs: the consistent partitions are given by $P_V = \mathit{partition}(V, C)$ and $P_W = \mathit{partition}(W, C)$. Given this, we provide the first algorithm for computing optimal payment functions for the welfare-maximizing allocation, and an analytical solution for a k-fair allocation.

In contrast to the case with free items, the number of allocated agents is *not* fixed but depends on c and v: the efficient allocation function does not assign the item to an agent whose value for the item is lower than its cost. Formally, an efficient mechanism in this setting will maximize the total value of the allocated agents minus the total cost; the number of items allocated this way is $m(v,c) = \max_i(i \mid v_i \geq c_i)$ and the value of the efficient allocation is $\sum_{i=1}^{m(v,c)}(v_i - c_i)$. Let $m'(w,c) = \max_i(i \mid w_i \geq c_i)$ denote the number of items allocated efficiently among $n-1$ agents with types $w = v_{-i}$ for some i. When clear from the context we drop the arguments from $m(v,c)$ and $m'(w,c)$ and talk about m and m'. The value of the threshold for the allocated agents under efficient allocation (see Theorem 1) is $\tau(v_{-i}) = \min((v_{-i})_{m'}, c_{m'+1})$, which for the allocated agents $i \leq m$ becomes $\tau(v_{-i}) = \max(v_{m+1}, c_m)$. Finally, we assume that at least one, but no more than $n-1$, items are allocated: i.e., $c_1 < v_1$ and $c_n > 1$.

5.2.1. Welfare-Maximizing Allocation

We now formulate the welfare-maximizing allocation problem in this domain. First, we extend the definition of the welfare ratio as follows:

$$\min_{v \in V} \frac{\sum_{i=1}^{m(v,c)} v_i - m(v,c)\max(v_{m+1}, c_{m(v,c)}) + \sum_{i=1}^{n} h(v_{-i})}{\sum_{i=1}^{m(v,c)}(v_i - c_i)}$$

Note that we fix the cost vector c and consider the worst ratio over all possible value profiles: we do not take the minimum over costs as that would obviously result in a zero ratio—when the first $n-1$ costs are the same, the ratio is zero as in the case with $n-1$ free items. The welfare-maximizing allocation problem is then defined by the following optimization program:

$$\max_{z \in \mathbb{R},\ h: W \to \mathbb{R}} z \quad \text{s.t.} \quad \forall\, v \in V \tag{9}$$

$$m = \text{argmax}_i(v_i \geq c_i) \tag{10}$$

$$\sum_{i=1}^{m} v_i - m\max(v_{m+1}, c_m) + \sum_{i=1}^{n} h(v_{-i}) \geq z \sum_{i=1}^{m}(v_i - c_i) \tag{11}$$

$$h(v_{-i}) \geq 0 \quad \forall i \tag{12}$$

$$\sum_{i=1}^{n} h(v_{-i}) \leq m\max(v_{m+1}, c_m) - \sum_{i=1}^{m} c_i \tag{13}$$

Equation 10 computes the number of allocated items for the current value profile. As before, (11) ensures that the ratio is achieved for all value profiles. Constraint (12) guarantees that the utility of each agent is non-negative (see footnote 16). Finally, the no-subsidy constraint is enforced in (13): the amount redistributed cannot exceed the total price paid less the cost of the allocated items.

The efficient allocation in this model satisfies Theorem 2 (with $P_V = \mathit{partition}(V, C)$ and $P_W = \mathit{partition}(W, C)$). Therefore, a *piecewise linear* welfare-maximizing mechanism can be obtained by solving (9)-(13) for the value profiles $\hat{P}_V$, and linearly interpolating the rebates at these—extreme—points across each region in P_W.

5.2.2. Fair Allocation

The notion of k-fairness was designed to guarantee that every agent gets a (maximal) share of the social welfare. Social welfare in the model with free items is a function of the values of the allocated agents, and k-fairness measures the share of each agent relative to the value of kth agent. In the model with costs, however, social welfare is reduced by the cost of the allocated items. Accordingly, a fair share now must be a function of this cost. To this end, we naturally generalize the k-fairness metric by subtracting the cost $\sum_{i=1}^{m} c_i$. Thus, for the allocation to be k-fair, the following inequalities must hold for each value profile $v \in V$:

$$m = \mathrm{argmax}_i (v_i \geq c_i) \tag{14}$$

$$h(v_{-i}) \geq \frac{1}{n}\left(m v_k - \sum_{i=1}^{m} c_i \right) \quad \forall i \tag{15}$$

$$\sum_{i=1}^{n} h(v_{-i}) \leq m \max(v_{m+1}, c_m) - \sum_{i=1}^{m} c_i \tag{16}$$

The generalized notion of k-fairness coincides with the regular k-fairness when m items have zero cost and no other items are available. Thus, the impossibility results for $k < m + 2$ still apply (when allocating items with costs, the number of allocated items is not fixed but depends on the value v and cost c profiles).

Solving the restricted problem (i.e., Equations 14-16 for the extreme points of the partition $P_V = \mathit{partition}(V, C)$) for a few different cost profiles, we derive a $(m + 2)$-fair mechanism stated in the theorem below.

Theorem 4. *The rebate function*

$$h(w) = \frac{1}{n}\left(m' \max(w_{m'+1}, c_{m'}) - \sum_{i=1}^{m'} c_i \right) \tag{17}$$

together with the efficient allocation define a $(m + 2)$-fair mechanism.

Proof We need to show that Equations 15-16 hold for $k = m + 2$ and the rebate function from Equation 17:

$$\frac{1}{n}\left(m' \max((v_{-i})_{m'+1}, c_{m'}) - \sum_{i=1}^{m'} c_i\right) \geq \frac{1}{n}\left(m v_{m+2} - \sum_{i=1}^{m} c_i\right) \quad \forall i \tag{18}$$

$$\sum_{i=1}^{n} \frac{1}{n}\left(m' \max((v_{-i})_{m'+1}, c_{m'}) - \sum_{i=1}^{m'} c_i\right) \leq m \max(v_{m+1}, c_m) - \sum_{i=1}^{m} c_i \tag{19}$$

We prove separately for the two following cases: the number of items m' allocated when agent i is missing can be either m or $m - 1$. Formally,

$$m'(v_{-i}) = \begin{cases} m(v_{-i}) & \text{if } (v_{-i})_m \geq c_m \\ m(v_{-i}) - 1 & \text{if } (v_{-i})_m \leq c_m \end{cases}$$

When $(v_{-i})_m \geq c_m$, the rebate $h(v_{-i})$ becomes

$$\frac{1}{n}\left(m \max((v_{-i})_{m+1}, c_m) - \sum_{i=1}^{m} c_i\right) \geq$$
$$\frac{1}{n}\left(m \max(v_{m+2}, c_m) - \sum_{i=1}^{m} c_i\right) \geq$$
$$\frac{1}{n}\left(m v_{m+2} - \sum_{i=1}^{m} c_i\right)$$

The first inequality follows as $(v_{-i})_{m+1}$ is v_{m+2} for $i \leq m + 1$ and $v_{m+1} \geq v_{m+2}$ for $i > m + 1$. Thus, Equation 15 is satisfied. Next we show that the no-subsidy Equation 16 holds as well.

$$\sum_{i=1}^{n} \frac{1}{n}\left(m \max((v_{-i})_{m+1}, c_m) - \sum_{i=1}^{m} c_i\right) \leq$$
$$\sum_{i=1}^{n} \frac{1}{n}\left(m \max(v_{m+1}, c_m) - \sum_{i=1}^{m} c_i\right) =$$
$$m \max(v_{m+1}, c_m) - \sum_{i=1}^{m} c_i$$

The first inequality holds as $(v_{-i})_{m+1}$ is $v_{m+2} \leq v_{m+1}$ for $i \leq m + 1$ and v_{m+1} for $i > m + 1$. Now, we turn to the second case $(v_{-i})_m \leq c_m$ with $m' = m - 1$. The rebate in this case is

$$\frac{1}{n}\left((m-1)\max((v_{-i})_m, c_{m-1}) - \sum_{i=1}^{m-1} c_i\right) \geq$$
$$\frac{1}{n}\left((m-1)\max((v_{-i})_m, c_{m-1}) - \sum_{i=1}^{m-1} c_i + ((v_{-i})_m - c_m)\right) \geq$$
$$\frac{1}{n}\left((m-1)\max(v_{m+1}, c_{m-1}) - \sum_{i=1}^{m-1} c_i + (v_{m+1} - c_m)\right) \geq$$
$$\frac{1}{n}\left(mv_{m+2} - \sum_{i=1}^{m} c_i\right)$$

We used $(v_{-i})_m \leq c_m$ to obtain the first inequality, and $(v_{-i})_m \geq v_{m+1}$ to get the second. Finally, we deal with Equation 16:

$$\sum_{i=1}^{n}\frac{1}{n}\left((m-1)\max((v_{-i})_m, c_{m-1}) - \sum_{i=1}^{m-1} c_i\right) \leq$$
$$\sum_{i=1}^{n}\frac{1}{n}\left((m-1)c_m - \sum_{i=1}^{m-1} c_i\right) =$$
$$\sum_{i=1}^{n}\frac{1}{n}\left(mc_m - \sum_{i=1}^{m} c_i\right) \leq$$
$$m\max(v_{m+1}, c_m) - \sum_{i=1}^{m} c_i$$

The first inequality follows from $(v_{-i})_m \leq c_m$ and $c_m \geq c_{m-1}$. □

Interestingly, the optimal rebate function is linear on the subdivision of W space, which is the same as the initial subdivision of the value space for $n-1$ agents. This subdivision has only $2n$ regions: there are n possible allocations (no agents is allocated, agent 1 is allocated, ..., all n-1 agents are allocated). Within an allocation region with m' allocated agents, there are two regions on which threshold is linear: on one the threshold is given by $w_{m'+1}$, on the other by $c_{m'}$.

6. Discussion and Future Work

In this paper we link optimality of payments to geometric properties of n and $(n-1)$-dimensional polytopes that we call consistent subdivisions. Using this characterization, we establish that piecewise linear payment functions are optimal, and reduce the problem of finding them to solving a linear program. Mechanism design problems that have no objective function but seek payments satisfying a combination of constraints (see Sections 5.1.2 and 5.2.2 for examples) are therefore reduced to solving a system of linear inequalities. These reductions lead to immediate solutions of mechanism design problems that are otherwise hard to tackle. Given this, our work can be viewed as an instance of *Automated Mechanism Design* [5, 13] that sets forth the objective of taking a mechanism design problem as an input, and outputting an optimal mechanism.

Even though our technique is algorithmic, we are sometimes able to derive "complete" analytical solutions. In more detail, optimal payments in mechanism design problems depend on the number of agents, and in allocation scenarios—on the number of items available (and if applicable, their costs). Given a configuration of these parameters (e.g., 2 free items and 5 agents) as input to our algorithm, we find optimal (e.g., maximizing social welfare under efficient allocation) payment functions. To avoid computing optimal payment functions for each configuration, it is desirable to obtain analytical solutions parameterized by configuration parameters. Such complete analytical solutions can be obtained using our approach by finding optimal payments functions for a few configurations and discerning how the optimal functions change with the configuration parameters. For example, the rebate functions in Sections 5.1.2 and 5.2.2 were derived this way.

Complete analytical solutions are desirable also because linear programs become computationally intractable as the values of configuration parameters increase. We note however, that our focus is not on computational efficiency, but rather is on the optimality of solutions. Indeed, consistent subdivisions may have an exponential number of regions in the number of agents n and the number of cost constants. In the case of welfare-maximizing payment functions for allocation with costs, we empirically observed for small values of n that almost all of the regions defined by consistent subdivisions are required to find an optimal payment function. However, in the case of k-fair mechanisms, most of these regions have the same optimal rebate function, and thus can be merged. In fact, an optimal payment function can be specified on a subdivision into just $2n$ regions (see Equation 17).

In the rest of this section, we comment on the applicability of our approach and directions for future work. Perhaps the most significant limitation is the reliance of the approach on finding consistent partitions. To this end, Theorem 3 provides a partial characterization of allocation functions that admit consistent partitions.

However, at this stage, a complete characterization remains an open question.[20] To remedy this limitation, in a working paper [9], we are building on the techniques developed here to find heuristic solutions when consistent partitions cannot be identified. In that paper, we also extend the concept of consistent partitions to multi-parameter domains.

Our technique is designed for deterministic mechanisms: i.e., a *deterministic* allocation function must be provided as an input. Considering randomized mechanisms and, more generally, optimizing over (potentially randomized) allocation function and payment function simultaneously is an open direction for future research. To date, the potential for advances in this direction has been demonstrated by showing that non-efficient allocation functions lead to much better mechanisms in certain cases [11, 7, 14]. However, these results only provide heuristic solutions.

Finally, we discuss the restriction to anonymous rebate functions. Apt *et al.* show in [1] how to convert n agent-specific rebate functions into a single rebate function used by all agents (i.e., into an anonymous rebate function). This anonymous function satisfies the same constraints as the agent-specific functions (the authors provide a proof for the no subsidy constraint, but the same logic applies to the ratio constraint; individual rationality/k-fairness is trivial). Thus, the restriction to anonymous rebate functions is innocuous in all of the applications considered in this paper.

[1] K. R. Apt, V. Conitzer, M. Guo, and E. Markakis. Welfare undominated groves mechanisms. In *WINE*, pages 426–437, 2008.

[2] A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *FOCS*, pages 482–491. IEEE, 2001.

[3] M.J. Bailey. The demand revealing process: To distribute the surplus. *Public Choice*, 91(2):107–26, April 1997.

[4] R. Cavallo. Optimal decision-making with minimal waste: Strategyproof redistribution of vcg payments. In *AAMAS*, Hakodate, Japan, 2006.

[5] V. Conitzer and T. Sandholm. Complexity of mechanism design. In *UAI*, pages 103–110, 2002.

[20] For instance, an open question is identifying consistent partitions for the public project model where the initial subdivision is specified by two polytopes obtained by cutting V with the hyperplane $\sum_i v_i = costofproject$.

[6] H. Crès, H. Moulin, and HEC Groupe. Commons with increasing marginal costs: random priority versus average cost. *Int. econ. review*, 44:1097–1115, 2003.

[7] G. de Clippel, V. Naroditskiy, and A. Greenwald. Destroy to save. In *EC*, 2009.

[8] J.A. De Loera, J. Rambau, and F. Santos. *Triangulations*. Springer, 2010.

[9] L. Dufton, V. Naroditskiy, M. Polukarov, and N. R. Jennings. Approximation of payments in dominant-strategy mechanisms for multi-parameter domains. Working Paper, 2012.

[10] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[11] M. Guo and V. Conitzer. Better redistribution with inefficient allocation in multi-unit auctions with unit demand. In *EC*, pages 210–219, 2008.

[12] M. Guo and V. Conitzer. Worst-case optimal redistribution of vcg payments in multi-unit auctions. *Games and Economic Behavior*, 67(1):69 − 98, 2009.

[13] M. Guo and V. Conitzer. Computationally feasible automated mechanism design: General approach and case studies. In *AAAI*, pages 1676–1679, 2010.

[14] M. Guo, V. Naroditskiy, V. Conitzer, A. Greenwald, and N. R. Jennings. Budget-balanced and nearly efficient randomized mechanisms: Public goods and beyond. In *WINE*, December 2011.

[15] G. Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1248, December 1968.

[16] R. Juarez. The worst absolute surplus loss in the problem of commons: random priority versus average cost. *Economic Theory*, 34(1):69–84, 2008.

[17] H. Moulin. Almost budget-balanced vcg mechanisms to assign multiple objects. *Journal of Economic Theory*, 144(1):96–119, 2009.

[18] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[19] N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.

[20] R. Porter, Y. Shoham, and M. Tennenholtz. Fair imposition. *Journal of Economic Theory*, 118(2):209 – 228, 2004.

[21] K. Roberts. The characterization of implementable social choice rules. *In Jean-Jacques Laffont, editor, Aggregation and Revelation of Preferences*, 1979.