Dominant Resource Fairness with Meta-Types

Steven Yin[*] , **Shatian Wang** , **Lingyi Zhang** and **Christian Kroer**
IEOR Department, Columbia University
{sy2737, sw3219, lz2573, christian.kroer}@columbia.edu

Abstract

Inspired by the recent COVID-19 pandemic, we study a generalization of the multi-resource allocation problem with heterogeneous demands and Leontief utilities. Unlike existing settings, we allow each agent to specify requirements to only accept allocations from a subset of the total supply for each resource. These requirements can take form in location constraints (e.g. A hospital can only accept volunteers who live nearby due to commute limitations). This can also model a type of substitution effect where some agents need 1 unit of resource A *or* B, both belonging to the same meta-type. But some agents specifically want A, and others specifically want B. We propose a new mechanism called Dominant Resource Fairness with Meta Types which determines the allocations by solving a small number of linear programs. The proposed method satisfies Pareto optimality, envy-freeness, strategy-proofness, and a notion of sharing incentive for our setting. To the best of our knowledge, we are the first to study this problem formulation, which improved upon existing work by capturing more constraints that often arise in real life situations. Finally, we show numerically that our method scales better to large problems than alternative approaches.

1 Introduction

The recent COVID-19 pandemic has brought forward many problems that are particularly relevant to the operations research and computer science communities. Among them, an often overlooked problem is the effective and fair allocation of resources, such as volunteer medical workers, ventilators, and emergency field hospital beds.

There are several key challenges to the medical resource allocation problem in the face of an infectious disease outbreak. First, utilities from different types of resources are not additive nor linear. For example, when there are enough nurses but not enough doctors, the marginal utility of having one additional nurse on staff is very low. Second, not all resources are accessible to all hospitals (referred to henceforth as *accessibility constraints*). For instance, the home location of each volunteer medical worker determines where she can commute to work; thus, she can only be assigned to hospitals within her commutable radius. Third, hospitals have different capacities (big medical centers versus small hospitals) and are in different stress levels (hospitals in an epicenter versus the ones in rural areas with few cases), so they should naturally be prioritized differently.

Another setting that has the above characteristics is the compute resource sharing problem with sub-types. For example, suppose a compute server has several compute nodes, and there are different types of GPU/CPU on the various nodes (e.g. NVIDIA vs. AMD GPU, Intel vs. AMD CPU). Some users look for specific hardware configurations (e.g. accept only Intel CPU) while others might be less selective (e.g. accept all CPU brands).

In this paper, we propose a new market mechanism that tackles the three challenges outlined above and achieves desirable properties including Pareto optimality, envy-freeness, strategy-proofness, and sharing incentive. In our numerical experiments, we demonstrate that compared to the Maximum Nash Welfare (MNW), and the Discrete MNW approach, our mechanism is cheaper to compute and enjoys theoretical properties that MNW approaches do not have.

2 Related Work

Recently, a flurry of papers have come out of the operations research, statistics, and computer science communities addressing resource allocations in the aftermath of a pandemic. For instance, Jalota *et al.* [2020] proposed a mechanism for allocating public goods that are capacity constrained due to social distancing protocols, focusing on achieving a market clearing outcome. Mehrotra *et al.* [2020] studied the allocation of ventilators under a stochastic optimization framework, minimizing the expected number of shortages in ventilators while also considering the cost of transporting ventilators. Zenteno [2013] combined influenza modeling techniques with robust optimization to handle workforce shortfall in a pandemic. These papers differ from our work in that they do not explicitly address any fairness considerations that we study in this paper.

[*] Contact Author

There has also been a growing interest in developing resource allocation mechanisms with fairness properties. Under a fairly general class of utility functions including the Leontief utility, computing market equilibrium under the fisher market setting (divisible goods) can be done using an Eisenberg-Gale (EG) convex program Eisenberg and Gale [1959]. Market equilibrium solutions satisfy Pareto optimality, proportionality, and envy-freeness. It is also known that an EG convex program implicitly maximizes Nash welfare, which is the product of all agents' utilities. However, MNW is generally not strategy-proof, and can be computationally expensive for large problems. Chen *et al.* [2006] studied the computation and approximation of market equilibria under the so-called "hybrid Linear-Leontief" utilities. They assume that there are m disjoint groups and each group contains several types of resources. An agent's utility is a linear combination of m Leontief utilities, each associated with a group. Note that although we use the terms "meta-type" and "group" in our problem formulation, our setting is different from theirs because there is only one Leontief utility per user in our setting.

For Leontief utilities, Ghodsi *et al.* [2011] introduced the Dominant Resource Fairness allocation mechanism (DRF) which in addition to the three properties satisfied by market equilibrium solutions, is also strategy-proof. Later Parkes *et al.* [2015] extended the setting to allow agents to be weighted and have zero demand over some resources while maintaining all four desiderata. Our paper generalizes the setting further by allowing accessibility constraints, which as mentioned in the introduction, arise naturally in many practical settings.

For indivisible resources, Caragiannis *et al.* [2019] showed that maximizing Nash welfare with integer constraints (Discrete MNW) satisfies envy-freeness up to one resource unit and has nice guarantees on the Max-Min Share ratio. However, similar to MNW, it is not strategy-proof and as we will see in the numerical section, it does not scale well to large number of agents and resource types. Although exact market equilibrium might not exist in indivisible settings, Budish [2011] showed that a close approximation of it exists in the unweighted, binary allocation case. This was later put into practice for course allocation in Budish *et al.* [2017]. However, the theory does not provide useful approximation bounds when assignments are not binary (e.g., each student only needs one seat from each class, but each hospital may require hundreds of doctors), and therefore it is not well suited to our setting.

3 Problem Formulation

For the remainder of the paper we use local medical personnel allocation as a running example, even though other resource allocation problems can be formulated in a similar fashion. We group resources into *meta-types*: doctors, nurses, ventilators, emergency field hospital beds, etc. Within each meta-type (e.g. doctors), we have *types* (e.g. doctors from the Bronx, doctors from Brooklyn, doctors from Manhattan, etc.[1]). We assume that demands are given over meta-types, but each agent sometimes can only receive allocation from a subset of the resources in a meta-type because of constraints such as location (e.g. a hospital is indifferent to where doctors assigned to it come from as long as they are within commutable radius). We refer to such subsets of resource types in each meta-type as the agents' demand *groups*.

Let $\Omega_1, \Omega_2, \ldots, \Omega_L$ denote the meta-types. Each meta-type Ω_l is a collection of resource types. We assume that $\Omega_i \cap \Omega_j = \emptyset$ for any two different meta-types i, j, which means that each resource type belongs to exactly one meta-type. Let R denote the set of all resource types: $R = \cup_{l \in [L]} \Omega_l$, and N denote the set of agents. We use $m = |R|$ and $n = |N|$ to denote the total number of resource types and agents. Each type of resource r has a finite supply of S_r. We assume that the supplies are normalized within each meta-type:

$$\sum_{r \in \Omega_l} S_r = 1 \ \ \forall l \in [L].$$

Each agent $i \in N$ submits a demand vector $[d_{i1}, \ldots, d_{iL}]$ where d_{il} denotes the fraction of meta-type l that agent i needs in order to get one unit of utility (one can think of this as each agent trying to complete as many units of work as possible, where each unit of work requires d_{il} units of meta-type l). Additionally, each agent has access to only a subset of resource types within each meta-type. We represent this accessibility constraint in the form of a set of demand groups. Let $G_i = \{g_l^i \subseteq \Omega_l : l \in [L], d_{il} > 0\}$, be the set of demand groups for agent i, where $g_l^i \subseteq \Omega_l$ is agent i's demand group for l, specifying the subset of resource types belonging to meta-type l that agent i can access. Note that we only include in G_i meta-types that agent i has non-zero demand of. This is to simplify notation in the later analysis. Intuitively, the introduction of meta-types models the substitution effects, and the introduction of demand groups models the accessibility constraints. When i is clear from the context, we sometimes use g_l instead of g_l^i to simplify the notation.

Following the setup in Parkes *et al.* [2015], we also allow agents to be weighted differently for *each* meta-type and we denote the weight of agent i for meta-type l as w_{il}. Having different weights for different meta-types makes the model more expressive: if we let $w_{i1} = \ldots = w_{iL}$, then this reduces to having a single priority weight for each agent. This weight can depend on factors such as agent i's contribution to the resource pool of meta-type l, as well as the size and stress level of agent i. In the case of medical supplies allocation, weights can represent how much each hospital is in need of extra resources. In the cloud compute setting, weights can represent how much money each user has paid for each meta-type of resource. Note that weights are fixed apriori, not self reported by the agents, nor determined by the allocation algorithm. We assume that weights are normalized within each meta-type: $\sum_{i \in N} w_{il} = 1$ for $l \in [L]$. Note that weights represent agents' priorities over the meta-types, not agents' preferences. Therefore they do not appear in the agents' utility functions, as we will define next.

Let x_i be the allocation vector of agent i: x_{ir} represents the assignment of resource type r to i. For each meta-type l, $\sum_{r \in g_l^i} x_{ir}$ is the fraction of the total supply of meta-type l

[1] Manhattan, the Bronx, and Brooklyn are three boroughs of New York City.

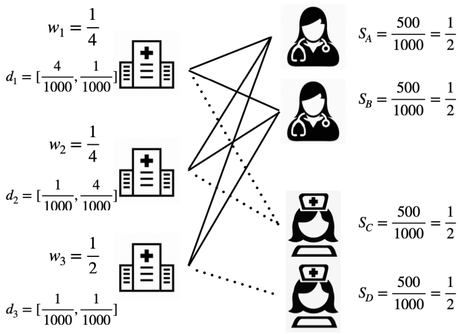
$w_1 = \frac{1}{4}$
$d_1 = [\frac{4}{1000}, \frac{1}{1000}]$
$w_2 = \frac{1}{4}$
$d_2 = [\frac{1}{1000}, \frac{4}{1000}]$
$w_3 = \frac{1}{2}$
$d_3 = [\frac{1}{1000}, \frac{1}{1000}]$
$S_A = \frac{500}{1000} = \frac{1}{2}$
$S_B = \frac{500}{1000} = \frac{1}{2}$
$S_C = \frac{500}{1000} = \frac{1}{2}$
$S_D = \frac{500}{1000} = \frac{1}{2}$

Figure 1: All three hospitals can accept both types of doctors. However, hospitals I and II can only accept Nurse type C, while hospital III accepts only Nurse type D.

that is assigned to agent i. The utility of agent i is then defined as:

$$u_i(x_i) := \min_{g_l \in G_i} \left\{ \frac{1}{d_{il}} \sum_{r \in g_l} x_{ir} \right\} \tag{1}$$

Since agent i needs d_{il} fraction of each meta-type l to finish one unit of work, $u_i(x_i)$ is the total units of work that agent i can finish given allocation vector x_i. This form of utility measure is called the *Leontief utility*.

We now give a concrete example, which is also illustrated in Figure 1. To keep the example simple we assume that each agent has the same weight over all meta-types: only one weight w_i is defined for each agent i.

Example 1. *Consider a case of three hospitals (agents) $\{1,2,3\}$ and two resource meta-types. The first meta-type consists of two types of doctors (resource A, B), and the second consists of two types of nurses (resource C, D): $\Omega_1 = \{A, B\}, \Omega_2 = \{C, D\}$. The normalized weights for the three hospitals are: $w_1 = w_2 = \frac{1}{4}, w_3 = \frac{1}{2}$. The supply for each type of doctor and nurse is 500. Thus, the total available units of each meta-type is $500 + 500 = 1000$, and $S_r = \frac{500}{1000} = \frac{1}{2} \quad \forall r \in \{A, B, C, D\}$. All three hospitals have access to both types of doctors but hospitals $1, 2$ only have access to nurse type C while the third hospital only has access to nurse type D: $G_1 = \{g_1^1 = \{A, B\}, g_2^1 = \{C\}\}, G_2 = \{g_1^2 = \{A, B\}, g_2^2 = \{C\}\}, G_3 = \{g_1^3 = \{A, B\}, g_2^3 = \{D\}\}$. For each unit of utility, hospital 1 demands 4 doctors and 1 nurse, hospital 2 demands 1 doctor and 4 nurses, and hospital 3 demands 1 doctor and 1 nurse. Since the total units of supply for each meta-type is 1000, $d_1 = [\frac{4}{1000}, \frac{1}{1000}]$, $d_2 = [\frac{1}{1000}, \frac{4}{1000}]$, $d_3 = [\frac{1}{1000}, \frac{1}{1000}]$.*

3.1 Desirable Properties

Now we formally define the properties studied in this paper.

Pareto optimality. An allocation mechanism is Pareto optimal if compared to the output allocation x, there does not exist another allocation x' where some agent is strictly better off without some other agent being strictly worse off: $\exists i \; s.t. \; u_i(x_i') > u_i(x_i) \implies \exists j \; s.t. \; u_j(x_j') < u_j(x_j)$.

Weighted envy-freeness. Given an allocation x_j for agent$j$, let $\tilde{x}_i$ be the same allocation adjusted to agent i's weights and demand groups, i.e., $\tilde{x}_{ir} = x_{jr} \frac{w_{il}}{w_{jl}}$ for all $r \in g_l^i, l \in [L]$, and $\tilde{x}_{ir} = 0$ otherwise. $u_i(\tilde{x}_i) - u_i(x_i)$ is how much i envies j. An allocation is weighted envy free if for any $i, j \in N$ this quantity is non-positive, i.e.,

$$u_i(\tilde{x}_i) - u_i(x_i) \leq 0.$$

Intuitively, this means an agent prefers her allocation over the allocation of any other agent scaled by the weight ratios of the two agents. Note that since there is a separate weight for every meta-type l, the allocations for each resource type r is scaled according to the corresponding weight for the meta-type that it belongs to.

Strategy-proofness. In the existing literature, agents can only be strategic by misreporting their demand vector. In our setting however, agents have the additional possibility of misreporting their accessibility constraints for the meta-types (e.g. One can report that she accepts both Intel and AMD CPUs but in fact her program only runs on Intel CPU). Our definition of strategy-proofness guards against both types of misreporting. Let x be the allocation returned by the mechanism under truthful reporting from all agents. Let x' be an allocation returned by the mechanism when all agents report truthfully except agent i reports an alternative demand vector and/or alternative demand groups. The mechanism is strategy-proof if $u_i(x_i) \geq u_i(x_i')$.

Sharing incentive. In settings where the supplies for each resource come from the participants' contribution, sharing incentive is satisfied when the resulting allocation gives each participant at least as much utility as she originally had. More specifically, for each $i \in N$ and $l \in [L]$, let s_{il} be the proportion of meta-type l contributed by agent i *that she can also access*. We can also think of s_{il} as the amount of "useful" resource agent i originally possessed of meta-type l (she might contribute more than s_{il} to the pool). Prior to reallocation of resources, agent i's utility would be

$$u_i^o := \min_{g_l \in G_i} \left\{ \frac{s_{il}}{d_{il}} \right\}.$$

Sharing incentive says that $u_i(x_i) \geq u_i^o \quad \forall i \in N$, where x_i is the output allocation of the algorithm (i.e., all agents have incentives to share (pool) their individual resources for reallocation).[2]

4 Dominant Resource Fairness with Meta-Types

Before describing the algorithm we first define some key concepts used in the algorithm:

$$l_i^* := \arg\min_{l \in [L]} \frac{w_{il}}{d_{il}} \quad d_{i*} := d_{il_i^*} \quad w_{i*} := w_{il_i^*}$$

Namely, l_i^* is the meta-type from which agent i demands the biggest proportional share, adjusted by her priority weights. We refer to l_i^* as the *dominant resource meta-type* for agent i.

[2] A closely related concept called proportionality is also often seen in the literature. We focus on Sharing Incentive in the main paper and include a discussion of proportionality in the Appendix.

d_{i*} is the proportional share demanded by agent i from its dominant resource meta-type to finish one unit of work.

We now present our fair allocation mechanism, which we call *Dominant Resource Fairness with Meta-Types* (DRF-MT). The mechanism proceeds in rounds and agents are gradually "eliminated". In each round t, we use the linear program described in (2) to maximize a fractional value y_t so that each *remaining agent i* ($i \in N_t$) receives at least $y_t \times w_{i*}$ fraction of the total supply from its' dominant resource meta-type l_i^*, and more generally $y_t \times w_{i*} \times d_{il}/d_{i*}$ of each demanded meta-type l.[3] Based on this solution, we eliminate at least one resource and one agent using Definition 1 and 2 (although the algorithm only needs to explicitly maintain a list of active/eliminated agents, not resources). For each agent i eliminated in round t, we set $\gamma_i = y_t$. We fix the fraction of dominant meta-type l_i^* assigned to agent i to $\gamma_i \times w_{i*}$, *without fixing the specific allocations of the resources*. We first observe the following fact (proof is in the Appendix):

Fact 1. *In any round t of Algorithm 1, the allocation constraints in Equation 2 for $i \notin N_t$ are tight for optimal solutions.*

This fact implies that when an agent is eliminated, her utility in the final allocation is fixed, even though the exact allocation is not. Not fixing the allocation is a deliberate algorithmic design choice because agents who are flexible with their demand groups should accommodate agents who are more restrictive (e.g. if agent 1 accepts both type A and B, and agent 2 only accepts type A, then we should allocate agent 1 mostly type B resource, and leave type A resource for agent 2). When the number of agents and resource types is large, it is difficult to characterize such dynamics explicitly. So it is crucial to not fix the allocation to the agents until the last iteration.

We will show that there is at least one new resource and one agent being eliminated in each round. Thus our algorithm requires at most $\min(m, n)$ rounds (in practice it often terminates in 2-3 rounds even with a large number of resources types and agents). Since each round requires solving a polynomial-sized linear program, the overall procedure can be run in polynomial time.

Let N_t, R_t be the set of active agents and resources at the beginning of round t. The LP for round t is defined in (2). Note that the ratio $\frac{d_{il}}{d_{i*}}$ is simply making sure that there is no waste in the allocation. For an agent who has been eliminated, $\frac{\gamma_i w_{i*}}{d_{i*}}$ is her final utility. If agent i is not yet eliminated after round t, then $\frac{y_t w_{i*}}{d_{i*}}$ represents how much utility she is currently guaranteed to receive (it will never decrease in later rounds, see Fact 2).

$$\begin{aligned}
&\max\ y_t \\
\text{s.t. }&\text{(active agents allocation constraints)} \\
&y_t \times w_{i*} \times \frac{d_{il}}{d_{i*}} \le \sum_{r \in g_l} x_{ir} \quad \forall i \in N_t, g_l \in G_i \\
&\text{(eliminated agents allocation constraints)} \\
&\gamma_i \times w_{i*} \times \frac{d_{il}}{d_{i*}} \le \sum_{r \in g_l} x_{ir} \quad \forall i \notin N_t, g_l \in G_i \qquad (2) \\
&\text{(supply constraints)} \\
&\sum_{i \in N} x_{ir} \le S_r \quad \forall r \in R \\
&\text{(non-negativity constraints)} \\
&x_{ir} \ge 0 \quad \forall i \in N, r \in R
\end{aligned}$$

Fact 2. *The optimal value for Equation 2 is non-decreasing over rounds: $y_1^* \le y_2^* \le \ldots$, where y_t^* is the optimal objective function value of the LP in round t.*

This follows because the constraints on eliminated agents are less restrictive than the constraints on active agents, and the set of active agents is decreasing over time.

Definition 1. Resource r is eliminated in round t if t is the first round in Algorithm 1 in which $\sum_{i \in N} x_{ir} = S_r$ for every optimal x.

By Fact 2 it is also easy to see that the set of remaining resources R_t decreases over time.

Definition 2. We give two equivalent definitions for eliminating agents:

- Agent i is eliminated in round t when there exists $g_l \in G_i$ such that $g_l \cap R_{t+1} = \emptyset$.
- Agent i is eliminated in round t when there exists $g_l \in G_i$ such that $y_t \times w_{i*} \times \frac{d_{il}}{d_{i*}} = \sum\limits_{r \in g_l} x_{ir}$ for every optimal x, y_t.

Intuitively, both definitions are saying that agent i can not improve her utility further in later rounds. Due to space constraint we defer the proof of their equivalence, and most of the other results to the Appendix. We include the proof of Claim 1 here because it is a good representation of the flavor of arguments used in other proofs. First we address the question of whether the DRF-MT can be efficiently implemented.

Claim 1. *In each round t of Algorithm 1, at least one remaining resource $r \in R_t$ and one remaining agent $i \in N_t$ is eliminated.*

Proof. Suppose no resource is eliminated in round t, then for each $r \in R_t$, there exists an optimal solution such that $\sum_{i \in N} x_{ir} < S_r$. Then the convex combination of these solutions gives us an optimal solution x^* that satisfies $\sum_{i \in N} x_{i,r}^* < S_r\ \forall r \in R_t$. However, by Definition 2, for every remaining agent $i \in N_t$, $g_l \cap R_t \neq \emptyset \forall g_l \in G_i$. So if we assign a little more of every active resource to every active agent, then the overall objective value would be higher. This contradicts the optimality of the LP.

[3] We also considered an alternative design that did not yield a strategy-proof and envy-free algorithm. See Appendix D .

Algorithm 1: Dominant Resource Fairness with Meta-Types (DRF-MT)

```
Input: Agents $N$, resources $R$, supplies $S_r \ \forall r \in R$,
  demand groups $G_i \ \forall i \in N$, normalized demands
  $d_{il} \ \forall i \in N, g_l \in G_i$, priority weights
  $w_{il} \ \forall i \in N, l \in [L]$
Initialize $N_0 = N$
for $t \leftarrow 0, 1, 2, ...$ do
    $y^*_t \leftarrow$ Solve (2)
    Update the remaining active agents $N_{t+1}$ (using
      Claim 2)
    for agent $i$ eliminated in this round do
        $\gamma_i \leftarrow y^*_t$
    end
    if $N_{t+1} = \emptyset$ then
        Solve Equation 2 and assign resources according
          to $x_{ir}$ with rounding
        break
    end
end
```

Now suppose some resource $r \in R_t$ is eliminated in round t but no agent is eliminated. Suppose this resource type is part of meta-type l. By the first definition in Definition 2, this means that for every i such that $x_{ir} > 0$, there exists $r' \in g^i_l$ such that r' is not eliminated. By the same convex combination argument above, we know that there is an optimal solution such that $\sum_i x_{ir'} < S_{r'}$ for every such r'. Then for every such agent we can remove ϵ allocation of r from her and replace it with ϵ allocation of the corresponding r'. This gives us an allocation that has the same objective as before without using up the entire supply of r, contradicting r being eliminated. □

This result shows that DRF-MT can be implemented efficiently by solving at most $\min(m, n)$ number of polynomial-size linear programs. However, it does not tell us how to find the eliminated agents. The following theorem says that we can do so by looking at the dual variables of the LP. Note that the algorithm does not need to explicitly maintain a list of active resources (Equation 2 does not depend on R_t).

Claim 2. *This claim has two parts:*

- *If the shadow price of an allocation constraint of an active agent in round t is positive, then its corresponding agent is eliminated in round t.*
- *In each round t, at least one allocation constraint corresponding to an agent in N_t has a positive shadow price.*

Now we state our main results.

Lemma 1. *DRF-MT is Pareto optimal.*

Lemma 2. *DRF-MT is weighted envy-free.*

Lemma 3. *DRF-MT is strategy-proof.*

The proofs for these three lemmas all involve a case analysis of different scenarios and showing that the undesirable outcomes violate either the optimality of the LP or the definition of eliminated resources/agents, similar to the arguments presented in the proof of Claim 1.

Lemma 4. *Assume that demands, weights and supplies are all rational numbers. If priority weights of the algorithm are set according to the each agent's accessible contribution to the resource pool (for each meta-type), then DRF-MT satisfies sharing incentive.*

In resource pooling settings, having a separate weight for each meta-type is crucial in proving this result (e.g. contributing a ton of hard drive space but no GPU should not give the agent high priority if GPU is its dominant/bottleneck resource type). The proof constructs a bipartite graph of supplies and demands of the resources, then uses Hall's Theorem Hall [1935] to show that there exists a feasible solution to the first round's LP that already gives every agent at least as much utility as they could get without participating in the pool. Since agents' utilities only increase in later rounds, the final allocation must also satisfy sharing incentive.

4.1 Integral Allocation from Rounding

So far we have implicitly assumed that the resources are divisible, and all fairness results are stated with respect to the *fractional assignment* output of Algorithm 1. In practice we round down the output to obtain the final assignment, since resources such as ventilators are indivisible. Each agent loses at most 1 unit of each type of resource through rounding. Since we focus on problems where each agent receives hundreds of units of each resource, the performance loss due to rounding is small. For example, starting with an envy-free fractional allocation, one agent can envy another by at most $2m$ items after rounding. In Example 1, m is 4, while the total allocations each agent receives are in the hundreds. So an envy of $2m$ items is not significant. Note that such divisibility assumption is also standard in existing DRF literature, which often focus on the compute resource sharing problem: even though CPU cores are discrete, it's common to treat the problem as a continuous problem since there is a large quantity of cores in a compute cluster.

There are many existing algorithms that focus on fair allocation of indivisible goods (e.g. Discrete MNW from Caragiannis *et al.* [2019]). Indivisible resource allocation is particularly important when the quantities of the resources are small (e.g. fairly assigning a car, a house, and a ring to two people). However, as is the case with most discrete optimization problems, these algorithms do not scale well to the sizes that we deal with in a pandemic with hundreds of hospitals and many types of resources. In settings where each agent receives hundreds of units of each resource, the performance loss due to rounding is often small compared to the dramatic increase in computational cost for solving Mixed Integer Programs (see Section 5 for a numerical comparison).

4.2 Connection to Previous Dominant Resource Fairness Algorithms

The core difference between our problem setup and the existing DRF settings (Ghodsi *et al.* [2011], Parkes *et al.* [2015]) is the addition of accessibility constraints. When $|\Omega_l| = 1$

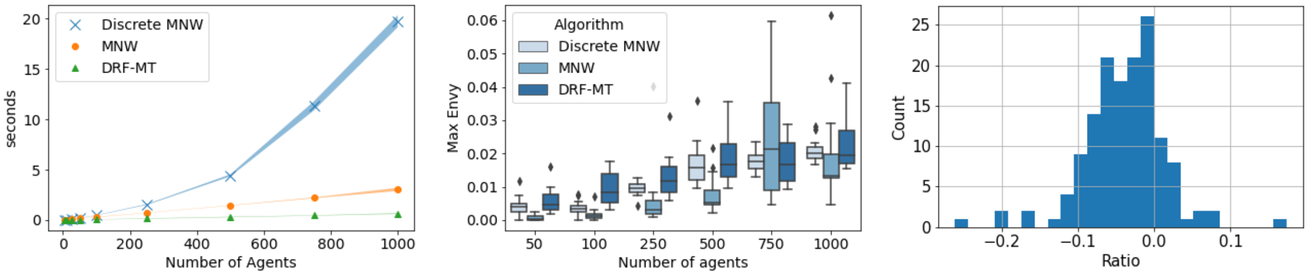

Figure 2: Left: Running time comparison. Middle: Normalized max envy comparison. Right: Distribution of normalized difference in social welfare between Discrete MNW and DRF-MT over all trials. Normalized difference is calculated by subtracting the social welfare of Discrete MNW from that of (rounded) DRF-MT and then dividing by the social welfare of Discrete MNW.

for each $l \in [L]$, both our problem formulation and the DRF-MT algorithm reduce to the problem and algorithm studied in those papers. Note that in this simplified setting one can write out the closed form solutions to the LPs, so no actual optimization needs to be performed. However, it is natural that resources come in different “flavors” and that agents have different constraints/preferences over these variations. So our formulation captures a much wider range of problems encountered in practice.

4.3 Alternative Fair Allocation Mechanisms

As discussed in Section 2 and Section 4.1, the other most suitable approaches in our setting are MNW and Discrete MNW. When the weights are equal, MNW is also commonly referred to as the Competitive Equilibrium with Equal Income (CEEI) approach. Without the accessibility constraints, MNW is known to be Pareto optimal, envy-free and satisfy sharing incentive. However, unlike DRF-MT, it is known that MNW is not strategy-proof (see Section 6 of Ghodsi *et al.* [2011] for an example). We show in Section 5 that our DRF-MT mechanism achieves almost as much social welfare (i.e. sum of utilities of all agents) as MNW, and also runs faster in practice.

4.4 Extension to Arbitrary Group Structure

We currently assume that resources and demands follow a meta-type/group/type structure. One might be interested in a general group structure where a demand group can contain any subset of all possible resources (not necessarily from a single meta-type). The problem with this kind of flexible group structure is that it opens up possibilities for people to cheat the system by misreporting their true demand structure (e.g. instead of reporting that they are indifferent to resource A and B, and that they only need one unit of either one to finish a unit of work, agents can claim that they need one unit each from both A and B to finish one unit of work). In particular, Dominant Resource Fairness based approaches will likely not work, since it is unclear how one would even define the dominant resource under such a general setting. We leave this as an open question for future work.

5 Numerical Experiments

We compare the algorithms on running time, normalized max envy, and social welfare. Normalized max envy is the maximum envy (see Section 3.1) between any pair of agents normalized as a fraction of each agent’s allocated utility. Social welfare is the sum of utilities of all agents. We fix a meta-type structure ($\Omega_1 = \{0\}, \Omega_2 = \{1, 2\}, \Omega_3 = \{3, 4, 5\}, \Omega_4 = \{6, 7, 8, 9\}$) and randomly generate the demands, group structures, and weights for the agents. For each choice of number of agents, we ran 16 trials. All three algorithms Allocations are rounded down for MNW and DRF-MT. All three algorithms allow specifying different agent weights and also observe the accessibility constraints. Gurobi [2021] and Mosek (Aps [2020]) are used to implement the algorithms. More details on the experimental setup and additional experiments can be found in Appendix B .

First we investigate scalability. As shown in Figure 2 (left), the running time for Discrete MNW quickly explodes while MNW and DRF-MT are much more scalable. DRF-MT runtime in particular grows very slowly. The error region represents one standard deviation from the mean.

Recall that DRF-MT is envy-free before rounding. We now investigate envy when the solution is rounded. Without accessibility constraints, MNW is also envy-free before rounding, and Discrete MNW satisfies envy-free up to one good. Figure 2 (middle) shows that all three algorithms have small max envy after rounding in practice ($< 4\%$ in most trials).

Finally, we compare the social welfare obtained under DRT-MT and Discrete MNW. Figure 2 (right) shows that in roughly 95% of the trials, DRF-MT obtained at least 90% as much social welfare as Discrete MNW. In Appendix B we show that MNW has slightly lower social welfare than Discrete MNW, so the above conclusion holds when DRF-MT is compared to rounded MNW as well.

In conclusion, compared to both Discrete MNW and MNW, DRF-MT 1) achieves almost as much social welfare, 2) has comparable level of max envy, 3) has the additional property of strategy-proofness (in the fractional case), and 4) is more scalable. An interesting avenue for future work is to determine the properties of the rounded variant of DRF-MT. A particularly interesting question would be whether one can show that approximate strategy-proofness holds when there is large supply of each item.

References

[ApS, 2020] MOSEK ApS. Mosek modeling cookbook. https://www.mosek.com, 2020. Accessed: 2021-05-20.

[Budish *et al.*, 2017] Eric Budish, Gérard P Cachon, Judd B Kessler, and Abraham Othman. Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research*, 65(2):314–336, 2017.

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

[Caragiannis *et al.*, 2019] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. *ACM Transactions on Economics and Computation (TEAC)*, 7(3):1–32, 2019.

[Chen *et al.*, 2006] Xi Chen, Li-Sha Huang, and Shang-Hua Teng. Market equilibria with hybrid linear-leontief utilities. In Paul Spirakis, Marios Mavronicolas, and Spyros Kontogiannis, editors, *Internet and Network Economics*, pages 274–285, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[Eisenberg and Gale, 1959] Edmund Eisenberg and David Gale. Consensus of subjective probabilities: The pari-mutuel method. *The Annals of Mathematical Statistics*, 30(1):165–168, 1959.

[Ghodsi *et al.*, 2011] Ali Ghodsi, Matei Zaharia, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: Fair allocation of multiple resource types. *NSDI'11: Proceedings of the 8th USENIX conference on Networked Systems Design and Implementation*, pages 323–336, 2011.

[Gurobi Optimization, 2021] LLC Gurobi Optimization. Gurobi optimizer reference manual. http://www.gurobi.com, 2021. Accessed: 2021-05-20.

[Hall, 1935] P. Hall. On representatives of subsets. *Journal of the London Mathematical Society*, s1-10(1):26–30, 1935.

[Jalota *et al.*, 2020] Devansh Jalota, Marco Pavone, and Yinyu Ye. Markets for efficient public good allocation. *ArXiv*, abs/2005.10765, 2020.

[Mehrotra *et al.*, 2020] Sanjay Mehrotra, Hamed Rahimian, Masoud Barah, Fengqiao Luo, and Karolina Schantz. A model for supply-chain decisions for resource sharing with an application to ventilator allocation to combat covid-19. *medRxiv*, 2020.

[Parkes *et al.*, 2015] David C. Parkes, Ariel D. Procaccia, and Nisarg Shah. Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. *ACM Trans. Econ. Comput.*, 3(1), March 2015.

[Zenteno, 2013] Ana Zenteno. *Models for managing surge capacity in the face of an influenza epidemic*. PhD thesis, Columbia University, 2013.