Resource Allocation with Budgets: Optimal Stable Allocations and Optimal Lotteries*

Hau Chan Jing Chen
Department of Computer Science, Stony Brook University
Stony Brook, NY 11777
{hauchan, jingchen}@cs.stonybrook.edu

November 6, 2014

Abstract

We introduce the resource allocation problem where a planner needs to purchase different resources from providers of different qualities and costs, and the planner allocates the resources/providers to consumers with different preferences. The planner has a budget that limits how much he can spend on the resources. He wants to maximize the social welfare generated from the consumers, while keeping his total expenditure for purchasing the resources within his budget. Previous studies have either focused on the resource acquisition part, with one buyer and many strategic sellers, or the resource allocation part, with one seller and many strategic buyers. This is for the first time both providers and consumers are included in the picture.

The consumers do not pay for the resources and will act to maximize their individual utilities. Thus the planner must use proper rationing tools to make sure that they will stick to the providers allocated to them. Two widely existing rationing tools are studied in this paper: waiting times and lotteries (and the combination of them).

We characterize (sometimes partially) the structures of optimal allocation schemes when different rationing tools are used, and we identify conditions under which lotteries can do better and under which waiting times can do better. We also settle the computation complexity for computing/approximating them. For resource allocation with waiting times, we show that the optimal solution is NP-hard to find, and we construct an FPTAS for it. For resource allocation with lotteries, we show that for a large class of the problem the optimal solution has a surprisingly simple structure, and can be solved by a linear program.

Following our results, neither waiting time nor lottery is absolutely better than the other in terms of generating social welfare. A planner should choose an appropriate tool based on the conditions that we identify. Our results then let the planner computes/approximates the corresponding optimal allocations efficiently. Indeed, our results are the first systematic study of both rationing tools when resource acquisition and resource allocation occur together (which is often the case in real life), and we provide useful approaches for future study on this more general and realistic model.

Keywords: budget, social welfare, stable allocations, NP-hardness, approximation, lotteries

*We thank several anonymous reviewers for their comments. The first author is supported by the NSF Graduate Research Fellowship.

1 Introduction

Resource acquisition and resource allocation are the two central problems in resource management, and they often occur together. A central *planner* of an organization must decide, efficiently and effectively, what resources to purchase/rent from which *providers* and how to allocate them to different *consumers* in the organization. For example, the planner can be a government or a company; the resources can be office supplies, transportation vehicles, human resources, etc; and the consumers can be individual employees or business units. Most studies in the mechanism design literature (see, e.g., [3, 8, 17, 22, 30, 32, 57, 59]) have focused on just one side of the acquisition-allocation story and ignored the other: either the planner already owns a set of items and is only concerned about distributing them to strategic consumers; or he is the consumer and knows his own preferences, and he is only concerned about getting them from strategic providers.

In this paper we keep *both* consumers and providers in the picture —a more realistic scenario for the planner. We consider the consumers to be the only strategic side, and the providers in our model are silent: they provide resources of different qualities and costs, but the information about the resources is publicly known. Thus the planner is facing (1) an external market where the resources are sold at market prices and (2) a set of internal consumers with different preferences about the resources who act to maximize their individual utilities. To the best of our knowledge, we are the first to systematically study this more general and realistic model.

An important feature of many resource management problems is that the planner has a *budget* on how much he can spend on the acquisition. The planner's goal is thus to maximize the social welfare of the whole organization —the sum of the consumers' utilities— while keeping his total expenditure within the budget. Another important feature of many such problems is the *absence of money transfer* from the consumers to the planner: the planner pays the providers for the resources but allocates them to the consumers free of charge.[1] Accordingly, when the planner's budget cannot afford to buy the best resource for each consumer (which is often the case), he must use proper rationing tools to enforce certain *fairness* among the consumers and ensure that they will stick to the providers allocated to them. Two classes of rationing tools have been widely studied in the literature: *waiting times* and *lotteries*, corresponding to the two major parts of this paper.

Waiting Times v.s. Lotteries. Waiting times (see, e.g., [27, 35, 38, 39, 40, 51, 58]) are widely used in real life when demand exceeds supply: at popular restaurants, in medical services, in job promotions, etc. Each consumer is free to choose which provider he wants to be served by and will be served after waiting for certain amount of time. A consumer's utility for being served at a provider is his *value* for that provider minus his waiting time there, and the fairness requirement in our model corresponds to the allocations being *stable*: that is, they are *envy-free*[2] and give each consumer a non-negative utility. Rationing by waiting times gives the consumers free choice, which is highly desired by our society. Also, the allocation is deterministic and the planner's goal (i.e., to maximize social welfare given his budget) is achieved with probability 1. However, waiting times incur a loss to the social welfare, since the time waited by the consumers does not benefit anybody.

Lotteries (see, e.g., [13, 20, 26, 47, 52, 62]) are also widely used in many scenarios: at school choice, at companies end-of-year parties, etc. All consumers are offered the same set of lotteries (in the simplest case there is only one), where each lottery is a probabilistic distribution over all providers. Each consumer chooses a lottery and gets a provider sampled according to it. Thus a consumer is not completely free to choose his provider: he can choose any lottery he wants but then

[1] In some cases, such as the allocation of medical resources to patients of a healthcare plan, payments of the consumers are not completely forbidden but highly constrained.

[2] The idea of envy-freeness has been widely adopted in mechanism design, see, e.g., [31, 37, 60, 53, 54, 41, 46, 24, 36]. In our model, it means no consumer wants to exchange his provider and waiting time with those of another consumer.

has to stick to the sampled provider. A consumer's utility, given a lottery, is his expected value for the sampled provider, and fairness is defined in terms of expected values. The planner's total cost and the social welfare are also measured in expectation. Lotteries can avoid the welfare-burning effect of waiting times and conceivably achieve better social welfare. However, the planner must be able to enforce the consumers to stick to the sampled providers, and it is possible that under certain coin tosses neither is the budget constraint satisfied nor is the social welfare optimal.

The two rationing tools can also be combined: a lottery can be associated with a waiting time, and a consumer choosing this lottery will wait for that amount of time and then get a sampled provider. A stable allocation is thus a special case of such allocations. Lottery allocations with waiting times are similar to lottery pricing schemes [13, 20, 52, 62], except that their goal is to maximize revenue while ours is to maximize social welfare with budget constraints. However, different from lottery pricing schemes where the structure and the computation complexity of optimal solutions are still not well understood in many cases, in this paper we characterize (sometimes partially) the structures of optimal allocation schemes for those use waiting times and those using lotteries, including those using both. Part of our results may bring new insights to lottery pricing.

1.1 Our Results

We consider the allocation of one type of resource that can be obtained from different providers. Here a provider is an abstract way of representing the resource with specific quality and cost, and it may or may not correspond to a physical institution. For example, in a hiring procedure, the providers can be candidates with different succeeding probabilities and salary requirements, or, in vehicle rental, the providers can be trucks of different volumes and prices (even if all of them are owned by U-Haul). A consumer's value for a provider is proportional to the provider's quality: this is typical in many real-life scenarios, such as in the examples above where quality means succeeding probability or transportation volume, in advertisements where quality means view-through or click-through rate [33, 63, 2, 7, 50], or in scenarios where quality means the probability of obtaining the same resource [4, 18, 28, 29, 42, 44]. Our models are formally defined for waiting times in Section 2 and for lotteries in Section 3. Below we summarize our main results. Due to the lack of space, most of the proofs are provided in the appendix.

Optimal Stable Allocations. For resource allocation with waiting times, we completely characterize the structure of optimal stable allocations and use this characterization to show the following.

Theorem 1. (restated) *It is NP-hard to compute an optimal stable allocation.*

Theorem 2. (restated) *There exists an FPTAS for the optimal stable allocation problem which, given any $\epsilon > 0$, runs in time $O((n+m)n^3m/\epsilon)$, where n and m are respectively the numbers of consumers and providers.*

To construct the desired FPTAS we introduce another problem, *ordered Knapsack*. Roughly speaking, this is a bounded Knapsack problem where the items' values are affected by the order in which the items are packed into the knapsack. We construct an FPTAS for this problem and show how to use it to approximate the optimal stable allocation. We believe that the ordered Knapsack problem itself is of independent interest and is worth further study. Detailed discussion on Theorems 1 and 2 are in Sections 2.1 and 2.2, respectively.

Optimal Lottery Allocations. If we consider the consumers as a continuous population represented by the interval $[0, 1]$, their values of the providers can be specified by a function v mapping each consumer $x \in [0, 1]$ to a non-negative real, such that the value of consumer x at a provider is $v(x)$ times the provider's quality. The function v is called the consumers' *valuation profile.*[3] It

[3]We do not call v the valuation *function* because it applies to all consumers, in which sense it corresponds to a valuation profile in the discrete case.

is somewhat surprising that, given the extremely rich structures of the possible lottery allocations (with or without waiting times associated), for a large class of resource allocation problems, the optimal lottery allocation has a very simple form: that is, there is a single distribution from which all consumers' providers are drawn, and no waiting time is imposed. We call such an allocation scheme a *randomized allocation*, and we have the following.

Theorem 4. (restated) *For any $v(x)$ such that $(1-x)v'(x)$ is non-increasing, the optimal randomized allocation is optimal among all lottery allocations, including those with waiting times.*

Notice that a randomized allocation does not require the consumers' providers be sampled independently: our result holds as long as the marginal distribution for each consumer is as specified by the allocation. In Section 3.1 we show that the optimal randomized allocation can be solved by a linear program, so does the optimal lottery allocation whenever the condition in Theorem 4 holds. In particular, the condition holds when $v(x)$ is *concave*, and for many other cases where it is neither concave nor convex, but does not increase "too fast". When the condition holds, the ratio between the social welfare of the optimal randomized allocation and that of the optimal stable allocation can be arbitrarily large. Moreover, the randomized allocation can be implemented so that the budget constraint is satisfied with probability 1. (See also Appendix D.) When the condition does not hold, there are cases (see Appendix F) where optimal stable allocations do better.

Interestingly, the condition in Theorem 4 has a very natural interpretation from another viewpoint. If we consider a single consumer who first has his "type" drawn uniformly at random from $[0, 1]$ and set his value to $v(x)$, then the condition in Theorem 4 holds if and only if the distribution of the consumer's value has *monotone hazard rate* (MHR). This immediately connects our result with lottery pricing schemes [19] with a single buyer and multiple items, where optimal pricing schemes are studied when the distributions of the buyer's values have MHR.

We generalize our result to settings where the consumers' values are not proportional to the providers' qualities. Here the randomized allocation may not be optimal among all lottery allocations, but we have the following.

Theorem 5. (restated) *Under similar conditions as in Theorem 4, the optimal randomized allocation is better than the optimal stable allocation.*

Theorem 5 is formalized in Section 3.2. Again, when the conditions do not hold there are cases (see Appendix F) where optimal stable allocations do better.

In sum, our results suggest that neither rationing tool is absolutely better than the other in terms of generating social welfare, and a planner should choose an appropriate tool based on the consumers' valuations as specified by Theorems 4 and 5. Our results allow the planner to compute/approximate the corresponding optimal allocations efficiently.

1.2 Related Work

The closest setting to ours is the one in [11], on purchasing and providing healthcare services. Since the authors allow arbitrary values of the patients for the hospitals, the NP-hardness for computing the optimal equilibrium there is much easier to show compared with our result on the complexity of optimal stable allocations. Also, in [11] the authors study optimal lotteries when there are two hospitals. Since we allow any number of providers, our results on optimal lotteries and randomized allocations have greatly generalized theirs. Besides [11], we are not aware of any other study where both resource acquisition and resource allocation are considered in the same model. There are, however, many studies focusing one side of the story.

On the one hand, procurement games with budget constraints, where the providers are strategic and there are no consumers, have been studied under the framework of *budget-feasible mechanisms* [59, 30, 22, 8, 17], where the goal is to maximize the buyer's value for the items he buy. *Frugal* *mechanisms* [61, 34, 45, 5, 16] also study procurement problems, but instead of imposing a budget constraint on the total cost, they look for solutions whose cost is as low as possible.

In unit-demand pricing schemes [12, 13, 19, 20, 21, 41, 44, 52, 62] n items are to be sold to m buyers and each buyer only wants one of them, similar to our case where each consumer needs to be assigned to one provider. Envy-freeness is a widely adopted solution concept there, but the goal of pricing schemes is to maximize revenue. In [28, 44], the authors study pricing problems where the buyers' valuations are similar to ours, and they characterize the optimal envy-free solutions for revenue [28, 44] and for total values [28] (what they call social welfare) in their settings. Although their characterizations are analogous to ours, their goals are different and they do not further study the computational complexity issues of the optimal solutions. While most works on pricing schemes study deterministic optimal item-pricing [12, 19, 21, 41, 44], a few consider lotteries [13, 20, 52, 62] and show that they can generate more revenue than deterministic item-pricing in various cases. However, the structures of optimal lottery pricing schemes are far from being well understood.

The social welfare studied in our model has also been studied in *money-burning mechanisms* [43], but for single-good auctions only. Moreover, the relation among providers and consumers can be considered as a *unit-demand auction* [57, 32], where prices correspond to waiting times in our model. Sometimes the goal of the auctioneer there is to maximize the same social welfare as ours [3]. However, a big difference between our model and unit-demand auctions is that in our case the numbers of items available from the providerd depend on their costs and the planner's budget, thus are not prefixed but need to be determined as part of the solution.

Finally, in scheduling problems a set of jobs need to be assigned to a set of machines, similar to the allocation problem we consider. A well studied objective there is *makespan* minimization [55, 49, 6, 24, 23] —that is, to minimize the maximum finishing time (which is considered as the cost) of the machines. There is no value associated with the jobs being carried out, neither is there a budget for the total cost. In [15] and [25], the authors introduce waiting times in order to reduce the makespan and/or the total completion time —a "money-burning-based" scheduling problem. Furthermore, a particular variant of the scheduling problem is online scheduling, see, e.g., [56], where the jobs arrive along time. Although in this paper we do not consider the online resource allocation problem, it would be very interesting to explore that direction.

2 Resource Allocation with Waiting Times

In the resource allocation problem, a *planner* allocates a particular type of resource to n *consumers*. The resource can be provided by m *providers*, where each provider j serves the resource with *quality* $q_j \geq 0$ and can serve each consumer assigned to him for a *cost* $c_j \geq 0$. For example, if the resource is computation power then the providers can be computer clusters, with each q_j being the processing speed and c_j the price for purchasing or renting such a cluster. We do not require c_j to be proportional to q_j: some providers may be more cost-efficient than others. Consumer i's values for the providers are proportional to their qualities: there exists a value $v_i \geq 0$ such that i's value for being served by provider j is $v_i q_j$. As pointed out in the Introduction, this valuation model captures many resource allocation problems and carries many interesting properties as shown by our results. We shall study more general valuation models in the last part of this paper.

A solution to the resource allocation problem is called an *allocation*. An allocation A consists of an *allocation function* $a : [n] \to [m]$ and a *waiting vector* $w = (w_1, \dots, w_m)$ where $w_j \geq 0$ for each $j \in [m]$: $a(i)$ is the provider to whom consumer i is assigned and w_j is the waiting time for any consumer assigned to provider j. The total *cost* of A is $C(A) = \sum_{i\in[n]} c_{a(i)}$. The planner has a *budget* $B \geq 0$, and A is *budget-feasible* if $C(A) \leq B$. Given A, each consumer i's *utility* is $v_i q_{a(i)} - w_{a(i)}$, and the *social welfare* of A is $SW(A) = \sum_{i\in[n]} v_i q_{a(i)} - w_{a(i)}$.

An allocation A is *stable* if for any consumer i, (1) $v_i q_{a(i)} - w_{a(i)} \geq 0$ and (2) for any provider j, $v_i q_{a(i)} - w_{a(i)} \geq v_i q_j - w_j$. Notice that we could have allowed different consumers to have different waiting times at the same provider. However, an allocation is *envy-free* if for any consumers i and i', $v_i q_{a(i)} - w_{a(i)} \geq v_i q_{a(i')} - w_{a(i')}$. Thus envy-freeness automatically implies that consumers assigned to the same provider have the same waiting time, and our model is without any loss of generality. We are interested in mechanisms that, given the consumers' values, output allocations that are stable and budget-feasible, and maximize the social welfare.

Definition 1. *A stable allocation A is* optimal *if $A \in \text{argmax}_{A' \text{ is stable and budget-feasible}} SW(A')$.*

Without loss of generality we assume $nc_{max} > B$ and $nc_{min} \leq B$, where c_{max} and c_{min} are the maximum and the minimum costs of the providers, respectively.[4] Moreover, it is often useful to consider optimal stable allocations with respect to a particular allocation function, as follows.

Definition 2. *For any allocation function a, a stable allocation $A = (a, w)$ is* optimal with respect to a *if $A \in \text{argmax}_{A'=(a,w') \text{ and } A' \text{ is stable}} SW(A')$.*

Note that a stable allocation optimal with respect to a is not required to be budget-feasible: the cost of such allocations is decided by a, thus either all of them are budget-feasible or none is.

2.1 The Hardness of Finding Optimal Stable Allocations

In this and next sections, for convenience we rename the consumers and the providers so that

$$v_1 \geq v_2 \geq \cdots \geq v_n \quad \text{and} \quad q_1 \geq q_2 \geq \cdots \geq q_m. \tag{1}$$

Theorem 1. *It is NP-hard to compute an optimal stable allocation.*

To prove Theorem 1, we first characterize the social welfare of optimal stable allocations. Our characterization is similar to that in [28, 44], but the exact formulas are very different from [44]. Indeed, [44]'s goal was to maximize the total payment, while our goal is to maximize social welfare. Our Definition 4 and Lemma 2 are equivalent to Lemma 2.1 in [28]. Thus, we only state our key lemmas here and refer the readers to Appendix A for the proofs

Definition 3. *An allocation function a is* ordered *if $a(1) \leq a(2) \leq \cdots \leq a(n)$.*

In Lemma 4 of Appendix A we show that it is sufficient to consider stable allocations $A = (a, w)$ with a ordered. For any such A,

$$q_{a(1)} \geq q_{a(2)} \geq \cdots \geq q_{a(n)} \quad \text{and} \quad w_{a(1)} \geq w_{a(2)} \geq \cdots \geq w_{a(n)}. \tag{2}$$

Definition 4. *For any ordered allocation function a, an allocation $A = (a, w)$ is* tight at a *if*

$$w_{a(n)} = 0 \text{ and } w_{a(i)} = (q_{a(i)} - q_{a(i+1)})v_{i+1} + w_{a(i+1)} \text{ for any } i < n.$$

Notice that, by the first part of Inequality 2, the $w_{a(i)}$'s in Definition 4 are all non-negative. Also notice that, being tight at a implies that for any $i < n$, $v_{i+1} q_{a(i+1)} - w_{a(i+1)} = v_{i+1} q_{a(i)} - w_{a(i)}$. That is, consumer $i + 1$ is indifferent between his utility at $a(i + 1)$ and that at $a(i)$.

Lemma 1. *For any ordered allocation function a, let $A = (a, w)$ be an allocation such that, A is tight at a and $w_j = v_1 q_1$ for any $j \notin a(\{1, \ldots, n\})$. Then A is stable.*

[4] If not all consumers have to get the resource, the planner can add a dummy provider with quality 0 and cost 0 (representing the option of not getting the resource) and the condition $nc_{min} \leq B$ is satisfied.

Lemma 2. *For any ordered allocation function a and any stable allocation $A = (a, w)$, A is optimal with respect to a if and only if it is tight at a.*

The following lemma shows that the social welfare of any stable allocation optimal with respect to a can be explicitly calculated from the consumers' values and the providers' qualities.

Lemma 3. *For any ordered allocation function a and any stable allocation $A = (a, w)$ optimal with respect to a, $SW(A) = \sum_{i<n} i \cdot q_{a(i)} \cdot (v_i - v_{i+1}) + n \cdot q_{a(n)} \cdot v_n$.*

Given the above lemma, we show how to reduce subset-sum to the resource allocation problem. We refer the readers to Appendix A for the remaining of the proof. Consider the decision version of the resource allocation problem:

$$\begin{aligned} DRA \;=\; & \{(q_1, \ldots, q_m, c_1, \ldots, c_m, v_1, \ldots, v_n, B, V) : \\ & \quad \text{there exists a stable budget-feasible allocation } A \text{ s.t. } SW(A) \geq V\}. \end{aligned}$$

It is clear that if one can find an optimal stable allocation for every instance of the resource allocation problem then one can decide DRA. We shall show that DRA is NP-complete by a reduction from the Subset-Sum problem:

$$\mathit{SubsetSum} = \left\{ (s_1, \ldots, s_n, T) : \text{ there exists } S \subseteq [n] \text{ s.t. } \sum_{i \in S} s_i = T \right\}.$$

Given an instance $\alpha = (s_1, \ldots, s_n, T)$ of $\mathit{SubsetSum}$, we assume without loss of generality that $s_1 \geq s_2 \geq \cdots \geq s_n$, and construct an instance $\gamma = (q_1, \ldots, q_m, c_1, \ldots, c_m, v_1, \ldots, v_n, B, V)$ of DRA as follows. Notice that we use the same symbol for both a variable and its binary representation, and the 1st bit refers to the rightmost bit.

- There are $m = 2n$ providers and n consumers.
- For each $i \in [n]$, $q_i = c_i = s_i \cdot 2^{n(\lceil \log n \rceil + 1)} + 2^{(n-i)(\lceil \log n \rceil + 1)}$. That is, q_i and c_i are obtained by appending $n(\lceil \log n \rceil + 1)$ bits of 0's to the right of the binary representation of s_i, and then set the $(n-i)(\lceil \log n \rceil + 1) + 1$st bit to 1.
- For each $i \in [n]$, $q_{n+i} = c_{n+i} = 2^{(n-i)(\lceil \log n \rceil + 1)}$. That is, q_{n+i} and c_{n+i} consist of one bit of 1 followed by $(n-i)(\lceil \log n \rceil + 1)$ bits of 0's. Notice that the unique bit of 1 in q_{n+i} and c_{n+i} is aligned with the unique bit of 1 after s_i in q_i and c_i.
- $B = V = T \cdot 2^{n(\lceil \log n \rceil + 1)} + \sum_{i \in [n]} 2^{(n-i)(\lceil \log n \rceil + 1)}$. That is, B and V are obtained by appending $n(\lceil \log n \rceil + 1)$ bits of 0's to the right of the binary representation of T, and then set the $(n-i)(\lceil \log n \rceil + 1) + 1$st bit to 1 for each $i \in [n]$.
- For each $i \in [n]$, $v_i = \sum_{k=i}^{n} \frac{1}{k}$.

It is easy to see that the construction takes polynomial time and that γ satisfies Inequality 1.

2.2 An FPTAS for Optimal Stable Allocations

Letting A^{opt} be the optimal stable allocation, we have the following.

Theorem 2. *There exists an algorithm for the resource allocation problem such that, given any $\epsilon > 0$, it runs in time $O((n + m)n^3 m/\epsilon)$ and outputs a stable budget-feasible allocation $A = (a, w)$ such that $SW(A) \geq (1 - \epsilon)SW(A^{opt})$.*

In order to prove Theorem 2, notice that by Lemma 3, for any ordered allocation function a we can define the *social welfare of* a, $SW(a)$, to be the social welfare of stable allocations optimal with respect to a. That is,

$$SW(a) = \sum_{i<n} i \cdot q_{a(i)} \cdot (v_i - v_{i+1}) + n \cdot q_{a(n)} \cdot v_n. \tag{3}$$

An allocation function a is *budget-feasible* if $C(a) = \sum_{i\in[n]} c_{a(i)} \leq B$.

Definition 5. *An ordered allocation function a is* optimal *if* $a \in \underset{a' \text{ is ordered and budget-feasible}}{\operatorname{argmax}} SW(a')$.

Given an ordered allocation function a, by Lemmas 1 and 2 we can construct, in time $O(m+n)$, a stable allocation A optimal with respect to a: that is, the allocation defined in Lemma 1. If a is optimal, then A is an optimal stable allocation. Thus to prove Theorem 2 it suffices to focus on approximating the optimal ordered allocation function.

Notice that if there exists a provider j such that $c_j < c_{j+1}$, then for any ordered allocation function a and for all consumers assigned to provider $j+1$, by reassigning them to j we get another ordered allocation a' such that $C(a') \leq C(a)$ and $SW(a') \geq SW(a)$, where the second inequality is by Equation 3 and because $q_j \geq q_{j+1}$. Accordingly, we can further focus on ordered allocation functions that do not assign any consumer to $j+1$. That is, we can assume without loss of generality that $c_1 \geq c_2 \geq \cdots \geq c_m$. Below we define a more general problem and construct an FPTAS for it, which will give us an FPTAS for the optimal ordered allocation function.

2.3 The Ordered Knapsack Problem

Definition 6. *The* ordered Knapsack *problem has m items, n players, and a budget B. Each item j has n copies, with cost c_j each. Each player i has value u_{ij} for item j. We have $c_1 \geq c_2 \geq \cdots \geq c_m$, $u_{i1} \geq u_{i2} \geq \cdots \geq u_{im}$ for each $i \in [n]$, and $nc_m \leq B < nc_1$. An* allocation *is a function $a : [n] \to [m]$ such that $a(1) \leq a(2) \leq \cdots \leq a(n)$. The* social welfare *of a is $SW(a) = \sum_{i\in[n]} u_{ia(i)}$, and the* cost *of a is $C(a) = \sum_{i\in[n]} c_{a(i)}$. The goal is to find an allocation with cost no larger than B and the maximum possible social welfare.*

Intuitively, the ordered-Knapsack problem has a knapsack where the order of the items packed in it affects their values —the "players" can be considered as ordered slots in the knapsack.[5] Following Equation 3, we can reduce the problem of the optimal ordered allocation function to the ordered Knapsack problem by taking, for any $j \in [m]$, $u_{ij} = iq_j(v_i - v_{i+1})$ for any $i < n$ and $u_{nj} = nq_jv_n$. Any allocation of the resulting ordered Knapsack problem is an ordered allocation function of the original resource allocation problem, with the same cost and the same social welfare. Thus, letting a^{opt} be the optimal allocation for the ordered Knapsack problem, to prove Theorem 2 it suffices to construct an FPTAS for a^{opt}.

Theorem 3. *There exists an algorithm for the ordered Knapsack problem such that, given any $\epsilon > 0$, it runs in time $O((n+m)n^3m/\epsilon)$ and outputs an allocation a with $C(a) \leq B$ and $SW(a) \geq (1-\epsilon)SW(a^{opt})$.*

[5] Such a scenario widely exists in real life. For example, in school choices the order may represent the priority of being admitted to different schools. Indeed, priority list has been widely studied in the Economics literature (see, e.g., [10, 1, 14]). But the model and the concerns there are different from ours, e.g., the optimization goal is usually not utilitarian, and there is no budget constraints. Thus we do not elaborate on this line of research. Also notice that the ordered Knapsack problem is quite different from the partially ordered Knapsack problem studied in [48]. In the latter each item has a fixed value and the outcome is a set instead of a function from players to items.

Theorem 3 is proved in Appendix B, where we first construct a dynamic program that computes the optimal allocation in pseudo-polynomial time, and then run it on properly scaled inputs to get the desired FPTAS. The proof of Theorem 2 is in Appendix B as well.

Remark 1. *In fact, we can construct a pseudo-polynomial time dynamic program directly for the resource allocation problem. Then one may try to scale down the providers' qualities q_j and consumers' values v_i separately and apply the dynamic program on the scaled inputs. However, when scaling everything back, the errors in the social welfare will accumulate multiplicatively, due to the terms $iq_{a(i)}(v_i - v_{i+1})$. Thus the desired approximation ratio cannot be guaranteed. The idea is to scale down each $iq_j(v_i - v_{i+1})$* as a whole, *but the resulted parameters may not lead to a well defined resource allocation problem with qualities and values. That is where the ordered Knapsack problem comes into play.*

3 Resource Allocation with Lotteries

If no randomness is allowed, the optimal stable allocation is the best we can hope. However, if the planner can ask the consumers to enter lotteries, the space of possible mechanisms becomes much larger and more social welfare can be obtained. In this section, we first characterize the structure of the optimal lotteries for a large sub-class of the resource allocation problem. Furthermore, for a class of valuations more general than what we currently consider, we characterize the conditions under which a particular lottery achieves more social welfare than the optimal stable allocation.

Although our results apply to the discrete case of n consumers, they are more succinct to state for a continuous population of consumers. Thus in the discussion below, we let the consumers be indexed by the interval $[0, 1]$, and use the valuation function $v(x)$ to specify the value of each consumer x. We assume $v(x)$ is strictly increasing and twice differentiable, so that the integrations and differentiations used below are always well defined.[6] Also, by shifting down all consumers' values by $v(0)$, we assume without loss of generality that $v(0) = 0$.

Definition 7. *A* lottery λ *for the resource allocation problem is a tuple of non-negative reals, $\lambda = (p_1, \ldots, p_m, w)$, such that $\sum_{j\in[m]} p_j \leq 1$. A* lottery scheme L *is a set of lotteries such that there exists $\lambda = (p_1, \ldots, p_m, w) \in L$ with $w = 0$.*

A consumer taking lottery λ will wait for time w and then be assigned to each provider j with probability p_j. Consumer x's *(expected) utility* under λ is $u(x, \lambda) = (\sum_{j\in[m]} p_j q_j v(x)) - w$.[7] Given L, each consumer chooses a lottery to maximize his own utility. That is, denoting by $\lambda^L(x) = (p^L_1(x), \ldots, p^L_m(x), w^L(x)) \in L$ the choice of consumer x, we have that for any $\lambda \in L$,

$$u(x, \lambda^L(x)) \geq u(x, \lambda). \tag{4}$$

The definition of a lottery scheme ensures $u(x, \lambda^L(x)) \geq 0$ for any x, and $u(0, \lambda^L(0)) = 0$.

Since for any two lotteries $\lambda_1, \lambda_2 \in L$, any convex combination $\alpha\lambda_1 + (1-\alpha)\lambda_2$ can be realized by a consumer choosing λ_1 with probability α and λ_2 with probability $1-\alpha$, without loss of generality we assume that L is convex. Accordingly, the consumers' choices are on the boundary of L.

[6] Our approach works as long as $v(x)$ is non-decreasing (which is without loss of generality since we can reorder and rename the consumers) and piece-wise twice differentiable. But in this more general setting the analysis is unnecessarily complicated without bringing in more interesting view points. Thus we stick to our current setting so as to highlight the key ideas. Moreover, following Inequality 1, we could have assumed that $v(x)$ is decreasing. But assuming $v(x)$ to be increasing will make the statements and the analysis of the results more succinct.

[7] If $\sum_j p_j < 1$ then with probability $1 - \sum_j p_j$ the consumer waits for time w but does not get any resource. If each consumer has to be served, then we just need to require $\sum_j p_j = 1$ in the definitions and our results still hold.

Since the consumers are infinite, each provider j's cost c_j denotes the cost for serving 1 unit of the population at j, and the *(expected) cost* of L is $C(L) = \int_0^1 \sum_{j\in[m]} p_j^L(x)c_j dx$. L is *budget-feasible* if $C(L) \leq B$. The *(expected) social welfare* of L is $SW(L) = \int_0^1 u(x, \lambda^L(x))dx$. We denote by L^{opt} the *optimal lottery scheme*, that is, $L^{opt} \in \text{argmax}_{L \text{ is budget-feasible}} SW(L)$. For each $x \in [0, 1]$, we denote by $\lambda^{opt}(x) = (p_1^{opt}(x), \dots, p_m^{opt}(x), w^{opt}(x))$ the choice of consumer x under L^{opt}.

A stable allocation $A = (a, w)$ is defined as before, except a is now a function on $[0, 1]$. It is easy to see that A is equivalent to a lottery scheme L which is the convex hull of a set of lotteries $\{\lambda_1, \dots, \lambda_m\}$: for each $j \in [m]$, $\lambda_j = (p_1, \dots, p_m, w_j)$, $p_j = 1$ and $p_{j'} = 0$ for any $j' \neq j$. Given L, each consumer x chooses $\lambda_{a(x)}$, which corresponds to being assigned to $a(x)$ with probability 1 after waiting $w_{a(x)}$. Thus we have $SW(L^{opt}) \geq SW(A^{opt})$, where A^{opt} is the optimal stable allocation.

Besides stable allocations, another class of lottery schemes is of particular interest: those with waiting time 0. Such a lottery scheme L reduces to a single lottery $(p_1, \dots, p_m, 0)$ whose expected quality $\sum_{j\in[m]} p_j q_j$ is the maximum in L, since this lottery maximizes all consumers' utilities over L. We call a lottery scheme of this form a *randomized allocation*, formally defined below.

Definition 8. *A* randomized allocation R *is a tuple of non-negative reals, $R = (p_1, \dots, p_m)$, such that $\sum_{j\in[m]} p_j \leq 1$.*

According to R, each consumer is assigned to each provider j with probability p_j and waiting time 0. The expected social welfare of R is $SW(R) = \int_0^1 \sum_{j\in[m]} p_j q_j v(x)dx$, and the expected cost is $C(R) = \sum_{j\in[m]} p_j c_j$. We denote by R^{opt} the *optimal randomized allocation*, that is, $R^{opt} \in \text{argmax}_{R \text{ is budget-feasible}} SW(R)$. It is immediate that $SW(L^{opt}) \geq SW(R^{opt})$, as a randomized allocation is a special lottery scheme.

3.1 Optimal Lottery Schemes

The structure of the optimal lottery scheme is hard to characterize in general, but as we show in the following theorem, for a large sub-class of the problem, the optimal randomized allocation is actually optimal among all lottery schemes.

Theorem 4. *For any $v(x)$ such that $(1 - x)v'(x)$ is non-increasing, $SW(R^{opt}) = SW(L^{opt})$.*

The proof of Theorem 4 is in Appendix C. The following shows that the class of valuation functions satisfying Theorem 4 is very broad: in particular it includes all concave valuation functions.

Corollary 1. *For any concave valuation function $v(x)$, $SW(R^{opt}) = SW(L^{opt})$.*

Proof. Letting $g(x) = (1 - x)v'(x)$, we have $g'(x) = -v'(x) + (1 - x)v''(x)$. Since $v(x)$ is concave, $v''(x) \leq 0$. Since $v'(x) > 0$ and $x \in [0, 1]$, we have $g'(x) \leq 0$. Thus $g(x)$ is non-increasing. $\square$

Clearly Theorem 4 applies to many other valuation functions that are not concave. For example, letting $v(x) = e^x - 1$, we have $(1 - x)v'(x) = (1 - x)e^x$, which is non-increasing on $[0, 1]$. Thus the optimal randomized allocation is optimal among all lottery schemes in this case. It is not hard to see that R^{opt} can be computed by a linear program, and can be implemented so that the budget constraint is satisfied with probability 1. Further more, when Theorem 4 applies, the ratio between $SW(R^{opt})$ and $SW(A^{opt})$ can be arbitrarily large. We elaborate these properties in Appendix D.

Theorem 4 in Terms of Monotone Hazard Rate. Interestingly, the condition in Theorem 4 has a very natural interpretation from another viewpoint. Consider an allocation problem where there is a single consumer and multiple providers. The planner's budget is lower than the cost of the consumer's favorite provider, thus he cannot simply be assigned there with probability 1. There is a distribution D from which the consumer's value is drawn: in particular, his "type" is uniformly distributed over $[0,1]$ and his value at type x is $v(x)$. All the concepts we have defined can be defined naturally for this Bayesian allocation problem. Letting $y = v(x)$, it is easy to see that for any value v_0 and $x_0 = v^{-1}(v_0)$, the cumulative distribution function is $F(v_0) = \Pr[y \leq v_0] = \Pr[v^{-1}(y) \leq x_0] = x_0 = v^{-1}(v_0)$, and the probability density function is $f(v_0) = F'(v_0) = \frac{1}{v'(x_0)}$. Accordingly, $(1-x_0)v'(x_0) = \frac{1-F(v_0)}{f(v_0)} = \frac{1}{h(v_0)}$, where $h(v) \triangleq \frac{f(v)}{1-F(v)}$ is the *hazard rate* of D. Recall that a distribution has *monotone hazard rate* (MHR) if the function h is non-decreasing. Thus $(1-x)v'(x)$ is non-increasing *if and only if* D has MHR, and we immediately have the following.

Corollary 2. *For any value distribution D that has MHR, $SW(R^{opt}) = SW(L^{opt})$.*

3.2 Randomized Allocations v.s. Stable Allocations

We now extend our approach to settings where the consumers' values are not proportional to the providers' qualities, but there are still orders among the providers and the consumers. More precisely, for each $j \in [m]$, let function $v_j(x)$ be the value that consumer $x \in [0,1]$ receives when assigned to provider j. Again by shifting each function $v_j(x)$ down by $v_j(0)$, we assume without loss of generality that $v_j(0) = 0$ for each j. We consider the cases where each $v_j(x)$ is strictly increasing and $v_1(x) \leq v_2(x) \leq \cdots \leq v_m(x)$ for each x.

As will become clear in the analysis, the key factors affecting the social welfare are actually not the consumers' values, but the *differences* among their values at different providers. Accordingly, for each $j \in [m]$, letting $f_j(x)$ be a function on $[0,1]$ that is strictly increasing, twice differentiable and $f_j(0) = 0$,[8] we consider the consumers' values such that $v_j(x) = \sum_{k=1}^{j} f_k(x)$ for any $x \in [0,1]$. Notice this setting includes that of Section 3.1 as a special case: by renaming the providers we have $q_1 \leq q_2 \leq \cdots \leq q_m$, and we can take $f_1(x) = q_1 v(x)$ and $f_j(x) = (q_j - q_{j-1})v(x)$ for any $j > 1$. All concepts we have considered can be naturally extended to this more general setting.

In this more general setting it is unclear how to compare the optimal lottery scheme and the optimal randomized allocation, but we still have the following.

Theorem 5. *If $(1-x)f_j'(x)$ is non-increasing for every $j \in [m]$, then $SW(R^{opt}) \geq SW(A^{opt})$.*

The proof of Theorem 5 uses related but different ideas from those for Theorem 4, and is provided in Appendix E. We have the following corollary.

Corollary 3. *If $f_j(x)$ is concave for every $j \in [m]$, then $SW(R^{opt}) \geq SW(A^{opt})$.*

Again, R^{opt} can be computed by a linear program, and when the conditions in Theorem 5 hold the ratio between $SW(R^{opt})$ and $SW(A^{opt})$ can be arbitrarily large. When the conditions do not hold, the relation between randomized allocations and stable allocations depend on the budget and the providers' costs, as shown in Appendix F. As a future direction, it would be interesting not only to characterize the conditions under which the optimal stable allocation does better, but also to quantify the ratio and/or difference between the social welfare of the two. Moreover, it is easy to see the FPTAS in Section 2.2 can be generalized to the setting of Section 3 with discrete consumers. Therefore whenever the optimal stable allocation is more preferable, one can approximate the optimal stable allocation efficiently. *Finally, the conditions in Theorem 5 can also be interpreted as monotone hazard rate in the corresponding single-player Bayesian allocation problem.*

[8] Again our approach works as long as each $f_j(x)$ is non-decreasing and piece-wise differentiable, but in such settings the analysis is unnecessarily complicated. Therefore we focus on the current setting.

References

[1] A. Abdulkadiroğlu, P. A. Pathak and A. E. Roth. The New York City High School Match. *American Economic Review, Papers and Proceedings*, 95:364-367, 2005.

[2] Z. Abrams, A. Ghosh, and E. Vee. Cost of conciseness in sponsored search auctions. in *WINE*, pp. 326-334, 2007.

[3] G. Aggarwal, S. Muthukrishnan, D. Pál and M. Pál. General auction mechanism for search advertising. In *Proc. 18th WWW*, pp. 241-250, 2009.

[4] S. Alaei, A. Malekian, and A. Srinivasan. On random sampling auctions for digital goods. In *EC*, pp. 187-196, 2009.

[5] A. Archer and É. Tardos. Frugal path mechanisms. *ACM Transactions on Algorithms*, 3(1), 2007.

[6] I. Ashlagi, S. Dobzinski and R. Lavi. An Optimal Lower Bound for Anonymous Scheduling Mechanisms. In *EC*, pp. 169-176, 2009.

[7] S. Athey and G. Ellison. Position Auctions with Consumer Search. *NBER Working Paper No. 15253*, 2009.

[8] X. Bei, N. Chen, N. Gravin and P. Lu. Budget Feasible Mechanism Design: From Prior-free to Bayesian. In *STOC*, pp. 449-458, 2012.

[9] L. Blumrosen, J. D. Hartline, and S. Nong. Position Auctions and Non-uniform Conversion Rates. In EC Workshop on Ad Auctions. 2008.

[10] A. Bogomolnaia and H. Moulin. A New Solution to the Random Assignment Problem. *Journal of Economic Theory*, 100: 295-328, 2001.

[11] M. Braverman, J. Chen and S. Kannan. Optimal Provision-after-wait in Healthcare. In *Proc. 5th ITCS*, pp. 541-542, 2014.

[12] P. Briest. Uniform budgets and the envy-free pricing problem. In *ICALP*, pp. 808-819, 2008.

[13] P. Briest, S. Chawla, R. Kleinberg, and S. M. Weinberg. Pricing randomized allocations. In *SODA*, pp. 585-597, 2010.

[14] E. Budish, Y. Che, F. Kojima and P. Milgrom. Designing Random Allocation Mechanisms: Theory and Applications. *American Economic Review*, 103(2): 585-623, 2013.

[15] I. Caragiannis. Efficient Coordination Mechanisms for Unrelated Machine Scheduling. In *SODA*, pp. 815-824, 2009.

[16] M. Cary, A. D. Flaxman, J. D. Hartline, and A. R. Karlin. Auctions for structured procurement. In *SODA*, pp. 304-313, 2008.

[17] H. Chan and J. Chen. Mechanism design for multi-unit procurements with budgets. In *WINE*, 2014.

[18] S. Chawla and J. D. Hardline. Auctions with Unique Equilibria. In *EC*, pp. 181-196, 2013.

[19] S. Chawla, J. D. Hartline, and R. D. Kleinberg. Algorithmic pricing via virtual valuations. In *EC*, pp. 243-251, 2007

[20] S. Chawla, D. L. Malec, and B. Sivan. The power of randomness in bayesian optimal mechanism design. In *EC*, pp. 149-158, 2010.

[21] X. Chen, I. Diakonikolas, D. Paparas, X. Sun, and M. Yannakakis. The complexity of optimal multidimensional pricing. In *SODA*, pp. 1319-1328, 2014.

[22] N. Chen, N. Gravin and P. Lu. On the Approximability of Budget Feasible Mechanisms. In *Proc. 22nd SODA*, pp. 685-699, 2011.

[23] G. Christodoulou and E. Koutsoupias. Mechanism Design for Scheduling. In *Bulletin of the European Association for Theoretical Computer Science (BEATCS)*, 97:3959, 2009.

[24] E. Cohen, M. Feldman, A. Fiat, H. Kaplan and S. Olonetsky. Envy-free Makespan Approximation. In *EC*, pp. 159-166, 2010.

[25] R. Cole, J. R. Correa, V. Gkatzelis, V. Mirrokni, and N. Olver. Inner Product Spaces for MinSum Coordination Mechanisms. In *STOC*, pp. 539-548, 2011.

[26] J. B. Cullen, B. A. Jacob, and S. Levitt. The Effect of School Choice on Participants: Evidence from Randomized Lotteries. *Econometric Society*, 74(5), pp 1191-1230, 2009.

[27] D. Dawson, H. Gravelle, R. Jacobs, S. Martin, and P. C. Smith. The effects of expanding patient choice of provider on waiting times: evidence from a policy experiment. *Health Economics*, 16, 113-128, 2007.

[28] N. R. Devanur, B. Q. Ha, and J. D. Hartline. Prior-free Auctions for Budgeted Agents. In *EC*, pp. 287-304, 2013.

[29] N. R. Devanur and J. D. Hartline. Limited and online supply and the bayesian foundations of prior-free mechanism design. In *EC*, pp. 41-50, 2009.

[30] S. Dobzinski, C. Papadimitriou and Y. Singer. Mechanisms for Complement-free Procurement. In *Proc. 12th EC*, pp. 273-282, 2011.

[31] L. E. Dubins and E. H. Spanier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1): 1-17, 1961.

[32] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*, Chp. 10, Cambridge University Press, 2010.

[33] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords. *American Economic Association*, 97(1):242-259, 2007.

[34] E. Elkind, A. Sahai and K. Steiglitz. Frugality in path auctions. In *SODA*, pp. 701-709, 2004.

[35] S. Felder. To wait or to pay for medical treatment? restraining ex-post moral hazard in health insurance. *Journal of Health Economics*, 27, 1418-1422, 2008.

[36] M. Feldman and J. Lai. Mechanisms and Impossibilities for Truthful, Envy-Free Allocations. *Algorithmic Game Theory, Lecture Notes in Computer Science*, pp. 120-131, 2012.

[37] D. K. Foley. Resource allocation and the public sector. *Yale Economic Studies*, 1967.

[38] H. Gravelle and L. Siciliani. Is waiting-time prioritisation welfare improving. *Health Economics*, 17, 167-184, 2008.

[39] H. Gravelle and L. Siciliani. Optimal quality, waits and cargoes in health insurance. *Journal of Health Economics*, 27, 663-674, 2008.

[40] H. Gravelle and L. Siciliani. Third degree waiting time discrimination: Optimal allocation of a public sector health care treatment under rationing by waiting. *Health Economics*, 18, 977-986, 2009.

[41] V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon and F. McSherry. On Profit-maximizing Envy-free Pricing. In *SODA*, pp. 1164-1173, 2005.

[42] B. Q. Ha and J. D. Hartline. Mechanism Design via Consensus Estimates, Cross Checking, and Profit Extraction. In *SODA*, pp. 887-895, 2012.

[43] J. D. Hartline and T. Roughgarden. Optimal mechanism design and money burning. In *Proc. 40th STOC*, pp. 75-84, 2008.

[44] J. D. Hartline and Q. Yan. Envy, Truth, and Profit. In *Proc. 12th EC*, pp. 243-252, 2011.

[45] A. R. Karlin, D. Kempe and T. Tamir. Beyond VCG: Frugality of truthful mechanisms. In *FOCS*, pp. 615-626, 2005.

[46] D. Kempe, A. Mu'Alem and M. Salek. Envy-Free Allocations for Budgeted Bidders. In *WINE*, pp. 537-544, 2009.

[47] O. Kesten and M. U. Ünever. A Theory of School-Choice Lotteries. Boston College Working Papers in Economics 737, Boston College Department of Economics, 2012.

[48] S. G. Kolliopoulos and G. Steiner. Partially ordered knapsack and applications to scheduling. *Discrete Applied Mathematics*, 155(8): 889-897, 2007.

[49] R. Lavi and C. Swamy. Truthful Mechanism Design for Multi-Dimensional Scheduling via Cycle Monotonicity. In *Proc. 8th EC*, pp. 252-261, 2007.

[50] R. P. Leme and É. Tardos. Pure and Bayes-Nash Price of Anarchy for Generalized Second Price Auction. In *FOCS*, pp. 735-744, 2010.

[51] J. D. Leshno. Dynamic matching in overloaded systems. Working paper.

[52] A. M. Manelli and D. R. Vincent. Bundling as an optimal selling mechanism for a multiple-good monopolist. *Journal of Economic Theory*, 127(1):1-35, 2006.

[53] E. Maskin and G. Feiwel. On the Fair Allocation of Indivisible Goods. In *Arrow and the Foundations of the Theory of Economic Policy (essays in honor of Kenneth Arrow)*, pp. 341-349. MacMillan, 1987.

[54] H. Moulin. *Fair Division and Collective Welfare*. MIT Press, 2004.

[55] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166-196, 2001.

[56] R. Porter. Mechanism design for online real-time scheduling. In *Proc. 5th EC*, pp. 61-70, 2004.

[57] L. S. Shapley and M. Shubik. The assignment game I: the core. *International Journal of Game Theory*, 1(2): 111-130, 1972.

[58] L. Siciliani and J. Hurst. Tackling excessive waiting times for elective surgery: a comparative analysis of policies in 12 oecd countries. *Health policy*, 72, 201-215, 2005.

[59] Y. Singer. Budget Feasible Mechanisms. In *Proc. 51st FOCS*, pp. 765-774, 2010.

[60] L. Svensson. On the existence of fair allocations. *Journal of Economics*, 43(3):301-308, 1983.

[61] K. Talwar. The price of truth: Frugality in truthful mechanisms. In *STACS*, pp. 608-619, 2003.

[62] J. Thanassoulis. Haggling over substitutes. *Jornal of Economic Theory*, 117:217-245, 2004.

[63] H. R. Varian. Position auctions. *International Journal of Industrial Organization*, 25(6):1163-1178, 2007.

A Proof of Theorem 1

We need the following two claims and an extra lemma to prove Theorem 1.

Claim 1. *For any stable allocation $A = (a, w)$ and consumers i and i' with $v_i > v_{i'}$, we have $q_{a(i)} \geq q_{a(i')}$.*

Proof. By the definition of stable allocations, we have

$$v_i q_{a(i)} - w_{a(i)} \geq v_i q_{a(i')} - w_{a(i')} \quad \text{and} \quad v_{i'} q_{a(i')} - w_{a(i')} \geq v_{i'} q_{a(i)} - w_{a(i)}.$$

Adding the two inequalities side by side and rearranging terms, we have $(v_i - v_{i'})(q_{a(i)} - q_{a(i')}) \geq 0$. Since $v_i > v_{i'}$, we have $q_{a(i)} \geq q_{a(i')}$ as desired. □

Claim 2. *For any stable allocation $A = (a, w)$ and consumers i and i' with $q_{a(i)} \geq q_{a(i')}$, we have $w_{a(i)} \geq w_{a(i')}$.*

Proof. Again by the definition of stable allocations we have $v_{i'} q_{a(i')} - w_{a(i')} \geq v_{i'} q_{a(i)} - w_{a(i)}$, which together with $q_{a(i)} \geq q_{a(i')}$ implies $w_{a(i)} - w_{a(i')} \geq v_{i'}(q_{a(i)} - q_{a(i')}) \geq 0$ —that is, $w_{a(i)} \geq w_{a(i')}$. □

The lemma below shows that without loss of generality we can focus on stable allocations with ordered allocation functions.

Lemma 4. *Given any stable allocation $A = (a, w)$, in polynomial time it can be modified so that: a is ordered, A is still stable, and the total cost and the utility of each consumer remain the same.*

Proof. Assume $a(i) > a(i + 1)$ for some $i \in [n]$. By Inequality 1 we have $q_{a(i)} \leq q_{a(i+1)}$ and $v_i \geq v_{i+1}$. If $v_i = v_{i+1}$ then

$$v_i q_{a(i)} - w_{a(i)} = v_{i+1} q_{a(i)} - w_{a(i)} \leq v_{i+1} q_{a(i+1)} - w_{a(i+1)} = v_i q_{a(i+1)} - w_{a(i+1)} \leq v_i q_{a(i)} - w_{a(i)},$$

where the inequalities are by the definition of stable allocations. Since the left end is the same as the right end, both inequalities above must be equalities. In particular, consumer i has the same utility at $a(i)$ and $a(i+1)$, and so does consumer $i+1$. Thus we can switch their assigned providers and the resulting A is stable, has the same total cost as before, the utility of each consumer remains the same, and yet $a(i) \leq a(i+1)$.

If $v_i > v_{i+1}$ then by Claim 1 it must be $q_{a(i)} = q_{a(i+1)}$. By the definition of stable allocations we have

$$v_i q_{a(i)} - w_{a(i)} \geq v_i q_{a(i+1)} - w_{a(i+1)} = v_i q_{a(i)} - w_{a(i+1)}$$

and

$$v_{i+1} q_{a(i+1)} - w_{a(i+1)} \geq v_{i+1} q_{a(i)} - w_{a(i)} = v_{i+1} q_{a(i+1)} - w_{a(i)},$$

which together imply $w_{a(i)} = w_{a(i+1)}$. Accordingly,

$$v_i q_{a(i)} - w_{a(i)} = v_i q_{a(i+1)} - w_{a(i+1)} \quad \text{and} \quad v_{i+1} q_{a(i+1)} - w_{a(i+1)} = v_{i+1} q_{a(i)} - w_{a(i)}.$$

Thus again we can switch i and $i+1$'s assigned providers and the resulting A is stable, has the same total cost as before, the utility of each consumer remains the same, and yet $a(i) \leq a(i+1)$.

With the discussion above, we have the following Algorithm 1 to make a ordered.

Algorithm 1: Making a Ordered

for k *from* n *to* 2 **do**
    $i = \max\{i' : i' \leq k, a(i') = \max_{j \leq k} a(j)\}$;
    while $i \neq k$ **do**
        $tmp = a(i)$, $a(i) = a(i+1)$, $a(i+1) = tmp$;
        $i = i + 1$;
    end while
end for

In Algorithm 1, in each loop k, variable i is initialized to be the largest consumer in $\{1, \ldots, k\}$ with the largest provider. If $i \neq k$, then $i+1 \leq k$ and $a(i) > a(i+1)$. As we have discussed, by switching $a(i)$ and $a(i+1)$, the resulting A is still stable, has the same total cost as before, and the utility of each consumer remains the same, yet the desired largest consumer becomes $i+1$. When $i = k$, we have $a(k) \geq a(i)$ for any $i \leq k$. Notice that loop k does not change $a(k+1), \ldots, a(n)$. Thus when loop k is finished we further have $a(k) \leq a(k+1) \leq \cdots \leq a(n)$. Accordingly, at the end of Algorithm 1 we have that: a is ordered, A is still stable, and the total cost and the utility of each consumer remain the same. This algorithm clearly runs in polynomial time, and Lemma 4 holds. $\square$

Lemma 1 (restated)**.** *For any ordered allocation function a, let $A = (a, w)$ be an allocation such that, A is tight at a and $w_j = v_1 q_1$ for any $j \notin a(\{1, \ldots, n\})$. Then A is stable.*

Proof. We start by showing that for any $i \in [n]$, provider $a(i)$ maximizes consumer i's utility given the waiting times. We first compare i's utility at $a(i)$ and his utility at $a(i')$ for any $i' \neq i$.

For any $i' < i$, since A is tight at a, we have $w_{a(i')} = (q_{a(i')} - q_{a(i'+1)}) v_{i'+1} + w_{a(i'+1)}$, that is,

$$q_{a(i')} v_{i'+1} - w_{a(i')} = q_{a(i'+1)} v_{i'+1} - w_{a(i'+1)}. \tag{5}$$

Since $i' + 1 \leq i$, by Inequality 1 we have $v_{i'+1} \geq v_i$. Since $q_{a(i')} \geq q_{a(i'+1)}$ by Inequality 2,

$$q_{a(i')}(v_{i'+1} - v_i) \geq q_{a(i'+1)}(v_{i'+1} - v_i). \tag{6}$$

Subtracting corresponding sides of Inequality 6 from those of Equation 5, we have

$$q_{a(i')} v_i - w_{a(i')} \leq q_{a(i'+1)} v_i - w_{a(i'+1)}.$$

Since this holds for all $i' < i$, we have

$$q_{a(1)}v_i - w_{a(1)} \leq q_{a(2)}v_i - w_{a(2)} \leq \cdots \leq q_{a(i-1)}v_i - w_{a(i-1)} = q_{a(i)}v_i - w_{a(i)},$$

where the equality is by Equation 5 with $i' = i - 1$. Thus i's utility at any $a(i')$ with $i' < i$ is no larger than his utility at $a(i)$.

Similarly, for any $i' > i$ we have $v_{i'} \leq v_i$ and

$$q_{a(i'-1)}v_{i'} - w_{a(i'-1)} = q_{a(i')}v_{i'} - w_{a(i')}. \tag{7}$$

Since $q_{a(i'-1)} \geq q_{a(i')}$,

$$q_{a(i'-1)}(v_{i'} - v_i) \leq q_{a(i')}(v_{i'} - v_i). \tag{8}$$

Subtracting corresponding sides of Inequality 8 from Equation 7, we have

$$q_{a(i'-1)}v_i - w_{a(i'-1)} \geq q_{a(i')}v_i - w_{a(i')}.$$

Since this holds for all $i' > i$, we have

$$q_{a(i)}v_i - w_{a(i)} \geq q_{a(i+1)}v_i - w_{a(i+1)} \geq \cdots \geq q_{a(n)}v_i - w_{a(n)}. \tag{9}$$

Thus i's utility at any $a(i')$ with $i' > i$ is no larger than his utility at $a(i)$.

It remains to show that for any provider $j \notin a(\{1, \ldots, n\})$, $q_{a(i)}v_i - w_{a(i)} \geq q_j v_i - w_j$. Since $v_1 \geq v_i$, $q_1 \geq q_j$, and $w_j = v_1 q_1$ by the construction of A, we have

$$q_j v_i - w_j = q_j v_i - q_1 v_1 \leq 0.$$

Since $v_i \geq v_n$ and $w_{a(n)} = 0$ by Definition 4, we have $q_{a(n)}v_i - w_{a(n)} \geq q_{a(n)}v_n \geq 0$. Thus Inequality 9 further implies

$$q_{a(i)}v_i - w_{a(i)} \geq 0 \geq q_j v_i - w_j,$$

as desired. Thus provider $a(i)$ maximizes i's utility given the waiting times. Notice that the inequality above also implies that consumer i's utility at $a(i)$ is non-negative.

In sum, A is stable and Lemma 1 holds. □

Lemma 2 (restated)**.** *For any ordered allocation function a and any stable allocation $A = (a, w)$, A is optimal with respect to a if and only if it is tight at a.*

Proof. We start by proving the "only if" part, and let $A = (a, w)$ be a stable allocation optimal with respect to a.

First, assume for the sake of contradiction that $w_{a(n)} > 0$. For any $j \in [m]$ such that there exists $i \in [n]$ with $a(i) = j$, let $w'_j = w_j - w_{a(n)}$, which is non-negative by the second part of Inequality 2. For any other j, let $w'_j = w_j$. It is easy to see that the allocation $A' = (a, w')$ is still stable: for each consumer i, the waiting time of $a(i)$ decreases by $w_{a(n)}$ and the waiting time of any other provider either decreases by the same amount or remains unchanged, thus $a(i)$ still maximizes i's utility under A'. Yet

$$\begin{aligned} SW(A') &= \sum_{i \in [n]} q_{a(i)}v_i - w'_{a(i)} = \sum_{i \in [n]} \left(q_{a(i)}v_i - w_{a(i)} + w_{a(n)}\right) \\ &= SW(A) + n w_{a(n)} > SW(A), \end{aligned}$$

contradicting the optimality of A. Thus $w_{a(n)} = 0$ as desired.

Second, notice that if $w_{a(i)} < (q_{a(i)} - q_{a(i+1)})v_{i+1} + w_{a(i+1)}$ for some $i < n$, then

$$q_{a(i+1)}v_{i+1} - w_{a(i+1)} < q_{a(i)}v_{i+1} - w_{a(i)},$$

and $a(i+1)$ does not maximize consumer $i+1$'s utility, contradicting the fact that A is stable. Thus for any $i < n$

$$w_{a(i)} \geq (q_{a(i)} - q_{a(i+1)})v_{i+1} + w_{a(i+1)}.$$

To prove that the two sides of the above inequality are actually equal, assume for the sake of contradiction that for some $i < n$

$$w_{a(i)} > (q_{a(i)} - q_{a(i+1)})v_{i+1} + w_{a(i+1)}.$$

Letting $\delta = w_{a(i)} - (q_{a(i)} - q_{a(i+1)})v_{i+1} - w_{a(i+1)}$, we have $\delta > 0$. Since $q_{a(i)} \geq q_{a(i+1)}$ by Inequality 2, we have $\delta \leq w_{a(i)}$.

For any provider $j \in a(\{1, 2, \ldots, i\})$, letting $w'_j = w_j - \delta$, we have $0 \leq w'_j < w_j$. For any other j, let $w'_j = w_j$. Letting $A' = (a, w')$, we show that A' is stable.

To begin with, for any consumer $i' \leq i$ and provider $j \in [m]$,

$$q_{a(i')}v_{i'} - w'_{a(i')} = q_{a(i')}v_{i'} - w_{a(i')} + \delta \geq q_j v_{i'} - w_j + \delta \geq q_j v_{i'} - w'_j, \tag{10}$$

where the first inequality is because A is stable and the second is because $w'_j \geq w_j - \delta$. Accordingly, the utility of i' is maximized at $a(i')$ in A'.

Now arbitrarily fix a consumer $i' \geq i+1$. For any provider $j \notin a(\{1, \ldots, i\})$, we have

$$q_{a(i')}v_{i'} - w'_{a(i')} \geq q_{a(i')}v_{i'} - w_{a(i')} \geq q_j v_{i'} - w_j = q_j v_{i'} - w'_j, \tag{11}$$

where the first inequality is because $w'_{a(i')} \leq w_{a(i')}$ and the equality is because $w'_j = w_j$.

It is left to consider i''s utility in A' at an arbitrary provider $j \in a(\{1, \ldots, i\})$. Applying Inequality 10 to consumer i, we have

$$q_j v_i - w'_j \leq q_{a(i)}v_i - w'_{a(i)}. \tag{12}$$

Since $j = a(i'')$ for some $i'' \leq i$, by Inequality 2 we have $q_j \geq q_{a(i)}$. Since $i < i'$, by Inequality 1 we have $v_i \geq v_{i'}$. Thus

$$q_j(v_i - v_{i'}) \geq q_{a(i)}(v_i - v_{i'}). \tag{13}$$

Subtracting corresponding sides of Inequality 13 from those of Inequality 12, we have

$$q_j v_{i'} - w'_j \leq q_{a(i)}v_{i'} - w'_{a(i)}, \tag{14}$$

that is, i''s utility at j is no larger than his utility at $a(i)$ in A'. We shall show that i''s utility at $a(i)$ is no larger than his utility at $a(i')$ in A'. To do so, by definition we have

$$w'_{a(i)} = w_{a(i)} - \delta = (q_{a(i)} - q_{a(i+1)})v_{i+1} + w_{a(i+1)},$$

namely,

$$q_{a(i)}v_{i+1} - w'_{a(i)} = q_{a(i+1)}v_{i+1} - w_{a(i+1)}. \tag{15}$$

By the hypothesis,

$$q_{a(i+1)}v_{i+1} - w_{a(i+1)} > q_{a(i)}v_{i+1} - w_{a(i)},$$

implying $a(i+1) \neq a(i)$. Since a is ordered, it must be $a(i+1) > a(i)$. Again since a is ordered, $a(\{1,\dots,i\}) \subseteq \{1,\dots,a(i)\}$. Thus

$$a(i+1) \notin a(\{1,\dots,i\}) \quad \text{and} \quad w'_{a(i+1)} = w_{a(i+1)}, \tag{16}$$

where the second part together with Equation 15 implies

$$q_{a(i)}v_{i+1} - w'_{a(i)} = q_{a(i+1)}v_{i+1} - w'_{a(i+1)}. \tag{17}$$

Since $i' \geq i+1$, by Inequality 1 we have $v_{i+1} \geq v_{i'}$. By Inequality 2 we have $q_{a(i)} \geq q_{a(i+1)}$. Thus

$$q_{a(i)}(v_{i+1} - v_{i'}) \geq q_{a(i+1)}(v_{i+1} - v_{i'}). \tag{18}$$

Subtracting corresponding sides of Inequality 18 from those of Equation 17, we have

$$q_{a(i)}v_{i'} - w'_{a(i)} \leq q_{a(i+1)}v_{i'} - w'_{a(i+1)}.$$

Since $a(i+1) \notin a(\{1,\dots,i\})$ according to the first part of Statement 16, by Inequality 11 with $j = a(i+1)$ we have

$$q_{a(i')}v_{i'} - w'_{a(i')} \geq q_{a(i+1)}v_{i'} - w'_{a(i+1)}.$$

Combining the two inequalities above, we have

$$q_{a(i')}v_{i'} - w'_{a(i')} \geq q_{a(i)}v_{i'} - w'_{a(i)}, \tag{19}$$

that is, i''s utility at $a(i)$ is no larger than his utility at $a(i')$ in A'.

Combining Inequalities 14 and 19, we have that for any $j \in a(\{1,\dots,i\})$,

$$q_{a(i')}v_{i'} - w'_{a(i')} \geq q_j v_{i'} - w'_j. \tag{20}$$

Combining Inequalities 10, 11 and 20, we have that A' is stable.

However,

$$\begin{aligned} SW(A') &= \sum_{i' \in [n]} q_{a(i')}v_{i'} - w'_{a(i')} \\ &= \sum_{i' \leq i} \left[q_{a(i')}v_{i'} - w_{a(i')} + \delta\right] + \sum_{i' \geq i+1} \left[q_{a(i')}v_{i'} - w_{a(i')}\right] \\ &\geq SW(A) + \delta > SW(A), \end{aligned}$$

contradicting the fact that A is optimal with respect to a. Therefore the hypothesis is false, and $w_{a(i)} = (q_{a(i)} - q_{a(i+1)})v_{i+1} + w_{a(i+1)}$ for any $i < n$, implying the "only if" part.

To prove the "if" part, notice that for any consumer i, the tightness of A at a has uniquely pinned down $w_{a(i)}$ and thus the utility of i at $a(i)$. Accordingly, all allocations (perhaps not even stable) that have allocation function a and are tight at a have the same social welfare. Thus any such allocation that is stable must be optimal with respect to a, as desired.

In sum, Lemma 2 holds. □

Lemma 3 (restated)**.** *For any ordered allocation function a and any stable allocation $A = (a, w)$ optimal with respect to a, $SW(A) = \sum_{i<n} i \cdot q_{a(i)} \cdot (v_i - v_{i+1}) + n \cdot q_{a(n)} \cdot v_n$.*

Proof. For any $k < n$, let $U_k = \sum_{i=k}^{n-1}\left[q_{a(i)}v_i - w_{a(i)}\right]$. By Lemma 2, A is tight at a. Thus by Definition 4 we have

$$\begin{aligned} U_{n-1} &= q_{a(n-1)}v_{n-1} - w_{a(n-1)} = q_{a(n-1)}v_{n-1} - (q_{a(n-1)} - q_{a(n)})v_n - w_{a(n)} \\ &= q_{a(n)}v_n + q_{a(n-1)}(v_{n-1} - v_n), \end{aligned}$$

and for any $k < n-1$,

$$\begin{aligned} U_k &= \sum_{i=k}^{n-1} q_{a(i)}v_i - \sum_{i=k}^{n-1} w_{a(i)} = \sum_{i=k}^{n-1} q_{a(i)}v_i - \sum_{i=k}^{n-1}\left[(q_{a(i)} - q_{a(i+1)})v_{i+1} + w_{a(i+1)}\right] \\ &= \sum_{i=k}^{n-1} q_{a(i)}(v_i - v_{i+1}) + \sum_{i=k+1}^{n} q_{a(i)}v_i - \sum_{i=k+1}^{n} w_{a(i)} \\ &= q_{a(n)}v_n + \sum_{i=k}^{n-1} q_{a(i)}(v_i - v_{i+1}) + \sum_{i=k+1}^{n-1} q_{a(i)}v_i - \sum_{i=k+1}^{n-1} w_{a(i)} \\ &= q_{a(n)}v_n + \sum_{i=k}^{n-1} q_{a(i)}(v_i - v_{i+1}) + U_{k+1}. \end{aligned}$$

Thus

$$\begin{aligned} SW(A) &= \sum_{i\in[n]} q_{a(i)}v_i - w_{a(i)} = q_{a(n)}v_n + U_1 = \cdots \\ &= nq_{a(n)}v_n + \sum_{k=1}^{n-1}\sum_{i=k}^{n-1} q_{a(i)}(v_i - v_{i+1}) = nq_{a(n)}v_n + \sum_{i=1}^{n-1}\sum_{k=1}^{i} q_{a(i)}(v_i - v_{i+1}) \\ &= \sum_{i=1}^{n-1} i \cdot q_{a(i)} \cdot (v_i - v_{i+1}) + n \cdot q_{a(n)} \cdot v_n, \end{aligned}$$

and Lemma 3 holds. □

Now we are ready to proof Theorem 1.

Proof of Theorem 1. Consider the decision version of the resource allocation problem:

$$\begin{aligned} DRA \;=\; &\{(q_1, \ldots, q_m, c_1, \ldots, c_m, v_1, \ldots, v_n, B, V) : \\ &\quad \text{there exists a stable budget-feasible allocation } A \text{ s.t. } SW(A) \geq V\}. \end{aligned}$$

It is clear that if one can find an optimal stable allocation for every instance of the resource allocation problem then one can decide DRA. We shall show that DRA is NP-complete by a reduction from the Subset-Sum problem:

$$SubsetSum = \left\{(s_1, \ldots, s_n, T) : \text{ there exists } S \subseteq [n] \text{ s.t. } \sum_{i\in S} s_i = T\right\}.$$

Given an instance $\alpha = (s_1, \ldots, s_n, T)$ of $SubsetSum$, we assume without loss of generality that $s_1 \geq s_2 \geq \cdots \geq s_n$, and construct an instance $\gamma = (q_1, \ldots, q_m, c_1, \ldots, c_m, v_1, \ldots, v_n, B, V)$ of DRA as follows. Notice that we use the same symbol for both a variable and its binary representation, and the 1st bit refers to the rightmost bit.

- There are $m = 2n$ providers and n consumers.
- For each $i \in [n]$, $q_i = c_i = s_i \cdot 2^{n(\lceil \log n \rceil + 1)} + 2^{(n-i)(\lceil \log n \rceil + 1)}$. That is, q_i and c_i are obtained by appending $n(\lceil \log n \rceil + 1)$ bits of 0's to the right of the binary representation of s_i, and then set the $(n-i)(\lceil \log n \rceil + 1) + 1$st bit to 1.
- For each $i \in [n]$, $q_{n+i} = c_{n+i} = 2^{(n-i)(\lceil \log n \rceil + 1)}$. That is, q_{n+i} and c_{n+i} consist of one bit of 1 followed by $(n-i)(\lceil \log n \rceil + 1)$ bits of 0's. Notice that the unique bit of 1 in q_{n+i} and c_{n+i} is aligned with the unique bit of 1 after s_i in q_i and c_i.
- $B = V = T \cdot 2^{n(\lceil \log n \rceil + 1)} + \sum_{i \in [n]} 2^{(n-i)(\lceil \log n \rceil + 1)}$. That is, B and V are obtained by appending $n(\lceil \log n \rceil + 1)$ bits of 0's to the right of the binary representation of T, and then set the $(n-i)(\lceil \log n \rceil + 1) + 1$st bit to 1 for each $i \in [n]$.
- For each $i \in [n]$, $v_i = \sum_{k=i}^{n} \frac{1}{k}$.

It is easy to see that the construction takes polynomial time and that γ satisfies Inequality 1. We have the following two lemmas.

Lemma 5. *$\gamma \in DRA \Rightarrow \alpha \in SubsetSum$.*

Proof. Let $A = (a, w)$ be an optimal stable allocation of γ. By definition, A is optimal with respect to a. By Claim 1 we assume without loss of generality that a is ordered. Thus by Lemma 3 we have

$$\begin{aligned} SW(A) &= \sum_{i<n} i \cdot q_{a(i)} \cdot (v_i - v_{i+1}) + n \cdot q_{a(n)} \cdot v_n = \sum_{i<n} i \cdot q_{a(i)} \cdot \frac{1}{i} + n \cdot q_{a(n)} \cdot \frac{1}{n} \\ &= \sum_{i \in [n]} q_{a(i)}. \end{aligned}$$

Since $q_j = c_j$ for any $j \in [m]$ and since A is budget-feasible,

$$SW(A) = \sum_{i \in [n]} c_{a(i)} = C(A) \le B = V.$$

Since $\gamma \in DRA$, we have $SW(A) \ge V$ and thus

$$SW(A) = V.$$

In particular, for any $j \in [n]$, $SW(A)$ has a 1 at the $(n-j)(\lceil \log n \rceil + 1) + 1$st bit preceded by $\lceil \log n \rceil$ bits of 0's. We now show that for any $j \in [n]$,

$$|\{i \in [n] : a(i) \in \{j, n+j\}\}| = 1, \tag{21}$$

that is, there is exactly one consumer assigned to either provider j or provider $n + j$.

To see why Equation 21 is true, notice that for any $k \in [n]$ there are $\lceil \log n \rceil$ bits of 0's between the $(n-k+1)(\lceil \log n \rceil + 1) + 1$st bit and the $(n-k)(\lceil \log n \rceil + 1) + 1$st bit in the binary representation of any q_j. Since there are n consumers, there is no carry to the $(n-k+1)(\lceil \log n \rceil + 1) + 1$st bit when computing $SW(A)$. Further notice that the only providers whose qualities contribute a 1 to the $(n-j)(\lceil \log n \rceil + 1) + 1$st bit of $SW(A)$ are providers j and $n + j$.

If more than one consumers are assigned to either j or $n + j$, then the $\lceil \log n \rceil$ bits preceding the $(n-j)(\lceil \log n \rceil + 1) + 1$st bit of $SW(A)$ cannot be all 0's, and $SW(A) \ne V$. If no consumer is assigned to either j or $n+j$, then the $(n-j)(\lceil \log n \rceil + 1) + 1$st bit of $SW(A)$ cannot be a 1, and again $SW(A) \neq V$. Thus there must be exactly one consumer assigned to either provider j or provider $n+j$, and Equation 21 holds.

By Equation 21, the two sets $S = \{j \in [n] : j \in a(\{1,\ldots,n\})\}$ and $S' = \{j \in [n] : n+j \in a(\{1,\ldots,n\})\}$ form a partition of $[n]$, and

$$\begin{aligned} SW(A) &= \sum_{i\in[n]} q_{a(i)} = \sum_{j\in S} q_j + \sum_{j\in S'} q_{n+j} \\ &= \sum_{j\in S}\left[s_j \cdot 2^{n(\lceil \log n\rceil+1)} + 2^{(n-j)(\lceil \log n\rceil+1)}\right] + \sum_{j\in S'} 2^{(n-j)(\lceil \log n\rceil+1)} \\ &= \sum_{j\in S} s_j \cdot 2^{n(\lceil \log n\rceil+1)} + \sum_{j\in[n]} 2^{(n-j)(\lceil \log n\rceil+1)}. \end{aligned}$$

Since $SW(A) = V = T \cdot 2^{n(\lceil \log n\rceil+1)} + \sum_{i\in[n]} 2^{(n-i)(\lceil \log n\rceil+1)}$, we have $\sum_{j\in S} s_j = T$. Thus $\alpha \in \mathit{SubsetSum}$ and Lemma 5 holds. □

Lemma 6. *$\alpha \in \mathit{SubsetSum} \Rightarrow \gamma \in \mathit{DRA}$.*

Proof. Since $\alpha \in \mathit{SubsetSum}$, there exists $S \subseteq [n]$ such that

$$\sum_{i\in S} s_i = T. \tag{22}$$

Let $k = |S|$ and $S = \{j_1,\ldots,j_k\}$, with $j_1 \leq j_2 \leq \cdots \leq j_k$. Further, let $S' = [n] \setminus S = \{j_{k+1},\ldots,j_n\}$, with $j_{k+1} \leq j_{k+2} \leq \cdots \leq j_n$. We construct an allocation $A = (a, w)$ as follows.

- $a(i) = j_i$ for any $i \leq k$, and $a(i) = n + j_i$ for any $i \geq k+1$.
- $w_{a(n)} = 0$, $w_{a(i)} = (q_{a(i)} - q_{a(i+1)})v_{i+1} + w_{a(i+1)}$ for any $i < n$, and $w_j = v_1 q_1$ for any $j \notin a(\{1,\ldots,n\})$.

Notice that

$$a(1) \leq a(2) \leq \cdots \leq a(n) = n + j_n \leq m. \tag{23}$$

Thus a is a well defined function from $[n]$ to $[m]$ and is ordered. Also notice that A is tight at a.

The cost of A is

$$\begin{aligned} C(A) &= \sum_{i\in[n]} c_{a(i)} = \sum_{i\leq k} c_{j_i} + \sum_{i\geq k+1} c_{n+j_i} \\ &= \sum_{i\leq k} s_{j_i} \cdot 2^{n(\lceil \log n\rceil+1)} + 2^{(n-j_i)(\lceil \log n\rceil+1)} + \sum_{i\geq k+1} 2^{(n-j_i)(\lceil \log n\rceil+1)} \\ &= \sum_{j\in S} s_j \cdot 2^{n(\lceil \log n\rceil+1)} + \sum_{j\in S} 2^{(n-j)(\lceil \log n\rceil+1)} + \sum_{j\in S'} 2^{(n-j)(\lceil \log n\rceil+1)} \\ &= T \cdot 2^{n(\lceil \log n\rceil+1)} + \sum_{j\in[n]} 2^{(n-j)(\lceil \log n\rceil+1)} = B, \end{aligned} \tag{24}$$

where the fifth equality is by Equation 22 and the last is by the definition of B. Thus A is budget-feasible.

Since A is tight at a, by Lemma 1, A is stable. By Lemma 2, A is optimal with respect to a. Thus

$$\begin{aligned} SW(A) &= \sum_{i=1}^{n-1} i \cdot q_{a(i)} \cdot (v_i - v_{i+1}) + n \cdot q_{a(n)} \cdot v_n \\ &= \sum_{i=1}^{n-1} i \cdot q_{a(i)} \cdot \frac{1}{i} + n \cdot q_{a(n)} \cdot \frac{1}{n} = \sum_{i\in[n]} q_{a(i)} = \sum_{i\in[n]} c_{a(i)} = C(A) = B = V, \end{aligned}$$

where the first equality is by Lemma 3, the sixth by Equation 24 and the others by the construction of γ. Therefore A is a stable budget-feasible allocation with $SW(A) \geq V$. Accordingly, $\gamma \in DRA$ and Lemma 6 holds. □

By Lemmas 5 and 6, $\alpha \in SubsetSum$ if and only if $\gamma \in DRA$. Therefore DRA is NP-complete and Theorem 1 holds. □

B Proof of Theorems 2 and 3

According to the discussion in Section 2.3, Theorem 2 depends on Theorem 3, therefore, we prove Theorem 3 first.

To prove Theorem 3, we first derive some lower/upper-bounds for $SW(a^{opt})$. For any player i, letting j_i be the smallest item that i can be assigned to in any budget-feasible allocation, we have

$$j_i = \min\{j \in [m] : ic_j + (n-i)c_m \leq B\}.$$

Indeed, if i is assigned to some item $j' < j_i$, then by definition the minimum cost of such allocations is achieved by assigning players $1, \ldots, i$ to item j' and all others to item m, leading to cost $ic_{j'} + (n-i)c_m > B$ by the definition of j_i. Notice that j_i is always well defined, as assigning all players to item m is budget-feasible.

For each $i \in [n]$, let a^i be the allocation which assigns players $1, \ldots, i$ to item j_i and all others to item m. We have that all the a^i's are budget-feasible and $a^i(1) \leq \cdots \leq a^i(n)$. Thus for each $i \in [n]$, by definition we have

$$SW(a^{opt}) \geq SW(a^i) \geq u_{ij_i}.$$

Also, by the definition of the j_i's we have $a^{opt}(i) \geq j_i$, and thus for any i,

$$u_{ia^{opt}(i)} \leq u_{ij_i}.$$

Accordingly, letting $V = \max_{i\in[n]} u_{ij_i}$, we have

$$nV \geq \sum_i u_{ij_i} \geq \sum_i u_{ia^{opt}(i)} = SW(a^{opt}) \geq V. \tag{25}$$

The following lemma shows the existence of a pseudo-polynomial time algorithm for ordered Knapsack.

Lemma 7. *There exists a dynamic program that runs in time $O((n+m)n^2mV)$ and computes an optimal allocation for the ordered Knapsack problem.*

Proof. For any allocation a and player i, let

$$SW(a,i) = \sum_{i'=i}^{n} u_{i'a(i')}$$

be the contribution of players $i,\ldots,n$ to $SW(a)$. For any $i \in [n]$, $j \in [m]$, and $s \in \{0,1,\ldots,nV\}$, we are interested in the *minimum cost*, denoted by $C(i)(j)(s)$, needed for players $i,\ldots,n$ to make contribution s to the social welfare, when player i is assigned to item j. More precisely, letting

$$SW(i,j) = \sum_{i'=i}^{n} u_{i'j}$$

be the contribution of players $i,\ldots,n$ when they are all assigned to j,

$$C(i)(j)(s) = \begin{cases} c_j + \min\limits_{\substack{a:j=a(i)\le a(i+1)\le\cdots\le a(n),\\ SW(a,i)\ge s}} \sum_{i'>i} c_{a(i')} & \text{if } SW(i,j) \ge s, \\ +\infty & \text{otherwise.} \end{cases} \tag{26}$$

Notice that $C(i)(j)(s) = +\infty$ means it is impossible for players $i,\ldots,n$ to make contribution s to the social welfare even if all of them are assigned to j, and thus impossible to make such contribution at j and providers after j. In practice, $+\infty$ can be replaced by $B+1$ (or any number larger than B and of polynomial length).

Also notice that, for any $s \le nV$, $\min_{j\in[m]} C(1)(j)(s)$ is the minimum cost of any allocation whose social welfare is at least s. Thus we immediately have the following claim whose proof is omitted.

Claim 3. *For any optimal allocation a,*

$$SW(a) = \max\{s : \min_{j\in[m]} C(1)(j)(s) \le B\}.$$

In order to compute the $C(i)(j)(s)$'s, we prove the following.

Claim 4. *$C(n)(j)(s) = c_j$ for any $j \in [m]$ and $s \le u_{nj}$; $C(i)(j)(0) = c_j + (n-i)c_m$ for any $i < n$ and $j \in [m]$; and for any $i < n$, $j \in [m]$ and $0 < s \le SW(i,j)$,*

$$C(i)(j)(s) = c_j + \min_{j'\ge j} C(i+1)(j')(\max\{s - u_{ij}, 0\}). \tag{27}$$

Finally, $C(i)(j)(s) = +\infty$ in all other cases.

Proof. We only prove Equation 27, since other equalities follow directly from the definition of the $C(i)(j)(s)$'s. Notice that for any allocation a with $a(i) = j$, $SW(a,i) \ge s$ if and only if $SW(a,i+1) \ge \max\{s - u_{ij}, 0\}$. For any $j' \ge j$, let

$$S_{j'} = \left\{a : j = a(i), j' = a(i+1) \le \cdots \le a(n), SW(a,i+1) \ge \max\{s - u_{ij}, 0\}\right\}.$$

We have

$$\{a : j = a(i) \le a(i+1) \le \cdots \le a(n), SW(a,i) \ge s\} = \cup_{j'\ge j} S_{j'}$$

and

$$C(i)(j)(s) = c_j + \min_{\substack{a:j=a(i)\le a(i+1)\le\cdots\le a(n),\\ SW(a,i)\ge s}} \sum_{i'>i} c_{a(i')} = c_j + \min_{j'\ge j} \min_{a\in S_{j'}} \left(c_{j'} + \sum_{i'>i+1} c_{a(i')}\right), \tag{28}$$

where $\min\limits_{a \in S_{j'}} \left(c_{j'} + \sum_{i'>i+1} c_{a(i')}\right)$ is defined to be $+\infty$ whenever $S_{j'} = \emptyset$. Notice that $S_{j'} = \emptyset$ implies $SW(i+1, j') < \max\{s - u_{ij}, 0\}$, and thus by Equation 26

$$C(i+1)(j')(\max\{s - u_{ij}, 0\}) = +\infty = \min_{a \in S_{j'}} \left(c_{j'} + \sum_{i'>i+1} c_{a(i')} \right). \tag{29}$$

Also notice that $S_{j'} \neq \emptyset$ for some j'. In fact, $s \leq SW(i, j)$ implies $SW(i+1, j) \geq \max\{s - u_{ij}, 0\}$, and thus $S_j \neq \emptyset$. For any $S_{j'} \neq \emptyset$, we have $SW(i+1, j') \geq \max\{s - u_{ij}, 0\}$, and

$$\begin{aligned} C(i+1)(j')(\max\{s - u_{ij}, 0\}) &= c_{j'} + \min_{\substack{a: j' = a(i+1) \leq \cdots \leq a(n), \\ SW(a, i+1) \geq \max\{s - u_{ij}, 0\}}} \sum_{i'>i+1} c_{a(i')} \\ &= \min_{a \in S_{j'}} \left(c_{j'} + \sum_{i'>i+1} c_{a(i')} \right), \end{aligned} \tag{30}$$

where the second equality is because that, given $a(i+1) = j' \geq j = a(i)$, neither $SW(a, i+1)$ nor $\sum_{i'>i+1} c_{a(i')}$ depends on $a(i)$.

Combining Equations 28, 29 and 30, we have

$$C(i)(j)(u) = c_j + \min_{j' \geq j} C(i+1)(j')(\max\{s - u_{ij}, 0\}),$$

and Claim 4 holds. □

Equation 27 immediately leads to a dynamic program computing all $C(i)(j)(s)$'s, with other equations in Claim 4 as initialization conditions. Since it takes $O(n)$ time to compute each $SW(i, j)$, by Claim 4 it takes $O(n + m)$ time to compute each $C(i)(j)(s)$ given the $C(i+1)(j')(s')$'s. Thus the dynamic program takes space $O(n^2 mV)$ and runs in time $O((n+m)n^2 mV)$. By Claim 3, given the $C(i)(j)(s)$'s, the social welfare of the optimal allocation can be computed in time $O(mnV)$.

Moreover, the dynamic program can keep track of the optimal j''s when computing the $C(i)(j)(s)$'s according to Equation 27. Once the $C(1)(j)(s)$ corresponding to the optimal social welfare is found, the dynamic program can trace back the stored j''s and recover the assigned item for each player, and thus compute the corresponding optimal allocation. The total space is still $O(n^2 mV)$ and the running time is still $O((n+m)n^2 mV)$.

We present the dynamic program in Algorithm 2, where for each $i < n$, $j \in [m]$ and $s \leq nV$, $\hat{a}(i)(j)(s)$ represents the item to which player $i+1$ is assigned to, in order for players $i, \ldots, n$ to make contribution s at cost $C(i)(j)(s)$. The correctness and the running time of Algorithm 2 follow from Claim 4 and the discussion above.

Accordingly, Lemma 7 holds. □

By scaling the players' values and running Algorithm 2 on the scaled input, we obtain an FPTAS for the ordered Knapsack problem, see below.

Theorem 3 (restated)**.** *There exists an algorithm for the ordered Knapsack problem such that, given any $\epsilon > 0$, it runs in time $O((n+m)n^3 m/\epsilon)$ and outputs an allocation a with $C(a) \leq B$ and $SW(a) \geq (1-\epsilon)SW(a^{opt})$.*

Proof. Given $c_1, \ldots, c_m$, $u_{11}, \ldots, u_{nm}$, B and $\epsilon > 0$, our algorithm oKnapsack works as follows. Let the j_i's and V be defined as before, $K = \frac{\epsilon V}{n}$, and $u'_{ij} = \lfloor \frac{u_{ij}}{K} \rfloor$ for any $i \in [n]$ and $j \in [m]$. Run Algorithm 2 with input $(c_1, \ldots, c_m, u'_{11}, \ldots, u'_{nm}, B)$ and return its output a.

Algorithm 2: A Dynamic Program for Ordered Knapsack

Input : cost c_j for each $j \in [m]$, value u_{ij} for each $i \in [n]$ and $j \in [m]$, and budget B.
Output: an optimal allocation a.

Initialization:
for i *from* 1 *to* n **do**
  $j_i = \min\{j \in [m] : ic_j + (n-i)c_m \le B\}$;
end for
$V = \max_{i \in [n]} u_{ij_i}$;
for j *from* 1 *to* m *and* s *from* 0 *to* nV **do**
  if $s \le u_{nj}$ **then**
    $C(n)(j)(s) = c_j$;
  else
    $C(n)(j)(s) = B + 1$;
  end if
end for
for i *from* 1 *to* $n-1$ *and* j *from* 1 *to* m **do**
  $C(i)(j)(0) = c_j + (n-i)c_m$; $\hat{a}(i)(j)(0) = m$;
end for

Compute $C(i)(j)(s)$ and $\hat{a}(i)(j)(s)$:
for i *from* $n-1$ *to* 1 **do**
  for j *from* 1 *to* m **do**
    $SW(i,j) = \sum_{i'=i}^{n} u_{i'j}$;
    for s *from* 1 *to* nV **do**
      if $s \le SW(i,j)$ **then**
        $\hat{j} = \text{argmin}_{j' \ge j}\, C(i+1)(j')(\max\{s - u_{ij}, 0\})$, with ties broken lexicographically;
        $C(i)(j)(s) = c_j + C(i+1)(\hat{j})(\max\{s - u_{ij}, 0\})$;
        $\hat{a}(i)(j)(s) = \hat{j}$;
      else
        $C(i)(j)(s) = B + 1$; (It doesn't matter what $\hat{a}(i)(j)(s)$ is in this case.)
      end if
    end for
  end for
end for

Compute a:
for s *from* nV *to* 0 **do**
  $\hat{j} = \text{argmin}_{j \in [m]}\, C(1)(j)(s)$, with ties broken lexicographically;
  if $C(1)(\hat{j})(s) \le B$ **then**
    $a(1) = \hat{j}$; **break**;
  end if
end for
for i *from* 1 *to* $n-1$ **do**
  $a(i+1) = \hat{a}(i)(\hat{j})(s)$; $s = \max\{s - u_{i\hat{j}}, 0\}$; $\hat{j} = a(i+1)$;
end for
return a

Since $u_{i1} \geq \cdots \geq u_{im}$ for any $i \in [n]$, we have $u'_{i1} \geq \cdots \geq u'_{im}$ for any i, and thus the input to Algorithm 2 is a valid instance of ordered Knapsack. Since the budget and the costs do not change, the j_i's computed on the scaled input are still the same as before. Let $V' = \max_{i\in[n]} u'_{ij_i}$ be the counterpart of V for the scaled input. We have $V' \leq \max_{i\in[n]} \frac{u_{ij_i}}{K} = \frac{V}{K} = \frac{n}{\epsilon}$. Thus Algorithm 2 runs in time $O((n+m)n^2mV') = O((n+m)n^3m/\epsilon)$, and so does the algorithm oKnapsack.

Below we analyze the approximation ratio. For any allocation a', let $SW(a')$ and $SW'(a')$ respectively be the social welfare of a' in the original ordered Knapsack problem and in the scaled input to Algorithm 2. We have

$$\begin{aligned} SW(a) &= \sum_{i\in[n]} u_{ia(i)} \geq K \sum_{i\in[n]} u'_{ia(i)} = K \cdot SW'(a) \geq K \cdot SW'(a^{opt}) = K \sum_{i\in[n]} u'_{ia^{opt}(i)} \\ &\geq K \sum_{i\in[n]} \Big(\frac{u_{ia^{opt}(i)}}{K} - 1\Big) = \sum_{i\in[n]} u_{ia^{opt}(i)} - nK = SW(a^{opt}) - \epsilon V \\ &\geq SW(a^{opt}) - \epsilon SW(a^{opt}) = (1-\epsilon) SW(a^{opt}), \end{aligned}$$

where the first and the third inequalities are by the definition of u'_{ij}'s, the second is because a is optimal under the scaled input, and the last is by Inequality 25.

In sum, Theorem 3 holds. □

Finally we can prove Theorem 2.

Proof of Theorem 2. It is easy to see that algorithm oKnapsack constructed in the proof of Theorem 3 can be applied to the resource allocation problem. Indeed, given an instance $\gamma = (q_1, \ldots, q_m, c_1, \ldots, c_m, v_1, \ldots, v_n, B)$ of resource allocation, for any $j \in [m]$, we can take

$$u_{ij} = iq_j(v_i - v_{i+1}) \text{ for any } i < n \text{ and } u_{nj} = nq_jv_n.$$

Then $\kappa = (c_1, \ldots, c_m, u_{11}, \ldots, u_{nm}, B)$ is a valid instance of ordered Knapsack. Moreover, any allocation a of κ is an ordered allocation function of γ with the same cost and the same social welfare, and vice versa. Thus a^{opt} is an optimal allocation function for γ and $SW(A^{opt}) = SW(a^{opt})$. By Theorem 3, the allocation a output by oKnapsack with input κ is budget-feasible and $SW(a) \geq (1-\epsilon)SW(a^{opt})$. Let $A = (a, w)$ be the allocation for γ that is tight at a and as defined by Lemma 1. We have that A is stable, budget-feasible, and optimal with respect to a, where the optimality follows from Lemma 2. Thus $SW(A) = SW(a) \geq (1-\epsilon)SW(a^{opt}) = (1-\epsilon)SW(A^{opt})$.

It takes $O(mn)$ time to construct κ from γ, and $O(n+m)$ time to construct A from a. Thus in total A can be computed in time $O(mn + (n+m)n^3m/\epsilon + n + m) = O((n+m)n^3m/\epsilon)$, and Theorem 2 holds. □

C Proof of Theorem 4

To prove Theorem 4, we start by showing several properties of lottery schemes. Recall that for any lottery scheme L, $\lambda^L(x) = (p_1^L(x), \ldots, p_m^L(x), w^L(x))$ denotes the lottery chosen by consumer x. The lemma below is the counterpart of Claim 1: the expected quality received by the consumers increases together with their values.

Lemma 8. *For any lottery scheme L, the function $\sum_{j\in[m]} p_j^L(x)q_j$ is non-decreasing.*

Proof. Let $x, x' \in [0,1]$ be such that $x < x'$. By Inequality 4,

$$u(x, \lambda^L(x)) \geq u(x, \lambda^L(x')) \quad \text{and} \quad u(x', \lambda^L(x')) \geq u(x', \lambda^L(x)).$$

That is,

$$\left(\sum_{j\in[m]} p_j^L(x) q_j v(x)\right) - w^L(x) \geq \left(\sum_{j\in[m]} p_j^L(x') q_j v(x)\right) - w^L(x') \tag{31}$$

and

$$\left(\sum_{j\in[m]} p_j^L(x') q_j v(x')\right) - w^L(x') \geq \left(\sum_{j\in[m]} p_j^L(x) q_j v(x')\right) - w^L(x). \tag{32}$$

Adding the two inequalities side by side and rearranging the terms, we have

$$\left[\sum_{j\in[m]} p_j^L(x) q_j - \sum_{j\in[m]} p_j^L(x') q_j\right] (v(x') - v(x)) \leq 0.$$

Since $v(x)$ is strictly increasing, we have $v(x) < v(x')$ and thus $\sum_{j\in[m]} p_j^L(x) q_j \leq \sum_{j\in[m]} p_j^L(x') q_j$. That is, the function $\sum_{j\in[m]} p_j^L(x) q_j$ is non-decreasing and Lemma 8 holds. □

Notice that the utility of consumer x depends on x only indirectly through $v(x)$, thus $\lambda^L(x)$ can be written as a vector of functions on $v(x)$: $\lambda^L(v(x)) = (p_1^L(v(x)), \ldots, p_m^L(v(x)), w^L(v(x)))$. We have the following.

Lemma 9. *For any lottery scheme L and any consumer $x \in [0,1]$,*

$$u(x, \lambda^L(x)) = \int_0^{v(x)} \sum_{j\in[m]} q_j p_j^L(\hat{v}) d\hat{v}.$$

Proof. Similar as before, by Inequalities 31 and 32 with $x' = x + \Delta$, we have

$$v(x) \left[\sum_{j\in[m]} q_j \left(p_j^L(x+\Delta) - p_j^L(x)\right)\right] \leq w^L(x+\Delta) - w^L(x)$$

and

$$v(x+\Delta) \left[\sum_{j\in[m]} q_j \left(p_j^L(x+\Delta) - p_j^L(x)\right)\right] \geq w^L(x+\Delta) - w^L(x).$$

Combining the two inequalities and letting $\Delta \to 0$, we have

$$v(x) \sum_{j\in[m]} q_j \cdot dp_j^L(x) = dw^L(x).$$

Accordingly,

$$\begin{aligned}
du(x, \lambda^L(x)) &= \left(\sum_{j\in[m]} q_j \cdot d\left(p_j^L(x) v(x)\right)\right) - dw^L(x) \\
&= \sum_{j\in[m]} q_j p_j^L(x) \cdot dv(x) + v(x) \sum_{j\in[m]} q_j \cdot dp_j^L(x) - v(x) \sum_{j\in[m]} q_j \cdot dp_j^L(x) \\
&= \sum_{j\in[m]} q_j p_j^L(x) \cdot dv(x) = \sum_{j\in[m]} q_j p_j^L(v(x)) v'(x) dx.
\end{aligned}$$

Integrating the first and the last terms with respect to x and changing variables, we have

$$u(x, \lambda^L(x)) = \int_0^x du(\hat{x}, \lambda^L(\hat{x})) = \int_0^x \sum_{j\in[m]} q_j p_j^L(v(\hat{x}))v'(\hat{x})d\hat{x} = \int_0^{v(x)} \sum_{j\in[m]} q_j p_j^L(\hat{v})d\hat{v}.$$

Thus Lemma 9 holds. $\square$

Now we are ready to prove our theorem.

Proof of Theorem 4. Since $SW(L^{opt}) \geq SW(R^{opt})$ by definition, it suffices to show

$$SW(R^{opt}) \geq SW(L^{opt}). \tag{33}$$

To do so, for any budget-feasible lottery scheme L, let $R = (p_1, \dots, p_m)$ be the randomized allocation where for any $j \in [m]$,

$$p_j = \int_0^1 p_j^L(x)dx.$$

That is, each p_j is the average of $p_j^L(x)$ over $[0, 1]$. It is easy to see $\sum_{j\in[m]} p_j = \int_0^1 \sum_{j\in[m]} p_j^L(x)dx \leq 1$, thus R is well defined. Also we have $C(R) = \sum_{j\in[m]} p_j c_j = \int_0^1 \sum_{j\in[m]} p_j^L(x)c_j dx = C(L) \leq B$, thus R is budget-feasible. Below we show

$$SW(R) \geq SW(L), \tag{34}$$

which, when applied to $L = L^{opt}$, implies Inequality 33.

Since $v(x)$ is strictly increasing and twice differentiable, $v^{-1}(\hat{v})$ is well defined for any $\hat{v} \in [0, v(1)]$. Thus we have

$$\begin{aligned}
SW(L) &= \int_0^1 u(x, \lambda^L(x))dx = \int_0^1 \int_0^{v(x)} \sum_{j\in[m]} q_j p_j^L(\hat{v})d\hat{v}dx = \int_0^{v(1)} \left(\sum_{j\in[m]} q_j p_j^L(\hat{v}) \int_{v^{-1}(\hat{v})}^1 dx \right) d\hat{v} \\
&= \int_0^{v(1)} \sum_{j\in[m]} q_j p_j^L(\hat{v})(1 - v^{-1}(\hat{v}))d\hat{v} = \int_0^1 \sum_{j\in[m]} q_j p_j^L(x)(1 - x)v'(x)dx, \qquad (35)
\end{aligned}$$

where the second equality is by Lemma 9, the last is by taking $\hat{v} = v(x)$, and all others are by definition or basic calculus. Moreover, letting $p = \sum_{j\in[m]} q_j p_j$, we have

$$\begin{aligned}
SW(R) &= \int_0^1 pv(x)dx = \int_0^1 \int_0^{v(x)} pd\hat{v}dx = \int_0^{v(1)} \int_{v^{-1}(\hat{v})}^1 pdxd\hat{v} = \int_0^{v(1)} p(1 - v^{-1}(\hat{v}))d\hat{v} \\
&= \int_0^1 p(1 - x)v'(x)dx. \qquad (36)
\end{aligned}$$

By Lemma 8, $\sum_{j\in[m]} q_j p_j^L(x)$ is non-decreasing. Thus

$$\sum_{j\in[m]} q_j p_j^L(0) = \int_0^1 \sum_{j\in[m]} q_j p_j^L(0)dx \leq \int_0^1 \sum_{j\in[m]} q_j p_j^L(x)dx = p \leq \int_0^1 \sum_{j\in[m]} q_j p_j^L(1)dx = \sum_{j\in[m]} q_j p_j^L(1),$$

and there exists $x_p \in [0, 1]$ such that

$$\sum_{j\in[m]} q_j p_j^L(x) \leq p \ \forall x \in [0, x_p) \quad \text{and} \quad \sum_{j\in[m]} q_j p_j^L(x) \geq p \ \forall x \in [x_p, 1].$$

Following Equations 35 and 36 we have

$$SW(R) - SW(L) = \int_0^1 \left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x)v'(x)dx$$

$$= \int_0^{x_p} \left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x)v'(x)dx + \int_{x_p}^1 \left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x)v'(x)dx. \quad (37)$$

The value of $\sum_{j\in[m]} q_j p_j^L(x_p)$ does not affect the value of the integration, and without loss of generality we assume it equals p.

For any $x \in [0, x_p]$, since $(1-x)v'(x)$ is non-increasing and $v'(x) > 0$, we have

$$(1-x)v'(x) \geq (1-x_p)v'(x_p) \geq 0.$$

Moreover, for any such x, since $p - \sum_{j\in[m]} q_j p_j^L(x) \geq 0$, we have

$$\left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x)v'(x) \geq \left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x_p)v'(x_p). \quad (38)$$

Similarly, for any $x \in [x_p, 1]$ we have $0 \leq (1-x)v'(x) \leq (1-x_p)v'(x_p)$ and $p - \sum_{j\in[m]} q_j p_j^L(x) \leq 0$, and thus

$$\left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x)v'(x) \geq \left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x_p)v'(x_p). \quad (39)$$

Combining Equation 37 with Inequalities 38 and 39, we have

$$\begin{aligned}
& SW(R) - SW(L) \\
\geq\ & \int_0^{x_p} \left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x_p)v'(x_p)dx + \int_{x_p}^1 \left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x_p)v'(x_p)dx \\
=\ & (1-x_p)v'(x_p) \int_0^1 \left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) dx = (1-x_p)v'(x_p) \left[p - \int_0^1 \sum_{j\in[m]} q_j p_j^L(x) dx \right] \\
=\ & (1-x_p)v'(x_p)(p-p) = 0,
\end{aligned}$$

implying Inequality 34. Thus Theorem 4 holds. □

D Important Properties of the Optimal Randomized Allocation

In this section we highlight several important properties of the optimal randomized allocation compared with general lottery schemes.

Computability. In general resource allocation problems, there may not be an efficient algorithm for finding an optimal lottery scheme. But the optimal randomized allocation is always defined by the following linear program.

$$\begin{aligned}
&\max_{p_1,\dots,p_m} \sum_{j\in[m]} p_j q_j \int_0^1 v(x)dx \\
\text{s.t.}\quad & p_j \geq 0 \ \forall j \in [m], \\
& \sum_{j\in[m]} p_j \leq 1, \\
& \sum_{j\in[m]} p_j c_j \leq B.
\end{aligned}$$

If $\int_0^1 v(x)dx$ has a closed form and can be computed in polynomial time, then R^{opt} can be computed in polynomial time. Otherwise, by computing $\int_0^1 v(x)dx$ numerically, R^{opt} can also be computed numerically. Thus when Theorem 4 applies, the optimal lottery scheme can be computed either analytically in polynomial time, or numerically.

Ex-post Budget-feasibility. A lottery scheme in general only satisfies the budget constraint in expectation, and it is possible that under some realization of the lotteries the total cost is much higher than the budget. Yet, given a randomized allocation $R = (p_1, \dots, p_m)$, the planner can first choose an ordering of the consumers uniformly at random, and then assign the first p_1 *fraction* of them to provider 1, the next p_2 fraction to provider 2, and so on. By doing so, each consumer is assigned to the providers according to the correct distribution $(p_1, \dots, p_m)$, thus the expected social welfare is that of R. While in any realized allocation the total cost is $\sum_{j\in[m]} p_j c_j$, exactly the expected cost of the randomized allocation, and thus the budget constraint is satisfied with probability 1.[9]

Advantage in Generating Social Welfare. When Theorem 4 applies, not only the social welfare of the optimal randomized allocation is greater than or equal to that of the optimal stable allocation, but the ratio between them can be arbitrarily large, since in the latter a lot of social welfare may be burnt by letting the consumers wait. As an example, consider the case where $v(x) = v_0$ is a positive constant, $q_1 = \epsilon \ll 1$, $1 \ll q_2 < \cdots < q_m$, $B \gg 1$, $c_1 = 1$, $c_2 = \cdots = c_m = \frac{B-\epsilon}{1-\epsilon}$. It is easy to see that one particular optimal stable allocation is to assign all consumers to provider 1 with waiting time 0, where the social welfare is $q_1 \int_0^1 v(x)dx = \epsilon v_0$ (assigning some consumers to better providers won't help, since all consumers must have the same utility anyway). While there is a randomized allocation that assigns each consumer to provider m with probability $1-\epsilon$ and to provider 1 with probability ϵ, resulting in total cost $(1-\epsilon)c_m + \epsilon c_1 = B$ and social welfare $((1-\epsilon)q_m + \epsilon q_1) \int_0^1 v(x)dx = ((1-\epsilon)q_m + \epsilon^2)v_0 \geq (1-\epsilon)v_0 \gg \epsilon v_0$.

To make $v(x)$ strictly increasing, just take $v(x) = \alpha x$ with some arbitrarily small $\alpha > 0$. The analysis is essentially the same as when $v(x)$ is a constant.

E Proof of Theorem 5

To prove Theorem 5 we first have the following claims.

[9] How to implement lotteries so that the desired constraints are satisfied ex-post is an important research topic in the Economics literature, see, e.g., [14].

Claim 5. *For any stable allocation $A = (a, w)$ and $x, x' \in [0, 1]$ with $x < x'$, we have $a(x) \le a(x')$ and $w_{a(x)} \le w_{a(x')}$.*

Proof. By the definition of stable allocations, we have

$$v_{a(x)}(x) - w_{a(x)} \ge v_{a(x')}(x) - w_{a(x')} \quad \text{and} \quad v_{a(x')}(x') - w_{a(x')} \ge v_{a(x)}(x') - w_{a(x)}.$$

Summing up the two inequalities side by side and rearranging terms, we have

$$v_{a(x')}(x') - v_{a(x')}(x) \ge v_{a(x)}(x') - v_{a(x)}(x),$$

that is

$$\sum_{k=1}^{a(x')} f_k(x') - f_k(x) \ge \sum_{k=1}^{a(x)} f_k(x') - f_k(x).$$

Notice that $f_j(x') - f_j(x) > 0$ for any $j \in [m]$, since $f_j(x)$ is strictly increasing and $x < x'$. Thus

$$a(x) \le a(x')$$

as desired. By definition, this inequality further implies $v_{a(x)}(x) \le v_{a(x')}(x)$. Thus $w_{a(x)} \le w_{a(x')}$, otherwise consumer x has better utility at $a(x')$ than at $a(x)$, contradicting the stability of A. Therefore Claim 5 holds. □

Claim 5 is the counterpart of Claims 1 and 2 in the current setting.

Claim 6. *For any stable allocation $A = (a, w)$, there exists $x_0, \cdots, x_m$ with $0 = x_0 \le x_1 \le x_2 \le \cdots \le x_{m-1} \le x_m = 1$, such that for any $j \in [m]$ and $x \in (x_{j-1}, x_j)$, $a(x) = j$.*

Moreover, if A is optimal with respect to a, then $w_1 = 0$ and for any $j > 1$,

$$w_j = v_j(x_{j-1}) - v_{j-1}(x_{j-1}) + w_{j-1} = f_j(x_{j-1}) + w_{j-1} = \cdots = \sum_{k=1}^{j} f_k(x_{k-1}).$$

Notice that when A is optimal with respect to a, for any $j > 1$, consumer x_{j-1} is indifferent between providers $j-1$ and j. Claim 6 is the counterpart of Lemmas 4, 1, and 2: to find the optimal stable allocation it suffices to focus on the choices of $x_0, \ldots, x_m$ and allocations whose waiting times are "tight" with respect to them. The first part of Claim 6 follows directly from Claim 5, the first equality of the second part is similar to the analysis of Lemma 2, and the remaining of the second part is by induction. Thus we omit the detailed proof here.

Now we are ready to prove our theorem.

Proof of Theorem 5. Let $A = (a, w)$ be a stable allocation that is budget-feasible and optimal with respect to a, and $x_0, \ldots, x_m$ as specified in Claim 6. That is, for each $j \in [m]$, consumers in (x_{j-1}, x_j) are assigned to provider j. Since there is a continuous population of consumers, it does not matter where consumers $x_0, x_1, \ldots, x_m$ are assigned to. Moreover, the cost of A is

$$C(A) = \sum_{j \in [m]} c_j (x_j - x_{j-1}) \le B.$$

Consider the randomized allocation $R = (p_1, \ldots, p_m)$ where $p_j = x_j - x_{j-1}$ for each $j \in [m]$. Notice that $p_j \ge 0$ for each j, and $\sum_{j \in [m]} p_j = \sum_{j \in [m]} x_j - x_{j-1} = x_m - x_0 = 1$. Thus R is well defined. Moreover, the cost of R is

$$C(R) = \sum_{j \in [m]} p_j c_j = \sum_{j \in [m]} c_j (x_j - x_{j-1}) = C(A) \le B,$$

and R is budget-feasible.

Similar to the proof of Theorem 4, we now show

$$SW(R) \geq SW(A), \tag{40}$$

which, when applied to $A = A^{opt}$, implies Theorem 5.

To prove Inequality 40, notice that

$$\begin{aligned}
SW(A) &= \int_0^1 v_{a(x)}(x) - w_{a(x)} dx = \sum_{j=1}^{m} \int_{x_{j-1}}^{x_j} v_j(x) - w_j dx \\
&= \sum_{j=1}^{m} \int_{x_{j-1}}^{x_j} \left[\sum_{k=1}^{j} f_k(x) - \sum_{k=1}^{j} f_k(x_{k-1}) \right] dx = \sum_{j=1}^{m} \int_{x_{j-1}}^{x_j} \sum_{k=1}^{j} [f_k(x) - f_k(x_{k-1})] dx \\
&= \sum_{k=1}^{m} \sum_{j=k}^{m} \int_{x_{j-1}}^{x_j} [f_k(x) - f_k(x_{k-1})] dx = \sum_{k=1}^{m} \int_{x_{k-1}}^{1} [f_k(x) - f_k(x_{k-1})] dx, \qquad (41)
\end{aligned}$$

where the third equality is by the definition of $v_j(x)$ and Claim 6.

Moreover, by definition the social welfare of R is

$$\begin{aligned}
SW(R) &= \int_0^1 \sum_{j=1}^{m} p_j v_j(x) dx = \int_0^1 \sum_{j=1}^{m} (x_j - x_{j-1}) \sum_{k=1}^{j} f_k(x) dx = \sum_{k=1}^{m} \int_0^1 \left[\sum_{j=k}^{m} (x_j - x_{j-1}) \right] f_k(x) dx \\
&= \sum_{k=1}^{m} \int_0^1 (1 - x_{k-1}) f_k(x) dx. \qquad (42)
\end{aligned}$$

We shall show

$$\int_0^1 (1 - x_{k-1}) f_k(x) dx \geq \int_{x_{k-1}}^{1} [f_k(x) - f_k(x_{k-1})] dx \tag{43}$$

for every $k \in [m]$, which together with Equations 41 and 42 implies Inequality 40. To do so, for any $k \in [m]$, consider the following function

$$g_k(y) = \int_0^1 (1-y) f_k(x) dx - \int_y^1 [f_k(x) - f_k(y)] dx = \int_0^1 (1-y) f_k(x) dx - \int_y^1 f_k(x) dx + (1-y) f_k(y)$$

for $y \in [0, 1]$. It suffices to show that $g_k(x_{k-1}) \geq 0$.

First, notice that

$$g_k(0) = \int_0^1 f_k(x) dx - \int_0^1 [f_k(x) - f_k(0)] dx = 0,$$

as $f_k(0) = 0$ by definition. Also,

$$g_k(1) = \int_0^1 0 dx - \int_1^1 [f_k(x) - f_k(1)] dx = 0.$$

Moreover,

$$g_k'(y) = -\int_0^1 f_k(x) dx + f_k(y) - f_k(y) + (1-y) f_k'(y) = (1-y) f_k'(y) - \int_0^1 f_k(x) dx.$$

Because $(1-y) f_k'(y)$ is non-increasing as required by Theorem 5, and because $\int_0^1 f_k(x) dx$ is a constant, we have that $g_k'(y)$ is non-increasing, that is, $g_k(y)$ is concave on $[0, 1]$. Since $g_k(0) = g_k(1) = 0$, we have $g_k(y) \geq 0$ for all $y \in [0, 1]$. Accordingly, $g_k(x_{k-1}) \geq 0$ and Inequality 43 holds. Thus Inequality 40 holds, and so does Theorem 5. □

F An Example where Stable Allocations Can Do Better

Consider two providers, 0 and 1, with costs $c_0 = 0$ and $c_1 > 0$ respectively. The valuation functions are $v_0(x) = 0$ and $v_1(x) = e^{2x} - 1$, and the budget is $B \in [0, c_1]$. (Strictly speaking we should take $v_0(x) = \epsilon x$ with some arbitrarily small $\epsilon > 0$, so that $v_0(x)$ is strictly increasing. But the idea is the same.)

First of all, $f_0(x) = 0$, $f_1(x) = v_1(x) - v_0(x) = e^{2x} - 1$, and $(1-x)f_1'(x) = 2(1-x)e^{2x}$ which is increasing when $x \leq 1/2$ and decreasing otherwise. Thus Theorem 5 does not apply here (neither does Theorem 4 as one can verify). Second, it is easy to see that the optimal randomized allocation is $R^{opt} = (p_0, p_1)$ where $p_1 = \frac{B}{c_1}$ and $p_0 = 1 - p_1$. We have $C(R^{opt}) = B$ and

$$SW(R^{opt}) = \frac{B}{c_1}\int_0^1 (e^{2x} - 1)dx = \frac{B(e^2 - 3)}{2c_1}.$$

For any stable allocation $A = (a, w)$ that is budget-feasible and optimal with respect to a, by Claim 6 there exists $x_1 \in [0, 1]$ such that consumers in $(0, x_1)$ are assigned to provider 0, consumers in $(x_1, 1)$ are assigned to provider 1, $w_0 = 0$ and $w_1 = f_1(x_1) = e^{2x_1} - 1$. Thus $C(A) = c_1(1 - x_1) \leq B$ and

$$SW(A) = \int_{x_1}^1 v_1(x) - w_1 dx = \int_{x_1}^1 e^{2x} - e^{2x_1} dx = \frac{e^2 + e^{2x_1}(2x_1 - 3)}{2}.$$

Accordingly, the optimal stable allocation A^{opt} is such that $x_1 = 1 - \frac{B}{c_1}$. Letting $r = \frac{B}{c_1}$, we have

$$SW(R^{opt}) = \frac{(e^2 - 3)r}{2} \quad \text{and} \quad SW(A^{opt}) = \frac{e^2 - e^{2(1-r)}(1 + 2r)}{2}.$$

It is easy to see (e.g., using Mathematica) that there exists $r_0 \approx 0.8$ such that $SW(R^{opt}) \geq SW(A^{opt})$ if $r \leq r_0$ and $SW(R^{opt}) < SW(A^{opt})$ otherwise. That is, if the budget is enough to serve 80 percent of the consumers at provider 1 then the optimal stable allocation has better social welfare, otherwise the optimal randomized allocation does better. Figure 1 shows the difference between the two for $r \in [0, 1]$. In this example, the optimal stable allocation only has a small advantage when $r > r_0$: $\max_r SW(A^{opt})/SW(R^{opt}) \approx 1.005$, which occurs at $r \approx 0.9$.

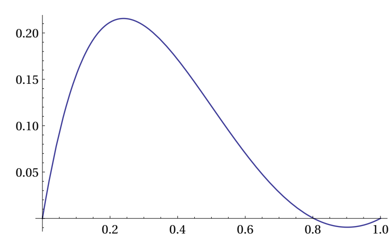

Figure 1: $SW(R^{opt}) - SW(A^{opt})$ as a function of $r = \frac{B}{c_1}$.