Provision-After-Wait with Common Preferences

Hau Chan
Department of Computer Science
Trinity University
San Antonio, TX 78212
hchan@trinity.edu

Jing Chen
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794
jingchen@cs.stonybrook.edu

ABSTRACT

We study the Provision-after-Wait problem in healthcare introduced by Braverman, Chen, and Kannan (2016). In this setting, patients seek a medical procedure, and the procedure can be performed by different hospitals of different costs. Each patient has a value for each hospital, and a budget-constrained government/planner pays for the medical expenses of the patients. The planner's goal is to find an optimal stable assignment that is envy-free and maximizes the social welfare while keeping the expenses within the budget.

In this work, we focus on the settings where the patients have a common preference of the hospitals. We show that computing the optimal stable assignment for maximizing social welfare is NP-hard. Furthermore, we construct a fully polynomial-time approximation scheme (FPTAS) that runs in time $O((n+m)n^3m/\epsilon)$, where m and n are the number of hospitals and patients, respectively. In order to develop the FPTAS, we have defined and studied a new problem, ordered Knapsack.

We also consider the setting where the planner uses lottery as a rationing tool. For a large sub-class of our settings, we show the conditions under which the optimal lottery scheme has a simple structure and generates more social welfare than the optimal stable assignment. Moreover, such optimal lottery scheme can be computed by a linear program.

Keywords

budget, NP-hardness, approximation, FPTAS, waiting, lottery, envy-free, assignment, matching

1. INTRODUCTION

We study the Provision-after-Wait problem in healthcare [5], which considers the interaction among the patients, the hospitals, and a planner. Each patient seeks a medical service, such as X-ray or MRI, and has different values for different hospitals. Each hospital has a cost for serving one patient, which must be paid. Yet, the patients do not pay for the service and, instead, the planner pays for all of them. However, the planner has a budget and might not be able to afford the costs incurred by all patients going to their most preferred hospitals. Thus, the planner decides how to distribute his budget among the hospitals and, thus, how many patients he can afford each hospital to serve (in one budget period, say a month or a year). Patients choose their favorite hospitals and, if a hospital is over-demanded, then naturally a line will form and a waiting-time is specified at that hospital: the amount of time each patient has to wait before getting served there. Given the waiting times, a patient selects the hospital that maximizes his utility, which is his value minus the waiting time. De facto, *the planner uses waiting times as a rationing tool* and sets the hospitals' waiting times such that when the patients pick their utility-maximizing hospitals, the social welfare is maximized while the total cost is within the budget. In other words, the planner wants to produce a *stable* assignment: where each patient has non-negative utility and gets the hospital that maximizes his utility (thus a stable assignment is automatically envy-free).

Even when each patient is "narcissistic" and only has positive value for a single distinct hospital, the problem of computing the optimal budget feasible social welfare is already NP-hard [5]. Beyond this, little is known about computing the optimal social welfare when the patients' preferences are structured. In this paper, we consider the setting where the patients have a *common preference* of the hospitals: that is, the patients have the same ranking over the hospitals and perceive the hospitals based on the same quality measurement. This type of preferences is considered in position auctions [3, 21] where the patients in our setting correspond to the advertisers there, the hospitals correspond to the slots, and all advisers know the click-through rates of all slots. Similarly, in our setting, the patients know a ranking and quantitative measurements of the hospitals (i.e., based on ranking and reviews from U.S. News Best Hospitals 2015-16[1] or Centers for Medicare and Medicaid Services[2] of the US government).

1.1 Our Results

In the common-preference setting, each patient has a value for receiving the medical service and each hospital has a quality factor that is publicly known. The value of a patient for a hospital is the product of the two: that is, a consumer's value for a provider is proportional to the provider's quality. This is typical in position auctions [3, 21] where quality means view-through or click-through rate, or in scenarios where quality means the probability of obtaining the same

[1] `http://health.usnews.com/best-hospitals`
[2] `http://www.cms.gov/`

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*
Copyright © 2016, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

resource [13, 15, 16]. In the Provision-after-Wait problem, a hospital's quality can be considered as a scale-down from a perfect hospital: the "star ratings" on the Internet for hospitals are illustrative examples for such measurements.

Our model and results for waiting times are formally presented in Sections 2 and 3, and those for lotteries are in Section 4. Below we summarize our main results. Due to lack of space, most of the proofs are provided in the full version of the paper [9].

Optimal Stable Assignments.

For computing an optimal stable assignment of the Provision-after-Wait problem with common preference, we have the following two theorems.

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

THEOREM 2 (restated). *There exists a fully polynomial-time approximation scheme (FPTAS) for the optimal stable assignment problem which, given any $\epsilon > 0$, runs in time $O((n+m)n^3 m/\epsilon)$, where n and m are respectively the number of patients and hospitals.*

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 assignment. We believe that the ordered Knapsack problem itself is of independent interest and is worth further study. Detailed discussions on Theorems 1 and 2 are in Sections 3.1 and 3.2, respectively. These results provide us a relatively complete understanding about the computation problem in our setting.

Moreover, it is worth pointing out that, our results are robust against small perturbations of the patients' valuations that may destroy the common-preference property: they remain true as long as the perturbed valuations keep a desired "ordering" among the patients. A more detailed discussion about this issue is in the full version.

Optimal Lottery Assignments.

In this part, we consider the patients as a continuous population instead of a discrete set, so that the solutions have a clean formula. Indeed, in the healthcare literature it is often assumed that there is a continuous population of patients, since in reality the number of patients is usually huge. By discretizing the patient space, similar results can be obtained for discrete settings.

Consider the patients as a continuous population represented by the interval $[0, 1]$. Their values for the medical service can be specified by a function v mapping each patient $x \in [0, 1]$ to a non-negative real, such that the value of patient x at a hospital is $v(x)$ times the hospital's quality. The function v is called the patients' *valuation profile*.[3] We assume v to be increasing, which is without loss of generality, since we can reorder and rename the patients: we can also do this in the discrete case, so that the value of patient 1 is smaller than that of patient 2, etc. Moreover, some regularity assumptions about v are typically used to ease the discussion, such as piece-wise continuity (see Section 4).

A lottery scheme is essentially a menu of randomized (hospital, waiting time) pairs for the patients to choose from, where the waiting times may or may not be non-zero. It is somewhat surprising that, given the extremely rich structures of the possible lottery schemes, for a large class of Provision-after-Wait problems, the optimal lottery scheme has a very simple form: that is, there is a single distribution from which all patients' hospitals are drawn, and no waiting time is imposed. We call such a lottery scheme a *randomized assignment*, and we have the following.

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

In Section 4.1 we show that the optimal randomized assignment can be solved by a linear program, so does the optimal lottery scheme 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 assignment and that of the optimal stable assignment can be arbitrarily large. This condition is tight in the sense that, when it does not hold, there are cases where optimal stable assignments generate more social welfare: we provide examples in the full version.

Moreover, it is worth pointing out that, given a randomized assignment, we do not need to sample the patients' hospitals independently: our result holds as long as the marginal distribution for each patient is as specified by the assignment. Thus the optimal randomized assignment can be implemented so that the budget constraint is satisfied with probability 1 (see Section 4.2).

Interestingly, the condition in Theorem 4 has a very natural interpretation from another aspect. If we consider a single patient whose "type" x is 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 resulting value has *monotone hazard rate* (MHR). This immediately connects our result with lottery pricing schemes [10] with a single buyer and multiple items, where optimal pricing schemes are studied when the distributions of the buyer's values have MHR. Indeed, besides being succinct, one benefit of consider a continuous population of patients is that it lets us see clearly the connection between our condition and MHR, which is typically studied for continuous distributions.

We further generalize our result to settings where the patients do not perceive the hospitals based on the same quality measurement but still have the same ranking for them. Here the randomized assignment may not be optimal among all lottery schemes, but we have the following.

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

Theorem 5 is formalized in Section 4.3. Moreover, when the conditions do not hold there are cases where optimal stable assignments do better.

In conclusion, we have studied the computation and the structure of the optimal Provision-after-Wait schemes when

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

waiting times and lotteries are used as rationing tools. Our results suggest that neither rationing tool is absolutely better than the other in terms of generating social welfare, and we have identified conditions that enable the comparison of their performances. A planner should choose an appropriate tool based on the patients' valuations as specified by Theorems 4 and 5. Our results then allow the planner to compute/approximate the optimal allocations efficiently.

1.2 Related Work

The Provision-after-Wait problem was introduced in [5]. 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 than ours. Also, in [5] the authors study optimal lotteries when there are two hospitals. Since we allow any number of hospitals, our results on optimal lotteries and randomized allocations are more general.

In unit-demand pricing schemes, n items are to be sold to m buyers and each buyer only wants one of them. This is similar to our case: each patient needs to be assigned to one hospital. Envy-freeness is a widely adopted solution concept there, but the goal of pricing schemes is to maximize revenue. In [13, 16], the authors study pricing problems where the buyers' valuations are similar to ours, and they characterize the optimal envy-free solutions for generating revenue [13, 16] and total value [13]. Their characterizations are analogous to ours, but their goals are different and they do not further study the computation complexity of the optimal solutions. While most works on pricing schemes study deterministic optimal item-pricing [6, 10, 12, 14, 16], a few consider lotteries [7, 11, 19, 20] 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.

Finally, one key difference we see between the Provision-after-Wait model and the model of [17], a classic study about matching, is that, in [17] there is only one "currency", the salary (in particular, gross product is measured in the same unit as salary); while in our model there are two "currencies", money and time, and they are not exchangeable since patients do not care about money paid to the hospital.

2. THE MODEL

In the Provision-after-Wait problem [5], there are n *patients*, indexed by $[n] = \{1, 2, ..., n\}$, and m *hospitals*, indexed by $[m] = \{1, 2, ..., m\}$. Each patient wants a single medical service, which can be provided by any one of the hospitals. For each hospital $j \in [m]$, there is a *cost* $c_j \in \mathbb{Z}^+$ for serving one patient.[4] Moreover, each patient $i \in [n]$ has a value $v_{ij} \in \mathbb{Z}^+$ for receiving the service in hospital $j \in [m]$. As mentioned earlier, the patients do not pay for the service. Instead, the *planner* (e.g., government) pays for everybody's service through some funding program (e.g., the Patient Protection and Affordable Care Act in United States—that is, Obamacare). The planner has a *budget* $B \in \mathbb{Z}^+$ that limits the total amount that he can spend. To ensure the planner does not spend more than the budget (and while ensuring free-choice of the patients), the planner determines the *waiting time* w_j for each hospital $j \in [m]$.The planner's goal is to produce a solution that maximizes *social welfare*.

It is worth pointing out that, although in the model it is the planner who decides the waiting times, it has been shown in [5] that waiting times derive endogenously from the budget constraint and the dynamics between patients and hospitals: the planner only needs to decide each hospital's supply given the budget, and the demand and supply together determine waiting times and social welfare. If a hospital is over-demanded and there is no budget to increase its supply, then its waiting time increases in the dynamics.

Solution Concept.

A solution to the Provision-after-Wait problem is called an *assignment*. An assignment A consists of an *assignment function* $a : [n] \to [m]$ and a *waiting vector* $w = (w_1, \ldots, w_m)$ where $w_j \geq 0$ for each $j \in [m]$: $a(i)$ is the hospital to whom patient i is assigned and w_j is the waiting time for any patient assigned to hospital 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 patient i's *utility* is $v_{ia(i)} - w_{a(i)}$,[5] and the *social welfare* of A is $SW(A) = \sum_{i\in[n]} v_{ia(i)} - w_{a(i)}$.

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

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

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 hospitals, respectively. Moreover, it is often useful to consider optimal stable assignments with respect to a particular assignment function, as follows.

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

Note that budget-feasibility is not required in Definition 2: the cost of all such assignments is solely decided by a, thus either all of them are budget-feasible or none is.

Common Preference.

In the general Provision-after-Wait problem, the patients' values for the hospitals can be arbitrary. For the special class of "narcissistic" valuations where each patient $i \in [n]$

[4]That is, one hospital provides the same service to all patients, while different hospitals may provide different services. For example, one "hospital" may actually represent X-ray services and another one may represent MRI services.

[5]Utilities are quasi-linear in waiting time because we measure patients' valuations as "willingness to wait", in parallel with "willingness to pay" in auctions, where utilities are quasi-linear in price. The planner doesn't need to intentionally delay care to preserve fairness: as mentioned previously, the desired waiting times will appear endogenously from the dynamics between hospitals and patients.

has a value $v_{ij} > 0$ for $j = i$ and $v_{ij} = 0$ for all $j \neq i$, the problem of computing an optimal stable assignment is already NP-hard [5]. Beyond this result, little is known for the computational complexity of the problem.

Here, we consider a different class of valuations where the patients have a *common preference* over the hospitals. In particular, for each hospital $j \in [m]$, there is a quality factor $q_j \in \mathbb{Z}^+$ that is publicly known. Without loss of generality, we can rename the hospitals such that $q_1 \geq q_2 \geq ... \geq q_m$. For each patient $i \in [n]$, there is a value $v_i \in \mathbb{Z}^+$ specifies i's happiness for getting the medical service. For each patient $i \in [n]$ and hospital $j \in [m]$, i's value for hospital j is $v_{ij} = v_i q_j$. Thus, $v_{i1} \geq v_{i2} \geq ... \geq v_{im}$ for all patients $i \in [n]$. The setting where a consumer's value for a provider is proportional to its value and the provider's quality has been explored in other context under different economic incentive settings [2, 13, 15, 16, 21]. We will consider this type of valuations throughout the paper and we will explicitly represent the value v_{ij} as $v_i q_j$ for clarity.

3. OPTIMAL STABLE ASSIGNMENTS

In this section, we show that (1) computing an optimal stable assignment is NP-hard; and (2) there is a fully polynomial time approximation scheme (FPTAS) that returns an assignment with social welfare ϵ-close to the optimum.

3.1 Finding Optimal Stable Assignments

In this and next sections, without loss of generality we rename the patients and the hospitals 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 assignment.*

To prove Theorem 1, we first characterize the social welfare of optimal stable assignments. Our approach is similar to that in [13, 16], but the formulas are different. Indeed, their goals were to maximize the total payment and/or the total value in the context of revenue maximization. In particular, our Definition 4 and Lemma 3 are equivalent to Lemma 2.1 in [13], but the remaining parts are new. In this section, we only state our key lemmas and we refer the readers to the full version [9] for a detailed analysis.

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

Lemma 1 shows that it is sufficient to consider stable assignments $A = (a, w)$ with a ordered.

LEMMA 1. *Given any stable assignment $A = (a, w)$, in polynomial time it can be modified so that: a beomes ordered, A is still stable, and the total cost and the utility of each patient remain the same.*

Therefore, for any stable assignment $A = (a, w)$ with a ordered, we can deduce that $q_{a(1)} \geq q_{a(2)} \geq \cdots \geq q_{a(n)}$ and $w_{a(1)} \geq w_{a(2)} \geq \cdots \geq w_{a(n)}$. In other words, patients with higher values are assigned to hospitals with higher quality, and hospitals with higher quality have higher waiting times than those with lower quality: indeed, if hospital i has higher quality than hospital j, but lower waiting time, then patients assigned to j want to deviate to i, since it gives them higher utilities.

DEFINITION 4. *For any ordered assignment function a, an assignment $A = (a, w)$ is* tight *at a if for any $i < n$*

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

Notice that, by the inequality on $q_{a(i)}$'s, 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, patient $i+1$ is indifferent between his utility at $a(i+1)$ and that at $a(i)$.

The following lemmas show that we only need to consider assignment A that is tight at a.

LEMMA 2. *For any ordered assignment function a, let $A = (a, w)$ be an assignment 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.*

LEMMA 3. *For any ordered assignment function a and any stable assignment $A = (a, w)$, 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 assignment optimal with respect to a can be explicitly calculated from the patients' values and the hospitals' qualities.

LEMMA 4. *For any ordered assignment function a and any stable assignment $A = (a, w)$ 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 lemmas, we now present the proof of Theorem 1.

PROOF OF THEOREM 1. Consider the decision version of the Provision-after-Wait problem: $DPaW = \{(q_1, \ldots, q_m, c_1, \ldots, c_m, v_1, \ldots, v_n, B, V)$: there exists a stable budget feasible assignment A such that $SW(A) \geq V\}$.

It is clear that if one can find an optimal stable assignment for every instance of the Provision-after-Wait problem then one can decide $DPaW$. We shall show that $DPaW$ is NP-complete by a reduction from the Subset-Sum problem: $SubsetSum = \{(s_1, \ldots, s_n, T)$: there exists $S \subseteq [n]$ such that $\sum_{i \in S} s_i = T\}$. 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 $DPaW$ 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 $m = 2n$ hospitals and n patients.
- For each $i \in [n]$, $q_i = c_i = s_i \cdot 2^{n(\lceil \log n \rceil + 1)} + 2^{(n-i)(\lceil \log n \rceil + 1)}$: 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 setting 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. *$\gamma \in DPaW \Rightarrow \alpha \in SubsetSum$.*

PROOF. Let $A = (a, w)$ be an optimal stable assignment of γ. By definition, A is optimal with respect to a. By Lemma 1 we assume without loss of generality that a is ordered. Thus by Lemma 4 we have $SW(A) = \sum_{i<n} i \cdot q_{a(i)} \cdot (v_i - v_{i+1}) + n \cdot q_{a(n)} \cdot v_n = \sum_{i<n} i \cdot q_{a(i)} \cdot \frac{1}{i} + n \cdot q_{a(n)} \cdot \frac{1}{n} = \sum_{i \in [n]} q_{a(i)}$. Since $q_j = c_j$ for any $j \in [m]$ and since A is budget feasible, $SW(A) = \sum_{i \in [n]} c_{a(i)} = C(A) \leq B = V$. Since $\gamma \in DPaW$, we have $SW(A) \geq V$ 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 $j \in [n]$,

$$|\{i \in [n] : a(i) \in \{j, n+j\}\}| = 1, \quad (2)$$

that is, there is exactly one patient assigned to either hospital j or hospital $n+j$.

To see why Equation 2 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 patients, 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 hospitals whose qualities contribute a 1 to the $(n-j)(\lceil \log n \rceil + 1) + 1$st bit of $SW(A)$ are hospitals j and $n+j$.

If more than one patients 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 $SW(A) \neq V$. If no patient 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 patient assigned to either hospital j or hospital $n+j$, and Equation 2 holds.

By Equation 2, 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 $SW(A) = \sum_{i \in [n]} q_{a(i)} = \sum_{j \in S} q_j + \sum_{j \in S'} q_{n+j} = \sum_{j \in S} s_j \cdot 2^{n(\lceil \log n \rceil + 1)} + \sum_{j \in [n]} 2^{(n-j)(\lceil \log n \rceil + 1)}$. 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 $\alpha \in SubsetSum$ and Lemma 5 holds. □

LEMMA 6. *$\alpha \in SubsetSum \Rightarrow \gamma \in DPaW$.*

PROOF. Since $\alpha \in SubsetSum$, there exists $S \subseteq [n]$ such that $\sum_{i \in S} s_i = T$. Let $k = |S|$ and $S = \{j_1, \ldots, j_k\}$, with $j_1 \leq j_2 \leq \cdots \leq j_k$. Further, let $S' = [n] \backslash S = \{j_{k+1}, \ldots, j_n\}$, with $j_{k+1} \leq j_{k+2} \leq \cdots \leq j_n$. We construct an assignment $A = (a, w)$ 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 $a(1) \leq a(2) \leq \cdots \leq a(n) = n + j_n \leq m$. 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 $C(A) = \sum_{i \in [n]} c_{a(i)} = \sum_{i \leq k} c_{j_i} + \sum_{i \geq k+1} c_{n+j_i} = T \cdot 2^{n(\lceil \log n \rceil + 1)} + \sum_{j \in [n]} 2^{(n-j)(\lceil \log n \rceil + 1)} = B$. Thus A is budget feasible.

Since A is tight at a, by Lemma 2, A is stable. By Lemma 3, A is optimal with respect to a. Thus $SW(A) = \sum_{i=1}^{n-1} i \cdot q_{a(i)} \cdot (v_i - v_{i+1}) + n \cdot q_{a(n)} \cdot v_n = \sum_{i=1}^{n-1} i \cdot q_{a(i)} \cdot \frac{1}{i} + n \cdot q_{a(n)} \cdot \frac{1}{n} = \sum_{i \in [n]} q_{a(i)} = \sum_{i \in [n]} c_{a(i)} = C(A) = B = V$. Therefore A is a stable budget feasible assignment with $SW(A) \geq V$. Accordingly, $\gamma \in DPaW$ and Lemma 6 holds. □

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

3.2 An FPTAS for Optimal Stable Assignments

Next, we show that there is an efficient algorithm that produces a stable assignment with social welfare arbitrarily close to the optimum. Letting A^{opt} be an optimal stable assignment, we have the following.

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

The complete proof of Theorem 2 is in the full version of the paper [9]. Below, we describe the general ideas. Notice that by Lemma 4, for any ordered assignment function a we can define the *social welfare of a*, $SW(a)$, to be the social welfare of stable assignments optimal with respect to a. That is, $SW(a) = \sum_{i<n} i \cdot q_{a(i)} \cdot (v_i - v_{i+1}) + n \cdot q_{a(n)} \cdot v_n$. An assignment function a is *budget feasible* if $C(a) = \sum_{i \in [n]} c_{a(i)} \leq B$.

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

Given an ordered assignment function a, by Lemmas 2 and 3, we can construct, in time $O(m+n)$, a stable assignment A optimal with respect to a: that is, the assignment defined in Lemma 2. If a is optimal, then A is an optimal stable assignment. Thus, to prove Theorem 2 it suffices to approximate the optimal ordered assignment function.

Notice that if there exists a hospital j such that $c_j < c_{j+1}$, then for any ordered assignment function a and for all patients assigned to hospital $j+1$, by reassigning them to j we get another ordered assignment a' such that $C(a') \leq C(a)$ and $SW(a') \geq SW(a)$. Accordingly, we can further focus on ordered assignment functions that do not assign any patient 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 assignment function.

3.3 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* assignment *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 *is $C(a) = \sum_{i \in [n]} c_{a(i)}$. The goal is to find an assignment with cost no larger than B and the maximum 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.[6] We can reduce the problem of the optimal ordered assignment 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 $u_{nj} = nq_j v_n$. Any assignment of the resulting ordered Knapsack problem is an ordered assignment function of the original Provision-after-Wait problem, with the same cost and the same social welfare. Thus, letting a^{opt} be the optimal assignment 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 assignment a with $C(a) \leq B$ and $SW(a) \geq (1-\epsilon)SW(a^{opt})$.*

PROOF OF THEOREM 3 (SKETCH). We first construct a dynamic program that computes the optimal assignment in pseudo-polynomial time, and then run it on properly scaled inputs to get the desired FPTAS. We refer the readers to the full version of the paper [9] for the complete proof, and here we only provide the objective function for our dynamic program. (We are abusing the notation C a bit, but it always represents the cost that we want to measure.)

For any assignment 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, let $SW(i,j) = \sum_{i'=i}^{n} u_{i'j}$ be the contribution of players $i,\ldots,n$ when they are all assigned to j. If $SW(i,j) \geq s$ then $C(i)(j)(s) = c_j + \min_{a:j=a(i)\leq\cdots\leq a(n), SW(a,i)\geq s} \sum_{i'>i} c_{a(i')}$, and $C(i)(j)(s) = +\infty$ otherwise. 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 items after j. In practice, $+\infty$ can be replaced by $B+1$ (or any number larger than B and of polynomial length). Moreover, we observe that (a) for any optimal assignment a, $SW(a) = \max\{s : \min_{j\in[m]} C(1)(j)(s) \leq B\}$; and (b) $C(n)(j)(s) = c_j$ for any $j \in [m]$ and $s \leq u_{nj}$, $C(i)(j)(0) = c_j + (n-i)c_m$ for any $i < n$ and $j \in [m]$, and for any $i < n$, $j \in [m]$ and $0 < s \leq SW(i,j)$,

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

Finally, $C(i)(j)(s) = +\infty$ in all other cases. The dynamic program thus computes the function C and the optimal assignment a. □

[6]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., [4, 1, 8]). 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 [18]. In the latter each item has a fixed value and the outcome is a set instead of a function from players to items.

REMARK 1. *In fact, we can construct a pseudo-polynomial time dynamic program directly for the Provision-after-Wait problem. Then one may try to scale down the hospitals' qualities q_j and patients' 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 Provision-after-Wait problem with qualities and values—that is where the ordered Knapsack problem comes into play.*

4. ASSIGNMENTS USING LOTTERIES

If no randomness is allowed, the optimal stable assignment is our best rationing tool. However, if the planner can ask the patients to enter lotteries, the space of possible mechanisms becomes much larger and better social welfare can be obtained. In this section, we first characterize the structure of the optimal lotteries for a large sub-class of the Provision-after-Wait 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 assignment.

Although our results apply to the discrete case of n patients, they are more succinct to state for a continuous population of patients. Thus, we let the patients be indexed by the interval $[0,1]$, and use the valuation function $v(x)$ to specify the value of each patient x. We assume $v(x)$ is strictly increasing and twice differentiable, so that the integrations and differentiations used below are always well defined.[7] Also, by shifting down all patients' values by $v(0)$, we assume without loss of generality that $v(0) = 0$.

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

A patient taking lottery λ will wait for time w and then be assigned to each hospital j with probability p_j. Patient x's *(expected) utility* under λ is $u(x,\lambda) = (\sum_{j\in[m]} p_j q_j v(x)) - w$.[8] Given L, each patient chooses a lottery to maximize his own utility. That is, denoting by $\lambda^L(x) = (p_1^L(x),\ldots,p_m^L(x), w^L(x)) \in L$ the choice of patient x, we have that for any $\lambda \in L$,

$$u(x,\lambda^L(x)) \geq u(x,\lambda). \quad (3)$$

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

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

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

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 patient 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 patients' choices are on the boundary of L.

Since the patients are infinitesimal, each hospital 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 $C(L) \le B$. 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*:

$$L^{opt} \in \operatorname*{argmax}_{L \text{ is budget feasible}} SW(L).$$

For each $x \in [0,1]$, we denote by $\lambda^{opt}(x) = (p_1^{opt}(x), \ldots, p_m^{opt}(x), w^{opt}(x))$ the choice of patient x under L^{opt}.

A stable assignment $A = (a, w)$ 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 $\{\lambda_1, \ldots, \lambda_m\}$: for each $j \in [m]$, $\lambda_j = (p_1, \ldots, p_m, w_j)$, $p_j = 1$ and $p_{j'} = 0$ for any $j' \neq j$. Given L, each patient x chooses $\lambda_{a(x)}$, which corresponds to being assigned to $a(x)$ with probability 1 after waiting $w_{a(x)}$. Thus we have $SW(L^{opt}) \ge SW(A^{opt})$, where A^{opt} is the optimal stable assignment.

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

Definition 8. *A* randomized assignment R *is a tuple of non-negative reals, $R = (p_1, \ldots, p_m)$, s.t. $\sum_{j\in[m]} p_j \le 1$.*

According to R, each patient is assigned to each hospital 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 assignment*, that is, $R^{opt} \in \operatorname{argmax}_{R \text{ is budget feasible}} SW(R)$. As a randomized assignment is a special lottery scheme, $SW(L^{opt}) \ge SW(R^{opt})$.

Notice that a lottery scheme is "stable in expectation", since a patient chooses a lottery that maximizes his expected utility. This is also true for a randomized assignment, since there is only one lottery to choose from. But a lottery scheme may not be "ex-post stable": the realized assignment may not maximize a patient's utility among all hospitals.

4.1 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 assignment is optimal among all lottery schemes.

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

Proof of Theorem 4 (Sketch). To show Theorem 4, we rely on the facts (as shown in the full version [9]) that (a) for any lottery scheme L, the function $\sum_{j\in[m]} p_j^L(x)q_j$ is non-decreasing, and (b) for any lottery scheme L and any patient $x \in [0,1]$, $u(x, \lambda^L(x)) = \int_0^{v(x)} \sum_{j\in[m]} q_j p_j^L(\hat{v})d\hat{v}$.

Since $SW(L^{opt}) \ge SW(R^{opt})$ by definition, we only need to show that $SW(R^{opt}) \ge SW(L^{opt})$. To show this, we first define, for any budget feasible lottery scheme L, $R = (p_1, \ldots, p_m)$ to be the randomized assignment 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 $[0,1]$. Letting $L = L^{opt}$, (a) and (b) imply $SW(R) \ge SW(L^{opt})$. Thus $SW(R^{opt}) \ge SW(L^{opt})$. □

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, $v''(x) \le 0$. Since $v'(x) > 0$ and $x \in [0,1]$, we have $g'(x) \le 0$. Thus $g(x)$ is non-increasing. □

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 assignment is optimal among all lottery schemes in this case.

Theorem 4 in Terms of Monotone Hazard Rate.

Interestingly, the condition in Theorem 4 has a very natural interpretation from another viewpoint. Consider an assignment problem where there is a single patient and multiple hospitals. The planner's budget is lower than the cost of the patient's favorite hospital, thus he cannot simply be assigned there with probability 1. There is a distribution D from which the patient's value is drawn: his *type* is uniformly distributed over $[0,1]$ and his value at type x is $v(x)$. 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 $F(v_0) = \Pr[y \le v_0] = \Pr[v^{-1}(y) \le x_0] = x_0 = v^{-1}(v_0)$, 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 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, $SW(R^{opt}) = SW(L^{opt})$.*

4.2 Important Properties of the Optimal Randomized Assignment

In this subsection, we further discuss several interesting properties of the optimal randomized assignment.

Computability.

In general Provision-after-Wait problems, there may not be an efficient algorithm for finding an optimal lottery scheme. But the optimal randomized assignment is always defined by the following linear program.

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

Since the integral $\int_0^1 v(x)dx$ is a constant in the linear program, the linear program can be solved in polynomial time. Of course, we need the value of $\int_0^1 v(x)dx$ so as to compute $SW(R^{opt})$. If $\int_0^1 v(x)dx$ has a closed form and can be computed in polynomial time, then $SW(R^{opt})$ can be computed in polynomial time. Otherwise, by computing $\int_0^1 v(x)dx$ numerically, $SW(R^{opt})$ can also be computed 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 assignment $R = (p_1, \dots, p_m)$, the planner can first choose an ordering of the patients uniformly at random, and then assign the first p_1 *fraction* of them to hospital 1, the next p_2 fraction to hospital 2, and so on. By doing so, each patient is assigned to the hospitals according to the correct distribution $(p_1, \dots, p_m)$, thus the expected social welfare equals $SW(R)$. While in any realized assignment the total cost is $\sum_{j\in[m]} p_j c_j$, exactly the expected cost of the randomized assignment, 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 assignment is no less that of the optimal stable assignment, but the ratio between them can be arbitrarily large, since in the latter a lot of social welfare may be burnt by letting the patients wait. As an example, consider the case where $v(x) = v_0$ is a positive constant, $q_1 = \epsilon \ll 1$, $1 \ll q_2 < \dots < q_m$, $B \gg 1$, $c_1 = 1$, $c_2 = \dots = c_m = \frac{B-\epsilon}{1-\epsilon}$. It is easy to see that one particular optimal stable assignment is to assign all patients to hospital 1 with waiting time 0, where the social welfare is $q_1 \int_0^1 v(x)dx = \epsilon v_0$ (assigning some patients to better hospitals won't help, since all patients must have the same utility anyway). While there is a randomized assignment that assigns each patient to hospital m with probability $1-\epsilon$ and to hospital 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.

4.3 Randomized v.s. Stable Assignments

Finally, we extend our approach to settings where the patients' values are not proportional to the hospitals' qualities, but there are still orders among the hospitals and the patients. More precisely, for each $j \in [m]$, let function $v_j(x)$ be the value that patient $x \in [0,1]$ receives when assigned to hospital 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 $v_1(x) \leq v_2(x) \leq \dots \leq v_m(x)$ for each x.

As will become clear in the analysis, the key factors affecting the social welfare are not the patients' values, but the *differences* among each patient's values at different hospitals. Accordingly, for each $j \in [m]$, letting $f_j(x)$ be a function on $[0,1]$ that is strictly increasing, twice differentiable and $f_j(0) = 0$,[10] we consider the patients' 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 4.1 as a special case: after renaming the hospitals s.t. $q_1 \leq q_2 \leq \dots \leq q_m$, take $f_1(x) = q_1 v(x)$ and $f_j(x) = (q_j - q_{j-1})v(x)\ \forall j > 1$.

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

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

Proof of Theorem 5 (Sketch). The proof uses related but different ideas from those for Theorem 4. We only present a key claim here.

Claim 1. *For any stable assignment $A = (a, w)$, there exists $x_0, \cdots, x_m$ with $0 = x_0 \leq x_1 \leq x_2 \leq \dots \leq x_{m-1} \leq x_m = 1$, s.t. 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 $j > 1$, $w_j = v_j(x_{j-1}) - v_{j-1}(x_{j-1}) + w_{j-1} = f_j(x_{j-1}) + w_{j-1} = \dots = \sum_{k=1}^{j} f_k(x_{k-1})$.

Let $A = (a, w)$ be a stable assignment that is budget feasible and optimal with respect to a, and $x_0, \dots, x_m$ as specified in Claim 1. Defining the randomized assignment $R = (p_1, \dots, p_m)$ where $p_j = x_j - x_{j-1}$ for each $j \in [m]$ and comparing $SW(R)$ and $SW(A)$ will prove our theorem. □

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 assignments and stable assignments depend on the budget and the hospitals' costs, as shown in the full version [9]. As a future direction, it would be interesting to not only characterize the conditions under which the optimal stable assignment does better, but also quantify the ratio or difference between the social welfare of the two. Moreover, it is easy to see the FPTAS in Section 3.2 can be generalized to the setting of Section 4.3 with discrete patients. Finally, the conditions in Theorem 5 can also be interpreted in terms of MHR in the corresponding single-player Bayesian assignment problem.

Acknowledgments

We thank several anonymous reviewers for their comments.

[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., [8].

[10] Again, our approach works as long as each $f_j(x)$ is non-decreasing and piecewise differentiable.

REFERENCES

[1] A. Abdulkadiroğlu, P. A. Pathak, and A. E. Roth. The New York City High School Match. *American Economic Review*, 95(2):364–367, May 2005.

[2] S. Alaei, A. Malekian, and A. Srinivasan. On random sampling auctions for digital goods. *ACM Trans. Econ. Comput.*, 2(3):11:1–11:19, July 2014.

[3] S. Athey and G. Ellison. Position Auctions with Consumer Search. NBER Working Papers 15253, National Bureau of Economic Research, Inc, Aug. 2009.

[4] A. Bogomolnaia and H. Moulin. A New Solution to the Random Assignment Problem. *Journal of Economic Theory*, 100(2):295–328, October 2001.

[5] M. Braverman, J. Chen, and S. Kannan. Optimal provision-after-wait in healthcare. *Mathematics of Operations Research*, 41(1):352–376, 2016.

[6] P. Briest. Uniform budgets and the envy-free pricing problem. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming, Part I*, ICALP '08, pages 808–819, Berlin, Heidelberg, 2008. Springer-Verlag.

[7] P. Briest, S. Chawla, R. Kleinberg, and S. M. Weinberg. Pricing randomized allocations. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 585–597, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

[8] E. Budish, Y.-K. Che, F. Kojima, and P. Milgrom. Designing random allocation mechanisms: Theory and applications. *American Economic Review*, 103(2):585–623, 2013.

[9] H. Chan and J. Chen. Provision-after-wait with common preferences, full version. `http://www.cs.stonybrook.edu/~jingchen/`, 2016.

[10] S. Chawla, J. D. Hartline, and R. Kleinberg. Algorithmic pricing via virtual valuations. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, EC '07, pages 243–251, New York, NY, USA, 2007. ACM.

[11] S. Chawla, D. L. Malec, and B. Sivan. The power of randomness in Bayesian optimal mechanism design. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, EC '10, pages 149–158, New York, NY, USA, 2010. ACM.

[12] X. Chen, I. Diakonikolas, D. Paparas, X. Sun, and M. Yannakakis. The complexity of optimal multidimensional pricing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1319–1328. SIAM, 2014.

[13] N. R. Devanur, B. Q. Ha, and J. D. Hartline. Prior-free auctions for budgeted agents. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, EC '13, pages 287–304, New York, NY, USA, 2013. ACM.

[14] V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On profit-maximizing envy-free pricing. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 1164–1173, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[15] B. Q. Ha and J. D. Hartline. Mechanism design via consensus estimates, cross checking, and profit extraction. *ACM Trans. Econ. Comput.*, 1(2):8:1–8:15, May 2013.

[16] J. Hartline and Q. Yan. Envy, truth, and profit. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, pages 243–252, New York, NY, USA, 2011. ACM.

[17] A. S. Kelso and V. P. Crawford. Job Matching, Coalition Formation, and Gross Substitutes. *Econometrica*, 50(6):1483–1504, November 1982.

[18] S. G. Kolliopoulos and G. Steiner. Partially ordered knapsack and applications to scheduling. *Discrete Appl. Math.*, 155(8):889–897, Apr. 2007.

[19] 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, March 2006.

[20] J. Thanassoulis. Haggling over substitutes. *Journal of Economic Theory*, 117(2):217–245, August 2004.

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