Walrasian Equilibria in Markets with Small Demands

Argyrios Deligkas* Themistoklis Melissourgos† Paul G. Spirakis‡

Abstract

We study the complexity of finding a Walrasian equilibrium in markets where the agents have k-demand valuations. These valuations are an extension of unit-demand valuations where a bundle's value is the maximum of its k-subsets' values. For unit-demand agents, where the existence of a Walrasian equilibrium is guaranteed, we show that the problem is in quasi-NC. For $k = 2$, we show that it is NP-hard to decide if a Walrasian equilibrium exists even if the valuations are submodular, while for $k = 3$ the hardness carries over to budget-additive valuations. In addition, we give a polynomial-time algorithm for markets with 2-demand single-minded valuations, or unit-demand valuations.

1 Introduction

One of the most significant problems in market design is finding pricing schemes that guarantee good social welfare under equilibrium. Evidently, the most compelling equilibrium notion in markets with indivisible items is a Walrasian eqilibrium, henceforth WE, [35]: an allocation of items to the agents and a pricing, such that every agent maximizes her utility and all items are allocated. By the First Welfare Theorem, WE has the nice property of maximizing social welfare. The existence of WE seems to heavily rely on the class of valuation functions of the agents. When parameterized by the valuation function class, the existence of WE is (relatively) clear due to Gul and Stracchetti [21] and Milgrom [29]: WE are guaranteed to exist only in the class of gross substitutes valuation functions. Two of the most central and interesting problems regarding WE are:

(a) decide if a WE exists,

(b) compute a WE (if it exists).

We study the aforementioned two problems when valuation functions are parameterized by an integer k which denotes the maximum bundle size k for which every agent is interested. Such a class of *k-demand* valuation functions can be seen as an extension of the unit-demand functions, where each agent, for a given bundle X values only the most valuable k-subset of X. The main idea behind k-demand valuations is that every agent has some capacity for utilising the items that is either endogenously or exogenously imposed. There are several real-life examples where more than k items have the same value as k of them: a supervisor can effectively supervise up to a limited number of students; a grant investigator can efficiently work up to a limited number of projects; a sports team is allowed to have up to a small number of foreign players (or at least a small number of native players) in the squad; one can hang only a certain number of paintings on their house's walls. We investigate the complexity of the aforementioned problems when we are restricted to the intersection of the standard valuation classes and the k-demand classes. Our results contain hardness results as well as efficient algorithms.

As an example of the effect that k-demand valuation functions have on the complexity of these problems, we present *unbalanced markets*. In such markets the available items are significantly more

*Royal Holloway University of London, UK. Email: argyrios.deligkas@rhul.ac.uk
†Technical University of Munich, Germany. Email: themistoklis.melissourgos@tum.de
‡University of Liverpool, UK, and University of Patras, Greece. Email: p.spirakis@liverpool.ac.uk

than the agents, or vice versa. We provide an algorithm for the aforementioned problems parameterized by k. Complemented by a result of Rothkopf [31], this algorithm concludes that for constant k and appropriate unbalancedness, these problems are in P.

1.1 Contribution

In this work we study WE under their classic definition with no relaxation or approximation notions involved. We introduce a hierarchy of valuation functions, parallel to the already existing one. Our valuation functions are called k-demand and are a generalization of unit-demand with parameter k that determines at most how many items from a bundle the agent cares about. By definition, it is easy to see that the class of j-demand is included in $(j+1)$-demand for any $j \in [m-1]$. The purpose of considering valuation functions from the intersection of some k-demand class and some other known class, is to refine the complexity of the WE-related problem.

Algorithms and hardness results on the existence of WE and/or the problem of computing one in the current literature show an interesting dependence on the parameter k that we define here. For example, existence of WE is guaranteed in the well studied case of unit-demand valuation functions (i.e. $k = 1$), and a WE can be computed in polynomial time [16, 26]. Non-existence of WE is established in [32] by proving that even WinnerDetermination is NP-hard and this is achieved for valuation functions according to which the agents are only interested in at most 2 items (i.e. $k = 2$). Furthermore, non-existence of WE and NP-hardness of WinnerDetermination is proven for single-minded agents via a reduction to instances where agents are interested in at most 3 items (i.e. $k = 3$) [10]. For each of the above cases of k we give improved results: we supplement the "easy" case, where $k = 1$, with a quasi-NC algorithm[1], and the "hard" cases with stronger NP-hardness results in the sense that ours imply the existing ones.

Mixing the standard valuations' hierarchy and the k-demand hierarchy results to a two-dimensional landscape of valuation classes that aims to break down the complexity of the WE-related problems. For example, a possible result could be that below some threshold of k and below some standard valuation class, deciding WE existence is in P. Our results however indicate that this is not the case: even for $k = 2$ and submodular functions WinnerDetermination is NP-hard, and therefore deciding existence of WE is also NP-hard. This is an improvement over the result of Roughgarden and Talgam-Cohen [32], where NP-hardness is proven for $k = 2$ but general functions. Our reduction is entirely different than the one in [32], and in particular, it is from the problem "3-bounded 3-dimensional matching" to a market with n agents, m items and 2-demand submodular valuations. Furthermore, in [25] WinnerDetermination is proven to be weakly NP-hard for budget-additive functions by reducing "knapsack" to a market with 2 agents, and m items. We show that the problem is strongly NP-hard for k-demand budget-additive functions even for $k = 3$. The case $k = 2$ for the latter problem remains open.

On the positive side, we show a clear dichotomy for the problem of deciding WE existence with single-minded agents. It was proven in [10] that WinnerDetermination is NP-hard, via a reduction from "exact cover by 3-sets" to a market with single-minded agents who actually used 3-demand valuations. We show that WinnerDetermination is solvable in polynomial time for single-minded agents with 2-demand valuations by a reduction to the maximum weight matching problem. Then, by the decomposition shown at the end of Section 2, one can find a WE pricing via an LP (if such a pricing exists).

1.2 Related Work

Existense of Walrasian Equilibria. The most general class of valuation functions for which existence of WE is guaranteed has been proved by Gul and Stracchetti [21] and Milgrom [29] to be gross substitutes. Other valuation classes (that can be seen as special market settings) outside gross substitutes that guarantee WE existence have also been discovered, including the "tree valuations" in [7], and the valuation classes of [4, 8, 9]. Interestingly, the former admits also a polynomial time algorithm.

Non-existence of WE has been shown for many valuation classes, mostly by constructing an ad hoc market that does not identify some particular pattern as responsible for the non-existence (e.g. [21, 25, 13]). Roughgarden and Talgam-Cohen in [32] reprove some of these results and show a systematic way of proving non-existence of WE for more general valuation and pricing classes via standard

[1]This is the first parallel algorithm for computing WE to the authors' knowledge.

complexity assumptions. The latter paper shows the remarkable relation between computability of seemingly arbitrary problems and existence of equilibria in markets. In fact, one of their results states that if for some class $\mathcal{V}$ of valuation functions WinnerDetermination is computationally harder than finding the demand for each agent, then there exist instances in $\mathcal{V}$ with no WE.

Computation of Walrasian Equilibria. On the computational side, in markets that do not guarantee existence of WE, the problem of deciding existence is NP-hard for all the most important valuation classes. This has been established by proving that WinnerDetermination for budget-additive valuations is NP-hard via the "knapsack" problem in [25] and via the strongly NP-hard problem "bin packing" in [32]. By the fact that a WE corresponds to an optimal allocation, it is immediate that existence of WE is at least as hard as WinnerDetermination. Since budget-additive functions are a subset of submodular functions, it seems that as soon as valuation functions are allowed to be more general than the class of gross substitutes, i.e. submodular, the problem is already NP-hard. Also, for the class of single-minded agents (which is incomparable to the rest of the classes), WinnerDetermination is NP-hard [10]. On the positive side, Rothkopf et al. [31] provide several classes of valuation functions where WinnerDetermination can be efficiently solved. In particular, they show that when there are logarithmically many items with respect to the number of agents, WinnerDetermination can be efficiently solved via dynamic programming. Sandholm [33] provides a comparison of several different methods for WinnerDetermination and experimentally evaluates them. It is also worth mentioning the "tollbooth" problem on trees, defined in [22] (see also [11]), for which, even though WE existence is not guaranteed, finding one (if it exists) is in P.

Relaxations/Approximations. Due to [21] and [29], existence of WE is guaranteed only in a restrictive class of functions, namely *gross substitutes*. This fact has ignited a line of works that, in essence, question the initially defined WE as being the equilibrium that occurs in actual markets. These works consider relaxed or approximate versions of WE. Some of the most interesting results on such relaxations are the following:

- If only 2/3 of the agents are required to be utility maximizers then a *relaxed Walrasian equilibrium* exists for single-minded agents ([10, 11]).
- If the seller is allowed to package the items into indivisible bundles prior to sale, not all items have to be sold, and additionally only half of the optimal social welfare is required (*Combinatorial Walrasian equilibrium*) then such an equilibrium exists for general valuation functions and can be found in polynomial time ([18]).
- If agents exhibit *endowment effect*, meaning that the agents' valuations for a bundle they already possess is multiplied by a factor a, then for any $a \geq 2$ there exists an *a-endowed equilibrium* for the class of submodular functions ([2]). For stronger notions of endowment, endowed equilibria exist even for XOS functions, and additionally, bundling guarantees equilibria for general functions ([17]).

Other works have also considered special classes of valuations that have as parameter the cardinality of the valuable bundles ([13] and [12]). However these valuation functions are not identical to ours. In [13] the valuation function of each agent, called *k-wise dependent*, is encoded in a hypergraph whose vertices are the items and each hyperedge has a positive or negative weight that determines the additional value of the bundle in case all of its adjacent vertices are a subset of the bundle. This class of valuations is incomparable to ours by definition. The model of [12] is the same as that of [13], as argued in the latter. Recently, Berger et al. in [5] introduced a hierarchy of valuation functions similar to ours, called "k-demand" that also generalize unit-demand functions. The same definition of functions appears also in [14]. However, those are a special case of our k-demand functions (i.e. also additive), and in fact they are gross substitutes.

The paper is organized in sections so that each deals with a particular value or group of values for k. We study unit-demand valuations in Section 3, 2-demand valuations in Section 4, 3-demand valuations in Section 5, and k-demand valuations for constant k and unbalanced markets in Section 6. We conclude with a discussion in Section 7.

2 Walrasian Equilibria and Valuation Functions

We consider markets with a set N of n agents and a set M of m items. Every agent i has a valuation function $v_i : 2^M \to \mathbb{R}_{\geq 0}$; for every subset, or bundle, of items $X \subseteq M$ agent i has value $v_i(X)$. A valuation function v_i is *monotone* if $X \subseteq Y$ implies $v_i(X) \leq v_i(Y)$, and it is *normalized* if $v_i(\emptyset) = 0$. In what follows, we assume that all the agents have monotone and normalized valuation functions.

There are many different valuation functions studied over the years and we focus on several of them.[2]

- Unit-demand (UD): for agent i there exist m values $v_{i1}, \ldots, v_{im}$ and $v_i(X) = \max_{j \in X} v_{ij}$, for every $X \subseteq M$.
- Additive (AD): for agent i there exist m values $v_{i1}, \ldots, v_{im}$ and $v_i(X) = \sum_{j \in X} v_{ij}$, for every $X \subseteq M$.
- Budget-additive (BA): for every agent i there exist $m+1$ values $v_{i1}, \ldots, v_{im}, B_i$, such that for every $X \subseteq M$ it is $v_i(X) = \min \left\{ B_i, \sum_{j \in X} v_{ij} \right\}$.
- Single-minded (SMi): for agent i there exist a set $X_i \subseteq M$ and a value B_i, such that $v_i(X) = B_i$, if $X_i \subseteq X$, and $v_i(X) = 0$, otherwise.
- Submodular (SubM): for agent i and every two sets of items X and Y it holds $v_i(X) + v_i(Y) \geq v_i(X \cup Y) + v_i(X \cap Y)$.
- Fractionally subadditive (XOS): for every agent there exist vectors $v_{i1}, \ldots v_{ik} \in \mathbb{R}^m$ and $v_i(X) = \max_{j \in [k]} \sum_{l \in X} v_{ik}(l)$, for every $X \subseteq M$.
- Subadditive (SubA): for agent i and every two sets of items X and Y it holds $v_i(X) + v_i(Y) \geq v_i(X \cup Y)$.

We will focus on constrained versions of the aforementioned valuation functions, where the cardinality of the sets an agent has value for is bounded by k. k-demand valuations naturally generalize unit-demand valuations, but, at the same time, they keep the structure of more complex valuation functions.

Definition 1 (k-demand valuation)**.** *A valuation function $v : 2^m \to \mathbb{R}_{\geq 0}$ is k-demand if for every bundle $X \subseteq M$ it holds that*

$$v(X) = \max_{\substack{X' \subseteq X \\ |X'| \leq k}} v(X').$$

A very important remark is that when k is constant the problems have succinct representation, namely polynomial in the number of agents and items, i.e. $\Theta\left(n \cdot m^k \cdot \log V\right)$, where V is the maximum valuation among all bundles and among all agents. This makes our setting computationally interesting and also removes the need for access to some *value oracle* or *demand oracle*: the former takes as input a bundle and returns its value, and the latter, for some indicated agent, takes a pricing as input and outputs the most preferable bundles for the agent. Having such oracles when k is constant is redundant since there are only $\sum_{j=1}^{k} \binom{m}{j} \in \Theta(m^k)$ many j-subsets of M, $j \leq k$, and an agent just needs to declare a value for each; then the algorithm with this input can compute in polynomial time the value of the agent for any bundle. Also, a demand oracle is not needed since, for a given pricing, one can compute efficiently the prices of all $\sum_{j=1}^{k} \binom{m}{j}$ bundles (these are the only ones that can maximize the utility of an agent; by considering a bundle Y with more than k items, its value will correspond to a bundle X with k items, but $p(Y) \geq p(X)$), and then (efficiently) search through them to find which ones yield the maximum utility to the agent. In contrast, a great line of works has studied the complexity of the WE-related problems, provided that value oracles and demand oracles are available (e.g. [6, 30, 21, 15, 27]).

An *allocation* $S = (S_0, S_1, \ldots, S_n)$ is a partition of M to $n+1$ disjoint bundles, where agent $i \in [n]$ gets bundle S_i. Items in S_0 are not allocated to any agent. The *social welfare* of allocation S is defined as $SW(S) = \sum_{i \in [n]} v_i(S_i)$. An allocation S is *optimal* if it maximizes the social welfare, i.e.,

[2]When we refer to a valuation function as *general* we mean that the value for any bundle does not depend on other bundles' values. It is clear that the set of general functions contains all other classes of functions.

$SW(S) \geq SW(S')$, for every possible allocation S'. A *pricing* $p = (p_1, \ldots, p_m)$ defines a price for every item, where $p_j \geq 0$ is the price of item j. For $X \subseteq M$, we denote $p(X) = \sum_{j \in X} p_j$. Given an allocation S and a pricing p, the *utility* of agent i is

$$u_i(S, p) := v_i(S_i) - p(S_i).$$

The *demand correspondence* of agent i with valuation v_i under pricing p, denoted $D(v_i, p)$, is the set of items that maximize the utility of the agent; formally $D(v_i, p) := \{S \subseteq M : u_i(S, p) \geq u_i(T, p) \text{ for all } T \subseteq M\}$. Any element of $D(v_i, p)$ is called *demand set* of agent i.

Definition 2 (Gross substitutes (GS)[24])**.** *A valuation function satisfies the gross substitutes property when for any price vectors $p \in \mathbb{R}^m$ and $S \in D(v, p)$, if p' is a price vector $p \leq p'$ (meaning that for all $l \in S$, $p_l \leq p'_l$), then there is a set $S' \in D(v, p')$ such that $S \cap \{j; p_j = p'_j\} \subseteq S'$.*

Intuitively, a valuation is gross substitute if after the increase of the prices of some items in some demand set S of an agent, the agent still has a demand set S' that contains the items with unchanged prices.

It is known that UD $\subset$ BA $\subset$ SubM, that AD $\subset$ GS $\subset$ SubM, and finally that SubM $\subset$ XOS $\subset$ SubA. Furthermore, SMi valuation functions are not contained in any of these valuation classes.

Definition 3 (Walrasian Equilibrium)**.** *An allocation $S = (S_0, S_1, \ldots, S_n)$ and a pricing $p = (p_1, \ldots, p_m)$ form a* Walrasian equilibrium *(WE), if the following two conditions hold.*

1. *For every agent i and any bundle $X \subseteq M$ it holds that $v_i(S_i) - p(S_i) \geq v_i(X) - p(X)$.*
2. *For every item $j \in S_0$ it holds that $p_j = 0$.*

Walrasian	
Input:	A market with n agents and m items, and a valuation function for each agent.
Task:	Decide whether the market possesses a Walrasian equilibrium, and if it does, compute one.

The *First Welfare Theorem* states that for any Walrasian equilibrium (S, p), partition S corresponds to an optimal allocation [26]. Hence, walrasian can be decomposed into the following two problems.

WinnerDetermination	
Input:	A market with n agents and m items, and a valuation function for each agent.
Task:	Find an optimal allocation S^* for the items.

WalrasianPricing	
Input:	A market with n agents and m items, a valuation function for each agent, and an optimal allocation S^*.
Task:	Find a pricing vector p such that (S^*, p) is a Walrasian equilibrium, or decide that there is no Walrasian equilibrium for the instance.

This decomposition highlights that a WE exists if and only if there exists a pricing vector p that satisfies the conditions of Definition 3 for any optimal allocation $S^* = (S^*_0, S^*_1, \ldots, S^*_n)$.

For k-demand valuation functions, the conditions of Definiton 3 (and therefore a solution to WalrasianPricing) correspond to the solution of the following linear system of m variables and $n \cdot \sum_{j=1}^{k} \binom{m}{j} + m$ equality/inequality constraints, where each constraint has at most $2k$ variables.

$$\begin{aligned} v_i(S^*_i) - p(S^*_i) \geq v_i(X) - p(X), &\quad \forall X \subseteq M, \text{ where } |X| \leq k, \forall i \in N \\ p_j \geq 0, &\quad \forall j \notin S^*_0 \\ p_j = 0, &\quad \forall j \in S^*_0. \end{aligned} \tag{1}$$

Note that when k is a constant, as mentioned earlier, the above constraints are $n \cdot \sum_{j=1}^{k} \binom{m}{j} + m$ which is at most linear in n and polynomial in m, since

$$\sum_{j=1}^{k} \binom{m}{j} \leq \sum_{j=1}^{k} \frac{m^j}{j!} \leq \sum_{j=1}^{k} \frac{k^j}{j!} \cdot \left(\frac{m}{k}\right)^j \leq e^k \cdot \sum_{j=1}^{k} \left(\frac{m}{k}\right)^j \leq e^k \cdot \left(\frac{m}{k}\right)^k.$$

We conclude that, for constant k, a solution to linear system (1) (and thus, WalrasianPricing) can be found in time polynomial in n and m by formulating it as an LP with objective function set to a constant. So, the problem of deciding the existence of WE and the problem of computing one (if it exists) essentially reduce to finding an optimal allocation S^*, i.e. WinnerDetermination. In Sections 4, 5, 6 we exploit the aforementioned fact and only investigate the complexity of WinnerDetermination.

3 Unit-demand Valuation Functions

The simplest case of markets is when the agents have unit-demand valuation functions. The existence of WE in this class of markets was shown in the seminal paper of Demange, Gale, and Sotomayor [16] via an algorithm that resembles the tâtonnement process. This algorithm is pseudopolynomial in general, and polynomial when the values of the agents are bounded by some polynomial. In [26] an algorithm (Algorithm 1) is presented and it is shown that a modification of it finds a WE in time $O(m^2 n + m^4 \log V)$, where V is the maximum valuation of any item across all agents.

In this section we show that walrasian in these markets is in quasi-NC. The complexity class quasi-NC is defined as quasi-$\mathtt{NC} = \mathbb{U}_{k\geq 0}$quasi-$\mathtt{NC}^k$, where quasi-$\mathtt{NC}^k$ is the class of problems having uniform circuits of quasi-polynomial size, $n^{\log^{O(1)} n}$, and polylogarithmic depth $O(\log^k n)$ [3]. Here "uniform" means that the circuit can be generated in polylogarithmic space. Put differently, quasi-NC contains problems that can be solved in polylogarithmic parallel time using quasi-polynomially many processors with shared memory.

In this class of markets WinnerDetermination can be reduced to a maximum weight matching on a complete bipartite graph. On the left side of the graph there exist n nodes corresponding to the agents, on the right side there are m nodes corresponding to the items and the weight of the edge (i, j) equals to the value of agent i for item j. The recent breakthrough of Fenner, Gurjar, and Thierauf [19] states that the maximum weight *perfect* matching in bipartite graphs is in quasi-NC when the edge-weights are bounded by some polynomial; later Svensson and Tarnawski [34] extended this result for general graphs. Thus, if we augment the bipartite graph that corresponds to the market by adding dummy items with zero value for every agent, or dummy agents with zero value for every item, we can guarantee that it contains a perfect matching without changing any optimal allocation. Then, we can use the algorithm of [19] and compute an optimal allocation in polylogarithmic time.

Given an optimal allocation, WalrasianPricing for these markets has a special structure. It is a linear feasibility problem with polynomially many inequalities and at most two variables per inequality. For this special type of feasibility systems there exists a quasi-NC algorithm [28].

Theorem 4. walrasian *in unit-demand markets with polynomial valuations is in quasi-*NC.

Proof. When shared memory is available, as in quasi-NC, we can solve WinnerDetermination in polylogarithmic parallel time via the algorithm of [19] and store it in the shared memory. Then, the processors will read the solution, build the linear system for WalrasianPricing and solve it in polylogarithmic time via the the algorithm of [28] on the shared memory. Hence, the composition of the two algorithms can be done in polylogarithmic time using quasi-polynomially many processors. □

Our result suggests a parallel algorithm that needs $O(\log^3(n))$ time which is significantly faster than any serial algorithm. On the other hand though, it requires $n^{\log(n)}$ processors in the worst case. We observe that this is the current best possible result, since any improvement would imply better parallel algorithms for other important problems like maximum weight matching and feasibility of systems with linear inequalities. We have to state though that it is open whether both aforementioned problems are in NC. On the other hand, it is known that the maximum weight problem in graphs with polynomial weights is in pseudo-deterministic RNC [1, 20]. Hence, a first improvement would be to place WalrasianPricing in pseudo-deterministic RNC.

4 2-demand Valuation Functions

In this section we resolve the complexity of deciding existence of WE for 2-demand valuation functions. As an example, consider the case where the football teams need to have at least 2 young native players in their squad. Each team knows exactly which pair of players wants and it does not want more young players due to capacity constraints. A version of 2-demand valuations, termed *pair-demand* valuations, was studied in [32], where every agent i has a value $v_i(j,k)$ for every pair of items and the value of i for a bundle S is $v_i(S) = \max_{j,k\in S} v_i(j,k)$. These are general valuation functions that can allow complementarities. We strengthen the results of [32] and prove that WinnerDetermination is NP-hard even when the valuation functions of the agents are 2-demand submodular and every agent has positive value for at most six items.

Theorem 5. WinnerDetermination *is strongly* NP*-hard even for 2-demand submodular functions.*

Proof. We reduce from 3-bounded 3-dimensional matching, termed 3DM(3). The input of a 3DM(3) instance consists of three sets X, Y, Z, where $|X| = |Y| = |Z|$, and a set S of triplets (hyperedges) (x, y, z) where $x \in X, y \in Y$, and $z \in Z$. In addition, every element of X, Y, Z appears in at most three triplets and every triplet shares at most one element with any other triplet. The task is to decide if there is a subset of non-intersecting triplets of S of cardinality $|X|$. The problem is known to be NP-complete [23].

For every element $x \in X$ we create an agent and for every element in $Y \cup Z$ we create an item. Let S_{x_j} denote the set of items that correspond to the jth triplet of S that x belongs to. Recall that there exist at most three such triplets. In addition, since any two triplets of S share at most one element, we have that S_{x_j}s are disjoint. Moreover, let S_x be the union of the elements from S_{x_j}s. Then, the valuation function of agent x for a subset of items T is defined as follows:

- $v_x(T) = 2$, if T contains some S_{x_j};
- $v_x(T) = 0$, if $|T \cap S_x| = 0$;
- $v_x(T) = 1$, if $|T \cap S_x| = 1$;
- $v_x(T) = 1.5$, if $|T \cap S_x| \geq 2$ and T does not contain any S_{x_j}.

Observe that if $|T \cap S_x| > 3$, then T will contain some S_{x_j}, hence the definition of the valuation function is complete. It is not hard to verify that v_x is indeed a 2-demand submodular function.

We claim that there is an allocation with welfare $2|X|$ if and only if the 3DM(3) instance is satisfiable. Firstly, assume that indeed the 3DM(3) instance has a solution S', i.e., S' contains $|X|$ non intersecting triplets in S. Then, if the triplet (x, y, z) belongs to S' we allocate the items that correspond to y and z to the agent that corresponds to x and the agent has value 2 for the bundle. Clearly, the allocation achieves welfare $2|X|$. For the other direction, assume that there is an allocation for the items with welfare $2|X|$. This means that every agent gets utility 2 from her allocated bundle. Then, by construction, each agent x alongside her allocated bundle corresponds to a triplet from S. Observe, that the allocation consists of non-overlapping bundles, hence we get $|X|$ non intersecting triplets in S. □

Theorem 5 implies that walrasian is NP-hard for any class of valuation functions that contains the class of 2-demand submodular valuations.

Corollary 6. walrasian *is strongly* NP*-hard even if all the agents have 2-demand submodular valuation functions.*

Closing the gap in single-minded valuations. In addition to the above hardness results we study single-minded agents with 2-demand valuations and we show that in this case walrasian is easy, contrary to the case of 3-demand valuations where it is NP-hard [10]. To prove this, for agents that are single-minded for bundles of size 2, we reduce WinnerDetermination to a maximum weight matching problem over a graph G. Every item corresponds to a vertex of G. For every pair of items that is the most preferable by an agent we create the corresponding edge with weight the value of the agent for the items; if there are more than one agents that want the same pair of items we keep only the weight for the highest valuation. Clearly, any maximum weight matching corresponds to an optimal allocation.

Next we show how to handle instances where every agent is either unit-demand or *multi-minded over a subset of size 2.* Recall, a unit-demand agent-might have positive value for various items. An agent i is multi-minded over a subset of size 2, if there exist items a_i and b_i and the agent has positive values only for the following three bundles: $\{a_i\}$, $\{b_i\}$, and $\{a_i, b_i\}$. Observe the this is a strict generalisation of 2-demand single minded. To achieve this, we extend the construction described above as follows. For every multi-minded agent i we add a new vertex x_i and the edges (x_i, a_i), (x_i, b_i), with weights $v_i(a_i)$ and $v_i(b_i)$ respectively. For every unit-demand agent i, we add a new vertex y_i and the edges (y_i, j), where j is a vertex that corresponds to item j, with weight $v_i(j)$; i.e. equal to the agent's value for item j. Again, a maximum weight matching for the constructed graph corresponds to an optimal allocation.

Theorem 7. WALRASIAN *is in* P *for markets where every agent is unit-demand or multi-minded over a subset of size 2.*

5 3-demand Valuation Functions

In this section we prove strong NP-hardness for WinnerDetermination for 3-demand budget-additive valuation functions. As an example for 3-demand budget additive valuation, we can think of departments within a university that want to hire staff members for their labs. The agents are the departments, the items are the staff members, and the available resources of each department's lab defines the budget. The value a department gets from a candidate equals the quantity of the resources the candidate is capable of utilizing. The department is allowed to hire at most 3 staff members, due to regulations imposed by the university.

Theorem 8. WinnerDetermination *is strongly* NP*-hard even when all the agents have identical 3-demand budget-additive valuation functions.*

Proof. We prove the theorem with a reduction from 3-partition. An instance of 3-partition consists of a multiset of $3n$ positive integers $a_1, a_2, \ldots, a_{3n}$ summing up to S. The question is whether the multiset can be partitioned into n triplets such that the elements of each triplet sum up to $B = \frac{S}{n}$. So, given an instance of 3-partition we create a WinnerDetermination instance with n agents and $3n$ items. All the agents have the same 3-demand budget-additive valuation: they have value a_i for item i and budget B.

The question we would like to decide is whether there exists an allocation with social welfare $n \cdot B$. It is not hard to see that if there is a solution to 3-partition, then there exists an allocation for WinnerDetermination with social welfare $n \cdot B$. On the other hand, observe that, due to the budget-additive valuations, social welfare $n \cdot B$ for the instance can be achieved only when there exists an allocation where every agent gets value B. In addition, since the agents have 3-demand valuation functions it means that any allocation that maximizes the social welfare, without loss of generality, allocates exactly three items to every agent; otherwise some agent gets more than 3 items and value gets wasted since, by definition of 3-demand valuation, the agent will only appreciate the 3 most valuable items. Hence, if there exists an allocation for the constructed instance with social welfare $n \cdot B$, necessarily, every agent gets exactly 3 items whose values sum up to B. This allocation trivially defines a solution to 3-partition. □

Corollary 9. WALRASIAN *is strongly* NP*-hard even if all the agents have identical 3-demand budget-additive valuation functions.*

6 Constant-demand Valuation Functions

In this section we study markets where the agents have k-demand valuation functions, where k is constant. Our results from the previous sections imply that deciding the existence of a WE is NP-hard even when $k = 2$ and the valuation functions are submodular. In addition, we showed that the problem is NP-hard for $k = 3$ even for budget-additive valuations [13]. This means that in order to get efficient algorithms we have to further restrict our market design in markets that retain constant demand k, but with either reduced number of agents, or reduced number of items. For this reason, we study *unbalanced* markets. A market is *unbalanced* if the number of available items is significantly larger than the number of agents, formally, $m \in \omega(n)$, or the other way around, $n \in \omega(m)$. For the case where, $m = O(\log n)$ and any k the dynamic programming approach of Rothkopf [31] solves WinnerDetermination in $O(n^3)$. Next we show a result for the case where the market is unbalanced in the opposite direction.

Theorem 10. *In markets with k-demand valuations, n agents and m items, where k and n are constant,* WinnerDetermination *is in* P*.*

Proof. We consider the unbalanced market where the number of available items m is a lot greater than the number of items $k \cdot n$ to be allocated. The number $k \cdot n$ comes from the fact that in an optimum allocation, not more than $k \cdot n$ items will be appreciated by the agents (by definition of the k-demand valuation function). Therefore, allocating more than these items does not improve the social welfare, thus, does not yield additional WE. In this case, we can find all possible subsets of size $k \cdot n$ of items, that is, all candidate sets of items to be allocated to the agents. Formally, we consider the set $I := \{L \subseteq M \mid |L| = k \cdot n\}$ that consists of all $(k \cdot n)$-subsets of M. It is $|I| = \binom{m}{k \cdot n} \in O((m - k \cdot n)^{k \cdot n})$, which is a polynomial in m when k and n are constant.

Observe now that, given a subset L of items with size $k \cdot n$, one can construct a $k+1$-uniform hypergraph, i.e. a hypergraph all of whose hyperedges have size $k+1$, in the following way. Have its vertex set be $L \cup N$, and for every k-subset L_k of L have a hyperedge $L_k \cup \{i\}$ for every $i \in N$. Also, assign to each hyperedge a weight equal to the valuation of agent i for the item bundle L_k, namely $v_i(L_k)$. On this graph one can run a brute-force algorithm to find a maximum weight $(k+1)$-dimensional matching in constant time, since the graph is of constant size. Then, by repeating the same routine for all $(k \cdot n)$-subsets of I in time polynomial in m, we pick the one that yields the maximum sum of weights in the matching. The optimal allocation of items to agents corresponds to the aforementioned optimum matching. The running time of this algorithm is $O(m^c)$ for some constant c, i.e. polynomial in the input size, since the input size is $\Omega\left(n \cdot \binom{m}{k} \cdot \log V\right)$ bits, where $V := \max_{\substack{i \in N \\ X \subseteq M}} v_i(X)$; that is because every agent has to declare how much her valuation is for every k-subset of items. $\square$

Corollary 11. *In markets with k-demand valuations, n agents and m items, where k and n are constant,* walrasian *is in* P*.*

7 Discussion

In this paper we study the complexity of computing Walrasian equilibria in markets with k-demand valuations. As we show, even for the smallest possible value for k, the problem of deciding WE existence remains NP-hard for the next greater well-studied class of valuations outside gross substitutes (submodular). Hence, we turn to the study of unbalanced markets and present a polynomial-time algorithm for k-demand general valuations, where k is constant.

For markets with $k = 1$, known as "matching markets", we prove that the problem is in quasi-NC. We view this as a very interesting result since all the known algorithms for the problem are highly sequential. Can we design an NC algorithm for the problem via a form of a simultaneous auction? This would be remarkable since it would imply that bipartite weighted matching is in NC. For $k = 2$ we show that WinnerDetermination is intractable even for submodular functions, and for $k = 3$ the hardness remains for an even stricter class, namely budget-additive functions. In order to completely resolve the complexity of 2-demand valuations, it remains to solve WinnerDetermination for 2-demand budget-additive valuations. Is the problem NP-hard, or is there a polynomial time algorithm for it? Answering this question would provide a complete dichotomy for the complexity of the problems WinnerDetermination and also Walrasian. For unbalanced markets with constant k, we covered the cases $m \in O(\log n)$ and $n \in \Theta(1)$. Are there efficient algorithms for any $n \in \omega(m)$ and $m \in \omega(n)$? Another very intriguing direction is to study approximate Walrasian equilibria. The recent results of Babaioff, Dobzinski, and Oren [2] and of Ezra, Feldman, and Friedler [17] propose some excellent notions of approximation. Can we get better results if we assume k-demand valuations?

Acknowledgements

The work of the second author was supported by the Alexander von Humboldt Foundation with funds from the German Federal Ministry of Education and Research (BMBF). The work of the third author was partially supported by the NeST initiative of the School of EEE and CS at the University of Liverpool, and by the EPSRC grant EP/P02002X/1.

References

[1] Nima Anari and Vijay V Vazirani. Matching is as easy as the decision problem, in the nc model. In *Proceedings of ITCS*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[2] Moshe Babaioff, Shahar Dobzinski, and Sigal Oren. Combinatorial auctions with endowment effect. In *Proc. of EC*, pages 73–90, 2018.

[3] DA Mix Barrington. Quasipolynomial size circuit classes. In *Proc. of the Seventh Annual Structure in Complexity Theory Conference*, pages 86–93. IEEE, 1992.

[4] Oren Ben-Zwi, Ron Lavi, and Ilan Newman. Ascending auctions and walrasian equilibrium. *CoRR*, abs/1301.1153, 2013.

[5] Ben Berger, Alon Eden, and Michal Feldman. On the power and limits of dynamic pricing in combinatorial markets. *CoRR*, abs/2002.06863, 2020.

[6] Sushil Bikhchandani and John W Mamer. Competitive equilibrium in an exchange economy with indivisibilities. *Journal of economic theory*, 74(2):385–413, 1997.

[7] Ozan Candogan, Asuman Ozdaglar, and Pablo A. Parrilo. Iterative Auction Design for Tree Valuations. *Operations Research*, 63(4):751–771, 2015.

[8] Ozan Candogan, Asuman E. Ozdaglar, and Pablo A. Parrilo. Iterative auction design for graphical valuations part ii : General graphs. 2014.

[9] Ozan Candogan and Sasa Pekec. Efficient allocation and pricing of multifeatured items. *Management Science*, 64:5521–5543, 2018.

[10] Ning Chen, Xiaotie Deng, and Xiaoming Sun. On complexity of single-minded auction. *Journal of Computer and System Sciences*, 69(4):675–687, 2004.

[11] Ning Chen and Atri Rudra. Walrasian equilibrium: Hardness, approximations and tractable instances. *Algorithmica*, 52(1):44–64, 2008.

[12] Yann Chevaleyre, Ulle Endriss, Sylvia Estivie, and Nicolas Maudet. Multiagent resource allocation in k-additive domains: Preference representation and complexity. *Annals of Operations Research*, 163(1):49–62, 2008.

[13] Vincent Conitzer, Tuomas Sandholm, and Paolo Santi. Combinatorial auctions with k-wise dependent valuations. In *Proc. of AAAI*, volume 5, pages 248–254, 2005.

[14] Vincent Conitzer and Hanrui Zhang. Learning the valuations of a k-demand agent. 2019.

[15] Sven de Vries, James Schummer, and Rakesh V Vohra. On ascending vickrey auctions for heterogeneous objects. *Journal of Economic Theory*, 132(1):95–118, 2007.

[16] Gabrielle Demange, David Gale, and Marilda Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.

[17] Tomer Ezra, Michal Feldman, and Ophir Friedler. A general framework for endowment effects in combinatorial markets. *arXiv preprint arXiv:1903.11360*, 2019.

[18] Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial walrasian equilibrium. *SIAM Journal on Computing*, 45(1):29–48, 2016.

[19] Stephen Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-NC. *SIAM Journal on Computing*, (0):STOC16–218, 2019.

[20] Shafi Goldwasser and Ofer Grossman. Bipartite perfect matching in pseudo-deterministic nc. In *Proc. of ICALP*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[21] Faruk Gul and Ennio Stacchetti. Walrasian Equilibrium with Gross Substitutes. *Journal of Economic Theory*, 87(1):95–124, 1999.

[22] Venkatesan Guruswami, Jason D Hartline, Anna R Karlin, David Kempe, Claire Kenyon, and Frank McSherry. On profit-maximizing envy-free pricing. In *Proc. of SODA*, pages 1164–1173, 2005.

[23] Viggo Kann. Maximum bounded 3-dimensional matching is max snp-complete. *Information Processing Letters*, 37(1):27–35, 1991.

[24] Alexander S. Kelso and Vincent P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica*, 50(6):1483–1504, 1982.

[25] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, May 2006.

[26] Renato Paes Leme. Gross substitutability: An algorithmic survey. *Games and Economic Behavior*, 106:294–316, 2017.

[27] Renato Paes Leme and Sam Chiu-wai Wong. Computing walrasian equilibria: Fast algorithms and structural properties. *Mathematical Programming*, pages 1–42, 2018.

[28] George S Lueker, Nimrod Megiddo, and Vijaya Ramachandran. Linear programming with two variables per inequality in poly-log time. In *Proc. of STOC*, pages 196–205, 1986.

[29] Paul Milgrom. Putting Auction Theory to Work: The Simultaneous Ascending Auction. *Journal of Political Economy*, 108(2):245–272, 2000.

[30] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129(1):192–224, 2006.

[31] Michael H Rothkopf, Aleksandar Pekeč, and Ronald M Harstad. Computationally manageable combinational auctions. *Management science*, 44(8):1131–1147, 1998.

[32] Tim Roughgarden and Inbal Talgam-Cohen. Why prices need algorithms. In *Proceedings of the sixteenth acm conference on economics and computation*, pages 19–36, 2015.

[33] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1):1 – 54, 2002.

[34] Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-nc. In *Proc. of FOCS*, pages 696–707, 2017.

[35] Léon Walras. *Éléments d'économie politique pure, ou, Théorie de la richesse sociale*. F. Rouge, 1896.