Computing a Walrasian Equilibrium in Iterative Auctions with Multiple Differentiated Items

Kazuo Murota[1], Akiyoshi Shioura[2], and Zaifu Yang[3]

[1] University of Tokyo, Tokyo 113-8656, Japan, murota@mist.i.u-tokyo.ac.jp
[2] Tohoku University, Sendai 980-8579, Japan, shioura@dais.is.tohoku.ac.jp
[3] University of York, York YO10 5DD, UK, zaifu.yang@york.ac.uk

Abstract. We address the problem of computing a Walrasian equilibrium price in an ascending auction with gross substitutes valuations. In particular, an auction market is considered where there are multiple differentiated items and each item may have multiple units. Although the ascending auction is known to find an equilibrium price vector in finite time, little is known about its time complexity. The main aim of this paper is to analyze the time complexity of the ascending auction globally and locally, by utilizing the theory of discrete convex analysis. An exact bound on the number of iterations is given in terms of the ℓ_∞ distance between the initial price vector and an equilibrium, and an efficient algorithm to update a price vector is designed based on a min-max theorem for submodular function minimization.

1 Introduction

We study an ascending auction, where given a set of discrete (or indivisible) items, the auctioneer aims to find an efficient allocation of items to bidders as well as market clearing prices of the items (see [5, 6] for surveys). In recent years, there has been a growing use of iterative auctions for items such as spectrum licenses in telecommunication, electrical power, landing slots at airports, etc. In this paper, we consider the setting where there are multiple indivisible items for sale and each item may have multiple units; this is more general than the single-unit setting used extensively in the literature. A fundamental concept in auctions is the Walrasian equilibrium (or competitive equilibrium), which is a pair of a price vector and an allocation of items satisfying a certain property (see below for the precise definition). The main aim of this paper is to analyze the problem of computing a Walrasian equilibrium with respect to the time complexity, by utilizing the theory of discrete convex analysis.

Multi-Item Auction and Walrasian Equilibrium The auction market model is formulated as follows. In the market, there are n types of items or goods, denoted by $N = \{1, 2, \ldots, n\}$, and m bidders, denoted by $M = \{1, 2, \ldots, m\}$. We have $u(i) \in \mathbb{Z}_+$ units available for each item $i \in N$. The case with $u(i) = 1$

($i \in N$) is referred to as the *single-unit auction* in this paper. We denote the integer interval as $[\mathbf{0}, u]_{\mathbb{Z}} = \{x \in \mathbb{Z}^n \mid \mathbf{0} \leq x \leq u\}$; note that $[\mathbf{0}, \mathbf{1}]_{\mathbb{Z}} = \{0, 1\}^n$. Each vector $x \in [\mathbf{0}, u]_{\mathbb{Z}}$ is often called a *bundle*; a bundle x corresponds to a (multi)-set of items, where $x(i)$ represents the multiplicity of item $i \in N$. Each bidder $j \in M$ has his valuation function $f_j : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{R}$; the value $f_j(x)$ represents the degree of satisfaction for a bundle x. An *allocation* of items is defined as a set of bundles $x_1, x_2, \ldots, x_m \in [\mathbf{0}, u]_{\mathbb{Z}}$ satisfying $\sum_{j=1}^{m} x_j = u$.

In an auction, we want to find an efficient allocation and market clearing prices. Given a price vector $p \in \mathbb{R}^n$, each bidder $j \in M$ wants to have a bundle x which maximizes the value $f_j(x) - p^\top x$. For $j \in M$ and $p \in \mathbb{R}^n$, define

$$V_j(p) = \max\{f_j(x) - p^\top x \mid x \in [\mathbf{0}, u]_{\mathbb{Z}}\}, \tag{1}$$

$$D_j(p) = \arg\max\{f_j(x) - p^\top x \mid x \in [\mathbf{0}, u]_{\mathbb{Z}}\}. \tag{2}$$

We call the function $V_j : \mathbb{R}^n \to \mathbb{R}$ and the set $D_j(p) \subseteq [\mathbf{0}, u]_{\mathbb{Z}}$ an *indirect utility function* and a *demand set*, respectively. On the other hand, the auctioneer wants to find a price vector under which all items are sold. Hence, all of the auctioneer and bidders are happy if we can find a pair of a price vector p^* and an allocation $x_1^*, x_2^*, \ldots, x_m^*$ satisfying the condition that $x_j^* \in D_j(p^*)$ for $j \in M$. Such a pair is called a *Walrasian equilibrium*; p^* is a *Walrasian equilibrium price vector* (see, e.g., [5, 6]). In this paper, we consider the problem of finding a Walrasian equilibrium in a multi-unit auction.

Although the Walrasian equilibrium possesses a variety of desirable properties, it does not always exist. It is known that a Walrasian equilibrium does exist in single-unit auctions under a natural assumption on bidder's valuation functions, called gross substitutes condition.

Gross Substitutes Condition and Discrete Concavity We say that function f_j satisfies *gross substitutes (GS) condition* if it satisfies the following:

> **(GS)** $\forall p, q \in \mathbb{R}^n_+$ with $p \leq q$, $\forall x \in D_j(p)$, $\exists y \in D_j(q)$
> such that $x(i) \leq y(i)$ ($\forall i \in N$ with $p(i) = q(i)$).

This condition means that a bidder still wants to get items that do not change in price after the prices of other items increase. The concept of GS condition is introduced in Kelso and Crawford [12] for a fairly general two-sided job matching model. Since then, this condition has been widely used in various models such as matching, housing, and labor markets (see, e.g., [2, 4–6, 8, 9, 14]). In particular, Gul and Stacchetti [9] show the existence of a Walrasian equilibrium in a *single-unit* auction if bidders' valuation functions satisfy the GS condition; they also show that the GS condition is an "almost" necessary condition for the existence of an equilibrium in a single-unit auction.

Various characterizations of GS condition are given in the literature of discrete convex analysis and auction theory [2, 8, 9]. Among others, Fujishige and Yang [8] revealed the relationship between GS condition and discrete concavity called M$^\natural$-concavity. A valuation function $f_j : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{R}$ is said to be *M$^\natural$-concave* (read "M-natural-concave") if it satisfies the following:

(M$^\natural$-EXC) $\forall x, y \in [\mathbf{0}, u]_{\mathbb{Z}}$, $\forall i \in \mathrm{supp}^+(x - y)$, $\exists k \in \mathrm{supp}^+(x - y) \cup \{0\}$:
$$f_j(x) + f_j(y) \leq f_j(x - \chi_i + \chi_k) + f_j(y + \chi_i - \chi_k).$$

Here, we denote $\mathrm{supp}^+(x) = \{i \in N \mid x(i) > 0\}$, $\mathrm{supp}^-(x) = \{i \in N \mid x(i) < 0\}$ for a vector $x \in \mathbb{R}^n$, $\chi_i \in \{0, 1\}^n$ is the characteristic vector of $i \in N$, and $\chi_0 = \mathbf{0} = (0, 0, \ldots, 0)$.

The concept of M$^\natural$-concave function is introduced by Murota and Shioura [18] as a class of discrete concave functions (independently of GS condition). It is an extension of the concept of M-concave function introduced by Murota [16]. The concepts of M$^\natural$-concavity/M-concavity play primary roles in the theory of discrete convex analysis [17].

It is shown by Fujishige and Yang [8] that GS condition and M$^\natural$-concavity are equivalent in the case of *single-unit* auctions.

Theorem 1.1. *A valuation function $f : \{0, 1\}^n \to \mathbb{R}$ defined on 0-1 vectors satisfies the GS condition if and only if it is an M$^\natural$-concave function.*

This result initiated a strong interaction between discrete convex analysis and auction theory; the results obtained in discrete convex analysis are used in auction theory ([4, 14], etc.), while auction theory provides discrete convex analysis with interesting applications (see, e.g., [19]).

The GS condition, however, is not sufficient for the existence of an equilibrium in a *multi-unit* setting. In the last decade, several researchers independently tried to derive conditions for valuation functions to guarantee the existence of an equilibrium in a multi-unit setting (see, e.g., [15, 19]). Murota and Tamura [19] derive a stronger version of GS condition by using the relationship with M$^\natural$-concavity, and prove the existence of an equilibrium in a more general setting (see also [17, Ch. 11]). In this paper, we use the following alternative condition given in [15], which is obtained by adding to (GS) an extra inequality:

(SGS) $\forall p, q \in \mathbb{R}^n_+$ with $p \leq q$, $\forall x \in D_j(p)$, $\exists y \in D_j(q)$ such that
$x(i) \leq y(i)$ ($\forall i \in N$ with $p(i) = q(i)$) and $\sum_{i \in N} x(i) \geq \sum_{i \in N} y(i)$.

The extra inequality $\sum_{i \in N} x(i) \geq \sum_{i \in N} y(i)$ means that if prices are increased, then a bidder wants less items than before. This condition turns out to be essentially equivalent to M$^\natural$-concavity (see Theorem 1.4 below), and also to the condition in [19]. Note that for valuation functions on $\{0, 1\}^n$, the SGS condition is equivalent to the GS condition (see [15]).

Throughout this paper we assume the following conditions for all bidders' valuation functions f_j ($j = 1, \ldots, m$):

(A0) f_j is monotone nondecreasing,
(A1) f_j satisfies the SGS condition,
(A2) f_j is concave-extensible,
(A3) f_j takes integer values.

The concave-extensibility (A2) is a natural condition for valuation functions [15, 19]; a valuation function $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{Z}$ is said to be *concave-extensible* if there exists a concave function $\bar{f}$ defined on $\{x \in \mathbb{R}^n \mid \mathbf{0} \leq x \leq u\}$ such that $\bar{f}(x) = f(x)$ for every $x \in [\mathbf{0}, u]_{\mathbb{Z}}$. The assumption (A3) can be removed if we want to compute an ε-approximate equilibrium price vector instead of an "exact" one; for $\varepsilon > 0$, an *ε-approximate equilibrium price vector* p is defined as a vector such that $\|p - p^*\|_\infty < \varepsilon$ for some equilibrium price vector p^*. For such a problem, all results in this paper can be easily adapted with slight modifications.

Iterative Auctions and Ascending Auctions The main theme of this paper is the computation of a Walrasian equilibrium in an ascending auction. We focus on an equilibrium price vector p^* since an allocation in the equilibrium can be computed efficiently once we obtain p^*. In the computation, we assume that bidders' valuation functions f_j are given implicitly by so-called *demand oracles*, i.e., we can get the information about demand set $D_j(p)$ for a price vector p, but no information is available about the function values of f_j. This assumption is very plausible, since bidders want to preserve their privacy about valuation functions and disclose only the information that is really needed.

In the auction literature an algorithm called the *iterative auction* (or dynamic auction, Walrasian auction, Walrasian tâtonnement process, etc.) is often used to find an equilibrium [5, 6]. An iterative auction computes an equilibrium price vector by iteratively updating a current price vector p by using the information on demand sets $D_j(p)$. The most natural and popular iterative auction is the *ascending auction*, in which the current price vector is increased monotonically. The ascending auction is a natural generalization of the classical English auction for a single item, and known to have various nice properties (see, e.g., [5, 6]); in particular, it is quite natural from the economic point of view, and easy to understand and implement.

In this paper, we consider the ascending auction[4] presented in Ausubel [1]. This algorithm can be seen as a simplified version of the one in Gul and Stacchetti [10], where the *Lyapunov function* defined by

$$L(p) = \textstyle\sum_{j=1}^{m} V_j(p) + u^\top p \qquad (p \in \mathbb{R}^n) \tag{3}$$

is used. It is known (see [1, 21]) that p^* is an equilibrium price vector if and only if it is a minimizer of the Lyapunov function and that there exists an integral minimizer $p^* \in \mathbb{Z}^n$ of the Lyapunov function. Based on this fact, the ascending auction in [1] tries to find a minimizer of the Lyapunov function. For $X \subseteq N$, we denote by $\chi_X \in \{0, 1\}^n$ the characteristic vector of X.

```
Algorithm Ascend
Step0: Set $p := p^\circ$, where $p^\circ \in \mathbb{Z}^n$ is a lower bound of some $p^* \in \arg\min L$
        (e.g., $p^\circ = \mathbf{0}$).
Step1: Find $X \subseteq N$ that minimizes $L(p + \chi_X)$.
```

[4] Our Ascend is slightly different from "Ascending Tâtonnement Algorithm" in [1] in the choice of X in Step 1; X is a minimal minimizer of $L(p + \chi_X)$ in [1], while it can be any minimizer in ours, which is easier to find than a minimal minimizer and does not increase the number of iterations. This is an additional merit of our algorithm.

Step2: If $L(p + \chi_X) = L(p)$, then output p and stop.
Step3: Set $p := p + \chi_X$ and go to Step 1.

It can be shown (cf. [1]) that this algorithm outputs an equilibrium price vector in a finite number of iterations. While the ascending auction has various nice properties (see, e.g., [5, 6]), it has a disadvantage that the initial price vector must be a lower bound of some equilibrium vector. Taking this into consideration, Ausubel [1] also propose an alternative iterative auction, which allows us to start with an arbitrary price vector, but has a drawback that the change of the price vector is not monotone.

Our Contribution The main aim of this paper is to theoretically analyze the ascending auction and other iterative auctions with respect to their time complexity. While computational experiments are often used to evaluate the practical performance of iterative auctions (see [3, 20]), there is no theoretical analysis of the time complexity, even in the case of the single-unit auction, except for the termination in finite time. This paper gives the first theoretical analysis in the case of multi-unit auctions.

The results in this paper consist of the following two:

(i) Tight bounds on the number of iterations of iterative auctions,
(ii) An efficient algorithm for the update of a price vector.

Our first result is the analysis of the number of iterations required by the algorithm Ascend. The upper bound established in this paper is useful in practice by providing bidders with an a priori guarantee for the time period of the auction process. The exact bound for the number of iterations in Ascend is given in terms of the distance between the initial price vector and a minimizer of the Lyapunov function L. For the analysis, we define

$$\hat{\mu}(p) = \min\{\|p^* - p\|_\infty \mid p^* \in \arg\min L,\ p^* \geq p\} \quad (p \in \mathbb{Z}^n).$$

The value $\hat{\mu}(p)$ remains the same or decreases by one in each iteration of the algorithm. Hence, if p° is the initial vector, then $\hat{\mu}(p^\circ) + 1$ is a lower bound for the number of iterations. We show that this bound is also an upper bound.

Theorem 1.2. *Suppose that the initial vector $p^\circ \in \mathbb{Z}^n$ in the algorithm* Ascend *is a lower bound of some minimizer of the Lyapunov function L. Then, the algorithm outputs a minimizer of L and terminates exactly in $\hat{\mu}(p^\circ)+1$ iterations.*

This result shows that the trajectory of a price vector generated by Ascend is the "shortest" path between the initial vector and a minimizer of the Lyapunov function. This reveals an additional advantage of the ascending auction. We also propose some other iterative auctions in this paper and derive tight bounds for the number of iterations in these algorithms.

Our second result is concerned with the update of a price vector. The algorithm Ascend and other iterative auctions considered in this paper update the price vector by using an optimal solution of the problem $\min_{X \subseteq N} L(p + \chi_X)$

or $\min_{X \subseteq N} L(p - \chi_X)$. It is known that these problems can be reduced to submodular function minimization (SFM, for short). Although polynomial-time algorithms are available for SFM [7, 17], they are quite slow and complicated.

We show that the SFM problems appearing in iterative auctions can be solved more efficiently than by a straightforward application of the existing SFM algorithms. We denote $U = \|u\|_\infty$.

Theorem 1.3. *For every integral vector $p \in \mathbb{Z}^n$, the problems $\min_{X \subseteq N} L(p + \chi_X)$ and $\min_{X \subseteq N} L(p - \chi_X)$ can be solved in $\mathrm{O}(mn^4 \log U \log(mnU))$ time.*

This improvement is achieved by exploiting the fact that valuation functions are given by demand sets and the submodular functions to be minimized can be represented in terms of demand sets as follows. For $x \in \mathbb{R}^n$ and $Y \subseteq N$, we denote $x(Y) = \sum_{i \in Y} x(i)$.

Proposition 1.1 (cf. [1, 17]). *For $p \in \mathbb{Z}^n$ and $X \subseteq N$, we have*

$$L(p + \chi_X) - L(p) = -\sum_{j \in M} \min\{y(X) \mid y \in D_j(p)\} + u(X),$$

$$L(p - \chi_X) - L(p) = \sum_{j \in M} \max\{y(X) \mid y \in D_j(p)\} - u(X).$$

The problem setting of SFM in terms of demand sets is interesting in its own right.

Proofs of the results in this paper are based on the following equivalence between the SGS condition and M$^\natural$-concavity.

Theorem 1.4. *Let $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{Z}$ be a concave-extensible function. Then, f satisfies the SGS condition if and only if it is an M$^\natural$-concave function.*

We also point out in Corollary 2.1 that the Lyapunov function has a discrete convexity called L$^\natural$-convexity. The concepts of M$^\natural$-concavity and L$^\natural$-convexity play primary roles in the theory of discrete convex analysis [17]. On the basis of these facts, we can make full use of rich results from discrete convex analysis to prove Theorems 1.2 and 1.3.

2 Property of Indirect Utility Functions

In this section, we show that the indirect utility function $V_j : \mathbb{R}^n \to \mathbb{R}$ given by (1) is an L$^\natural$-convex function. Function $g : \mathbb{R}^n \to \mathbb{R}$ is said to be *L$^\natural$-convex* [17] if for every $p, q \in \mathbb{R}^n$ and every nonnegative $\lambda \in \mathbb{R}_+$, it holds that

$$g(p) + g(q) \geq g((p + \lambda \mathbf{1}) \wedge q) + g(p \vee (q - \lambda \mathbf{1})),$$

where $\mathbf{1} = (1, 1, \ldots, 1)$ and for $p, q \in \mathbb{R}^n$ the vectors $p \wedge q$ and $p \vee q$ denote, respectively, the vectors obtained by component-wise minimum and maximum of p and q. It is easy to see that an L$^\natural$-convex function is a submodular function on $\mathbb{R}^n$, i.e., it satisfies $g(p) + g(q) \geq g(p \wedge q) + g(p \vee q)$ $(\forall p, q \in \mathbb{R}^n)$.

Theorem 2.1. *The indirect utility function $V_j : \mathbb{R}^n \to \mathbb{R}$ is an $L^\natural$-convex function.*

Proof. The assumptions (A1) and (A2) imply the $\mathrm{M}^\natural$-concavity of valuation function f_j by Theorem 1.4. Hence, the indirect utility function V_j is $\mathrm{L}^\natural$-convex by the conjugacy theorem in discrete convex analysis [17, Ch. 8]. □

From this property we obtain the $\mathrm{L}^\natural$-convexity of the Lyapunov function L given by (3) since any linear function is also an $\mathrm{L}^\natural$-convex function and $\mathrm{L}^\natural$-convexity is closed under the addition of functions.

Corollary 2.1. *The Lyapunov function $L : \mathbb{R}^n \to \mathbb{R}$ is an $L^\natural$-convex function. In particular, L is a submodular function.*

It follows from Corollary 2.1 and the integrality assumption (A3) for valuation functions f_j that $\arg\min L$ is an integral polyhedron by the results in discrete convex analysis [17]. Since $\arg\min L$ is exactly the same as the set of equilibrium price vectors [1, 21], this observation implies the known fact that there exists an integral equilibrium price vector.

3 Analysis for Number of Iterations in Iterative Auctions

In this section, we consider the algorithm Ascend and several other iterative auction algorithms for finding an integral equilibrium price vector, and analyze the number of iterations.

We first show that there exists an integral equilibrium price vector in the finite interval $[\mathbf{0}, \bar{p}]_{\mathbb{Z}}$, where $\bar{p} \in \mathbb{Z}^n_+$ is given by $\bar{p}(i) = \max_{j \in M}\{f_j(\chi_i) - f_j(\mathbf{0})\}$. Note that $\bar{p}$ can be easily computed from bidders' valuation functions.

Proposition 3.1. *There exists an equilibrium price vector p^* with $p^* \in [\mathbf{0}, \bar{p}]_{\mathbb{Z}}$.*

Hence, the number of iterations of the algorithm Ascend is at most $\sum_{i \in N} \bar{p}(i)$. We will see below that the bounds for the number of iterations in Ascend and other iterative auction algorithms are much smaller than $\sum_{i \in N} \bar{p}(i)$.

As stated in Theorem 1.2, the number of iterations in Ascend is $\hat{\mu}(p^\circ) + 1$. Its proof is quite nontrivial and can be done with the aid of some known results in discrete convex analysis. Note that any algorithm requires at least $\hat{\mu}(p^\circ) + 1$ iterations if it increases the price vector by a 0-1 vector in each iteration. Hence, the algorithm Ascend is the fastest among all iterative auction algorithms of this type, and the trajectory of the price vector is a "shortest" path from the initial vector to an equilibrium. In addition, since $\hat{\mu}(p^\circ) \leq \max_{i \in N}\{\bar{p}(i) - p^\circ(i)\}$, we can guarantee that the algorithm terminates in at most $\max_{i \in N}\{\bar{p}(i) - p^\circ(i)\} + 1$ iterations; note that this bound can be computed in advance before executing the algorithm.

Similarly to Ascend, we can consider an algorithm Descend as in [1], where the price vector is decreased by a vector $\chi_X \in \{0, 1\}^n$ which is a minimizer of $L(p - \chi_X)$. It is easy to see that algorithm Descend enjoys similar properties as Ascend. We define $\check{\mu}(p) = \min\{\|p^* - p\|_\infty \mid p^* \in \arg\min L,\ p^* \leq p\}$ for $p \in \mathbb{Z}^n$.

Theorem 3.1. *Suppose that the initial vector $p^\circ \in \mathbb{Z}^n$ in the algorithm* Descend *is a upper bound of some minimizer of the Lyapunov function L. Then, the algorithm outputs a minimizer of L and terminates exactly in $\check{\mu}(p^\circ)+1$ iterations.*

An advantage of algorithms Ascend and Descend is that a price vector is updated monotonically, which is an important property from the viewpoint of auctions. They, however, have a drawback that the initial price vector should be a lower or upper bound for some minimizer of Lyapunov function L. In contrast, the following two algorithms can start from any initial price vector and find an equilibrium. Therefore, the number of iterations can be small if we can choose an initial vector that is close to some minimizer of L.

The next algorithm TwoPhase can be seen as an application of Ascend with an arbitrary initial vector, followed by Descend. The algorithm has a merit that the price vector is updated "almost" monotonically.

```
Step 0: Set $p := p^\circ$, where $p^\circ \in \mathbb{Z}^n$ is a vector with $p^\circ \in [\mathbf{0}, \bar{p}]_{\mathbb{Z}}$.
        Go to Ascending Phase.
Ascending Phase:
Step A1: Find $X \subseteq N$ that minimizes $L(p + \chi_X) - L(p)$.
Step A2: If $L(p + \chi_X) = L(p)$, then go to Descending Phase.
Step A3: Set $p := p + \chi_X$ and go to Step A1.
Descending Phase:
Step D1: Find $X \subseteq N$ that minimizes $L(p - \chi_X) - L(p)$.
Step D2: If $L(p - \chi_X) = L(p)$, then output $p$ and stop.
Step D3: Set $p := p - \chi_X$ and go to Step D1.
```

A version of this algorithm specialized to valuation functions defined on $\{0,1\}^n$ coincides with the one in [21]. Another algorithm called "Global Walrasian tâtonnement algorithm" in [1] repeats ascending and descending phases until a minimizer of L is found; our analysis shows that this algorithm terminates after only one ascending phase and only one descending phase.

To analyze the number of iterations required by TwoPhase, we define

$$\mu(p) = \min\{\|p^* - p\|_\infty^+ + \|p^* - p\|_\infty^- \mid p^* \in \arg\min L\} \quad (p \in \mathbb{Z}^n),$$
$$\|p^* - p\|_\infty^+ = \max_{i \in N} \max(0, p^*(i) - p(i)), \ \|p^* - p\|_\infty^- = \max_{i \in N} \max(0, -p^*(i) + p(i)).$$

The value $\mu(p)$ can be regarded as the "distance" between the vector p and a minimizer of L. By definition, $\mu(p)$ remains the same or decreases by one if p is updated by adding or subtracting a 0-1 vector. Hence, the algorithm TwoPhase requires at least $\mu(p^\circ) + 1$ iterations. In the following, we show that the number of iterations is bounded by $3\mu(p^\circ) + 2$.

Theorem 3.2. *The algorithm* TwoPhase *outputs an equilibrium price vector in at most $3\mu(p^\circ)+2$ iterations; more precisely, the ascending (resp., descending) phase terminates in at most $\mu(p^\circ) + 1$ iterations (resp. $2\mu(p^\circ) + 1$ iterations).*

We finally consider the algorithm Greedy, which can be seen as the steepest descent (or greedy) algorithm for the minimization of the Lyapunov function.

```
Step 0: Set $p := p^\circ$, where $p^\circ \in \mathbb{Z}^n$ is a vector with $p^\circ \in [\mathbf{0}, \bar{p}]_{\mathbb{Z}}$.
Step 1: Find $\varepsilon \in \{+1, -1\}$ and $X \subseteq N$ that minimize $L(p + \varepsilon\chi_X)$.
Step 2: If $L(p + \varepsilon\chi_X) = L(p)$, then output $p$ and stop.
Step 3: Set $p := p + \varepsilon\chi_X$ and go to Step 1.
```

This can be seen as an application of the steepest descent algorithm for L$^\natural$-convex function minimization (see [17]), for which the number of iterations is analyzed in [13]. We give a refined analysis of this algorithm in terms of the "distance" between the initial vector and a minimizer of L.

Theorem 3.3. *The algorithm* Greedy *outputs an equilibrium price vector and terminates exactly in $\mu(p^\circ) + 1$ iterations.*

As mentioned above, any iterative auction algorithm of this type requires at least $\mu(p^\circ) + 1$ iterations. Theorem 3.3 shows that Greedy is the fastest among all iterative auction algorithms of this type, and the trajectory of a price vector is a "shortest" path from the initial vector to an equilibrium. Although Greedy has such merits in the choice of the initial vector and in the number of iterations, it also has a drawback that it may repeat the increment and decrement of the price vector many times, which is not a desirable behavior from the viewpoint of auction.

It should be noted that the algorithms as well as their analysis in this section can be applied not only to the Lyapunov function but also to any general L$^\natural$-convex function since our proofs do not rely on any special structure of the Lyapunov function. In particular, the key property used in our proofs is the following property of L$^\natural$-convex functions.

Proposition 3.2 ([17, Theorem 7.7]). *Let $g : \mathbb{R}^n \to \mathbb{R}$ be an L$^\natural$-convex function. For every integral $p, q \in \mathbb{Z}^n$ with $\mathrm{supp}^+(p - q) \neq \emptyset$, it holds that $g(p) + g(q) \geq g(p - \chi_X) + g(q + \chi_X)$, where $X = \arg\max_{i \in N}\{p(i) - q(i)\}$.*

Finally, we point out that in all of the iterative auction algorithms considered in this paper we use *linear* and *anonymous* pricing rule, meaning that the price of any bundle x of goods is equal to $p^\top x$ and is the same for all bidders. In this case, we need to impose conditions on the valuation of bidders to guarantee that iterative auction algorithms work. On the other hand, so-called combinatorial auction algorithms use *nonlinear* and *discriminatory* pricing rule, i.e., the price $p(x, i)$ of a bundle x of goods depends on x and bidder i and is not linear. In this case, iterative auction algorithms work with more general valuation functions, although auction algorithms of this type are difficult to use in practice.

4 Efficient Update of Price Vector

For an update of the price vector in the ascending auction and other iterative auctions, we repeatedly solve the local optimization problems $\min_{X \subseteq N} L(p+\chi_X)$ and $\min_{X \subseteq N} L(p-\chi_X)$ for some integral $p \in \mathbb{Z}^n$, both of which can be reduced to submodular function minimization (SFM, for short). Indeed, the former problem can be reduced to the minimization of a set function given by

$$\rho_L(X) = L(p + \chi_X) - L(p) \qquad (X \subseteq N), \tag{4}$$

which is submodular since the Lyapunov function L is submodular by Corollary 2.1. The latter problem can also be reduced to SFM. In this section, we show that by using demand sets $D_j(p)$ obtained from bidders, these problems can be solved faster than a straightforward application of SFM algorithms.

In the following, we consider minimization of ρ_L given by (4). Throughout this section, we assume that for a given integral vector $p \in \mathbb{Z}^n$ and $j \in M$, a vector $x_j^\circ \in D_j(p)$ is available and the membership test in $D_j(p)$ can be done in constant time. This means that the evaluation of $\rho_L(X)$ requires solving optimization problems on $D_j(p)$, which can be done in $\mathrm{O}(mn^2 \log U)$ time, where $U = \|u\|_\infty$. Recall that SFM is solvable in polynomial time [7, 17], provided the function value can be evaluated in polynomial time.

Almost all "combinatorial" polynomial-time algorithms for SFM are based on the following min-max formula (see, e.g., [7, 17]). For a submodular function $\rho : 2^N \to \mathbb{Z}$, we define a set

$$B(\rho) = \{x \in \mathbb{Z}^n \mid x(Y) \le \rho(Y) \ (\forall Y \subseteq N),\ x(N) = \rho(N)\},$$

which is called the *base polyhedron* associated with ρ.

Proposition 4.1. *For an integer-valued submodular function $\rho : 2^N \to \mathbb{Z}$,*

$$\min\{\rho(X) \mid X \subseteq N\} = \max\left\{ \textstyle\sum_{i \in N} \min\{0, x(i)\} \;\middle|\; x \in B(\rho)\right\} \tag{5}$$

holds. Moreover, if $x^* \in B(\rho)$ is an optimal solution of the maximization problem on the right-hand side of (5), then every set $X^* \subseteq N$ with $\{i \in N \mid x^*(i) < 0\} \subseteq X^*$ and $\{i \in N \mid x^*(i) > 0\} \cap X^* = \emptyset$ is a minimizer of ρ.

Solving the maximization problem in (5) requires the membership test in $B(\rho)$. For the efficient membership test in $B(\rho)$, the existing polynomial-time algorithms use a technique to represent a vector x as a convex combination of extreme points in $B(\rho)$, which makes the algorithms slow and complicated. The fastest (weakly-)polynomial algorithm runs in $\mathrm{O}((n^4\mathrm{EO} + n^5) \log \mathit{\Gamma})$ time [11], where $\mathit{\Gamma}$ is an upper bound on $|\rho(X)|$ and EO denotes the time for function evaluation; $\mathit{\Gamma} = mnU$ and EO$= \mathrm{O}(mn^2 \log U)$ in our case.

We show that the minimization of ρ_L can be solved faster by using a representation of the base polyhedron $B(\rho_L)$ in terms of demand sets $D_j(p)$.

Proposition 4.2. *It holds that $B(\rho_L) = \{u - \sum_{j \in M} x_j \mid x_j \in \widetilde{D}_j(p) \ (j \in M)\}$, where $\widetilde{D}_j(p)$ is the set of minimal vectors in $D_j(p)$ for $j \in M$.*

This formula can be proven by using the representation of $L(p + \chi_X) - L(p)$ in Proposition 1.1. Note that $\widetilde{D}_j(p)$ is a base polyhedron. By Propositions 4.1 and 4.2, the minimization of ρ_L is equivalent to the problem

$$\max\left\{ \textstyle\sum_{i \in N} \min\{0, x(i)\} \;\middle|\; x = u - \sum_{j \in M} x_j,\ x_j \in \widetilde{D}_j(p) \ (j \in M)\right\}.$$

Based on this observation, we can prove Theorem 1.3. The established bound $\mathrm{O}(mn^4 \log U \log(mnU))$ is smaller than the bound $\mathrm{O}(mn^6 \log U \log(mnU))$ obtained by a straightforward application of the SFM algorithm in [11].

Acknowledgements. This research is supported by KAKENHI (21360045, 21740060, 24500002) and the Aihara Project, the FIRST program from JSPS.

References

1. L. M. Ausubel. An efficient dynamic auction for heterogeneous commodities. American Economic Review, 96 (2006) 602–629.
2. L. M. Ausubel and P. Milgrom. Ascending auctions with package bidding. Front. Theor. Econ., 1 (2002) Article 1.
3. M. Bichler, P. Shabalin, and A. Pikovsky. A computational analysis of linear price iterative combinatorial auction formats. Inform. Syst. Res. 20 (2009), 33–59.
4. M. Bing, D. Lehmann, and P. Milgrom. Presentation and structure of substitutes valuations. Proc. EC 2004, 238–239.
5. L. Blumrosen and N. Nisan. Combinatorial auction. in: N. Nisan, et al. (eds.) Algorithmic Game Theory, pp. 267–299. Cambridge Univ. Press, 2007.
6. P. Cramton, Y. Shoham, R. Steinberg. Combinatorial Auctions. MIT Press, 2006.
7. S. Fujishige. Submodular Functions and Optimization. 2nd Edition. Elsevier, 2005.
8. S. Fujishige and Z. Yang. A note on Kelso and Crawford's gross substitutes condition. Math. Oper. Res. 28 (2003) 463–469.
9. F. Gul and E. Stacchetti. Walrasian equilibrium with gross substitutes. J. Econ. Theory 87 (1999) 95–124.
10. F. Gul and E. Stacchetti. The English auction with differentiated commodities. J. Economic Theory 92 (2000) 66–95.
11. S. Iwata. A faster scaling algorithm for minimizing submodular functions. SIAM J. Comp. 32 (2002) 833–840.
12. A. S. Kelso and V. P. Crawford. Job matching, coalition formation and gross substitutes. Econometrica 50 (1982) 1483–1504.
13. V. Kolmogorov and A. Shioura. New algorithms for convex cost tension problem with application to computer vision. Discrete Optimization 6 (2009) 378–393.
14. B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. Games Econom. Behav. 55 (2006) 270–296.
15. P. Milgrom and B. Strulovici. Substitute goods, auctions, and equilibrium. J. Economic Theory 144 (2009) 212–247.
16. K. Murota. Convexity and Steinitz's exchange property. Adv. Math. 124 (1996) 272–311.
17. K. Murota. Discrete Convex Analysis. SIAM, Philadelphia, 2003.
18. K. Murota and A. Shioura. M-convex function on generalized polymatroid. Math. Oper. Res. 24 (1999) 95–105.
19. K. Murota and A. Tamura. New characterizations of M-convex functions and their applications to economic equilibrium models with indivisibilities. Discrete Appl. Math. 131 (2003) 495–512.
20. D.C. Parkes and L.H. Ungar. Iterative combinatorial auctions: theory and practice. Proc. 17th National Conference on Artificial Intelligence (AAAI-00), 74–81, 2000.
21. N. Sun and Z. Yang. A double-track adjustment process for discrete markets with substitutes and complements. Econometrica 77 (2009) 933–952.