THE UNIVERSITY *of York*

Discussion Papers in Economics

No. 13/13

Computing a Walrasian Equilibrium in Iterative Auctions with Multiple Differentiated Items

Kazuo Murota, Akiyoshi Shioura, and Zaifu Yang

Department of Economics and Related Studies
University of York
Heslington
York, YO10 5DD

Computing a Walrasian Equilibrium in Iterative Auctions with Multiple Differentiated Items*

Kazuo MUROTA
Graduate School of Information Science and Technology
University of Tokyo, Tokyo 113-8656, Japan

Akiyoshi SHIOURA
Graduate School of Information Sciences
Tohoku University, Sendai 980-8579, Japan

Zaifu YANG
Department of Economics
University of York, York YO10 5DD, UK

June 2013

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 goods and each good 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.

*This research is partially supported by KAKENHI (21360045, 21740060, 24500002), the Global COE program "The Research and Training Center for New Development in Mathematics", and the Aihara Project, the FIRST program from JSPS. This paper is also registered as technical report at the Department of Mathematical Informatics Graduate School of Information Science and Technology University of Tokyo.

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 a 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 fundamental 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. Each f_j is assumed to be a monotone nondecreasing and non-negative *integer-valued* function. 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 a competitive price vector. 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

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

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

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 completely. 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 called 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 [13] 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, 5, 6, 8, 9, 15]). 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 and propose two equivalent conditions to GS; 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 them, 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 [19] 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 [17]. The concepts of M$^\natural$-concavity/M-concavity play primary roles in the theory of discrete convex analysis [18].

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, 15], etc.), while auction theory provides discrete convex analysis with interesting applications (see, e.g., [20]).

The GS condition, however, is not sufficient for the existence of an equilibrium in a *multi-unit* setting (see, e.g., [16]). 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., [16, 20]). Murota and Tamura [20] derive a stronger version of GS condition by using the relationship with M$^\natural$-concavity, and prove the existence of an equilibrium in more general setting (see also [18, Ch. 11]). In this paper, we use the following alternative condition given in [16], which is obtained by adding to (GS) an extra inequality:

(SGS) $\forall p, q \in \mathbb{R}^n_+$ with $p \leq q$, $\forall x \in D_j(p)$, $\exists y \in D_j(q)$
s.t. $x(i) \leq y(i)$ ($\forall i \in N$ 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 equivalent to M$^\natural$-concavity (see Theorem 1.5 below), and also to the condition in [20]. Note that for valuation functions on $\{0,1\}^n$, the SGS condition is equivalent to the GS condition (see [16]). Throughout this paper we assume that all of the bidders' valuation functions satisfy the SGS condition. Examples of such valuation functions are given in Section A.1.

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 the computation of 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 of 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[1] 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

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

is used. It is known (see [1, 22]) 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 $X \in 2^N$, we denote by $\chi_X \in \{0,1\}^n$ the characteristic vector of X.

Algorithm Ascend
Step0: Set $p{:=}p^\circ$, where p° is a lower bound of some $p^* {\in} \arg\min L$ (e.g., $p^\circ = \mathbf{0}$).
Step1: Find $X \subseteq N$ that minimizes $L(p + \chi_X)$.
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, 21]), 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 (see, e.g., [11] for related results in the case of *single-item* auctions where there is only one item). This paper gives the first theoretical analysis in the case of multi-unit auctions.

The results in this paper consist of the following two:

[1]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.

(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

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

It is easy to see that 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° 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 in $\hat{\mu}(p^\circ) + 1 = \|p^* - p^\circ\|_\infty + 1$ 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 a new advantage of the ascending auction in addition to various known properties. 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 concerns 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 $\min\{L(p + \chi_X) \mid X \subseteq N\}$ or $\min\{L(p - \chi_X) \mid X \subseteq N\}$. It is known that these problems can be reduced to the submodular function minimization (SFM, for short). Although polynomial-time algorithms are available for SFM [7, 18], they are quite slow and complicated.

In this paper, 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 $p \in [\mathbf{0}, u]_{\mathbb{Z}}$, the problems $\min\{L(p+\chi_X) \mid X \subseteq N\}$ and $\min\{L(p-\chi_X) \mid X \subseteq N\}$ can be solved in $\mathrm{O}(mn^4 \log U \log(mnU))$ time.*

This improvement is achieved by using the merit that valuation functions are given by demand oracles. Our SFM problems are interesting in their own right since the submodular functions to be minimized can be represented as follows by using demand sets; this representation admits a faster algorithm. For $x \in \mathbb{R}^n$ and $Y \subseteq N$, we denote $x(Y) = \sum_{i \in Y} x(i)$.

Lemma 1.4. *For $p \in \mathbb{Z}^n_+$ and $X \subseteq N$, we have*

$$L(p + \chi_X) - L(p) = -\sum_{j \in M} \min\{y(X) \mid y \in D_j(p)\} + u(X), \quad (4)$$

$$L(p - \chi_X) - L(p) = -\sum_{j \in M} \max\{y(X) \mid y \in D_j(p)\} + u(X). \quad (5)$$

Our Technique Our proofs of Theorems 1.2 and 1.3 are based on the equivalence between the SGS condition and M$^\natural$-concavity. To show this, we assume the concave-extensibility for valuation functions as in [16]; 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 $x \in [\mathbf{0}, u]_\mathbb{Z}$.

Theorem 1.5. *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.*

Proof is given in Section A.2 in Appendix. We also point out that the Lyapunov function has a discrete convexity called L$^\natural$-convexity. The definition of L$^\natural$-convexity and the proof of the following theorem are given in Section 2.

Theorem 1.6. *Suppose that each valuation function $f_j : [\mathbf{0}, u]_\mathbb{Z} \to \mathbb{Z}$ ($j \in M$) is concave-extensible and satisfies the SGS condition. Then, the Lyapunov function $L : \mathbb{R}^n \to \mathbb{R}$ is an L$^\natural$-convex function. In particular, L is a submodular function.*

The concepts of M$^\natural$-concavity and L$^\natural$-convexity play primary roles in the theory of discrete convex analysis [18]. On the basis of Theorems 1.5 and 1.6, we can make full use of rich results from discrete convex analysis.

Throughout this paper, we mainly assume that valuation functions take integer value. This assumption can be removed if we 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^*. In such a case, all results in this paper can be easily extended with slight modification; see Section A.11 for details.

2 Property of Indirect Utility Functions

In this section, we show that the indirect utility function $V(p) = \max\{f(x) - p^\top x \mid x \in [\mathbf{0}, u]_\mathbb{Z}\}$ is an L$^\natural$-convex function. This observation plays a crucial role in the analysis of iterative auction algorithms. Since there exists an integral equilibrium price vector (see Section 3), we may regard indirect utility functions (and the Lyapunov function) as those defined on integral vectors $\mathbb{Z}^n$, although they are originally defined on $\mathbb{R}^n$.

Function $g : \mathbb{Z}^n \to \mathbb{R}$ is said to be *L$^\natural$-convex* [18] if for every $p, q \in \mathbb{Z}^n$ and every $\lambda \in \mathbb{Z}_+$, 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 $\mathbb{Z}^n$, i.e., it satisfies

$$g(p) + g(q) \geq g(p \wedge q) + g(p \vee q) \quad (\forall p, q \in \mathbb{Z}^n).$$

Note that a function $g : \mathbb{Z}^n \to \mathbb{R}$ is L$^\natural$-convex if and only if it satisfies the following *discrete mid-point convexity* (see [18]):

$$g(p) + g(q) \geq g(\lceil (p + q)/2 \rceil) + g(\lfloor (p + q)/2 \rfloor) \qquad (\forall p, q \in \mathbb{Z}^n),$$

where $\lceil x \rceil$ and $\lfloor x \rfloor$ denote, respectively, the integer vectors obtained from $x \in \mathbb{R}^n$ by component-wise round-up and round-down to the nearest integer.

Proposition 2.1. *If $f_j : [\mathbf{0}, u]_\mathbb{Z} \to \mathbb{Z}$ is an M$^\natural$-concave function, then its indirect utility function $V_j : \mathbb{Z}^n \to \mathbb{Z}$ is an L$^\natural$-convex function.*

Using this property (proof is given in Section A.2), we can prove Theorem 1.6 on the L$^\natural$-convexity of the Lyapunov function L given by (3), as follows.

Since we assume that each valuation function f_j is concave-extensible and satisfies the SGS condition, it is M$^\natural$-concave by Theorem 1.5. Hence, the indirect utility function V_j of f_j is L$^\natural$-convex by Proposition 2.1. Since any linear function is also an L$^\natural$-convex function and L$^\natural$-convexity is closed under the addition of functions, the Lyapunov function L is also an L$^\natural$-convex function.

3 Analysis for Number of Iterations in Iterative Auctions

In this section, we consider the algorithm Ascend and several other iterative auction algorithms, and analyze the number of iterations.

Before starting the analysis, we firstly show that there exists an equilibrium price vector which is an integral vector contained 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. Proof is given in Section A.3.

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

Hence, the zero vector $\mathbf{0}$ can be used as an initial vector p° in the algorithm Ascend, and the number of iterations 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 $\sum_{i \in N} \bar{p}(i)$.

We firstly show the statement of Theorem 1.2 that 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; see Section A.4. Note that any algorithm requires at least $\hat{\mu}(p^\circ) + 1$ iterations if it increases a 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 a price vector is a "shortest" path from the initial vector to an equilibrium. In addition, since $\hat{\mu}(p^\circ) \le \max_{i \in N}\{\bar{p}(i) - p^\circ(i)\}$, we can guarantee that the algorithm terminates in at most $\max_{i \in N}\{\bar{p}(i) - p^\circ(i)\} + 1$ iterations.

Similarly to Ascend, we can consider an algorithm Descend as in [1], where a 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

$$\check{\mu}(p) = \min\{\|p^* - p\|_\infty \mid p^* \in \arg\min L,\ p^* \le p\}.$$

Theorem 3.2. *Suppose that the initial vector p° 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 in $\check{\mu}(p^\circ) + 1 = \|p^* - p^\circ\|_\infty + 1$ 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 which 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 a price vector is updated "almost" monotonically.

Step 0: Let $p^\circ \in [\mathbf{0}, \bar{p}]_{\mathbb{Z}}$ be any vector. Set $p := p^\circ$. 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 specialized version of this algorithm to valuation functions defined on $\{0, 1\}^n$ coincides with the one in [22]. An 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 $3\mu(p^\circ) + 2$; proof is given in Section A.5.

Theorem 3.3. *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: Let $p^\circ \in [\mathbf{0}, \bar{p}]_\mathbb{Z}$ be any vector. Set $p := p^\circ$.
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 [18]), for which the number of iterations is analyzed in [14]. We give a refined analysis of this algorithm in terms of the "distance" between the initial vector and a minimizer of L. Proof is given in Section A.6.

Theorem 3.4. *The algorithm* Greedy *outputs an equilibrium price vector in at most $\mu(p^\circ) + 1$ iterations.*

As mentioned above, any iterative auction algorithm of this type requires at least $\mu(p^\circ) + 1$ iterations. Theorem 3.4 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. Hence, Greedy has advantages in the choice of the initial vector and in the number of iterations, but also has a disadvantage that it may repeat the increment and decrement of a price vector many times, which is not a good behavior from the viewpoint of auction.

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

Proposition 3.5 ([18, Theorem 7.7])**.** *Let $g : \mathbb{Z}^n \to \mathbb{R}$ be an $L^\natural$-convex function. For every $p, q \in \mathbb{Z}^n$ with $\mathrm{supp}^+(p-q) \neq \emptyset$, it holds that*

$$g(p) + g(q) \geq g(p - \chi_X) + g(q + \chi_X),$$

where $X = \arg\max_{i \in N}\{p(i) - q(i)\}$.

4 Efficient Update of Price Vector

For the update of a price vector in the ascending auction and other iterative auctions, we repeatedly solve the local optimization problems $\min\{L(p + \chi_X) \mid X \subseteq N\}$ and $\min\{L(p - \chi_X) \mid X \subseteq N\}$, both of which can be reduced to the submodular function minimization (SFM, for short). Indeed, the former problem can be reduced to the minimization of a set function given by $\rho(X) = L(p + \chi_X) - L(p)$ $(X \in 2^N)$, which is submodular since the Lyapunov function L is submodular by Theorem 1.6. The latter problem can be also reduced to SFM in the same way. 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 the former problem only since the latter can be solved in the same way. Recall that ρ can be represented as (4) (see, e.g., [1]; see also Section A.7). Throughout this section, we assume that demand sets $D_j(p)$ for $j \in M$ are given; this means, in particular, that for each $j \in M$, 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 $\rho(X)$ requires solving optimization problems on $D_j(p)$, which can be done in $\mathrm{O}(mn^2 \log U)$ time, where $U = \|u\|_\infty$ (see Section A.8 for a proof).

Recall that SFM is solvable in polynomial time [7, 18], provided that 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, 18]). For a submodular function $\rho : 2^N \to \mathbb{Z}$, we define a set

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

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

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

$$\min\{\rho(X) \mid X \in 2^N\} = \max\Bigl\{\sum_{i\in N} \min\{0, x(i)\} \Bigm| x \in B(\rho)\Bigr\} \qquad (6)$$

holds. Moreover, if $x^* \in B(\rho)$ is an optimal solution of the maximization problem in the right-hand side of (6)*, then every set $X^* \in 2^N$ 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.*

Although the maximization problem in (6) is useful in solving SFM, it has a drawback that it requires the membership test in $B(\rho)$, which is known to be essentially equivalent to solving the original SFM. 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 $\mathrm{O}((n^4\mathrm{EO} + n^5)\log\Gamma)$ time [12], where Γ is an upper bound on $|\rho(X)|$ and EO denotes the time for function evaluation; $\Gamma = mnU$ and EO$=$ $\mathrm{O}(mn^2 \log U)$ in our case (see Section A.8).

In the following, we show that the minimization of ρ can be solved more efficiently by using the representation (4) of ρ. The next property states that the base polyhedron $B(\rho)$ can be represented explicitly by using demand sets $D_j(p)$. Proof is given in Section A.9.

Lemma 4.2. *For a submodular function ρ given by* (4)*, it holds that*

$$B(\rho) = \{u - \sum_{j\in M} x_j \mid x_j \in \widetilde{D}_j(p) \ (j \in M)\},$$

where $\widetilde{D}_j(p)$ is the set of minimal vectors in $D_j(p)$ for $j \in M$.

Note that $\widetilde{D}_j(p)$ is a base polyhedron (see Section A.7). By Proposition 4.1 and Lemma 4.2, the minimization of ρ is equivalent to the problem

$$\max\Bigl\{\sum_{i\in N} \min\{0, x(i)\} \Bigm| x = u - \sum_{j\in M} x_j,\ x_j \in \widetilde{D}_j(p) \ (j \in M)\Bigr\}.$$

Based on this observation, we can prove Theorem 1.3; proof is given in Section A.10. The bound $\mathrm{O}(mn^4 \log U \log(mnU))$ in Theorem 1.3 is smaller than the bound $\mathrm{O}(mn^6 \log U \log(mnU))$ obtained by a straightforward application of the SFM algorithm in [12].

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] P. J.-J. Herings, R. Müller, and D. Vermeulen. Bisection auctions. ACM SIGecom Exchanges 8 (2009) Article No. 6.

[12] S. Iwata. A faster scaling algorithm for minimizing submodular functions. SIAM J. Comp. 32 (2002) 833–840.

[13] A. S. Kelso and V. P. Crawford. Job matching, coalition formation and gross substitutes. Econometrica 50 (1982) 1483–1504.

[14] V. Kolmogorov and A. Shioura. New algorithms for convex cost tension problem with application to computer vision. Discrete Optimization 6 (2009) 378–393.

[15] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. Games Econom. Behav. 55 (2006) 270–296.

[16] P. Milgrom and B. Strulovici. Substitute goods, auctions, and equilibrium. J. Economic Theory 144 (2009) 212–247.

[17] K. Murota. Convexity and Steinitz's exchange property. Adv. Math. 124 (1996) 272–311.

[18] K. Murota. Discrete Convex Analysis. SIAM, Philadelphia, 2003.

[19] K. Murota and A. Shioura. M-convex function on generalized polymatroid. Math. Oper. Res. 24 (1999) 95–105.

[20] 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.

[21] 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.

[22] N. Sun and Z. Yang. A double-track adjustment process for discrete markets with substitutes and complements. Econometrica 77 (2009) 933–952.

A Appendix

A.1 Examples of Valuation Functions with SGS Condition

We present various examples of valuation functions $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{R}$ with the SGS condition (or equivalently, functions with M$^\natural$-concavity).

A simplest example of M$^\natural$-concave function is a linear function $f(x) = a^\top x \ x \in [\mathbf{0}, u]_{\mathbb{Z}}$ with a vector $a \in \mathbb{R}^n$. Below we give some nontrivial examples. See [18] for more examples of M$^\natural$-concave functions.

Example 1 (Laminar concave functions)**.** Let $\mathcal{T} \subseteq 2^N$ be a laminar family, i.e., $X \cap Y = \emptyset$ or $X \subseteq Y$ or $X \supseteq Y$ holds for every $X, Y \in \mathcal{T}$. For $Y \in \mathcal{T}$, let $\varphi_Y : \mathbb{Z}_+ \to \mathbb{R}$ be a univariate concave function. Define a function $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{R}$ by

$$f(x) = \sum_{Y \in \mathcal{T}} f_Y(x(Y)) \qquad (x \in [\mathbf{0}, u]_{\mathbb{Z}}),$$

which is called a *laminar concave function* [18]. Every laminar concave function is an M$^\natural$-concave function. □

Example 2 (Maximum-weight bipartite matching and its extension)**.** Consider a complete bipartite graph G with two vertex sets N and J, where N and J correspond to workers and jobs, respectively. We assume that for every $(i, v) \in N \times J$, profit $w(i, v) \in \mathbb{R}_+$ can be obtained by assigning worker i to job v. Consider a matching $M \subseteq N \times J$ between workers and jobs which maximizes the total profit. Define $f : \{0, 1\}^n \to \mathbb{R}$ by

$$f(\chi_X) = \max\{ \sum_{(i,v) \in M} w(i, v) \mid \exists M : \text{matching in } G \text{ s.t. } \partial_N M = X\} \ (X \in 2^N),$$

where $\partial_N M$ denotes the set of vertices in N covered by edges in M. Then, f is an M$^\natural$-concave function.

We consider a more general setting where each i corresponds to a *type* of workers and there are $u(i) \in \mathbb{Z}_+$ workers of type i. In a similar way as above, we can define a function $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{R}$ by

$$\begin{aligned} f(x) \quad = \quad & \max\{\sum_{i \in N} \sum_{v \in J} w(i, v) a(i, v) \mid \exists a : N \times J \to \mathbb{Z}_+ \\ & \qquad \text{s.t. } \sum_{v \in J} a(i, v) = x(i) \ (\forall i \in N)\} \quad (x \in [\mathbf{0}, u]_{\mathbb{Z}}). \end{aligned}$$

This f is an M$^\natural$-concave function.

A much more general example of M$^\natural$-concave functions can be obtained from the maximum-weight network flow problem (see [18]). □

Example 3 (Quadratic functions)**.** *Let $A = (a(i,k) \mid i,k \in N) \in \mathbb{R}^{N\times N}$ be a symmetric matrix, i.e., $a(i,k) = a(k,i)$ for $i,k \in N$. A quadratic function $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{R}$ given by*

$$f(x) = \sum_{i\in N}\sum_{k\in N} a(i,k)x(i)x(k) \quad (x \in [\mathbf{0}, u]_{\mathbb{Z}})$$

is M$^\natural$-concave if the matrix A satisfies the following condition:

$$a(i,k) \le 0 \ (\forall i,k \in N), \quad a(i,k) \le \max\{a(i,\ell), a(k,\ell)\} \text{ if } \{i,k\} \cap \{\ell\} = \emptyset.$$

In particular, a quadratic function $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{R}$ given by

$$f(x) = \sum_{i\in N} a(i)x(i)^2 + b\sum_{i<k} x(i)x(k) \quad (x \in [\mathbf{0}, u]_{\mathbb{Z}})$$

with $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$ is M$^\natural$-concave if a and b satisfies the condition $0 \ge b \ge 2\max_{i\in N} a(i)$ [18]. □

Example 4 (Maximum-value functions)**.** *Given a nonnegative vector $w \in \mathbb{R}^n_+$, we define a function $f : \{0,1\}^n \to \mathbb{R}_+$ by*

$$f(\chi_X) = \begin{cases} \max\{a(i) \mid i \in X\} & (\text{if } X \ne \emptyset), \\ 0 & (\text{if } X = \emptyset). \end{cases}$$

This corresponds to a valuation function of a bidder who wans only one item. We can show that f is an M$^\natural$-concave function [18]. □

Example 5 (Weighted rank functions)**.** Let $\mathcal{I} \subseteq 2^N$ be the family of independent sets of a matroid, and $w \in \mathbb{R}^n_+$. Define a function $f : \{0,1\}^n \to \mathbb{R}_+$ by

$$f(\chi_X) = \max\{w(Y) \mid Y \subseteq X,\ Y \in \mathcal{I}\} \quad (X \in 2^N),$$

which is called the *weighted rank function.* If $w(i) = 1$ ($i \in N$), then f is an ordinary rank function of the matroid $(N, \mathcal{I})$. Every weighted rank function is an M$^\natural$-concave function [**?**]. □

A.2 Proof of M$^\natural$-concavity and L$^\natural$-convexity

We firstly give a proof of Theorem 1.5. Let $\tilde{N} = \{(i,\beta) \mid i \in N,\ 1 \le \beta \le u(i)\}$. Given a function $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{Z}$, we define a function $\tilde{f} : \{0,1\}^{\tilde{N}} \to \mathbb{Z}$ as follows:

$$\text{for } \tilde{x} \in \{0,1\}^{\tilde{N}}, \qquad \tilde{f}(\tilde{x}) = f(x), \text{ where } x(i) = \sum_{\beta=1}^{u(i)} \tilde{x}(i,\beta) \ (i \in N). \quad (7)$$

It is known that the SGS condition for f is equivalent to the GS condition for $\tilde{f}$.

Proposition A.1 ([16]). *A function $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{Z}$ satisfies the SGS condition if and only if the function $\tilde{f} : \{0,1\}^{\tilde{N}} \to \mathbb{Z}$ defined by* (7) *satisfies the GS condition.*

We can also show the following; the proof is rather straightforward from the definition of M$^\natural$-concavity in Section 2 and therefore omitted.

Proposition A.2. *A function $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{Z}$ is M$^\natural$-concave if and only if the function $\tilde{f} : \{0,1\}^{\tilde{N}} \to \mathbb{Z}$ defined by* (7) *is M$^\natural$-concave.*

It is also known that for a valuation function on 0-1 vectors, the GS condition is equivalent to M$^\natural$-concavity.

Proposition A.3 ([8]). *A function $\tilde{f} : \{0,1\}^{\tilde{N}} \to \mathbb{Z}$ satisfies the condition* (GS) *if and only if it is an M$^\natural$-concave function.*

Theorem 1.5 can be obtained immediately by combining Propositions A.1, A.2, and A.3.

We then prove Proposition 2.1 by using the following relation between M$^\natural$-concavity and L$^\natural$-convexity.

Proposition A.4 ([18]). *Let $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{Z}$ be a function. Then, f is an M$^\natural$-concave function if and only if the function $g : \mathbb{Z}^n \to \mathbb{Z}$ defined by*

$$g(p) = \max\{f(x) - p^\top x \mid x \in [\mathbf{0}, u]_{\mathbb{Z}}\} \qquad (p \in \mathbb{Z}^n)$$

is an L$^\natural$-convex function.

Proposition 2.1 follows immediately from this property.

A.3 Proof of Proposition 3.1

We firstly recall that the existence of an integral equilibrium price vector is shown in [1]. Let $p^* \in \mathbb{Z}^n$ be an equilibrium price vector, and suppose that $p^*(i) > \bar{p}(i)$ for some fixed $i \in N$. Let p' be the vector obtained from p^* by replacing the component $p^*(i)$ with $\bar{p}(i)$. We show that the vector p' is also an equilibrium price vector. In the proof, we use the following property.

Proposition A.5 ([18]). *Let $f : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{R}$ be an M$^\natural$-concave function. Then, f is a submodular function on $[\mathbf{0}, u]_{\mathbb{Z}}$. In particular, for $x, y \in [\mathbf{0}, u]_{\mathbb{Z}}$ with $x \leq y$ and $i \in N$ with $y(i) < u(i)$, it holds that $f(x + \chi_i) - f(x) \geq f(y + \chi_i) - f(y)$.*

Let $x_1^*, x_2^*, \ldots, x_m^*$ be an allocation in a Walrasian equilibrium. That is, $x_1^* + x_2^* + \cdots + x_m^* = u$ and $x_j^* \in D_j(p^*)$ for all $j \in M$. To show that the vector p' is also an equilibrium price vector, it suffices to prove that $x_j^* \in D_j(p')$ for all $j \in M$.

Claim 1: $x^*_j(i) = 0$ for all $j \in M$.
[Proof of Claim 1] For every $x \in [\mathbf{0}, u]_{\mathbb{Z}}$ with $x(i) > 0$, it holds that

$$\begin{aligned} &\{f_j(x) - (p^*)^\top x\} - \{f_j(x - \chi_i) - (p^*)^\top (x - \chi_i)\} \\ &= f_j(x) - f_j(x - \chi_i) - p^*(i) \\ &< \{f_j(x) - f_j(x - \chi_i)\} - \{f_j(\chi_i) - f_j(\mathbf{0})\} \le 0, \end{aligned}$$

where the first inequality is by $p^*(i) > \bar{p}(i) \ge f_j(\chi_i) - f_j(\mathbf{0})$, and the last inequality by Proposition A.5. Hence, we have

$$f_j(x) - (p^*)^\top x < f_j(x - \chi_i) - (p^*)^\top (x - \chi_i)$$

for every $x \in [\mathbf{0}, u]_{\mathbb{Z}}$ with $x(i) > 0$. This implies that if $x^* \in D_j(p^*)$ then $x^*(i) = 0$. Hence, the claim follows. [End of Claim 1]

Claim 2: There exists some $x^* \in D_j(p')$ such that $x^*(i) = 0$.
[Proof of Claim 2] In a similar way as in the proof of Claim 1, we can show that

$$f_j(x) - (p')^\top x \le f_j(x - \chi_i) - (p')^\top (x - \chi_i)$$

for every $x \in [\mathbf{0}, u]_{\mathbb{Z}}$ with $x(i) > 0$. This implies the claim. [End of Claim 2]

By Claims 1 and 2, we have $x^*_j \in D_j(p')$ if and only if

$$f_j(x^*_j) - (p')^\top x^*_j \ge f_j(y) - (p')^\top y$$

holds for all $y \in [\mathbf{0}, u]_{\mathbb{Z}}$ with $y(i) = 0$. This inequality can be shown as follows:

$$\begin{aligned} &\{f_j(x^*_j) - (p')^\top x^*_j\} - \{f_j(y) - (p')^\top y\} \\ &= \{f_j(x^*_j) - (p^*)^\top x^*_j\} - \{f_j(y) - (p^*)^\top y\} \ge 0, \end{aligned}$$

where the equality is by $y(i) = x^*_j(i) = 0$ and the inequality is by $x^*_j \in D_j(p^*)$. Hence, we have $x^*_j \in D_j(p')$.

By repeating the argument above, we may assume that $p^* \le \bar{p}$ holds. In a similar way, we can show that there exists an equilibrium price vector p^* satisfying both of $p^* \le \bar{p}$ and $p^* \ge \mathbf{0}$.

A.4 Proof of Theorem 1.2 for Algorithm Ascend

Theorem 1.2 can be proved by using the following property repeatedly.

Proposition A.6. *Let $p \in [\mathbf{0}, \bar{p}]_{\mathbb{Z}}$ be a vector with $\hat{\mu}(p) > 0$, and $X \subseteq N$ be a set minimizing the value $L(p + \chi_X)$. Then, there exists a minimizer p^* of L satisfying*

$$p^* \ge p + \chi_X, \quad \|p^* - (p + \chi_X)\|_\infty = \hat{\mu}(p + \chi_X) = \hat{\mu}(p) - 1.$$

Proof. The inequality $\hat{\mu}(p + \chi_X) \geq \hat{\mu}(p) - 1$ can be shown as follows. By the triangle inequality, we have $\|p^* - (p + \chi_X)\|_\infty \geq \|p^* - p\|_\infty - 1$ for every $p^* \in \mathbb{Z}^n$. Taking the minimum over all $p^* \in \arg\min L$ with $p^* \geq p + \chi_X$, we obtain

$$\begin{aligned} \hat{\mu}(p + \chi_X) &\geq \min\{\|p^* - p\|_\infty \mid p^* \in \arg\min L,\ p^* \geq p + \chi_X\} - 1 \\ &\geq \min\{\|p^* - p\|_\infty \mid p^* \in \arg\min L,\ p^* \geq p\} - 1 \ = \ \hat{\mu}(p) - 1. \end{aligned}$$

In the following, we show that there exists a minimizer p^* of L satisfying

$$p^* \geq p + \chi_X, \quad \|p^* - (p + \chi_X)\|_\infty \leq \hat{\mu}(p) - 1.$$

Note that $\|p^* - (p + \chi_X)\|_\infty \geq \hat{\mu}(p + \chi_X)$ holds for such p^*.

Let $\hat{p}$ be a vector in the set $\{p^* \in \arg\min L \mid p^* \geq p\}$ satisfying $\|\hat{p} - p\|_\infty = \hat{\mu}(p)$, and assume that $\hat{p}$ is minimal among all such vectors. We denote

$$A = \arg\max_{i \in N}\{\hat{p}(i) - p(i)\}.$$

Since $\|\hat{p} - p\|_\infty = \hat{\mu}(p) > 0$, we have $p \neq \hat{p}$ and $A \neq \emptyset$.

Claim 1: We have $A \subseteq X$.
[Proof of Claim 1] Assume, to the contrary, that $A \setminus X \neq \emptyset$ holds. Since $\hat{p} \neq p$, it holds that $\mathrm{supp}^+(\hat{p} - p) \neq \emptyset$. Since $A \subseteq \mathrm{supp}^+(\hat{p} - p)$, we have

$$\mathrm{supp}^+(\hat{p} - (p + \chi_X)) \supseteq A \setminus X \neq \emptyset.$$

We also have

$$\arg\max_{i \in N}\{\hat{p}(i) - (p + \chi_X)(i)\} = A \setminus X.$$

Hence, Proposition 3.5 implies that

$$L(\hat{p})+L(p+\chi_X) \geq L(\hat{p}-\chi_{A\setminus X})+L(p+\chi_X+\chi_{A\setminus X}) = L(\hat{p}-\chi_{A\setminus X})+L(p+\chi_{X\cup A}). \tag{8}$$

Since $A \subseteq \mathrm{supp}^+(\hat{p} - p)$, we have $\hat{p} \geq \hat{p} - \chi_{A\setminus X} \geq p$, implying that $L(\hat{p}) < L(\hat{p} - \chi_{A\setminus X})$ by the choice of $\hat{p}$. This inequality, together with (8), implies that $L(p + \chi_X) > L(p + \chi_{X\cup A})$, a contradiction to the choice of X. Hence, we have $A \subseteq X$. [End of Proof of Claim 1]

Suppose firstly that the condition $\hat{p} \geq p + \chi_X$ holds. Then, we have

$$\hat{\mu}(p + \chi_X) \leq \|\hat{p} - (p + \chi_X)\|_\infty = \|\hat{p} - p\|_\infty - 1 = \hat{\mu}(p) - 1,$$

where the first equality is by Claim 1.

We next consider the case where the condition $\hat{p} \geq p + \chi_X$ fails, i.e., it holds that

$$B \cap X \neq \emptyset, \qquad \text{where } B = \{i \in N \mid \hat{p}(i) = p(i)\}.$$

Since $\hat{p} \geq p$, we have

$$\hat{p}(i) = p(i) \quad (\forall i \in B), \qquad \hat{p}(i) > p(i) \quad (\forall i \in N \setminus B), \tag{9}$$

from which $\hat{p} + \chi_{B\cap X} \geq p + \chi_X$ follows.

We now show that $\hat{p}+\chi_{B\cap X} \in \arg\min L$ holds. The condition (9) implies

$$\arg\max_{i\in N}\{(p+\chi_X)(i) - \hat{p}(i)\} = B \cap X \neq \emptyset.$$

Hence, it follows from Proposition 3.5 that

$$L(p+\chi_X)+L(\hat{p}) \geq L(p+\chi_X-\chi_{B\cap X})+L(\hat{p}+\chi_{B\cap X}) = L(p+\chi_{X\setminus B})+L(\hat{p}+\chi_{B\cap X}). \tag{10}$$

By the choice of X, we have $L(p+\chi_X) \leq L(p+\chi_{X\setminus B})$, which, together with (10), implies that $L(\hat{p}) \geq L(\hat{p}+\chi_{B\cap X})$, i.e., $\hat{p}+\chi_{B\cap X}$ is also a minimizer of L.

We have $A \cap B = \emptyset$ since

$$\hat{p} \neq p, \quad A = \arg\max_{i\in N}\{\hat{p}(i) - p(i)\}, \quad B = \arg\min_{i\in N}\{\hat{p}(i) - p(i)\}.$$

Hence, it holds that $A \subseteq X \setminus B$, implying that

$$\hat{\mu}(p+\chi_X) \leq \|(\hat{p}+\chi_{B\cap X}) - (p+\chi_X)\|_\infty = \|\hat{p}-p\|_\infty - 1 = \hat{\mu}(p) - 1.$$

□ □

A.5 Proof of Theorem 3.3 for Algorithm TwoPhase

Let $\hat{p}$ be the price vector at the end of the ascending phase and $\check{p}$ be the output of the algorithm. Also, let p^* be a minimizer of function L such that

$$\|p^* - p\|^+_\infty + \|p^* - p\|^-_\infty = \mu(p^\circ).$$

Note that for every $q \in \arg\min L$ satisfying

$$\left.\begin{array}{ll} p^\circ(i) \leq q(i) \leq p^*(i) & \text{if } p^\circ(i) \leq p^*(i), \\ p^\circ(i) \geq q(i) \geq p^*(i) & \text{if } p^\circ(i) \geq p^*(i), \end{array}\right\} \tag{11}$$

we have $\|q-p\|^+_\infty + \|q-p\|^-_\infty = \mu(p^\circ)$. Hence, we may assume that p^* satisfies the condition that

$$\text{there exists no } q \in \arg\min L \text{ with } q \neq p^* \text{ satisfying (11).}$$

We now show several lemmas below, from which Theorem 3.3 follows.

Lemma A.7. *The vector $\hat{p}$ satisfies $\hat{p} \in \arg\min\{L(p) \mid p \in \mathbb{Z}^n,\ p \geq p^\circ\}$ and $\hat{p} \geq p^*$. Moreover, the number of iterations in the ascending phase is equal to $\|\hat{p} - p^\circ\|_\infty + 1$.*

Proof. The behavior of the ascending phase is the same as that of the algorithm Ascend applied to the function $\hat{L} : \mathbb{Z}^n \to \mathbb{Z} \cup \{+\infty\}$ given by

$$\hat{L}(p) = \begin{cases} L(p) & (\text{if } p \geq p^\circ), \\ +\infty & (\text{otherwise}). \end{cases}$$

It is obvious that p° is a lower bound of all minimizers of $\hat{L}$. This observation, together with Theorem 1.2, implies that $\hat{p}$ is a minimizer of the function $\hat{L}$, i.e., it holds that

$$\hat{p} \in \arg\min\{L(p) \mid p \in \mathbb{Z}^n,\ p \geq p^\circ\}. \tag{12}$$

Theorem 1.2 also implies that the ascending phase terminates in $\|\hat{p}-p^\circ\|_\infty+1$ iterations.

We then prove $\hat{p} \geq p^*$. Assume, to the contrary, that $\hat{p} \not\geq p^*$. Then, we have $\mathrm{supp}^+(p^* - \hat{p}) \neq \emptyset$, and therefore Proposition 3.5 implies that

$$L(p^*) + L(\hat{p}) \geq L(p^* - \chi_X) + L(\hat{p} + \chi_X), \tag{13}$$

where

$$X = \arg\max_{i \in N}\{p^*(i) - \hat{p}(i)\} \subseteq \mathrm{supp}^+(p^* - \hat{p}).$$

Since $\hat{p} + \chi_X \geq \hat{p} \geq p^\circ$, we have $L(\hat{p} + \chi_X) \geq L(\hat{p})$ by (12), which, together with (13), implies $L(p^* - \chi_X) \leq L(p^*)$, i.e., $p^* - \chi_X \in \arg\min L$. This, however, is a contradiction to the choice of p^* since

$$p^*(i) \geq p^*(i) - 1 \geq \hat{p}(i) \geq p^\circ(i) \qquad (\forall i \in X).$$

□ □

Lemma A.8. $\|\hat{p} - p^\circ\|_\infty \leq \mu(p^\circ)$ *holds.*

Proof. In this proof, we may assume that $\hat{p}$ is a minimal vector in $\arg\min\{L(p) \mid p \in \mathbb{Z}^n,\ p \geq p^\circ\}$; if $\hat{p}$ is not minimal, then there exists a minimal q in $\arg\min\{L(p) \mid p \in \mathbb{Z}^n,\ p \geq p^\circ\}$ such that $q \leq \hat{p}$, which satisfies $\|q - p^\circ\|_\infty = \|\hat{p} - p^\circ\|_\infty$. Since $\mu(p^\circ) = \|p^* - p^\circ\|_\infty^+ + \|p^* - p^\circ\|_\infty^-$, it suffices to prove that

$$\|\hat{p} - p^\circ\|_\infty \leq \|p^* - p^\circ\|_\infty^+ + \|p^* - p^\circ\|_\infty^-$$

holds.

If $p^* \geq p^\circ$, then the choice of $\hat{p}$ implies that $\hat{p} \in \arg\min L$ and $\|\hat{p}-p^\circ\|_\infty = \|p^* - p^\circ\|_\infty = \mu(p^\circ)$. Hence, we may assume that $\mathrm{supp}^+(p^\circ - p^*) \neq \emptyset$, from which follows that $\mathrm{supp}^+(\hat{p} - p^*) \neq \emptyset$.

Let $X = \arg\max_{i \in N}\{\hat{p}(i) - p^*(i)\}$. By Proposition 3.5, it holds that

$$L(\hat{p}) + L(p^*) \geq L(\hat{p} - \chi_X) + L(p^* + \chi_X). \tag{14}$$

Assume, to the contrary, that $\hat{p}(i) > p^\circ(i)$ for all $i \in X$. Since $p^* \in \arg\min L$, we have $L(p^* + \chi_X) \geq L(p^*)$, which, combined with (14), implies $L(\hat{p} - \chi_X) \leq L(\hat{p})$. It follows from this inequality and the definition of $\hat{p}$ that $L(\hat{p} - \chi_X) = L(\hat{p})$, a contradiction to the minimality of $\hat{p}$ since $\hat{p} - \chi_X \geq p^\circ$.

Hence, we have $\hat{p}(k) = p^\circ(k)$ for some $k \in X$. It holds that

$$\begin{aligned} p^\circ(k) - p^*(k) \;=\; \hat{p}(k) - p^*(k) &= \max_{i \in N}\{\hat{p}(i) - p^*(i)\} \\ &\geq \max_{i \in N}\{p^\circ(i) - p^*(i)\} \\ &= \|p^* - p^\circ\|_\infty^- \;\geq\; p^\circ(k) - p^*(k). \end{aligned}$$

Hence, all the inequalities in this formula holds with equality. From this formula follows that

$$\begin{aligned} \{\hat{p}(i) - p^\circ(i)\} - \{p^*(i) - p^\circ(i)\} &= \hat{p}(i) - p^*(i) \\ &\leq \max_{i' \in N}\{\hat{p}(i') - p^*(i')\} \;=\; \|p^* - p^\circ\|_\infty^- \end{aligned}$$

for every $i \in N$, implying that

$$\hat{p}(i) - p^\circ(i) \leq \{p^*(i) - p^\circ(i)\} + \|p^* - p^\circ\|_\infty^- \leq \|p^* - p^\circ\|_\infty^+ + \|p^* - p^\circ\|_\infty^-.$$

Hence, we have

$$\|\hat{p} - p^\circ\|_\infty \leq \|p^* - p^\circ\|_\infty^+ + \|p^* - p^\circ\|_\infty^-.$$

□ □

Lemma A.9. *The vector* $\check{p}$ *satisfies* $\check{p} \in \arg\min L$. *Moreover, the number of iterations in the descending phase is at most* $\|\hat{p} - p^*\|_\infty + 1$.

Proof. The behavior of the descending phase is the same as that of the algorithm DESCEND applied to function L with the initial vector $\hat{p}$. Since $\hat{p}$ is an upper bound of the minimizer p^* of L by Lemma A.7, this observation and Theorem 3.2 imply that $\check{p} \in \arg\min L$. Theorem 3.2 also implies that the descending phase terminates in $\check{\mu}(\hat{p}) + 1 = \|\check{p} - \hat{p}\|_\infty + 1$ iterations. Since p^* is a minimizer of L with $p^* \leq \hat{p}$, we have $\check{\mu}(\hat{p}) \leq \|p^* - \hat{p}\|_\infty$. This concludes the proof. □ □

Lemma A.10. $\|\hat{p} - p^*\|_\infty \leq 2\mu(p^\circ)$ *holds.*

Proof. We have

$$\begin{aligned} \|\hat{p} - p^*\|_\infty &\leq \|p^* - p^\circ\|_\infty + \|\hat{p} - p^\circ\|_\infty \\ &\leq \{\|p^* - p^\circ\|_\infty^+ + \|p^* - p^\circ\|_\infty^-\} + \|\hat{p} - p^\circ\|_\infty \\ &\leq 2\mu(p^\circ), \end{aligned}$$

where the first inequality is by the triangle inequality and the third is by Lemma A.8 and the definition of p^*. □ □

Theorem 3.3 follows from Lemmas A.7, A.8, A.9, and A.10 shown above.

A.6 Proof of Theorem 3.4 for Algorithm Greedy

We firstly show that the output of algorithm Greedy is indeed a minimizer of the Lyapunov function L.

Proposition A.11 ([18])**.** *Let $g : \mathbb{Z}^n \to \mathbb{R}$ be an $L^\natural$-convex function. For every $p^* \in \mathbb{Z}^n$, we have $p^* \in \arg\min g$ if and only if $g(p + \varepsilon\chi_X) \geq g(p)$ for every $X \in 2^N$ and $\varepsilon \in \{+1, -1\}$.*

Recall that L is an $\mathrm{L}^\natural$-convex function by Theorem 1.6. Since the output p of the algorithm Greedy satisfies the condition

$$L(p + \varepsilon\chi_X) \geq L(p) \qquad (\forall X \in 2^N,\ \varepsilon \in \{+1, -1\}),$$

it is a minimizer of L by Proposition A.11.

To obtain the bound for the number of iterations in Theorem 3.4, it suffices to apply the following property repeatedly. The proof is similar to (but more difficult than) the one for Proposition A.6.

Proposition A.12. *Let $p \in [\mathbf{0}, \bar{p}]_{\mathbb{Z}}$ be a vector with $\mu(p) > 0$. Suppose that $\varepsilon \in \{+1, -1\}$ and $X \subseteq N$ minimize the value $L(p+\varepsilon\chi_X)$. Then, there exists a minimizer p^* of L satisfying*

$$\|p^* - (p + \varepsilon\chi_X)\|_\infty = \mu(p + \varepsilon\chi_X) = \mu(p) - 1.$$

Proof. We consider the case with $\varepsilon = +1$ since the case with $\varepsilon = -1$ can be dealt with similarly. For every $d \in \mathbb{Z}^n$ and $Y \subseteq N$, we have

$$\|d - \chi_Y\|_\infty^+ \geq \|d\|_\infty^+ - 1, \qquad \|d - \chi_Y\|_\infty^- \geq \|d\|_\infty^-.$$

Hence, it holds that

$$\begin{aligned} \mu(p + \chi_X) &= \min\{\|p^* - (p + \chi_X)\|_\infty^+ + \|p^* - (p + \chi_X)\|_\infty^- \mid p^* \in \arg\min L\} \\ &\geq \min\{\|p^* - p\|_\infty^+ + \|p^* - p\|_\infty^- \mid p^* \in \arg\min L\} - 1 \\ &= \mu(p) - 1. \end{aligned}$$

In the following, we show that there exists a minimizer p^* of L satisfying

$$p^* \geq p + \chi_X, \quad \|p^* - (p + \chi_X)\|_\infty^+ + \|p^* - (p + \chi_X)\|_\infty^- \leq \mu(p) - 1.$$

Note that $\|p^* - (p + \chi_X)\|_\infty \geq \mu(p + \chi_X)$ holds for such p^*.

We denote

$$S_* = \{p^* \in \arg\min L \mid \|p_* - p\|_\infty^+ + \|p_* - p\|_\infty^- = \mu(p)\}.$$

Let $\hat{p}$ be a vector in S_*; note that for every $p^* \in \arg\min L$ satisfying

$$p(i) \leq p^*(i) \leq \hat{p}(i) \quad \text{if } p(i) \leq \hat{p}(i), \qquad p(i) \geq p^*(i) \geq \hat{p}(i) \quad \text{if } p(i) > \hat{p}(i), \tag{15}$$

we have $p^* \in S_*$. Hence, we may assume that $\hat{p}$ satisfies the following condition:

there exists no $p^* \in \arg\min L$ with $p^* \neq \hat{p}$ satisfying (15).

We denote

$$A = \arg\max_{i \in N} \{\hat{p}(i) - p(i)\}, \qquad B = \arg\min_{i \in N} \{\hat{p}(i) - p(i)\}.$$

Claim 1: There exists some $p^* \in S_*$ such that $\max_{i \in N}\{p_*(i) - p(i)\} > 0$.
[Proof of Claim 1] Assume, to the contrary, that

$$p_* \leq p \qquad (\forall p^* \in S_*). \tag{16}$$

Then, $\mu(p) = \|\hat{p} - p\|_\infty$ and $\hat{p}$ is a maximal vector in $\arg\min L$ by the choice of $\hat{p}$. By $\mathrm{L}^\natural$-convexity of L, we have

$$L(p + \chi_X) + L(\hat{p}) \geq L((p + \chi_X - \mathbf{1}) \vee \hat{p}) + L((p + \chi_X) \wedge (\hat{p} + \mathbf{1})). \tag{17}$$

Suppose that $\hat{p}(i) < p(i)$ holds for all $i \in N$. Hence, we have

$$(p + \chi_X - \mathbf{1}) \vee \hat{p} = p - \chi_{N \setminus X}, \qquad (p + \chi_X) \wedge (\hat{p} + \mathbf{1}) = \hat{p} + \mathbf{1},$$

which, together with (17), implies

$$L(p + \chi_X) + L(\hat{p}) \geq L(p - \chi_{N \setminus X}) + L(\hat{p} + \mathbf{1}). \tag{18}$$

By the choice of $\varepsilon = +1$ and X, we have $L(p + \chi_X) \leq L(p - \chi_{N \setminus X})$. From this and (18) follows that $L(\hat{p}) \geq L(\hat{p} + \mathbf{1})$, implying that $\hat{p} + \mathbf{1} \in \arg\min L$. This, however, is a contradiction to the maximality of $\hat{p}$.

We then consider the case with $\max_{i \in N}\{\hat{p}(i) - p(i)\} = 0$. Then, we have

$$(p + \chi_X - \mathbf{1}) \vee \hat{p} = p - \chi_{N \setminus (X \cup A)}, \qquad (p + \chi_X) \wedge (\hat{p} + \mathbf{1}) = \hat{p} + \chi_{(N \setminus A) \cup (X \cap A)},$$

which, together with (17), implies

$$L(p + \chi_X) + L(\hat{p}) \geq L(p - \chi_{N \setminus (X \cup A)}) + L(\hat{p} + \chi_{(N \setminus A) \cup (X \cap A)}). \tag{19}$$

By the choice of $\varepsilon = +1$ and X, we have $L(p + \chi_X) \leq L(p - \chi_{N \setminus (X \cup A)})$. From this and (19) follows that $L(\hat{p}) \geq L(\hat{p} + \chi_{(N \setminus A) \cup (X \cap A)})$, implying that $\hat{p} + \chi_{(N \setminus A) \cup (X \cap A)} \in \arg\min L$. This, however, is a contradiction to the fact that $\hat{p}$ is a maximal vector in $\arg\min L$. [End of Proof of Claim 1]

By Claim 1, we may assume that $\max_{i \in N}\{\hat{p}(i) - p(i)\} > 0$.

Claim 2: We have $A \subseteq X$.
[Proof of Claim 2] Assume, to the contrary, that $A \setminus X \neq \emptyset$ holds. Since $A \subseteq \mathrm{supp}^+(\hat{p} - p)$, we have

$$\mathrm{supp}^+(\hat{p} - (p + \chi_X)) \supseteq A \setminus X \neq \emptyset.$$

We also have

$$\arg\max_{i \in N}\{\hat{p}(i) - (p + \chi_X)(i)\} = A \setminus X.$$

Hence, Proposition 3.5 implies that

$$L(\hat{p})+L(p+\chi_X) \geq L(\hat{p}-\chi_{A\setminus X})+L(p+\chi_X+\chi_{A\setminus X}) = L(\hat{p}-\chi_{A\setminus X})+L(p+\chi_{X\cup A}). \tag{20}$$

Since the vector $p^* = \hat{p} - \chi_{A\setminus X}$ satisfies the condition (15), we have $L(\hat{p}) < L(\hat{p} - \chi_{A\setminus X})$ by the choice of $\hat{p}$. This inequality, together with (20), implies that $L(p + \chi_X) > L(p + \chi_{X\cup A})$, a contradiction to the choice of X. Hence, we have $A \subseteq X$. [End of Proof of Claim 2]

To show the inequality $\mu(p+\chi_X) \leq \mu(p)-1$, we firstly consider the case where $\min_{i\in N}\{\hat{p}(i) - p(i)\} > 0$ or $B \cap X = \emptyset$ holds. Then, we have

$$\|\hat{p} - (p + \chi_X)\|^-_\infty = \|\hat{p} - p\|^-_\infty.$$

By Claim 2, we have

$$\|\hat{p} - (p + \chi_X)\|^+_\infty = \|\hat{p} - p\|^+_\infty - 1.$$

Therefore, we can obtain the desired inequality as follows:

$$\begin{aligned} \mu(p + \chi_X) &\leq \|\hat{p} - (p + \chi_X)\|^+_\infty + \|\hat{p} - (p + \chi_X)\|^-_\infty \\ &= \|\hat{p} - p\|^+_\infty + \|\hat{p} - p\|^-_\infty - 1 \ = \ \mu(p) - 1. \end{aligned}$$

We then consider the case with $\min_{i\in N}\{\hat{p}(i) - p(i)\} > 0$ and $B \cap X = \emptyset$. We claim that $\hat{p} + \chi_{B\cap X} \in \arg\min L$ holds.

Since $\min_{i\in N}\{\hat{p}(i) - p(i)\} \leq 0$ and $B \cap X \neq \emptyset$ hold, we have $\mathrm{supp}^+((p + \chi_X) - \hat{p}) \neq \emptyset$. Since

$$\arg\max_{i \in N}\{(p + \chi_X)(i) - \hat{p}(i)\} = B \cap X,$$

it follows from Proposition 3.5 that

$$L(p+\chi_X)+L(\hat{p}) \geq L(p+\chi_X-\chi_{B\cap X})+L(\hat{p}+\chi_{B\cap X}) = L(p+\chi_{X\setminus B})+L(\hat{p}+\chi_{B\cap X}). \tag{21}$$

By the choice of X, we have $L(p+\chi_X) \leq L(p+\chi_{X\setminus B})$, which, together with (21), implies that $L(\hat{p}) \geq L(\hat{p}+\chi_{B\cap X})$, i.e., $\hat{p}+\chi_{B\cap X}$ is also a minimizer of g.

Put $p^* = \hat{p}+\chi_{B\cap X}$. If $\min_{i\in N}\{\hat{p}(i)-p(i)\} < 0$, then p^* satisfies the condition (15), a contradiction to the choice of $\hat{p}$. Hence, we have $\min_{i\in N}\{\hat{p}(i) - p(i)\} = 0$. This implies that $A \cap B = \emptyset$ since $\max_{i\in N}\{\hat{p}(i) - p(i)\} > 0$. Therefore, we have $A \subseteq X \setminus B$ by Claim 2, from which follows that

$$\|(\hat{p} + \chi_{B\cap X}) - (p + \chi_X)\|^+_\infty = \|\hat{p} - p - \chi_{X\setminus B}\|^+_\infty = \|\hat{p} - p\|^+_\infty - 1.$$

We also have

$$\|(\hat{p}+\chi_{B\cap X})-(p+\chi_X)\|_\infty^- = \|(\hat{p}-p-\chi_{X\setminus B}\|_\infty^- = \|\hat{p}-p\|_\infty^-.$$

Hence, it holds that

$$\begin{aligned}\mu(p+\chi_X) &\leq \|(\hat{p}+\chi_{B\cap X})-(p+\chi_X)\|_\infty^+ + \|(\hat{p}+\chi_{B\cap X})-(p+\chi_X)\|_\infty^- \\ &= (\|\hat{p}-p\|_\infty^+ - 1) + \|\hat{p}-p\|_\infty^- \ = \ \mu(p)-1.\end{aligned}$$

□ □

A.7 Structure of Demand Sets

To prove Lemma 1.4, we show some properties of demand sets.

For $j \in M$ and $p \in \mathbb{Z}^n$, we define set functions $\eta_j^p, \zeta_j^p : 2^N \to \mathbb{Z}$ by

$$\eta_j^p(X) = V_j(p+\chi_X) - V_j(p), \qquad \zeta_j^p(X) = V_j(p-\chi_X) - V_j(p) \qquad (X \in 2^N).$$

Since each $V_j(p)$ is a submodular function in p by Proposition 2.1, both of η_j and ζ_j are submodular set functions. Recall that $\widetilde{D}_j(p)$ is the set of minimal vectors in the demand set $D_j(p)$.

Proposition A.13. *For $j \in M$ and $p \in \mathbb{Z}^n$, the demand set $D_j(p)$ is a g-polymatroid given as*

$$D_j(p) = \{x \in \mathbb{Z}^n \mid -\eta_j^p(Y) \leq x(Y) \leq \zeta_j^p(Y) \ (\forall Y \in 2^N)\} \tag{22}$$

and the set $\widetilde{D}_j(p)$ is a base polyhedron given as $\widetilde{D}_j(p) = -B(\eta_j^p)$.

We sometimes omit the superscript p of η_j^p and ζ_j^p if it is clear from the context. In the following, we will show the equation (22) and that the submodular functions η_j and ζ_j satisfy the following condition:

$$\eta_j(X) + \zeta_j(Y) \geq \eta_j(X \setminus Y) + \zeta_j(Y \setminus X) \qquad (\forall X, Y \in 2^N). \tag{23}$$

From these facts follows that $D_j(p)$ is a g-polymatroid (see [7]). Moreover, since $D_j(p)$ is a g-polymatroid given by (22), the set of minimal vectors in $D_j(p)$ is a base polyhedron given by $-B(\eta_j)$, i.e., $\widetilde{D}_j(p) = -B(\eta_j)$ (see [7]); note that if B is a base polyhedron, then $-B$ is also a base polyhedron.

We firstly prove (23). Since the function V_j is an L$^\natural$-convex function, it satisfies the discrete mid-point convexity

$$V_j(p') + V_j(q') \geq V_j(\lceil (p'+q')/2 \rceil) + V_j(\lfloor (p'+q')/2 \rfloor) \qquad (\forall p', q' \in \mathbb{Z}^n),$$

Hence, it follows that

$$\begin{aligned}\eta_j(X) + \zeta_j(Y) &= \{V_j(p+\chi_X) - V_j(p)\} + \{V_j(p-\chi_Y) - V_j(p)\} \\ &\geq V_j(\left\lceil \frac{(p+\chi_X)+(p-\chi_Y)}{2} \right\rceil) + V_j(\left\lfloor \frac{(p+\chi_X)+(p-\chi_Y)}{2} \right\rfloor) - 2V_j(p) \\ &= V_j(p+\chi_{X\setminus Y}) + V_j(p+\chi_{Y\setminus X}) - 2V_j(p) \\ &= \eta_j(X \setminus Y) + \zeta_j(Y \setminus X).\end{aligned}$$

To prove (22), we use the following properties concerning M$^\natural$-convex and L$^\natural$-convex functions. For a function $g : \mathbb{Z}^n \to \mathbb{Z} \cup \{+\infty\}$, we define

$$\partial_{\mathbb{Z}} g(p) = \{x \in \mathbb{Z}^n \mid p \in \arg\min\{g(q) - x^\top q \mid q \in \mathbb{Z}^n\}\} \qquad (p \in \mathbb{Z}^n),$$

which we call the *integer subdifferential* of g at p.

Proposition A.14 ([18]). *Let $g : \mathbb{Z}^n \to \mathbb{Z} \cup \{+\infty\}$ be an integer-valued L$^\natural$-convex function. For $p \in \mathbb{Z}^n$, it holds that*

$$\partial_{\mathbb{Z}} g(p) = \{x \in \mathbb{Z}^n \mid g(p) - g(p - \chi_Y) \le x(Y) \le g(p + \chi_Y) - g(p) \ (\forall Y \in 2^N)\}.$$

Proposition A.15 ([18]). *Let $h : \mathbb{Z}^n \to \mathbb{Z} \cup \{+\infty\}$ be an integer-valued M$^\natural$-convex function such that* $\mathrm{dom}\, h$ *is bounded, and $g : \mathbb{Z}^n \to \mathbb{Z}$ be a function given by*

$$g(p) = \max\{p^\top x - h(x) \mid x \in \mathbb{Z}^n\} \qquad (p \in \mathbb{Z}^n). \tag{24}$$

Then, g is an L$^\natural$-convex function.

Proposition A.16 ([18]). *Let $h : \mathbb{Z}^n \to \mathbb{Z} \cup \{+\infty\}$ be an integer-valued M$^\natural$-convex function such that* $\mathrm{dom}\, h$ *is bounded, and $g : \mathbb{Z}^n \to \mathbb{Z}$ be a function given by* (24)*. Then, it holds that*

$$\arg\min\{h(x) - p^\top x \mid x \in \mathbb{Z}^n\} = \partial_{\mathbb{Z}} g(p)$$

for every $p \in \mathbb{Z}^n$.

The equation (22) can be proven as follows. We consider the function $h = -f_j$, which is an M$^\natural$-convex function by Theorem 1.5. We also define a function $g : \mathbb{Z}^n \to \mathbb{Z}$ by (24), which is an L$^\natural$-convex function by Proposition A.15. Note that for $p \in \mathbb{Z}^n$, we have

$$g(p) = \max\{p^\top x - h(x) \mid x \in [\mathbf{0}, u]_{\mathbb{Z}}\} = \max\{f_j(x) + p^\top x \mid x \in [\mathbf{0}, u]_{\mathbb{Z}}\} = V_j(-p). \tag{25}$$

For $p \in \mathbb{Z}^n$, we have

$$D_j(p) = \arg\max\{f_j(x) - p^\top x \mid x \in \mathbb{Z}^n\} = \arg\min\{h(x) + p^\top x \mid x \in \mathbb{Z}^n\}. \tag{26}$$

It follows from Proposition A.16 that

$$\arg\min\{h(x) + p^\top x \mid x \in \mathbb{Z}^n\} = \partial_{\mathbb{Z}} g(-p). \tag{27}$$

By Proposition A.14 and (25), we have

$$\begin{aligned} \partial_{\mathbb{Z}} g(-p) &= \{x \in \mathbb{Z}^n \mid g(-p) - g(-p - \chi_Y) \le x(Y) \le g(-p + \chi_Y) - g(-p) \ (\forall Y \in 2^N)\} \\ &= \{x \in \mathbb{Z}^n \mid V_j(p) - V_j(p + \chi_Y) \le x(Y) \le V_j(p - \chi_Y) - V_j(p) \ (\forall Y \in 2^N)\} \\ &= \{x \in \mathbb{Z}^n \mid -\eta_j^p(Y) \le x(Y) \le \zeta_j^p(Y) \ (\forall Y \in 2^N)\}, \end{aligned} \tag{28}$$

where we use the fact that g is an L$^\natural$-convex function. The equation (22) follows immediately from (26), (27), and (28). This concludes the proof of Proposition A.13.

We note that from Proposition A.13 we have

$$V_j(p + \chi_X) - V_j(p) = \eta_j^p(X) = -\min\{y(X) \mid y \in D_j(p)\} \qquad (\forall X \in 2^N),$$

from which the formula (4) follows.

A.8 Algorithms for Demand Sets

We prove some algorithmic properties of demand sets $D_j(p)$ used in Section 4.

Let D be a g-polymatroid such that $D \subseteq [\mathbf{0}, u]_{\mathbb{Z}}$, and assume that the membership test in D can be done in constant time. Note that a demand set $D_j(p)$ satisfies these conditions for every $j \in M$ and $p \in \mathbb{R}^n_+$ (see Section A.7).

We firstly show the following property:

Proposition A.17. *For a given $x \in D$ and $i, i' \in N$, the values*

$$\max\{\alpha \mid x+\alpha\chi_i \in D\}, \quad \max\{\alpha \mid x-\alpha\chi_i \in D\}, \quad \max\{\alpha \mid x+\alpha(\chi_i-\chi_{i'}) \in D\}$$

can be computed in $\mathrm{O}(\log U)$ *time.*

Proof. Since D is given as a set of integral vectors in a polyhedron, the values can be computed by using binary search combined with the membership test in D. Since the values are nonnegative integer at most U, the running time is $\mathrm{O}(\log U)$. □ □

Let $\tilde{D}$ be the set of minimal vectors in D, which is a base polyhedron (see [7]). Given a vector in D, we can compute a vector in $\tilde{D}$ efficiently.

Proposition A.18. *Given a vector $x^\circ \in D$, a vector in $\tilde{D}$ can be computed in* $\mathrm{O}(n \log U)$ *time.*

Proof. A vector in $\tilde{D}$ can be computed by the following algorithm:

Step 0: Set $x := x^\circ$.
Step 1: For $i = 1, 2, \ldots, n$ do the following:
Compute $\alpha_i = \max\{\alpha \mid x - \alpha\chi_i \in D\}$, and set $x := x - \alpha_i\chi_i$.
Step 2: Output x.

It is shown (see, e.g., [7]) that the output of the algorithm satisfies $x \in \tilde{D}$. From Proposition A.17 follows that the running time of the algorithm is $\mathrm{O}(n \log U)$. □ □

Proposition A.19. *Suppose that a vector $x^\circ \in D$ is given. For $X \in 2^N$, the values* $\min\{y(X) \mid y \in D\}$ *and* $\min\{y(X) \mid y \in \tilde{D}\}$ *can be solved in* $\mathrm{O}(n^2 \log U)$ *time.*

Proof. It is known that there exists an optimal solution y^* of the former problem such that $y^* \in \tilde{D}$ [7]. Hence, it suffices to consider the latter problem only. An optimal solution of $\min\{y(X) \mid y \in \tilde{D}\}$ can be computed by the following greedy algorithm [7], where we assume, without loss of generality, that $X = \{1, 2, \ldots, t\}$ for some t:

Step 1: Compute a vector x in $\tilde{D}$.
Step 2: For $i = 1, 2, \ldots, t$ do
For $k = t + 1, t + 2, \ldots, n$ do the following:
Compute $\alpha_{ik} = \max\{\alpha \mid x - \alpha(\chi_i - \chi_k) \in D\}$,
and set $x := x - \alpha_{ik}(\chi_i - \chi_k)$.
Step 3: Output x.

Note that the vector x remains in $\tilde{D}$ during the execution of the algorithm. By Proposition A.18, Step 1 can be done in $\mathrm{O}(n \log U)$ time. By Proposition A.17, each value α_{ik} can be computed in $\mathrm{O}(\log U)$ time. Hence, Step 2 requires $\mathrm{O}(n^2 \log U)$ time in total. This concludes the proof. □ □

A.9 Proof of Lemma 4.2

We show that

$$B(\rho) = \{u - \sum_{j \in M} x_j \mid x_j \in \widetilde{D}_j(p) \ (j \in M)\} \tag{29}$$

holds.

Recall that the submodular function ρ is given as

$$\rho(X) = L(p + \chi_X) - L(p) = \sum_{j=1}^{m} \eta_j(X) + u(X) \qquad (X \in 2^N).$$

Note that each η_j is a submodular function. Hence, it holds that

$$\begin{aligned} B(\rho) &= \{x \in \mathbb{Z}^n \mid x(Y) \le \sum_{j=1}^{m} \eta_j(Y) + u(Y) \ (\forall Y \subseteq N), \ x(N) = \sum_{j=1}^{m} \eta_j(N) + u(N)\} \\ &= \{y + u \mid y \in \mathbb{Z}^n, \ y(Y) \le \sum_{j=1}^{m} \eta_j(Y) \ (\forall Y \subseteq N), \ y(N) = \eta_j(N)\} \\ &= \{y + u \mid y \in B(\sum_{j=1}^{m} \eta_j)\}. \end{aligned}$$

Since each η_j is a submodular function, we have

$$B(\sum_{j=1}^{m} \eta_j) = \bigoplus_{j=1}^{m} B(\eta_j),$$

where $\bigoplus$ denotes the direct sum of sets (see [7]). By Proposition A.13, we have $\widetilde{D}_j(p) = -B(\eta_j)$. Hence, the equation (29) follows.

A.10 A Fast Algorithm for Submodular Function Minimization

We present an efficient algorithm for the minimization of the submodular function ρ given by (4), which is obtained by simplifying the existing SFM algorithms. Our idea is as follows. As mentioned in Section 4, the existing polynomial-time algorithms use a technique to represent a vector $x \in B(\rho)$ as a convex combination of extreme points in $B(\rho)$. i.e., x is represented as

$$x = \sum_{j=1}^{r} \lambda_j y_j \tag{30}$$

with some extreme points $y_1, y_2, \ldots, y_r$ of $B(\rho)$ and $\lambda_j \in \mathbb{R}$ with $0 \leq \lambda_j \leq 1$ and $\sum_{j=1}^{r} \lambda_j = 1$. Instead, we use the representation of a vector in $B(\rho)$ as shown in Lemma 4.2, i.e., $x \in B(\rho)$ is represented as

$$x = u - \sum_{j \in M} x_j \tag{31}$$

with $x_j \in \widetilde{D}_j(p)$ $(j \in M)$. This representation makes it easier to check the membership in $B(\rho)$ and an update of x. Below we show that with this representation the weakly-polynomial time algorithm proposed by Iwata, Fleischer, and Fujishige [A1] (IFF algorithm, for short) can be simplified and made faster in a nontrivial way.

We briefly review the IFF algorithm. The IFF algorithm maintains two kinds of variables: a vector $x \in \mathbb{R}^n$ and a flow vector $\varphi \in \mathbb{R}^A$, where A is the arc set of the complete directed graph $G = (N, A)$ on the node set N, i.e., $A = \{(v, w) \mid v, w \in N,\ v \neq w\}$. The vector x always satisfies the condition $x \in B(\rho)$ and is represented as a convex combination of extreme points in $B(\rho)$. The algorithm also maintains two disjoint node subsets $S, T \subseteq N$, and a subgraph $G^\circ = (N, A^\circ)$ with the arc set $A^\circ = \{(v, w) \mid (v, w) \in A,\ \varphi(v, w) = 0\}$.

The algorithm tries to find a directed path in G° from S to T, and if such a path exists, then it updates the flow φ by using a procedure called Augment. If such a path does not exist, then let W be the set of nodes currently reachable from S in G°. Then, it holds that $W \cap T = \emptyset$. The algorithm tries to enlarge the set W by using a procedure called Double-Exchange, which updates both of x and φ as follows.

The procedure Double-Exchange is applied to a triple (j, v, w), where j is an index of an extreme point of $B(\rho)$ in (30) and $v, w \in N$ are nodes satisfying $v \in W$, $w \in N \setminus W$, $\varphi(v, w) > 0$, and some additional condition. Such a triple is called an *active triple*. For the vector $y_j \in B(\rho)$ and $v, w \in N$, the exchange capacity of $\tilde{c}(y_j, v, w)$ is given as

$$\tilde{c}(y_j, v, w) = \max\{\beta \mid y_j + \beta(\chi_v - \chi_w) \in D_j(p)\}.$$

The procedure updates x and φ as $x := x + \alpha(\chi_v - \chi_w)$ and $\varphi(v, w) := \varphi(v, w) - \alpha$, so that $z = x + \partial\varphi$ remains unchanged, where $\alpha = \min\{\varphi(v, w), \lambda_j \tilde{c}(y_j, v, w)\}$ and λ_j is the coefficient in (30). According to the update of x, vectors and coefficients in the representation (30) are modified appropriately. In particular, the vector y_j is updated as $y_j := y_j + \tilde{c}(y_j, v, w)(\chi_v - \chi_w)$. We call Double-Exchange *saturating* if $\alpha = \lambda_j \tilde{c}(y_j, v, w)$ and *nonsaturating* otherwise.

Note that the size r of the set of vectors $y_1, y_2, \ldots, y_r$ may be increased by procedure Double-Exchange. To keep this size to be $\mathrm{O}(n)$, the IFF algorithm sometimes applies a procedure called Reduce; given a representation (30), procedure Reduce reduce the number of extreme points in this representation by a variant of Gaussian elimination.

Proposition A.20 ([A1]). *The IFF algorithm finds a minimizer of an integer-valued submodular function $\rho : 2^N \to \mathbb{Z}$ in $\mathrm{O}(n^5 \log \Gamma \cdot \mathrm{EO})$ time, where Γ is an upper bound on $|\rho(X)|$ and* EO *denotes the time for function evaluation.*

The time complexity of the IFF algorithm shown above can be derived from the following properties:

Proposition A.21 ([A1]). *The IFF algorithm satisfies the following:*
(i) *The number of calls to* Augment *is $\mathrm{O}(n^2 \log \Gamma)$.*
(ii) *Between calls to* Augment*, there are at most $n-1$ calls to nonsaturating* Double-Exchange.
(iii) *Between calls to* Augment*, there are at most $\mathrm{O}(n^3)$ calls to saturating* Double-Exchange.
(iv) *Each call to* Double-Exchange *requires $\mathrm{O}(\mathrm{EO})$ time.*
(v) *Each call to* Reduce *requires $\mathrm{O}(n^3)$ time.*

Note that the proof of Claim (iii) above is based on the following observation.

Proposition A.22 ([A1]).
(i) Double-Exchange(j, v, w) *makes an active triple (j, v, w) inactive.*
(ii) *Between calls to* Augment*, any inactive triple does not become active.*

In our case, the IFF algorithm is modified as follows. As in the original IFF algorithm, the modified IFF algorithm maintains two kinds of variables

x and φ, where we modify the representation of x to (31). Due to this modification, we do not need the procedure Reduce any more since the number of vectors used in the representation (31) is always the same.

We also modify the procedure Double-Exchange. In the modified IFF algorithm, we call a triple (j, v, w) with $j \in M$ and $v, w \in N$ an *active triple* if it satisfies $v \in W$, $w \in N \setminus W$, and $\varphi(v, w) > 0$, and $\tilde{c}(y_j, w, v) > 0$. Procedure Double-Exchange is applied to an active triple (j, v, w) and it updates y_j and φ as $y_j := y_j - \alpha(\chi_v - \chi_w)$ and $\varphi(v, w) := \varphi(v, w) - \alpha$, where $\alpha = \min\{\varphi(v, w), \tilde{c}(y_j, w, v)\}$. According to the update of y_j and the representation (31), vector x is also updated as $x := x + \alpha(\chi_v - \chi_w)$. In this case, we call Double-Exchange *saturating* if $\alpha = \tilde{c}(y_j, w, v)$ and *nonsaturating* otherwise.

The time complexity of the modified IFF algorithm can be derived from the following properties. Recall that $\Gamma = mnU$ in our case.

Proposition A.23. *The modified IFF algorithm satisfies the following:*
(i) *The number of calls to* Augment *is* $\mathrm{O}(n^2 \log(mnU))$.
(ii) *Between calls to* Augment*, there are at most* $n - 1$ *calls to nonsaturating* Double-Exchange.
(iii) *Between calls to* Augment*, there are at most* mn^2 *calls to saturating* Double-Exchange.
(iv) *Each call to* Double-Exchange *requires* $\mathrm{O}(\log U)$ *time. In particular, nonsaturating* Double-Exchange *can be done in* $\mathrm{O}(1)$ *time.*

Proof. Claims (i) and (ii) can be shown in the same way as in Proposition A.21. Claim (iii) follows from Proposition A.24 below and the fact that the number of triples is at most mn^2. We now prove claim (iv). Procedure Double-Exchange can be done in $\mathrm{O}(\log U)$ time since the exchange capacity $\tilde{c}(y_j, w, v)$ can be computed in $\mathrm{O}(\log U)$ time by Proposition A.17. To reduce the running time in the case of nonsaturating Double-Exchange, we first check if $y_j - \varphi(v, w)(\chi_v - \chi_w) \in D_j(p)$. If $y_j - \varphi(v, w)(\chi_v - \chi_w) \in D_j(p)$ holds, then we have $\tilde{c}(y_j, w, v) \geq \varphi(v, w)$ and therefore this call to Double-Exchange is nonsaturating and we do not compute the value $\tilde{c}(y_j, w, v)$. That is, the value $\tilde{c}(y_j, w, v)$ is computed only when $y_j - \varphi(v, w)(\chi_v - \chi_w) \notin D_j(p)$ holds. Hence, claim (iv) follows. □ □

We then prove the following properties needed in the proof of claim (iii) above.

Proposition A.24. *The modified IFF algorithm satisfies the following:*
(i) Double-Exchange(j, v, w) *makes an active triple* (j, v, w) *inactive.*
(ii) *Between calls to* Augment*, any inactive triple does not become active.*

Proof. The claim (i) follows from the fact that after the call to Double-Exchange(j, v, w) we have either $\varphi(v, w) = 0$ or $\tilde{c}(y_j, v, w) = 0$ (or both). To prove (ii), we need an additional rule concerning the order of triples applied to Double-Exchange: if we apply Double-Exchange to some active triple (j, v, w) for some $j \in M$ and $v \in W$, then we apply Double-Exchange to active triples (j, v, w') for all $w' \in N \setminus W$ sequentially. Then, triples (j, v, w') with $w' \in N \setminus W$ are now all inactive, and it can be shown that these triples (j, v, w') cannot become active by the following calls to Double-Exchange, which can be shown by using the fact that the value $y_j(i)$ does not increase for $i \in W$ and does not decrease for $i \in N \setminus W$. □ □

Hence, we obtain Theorem 1.3 on the time complexity of the algorithm.

A.11 Ascending Auction for Real-Valued Valuation Functions

We show that a variant of the ascending auction finds an ε-approximate equilibrium price vector in the case of real-valued valuation functions with the SGS condition.

Suppose that for $j \in M$, $f_j : [\mathbf{0}, u]_{\mathbb{Z}} \to \mathbb{R}$ is a real-valued valuation function with the SGS condition which is concave-extensible. We define the Lyapunov function $L : \mathbb{R}^n \to \mathbb{R}$ by (3); we here regard L as a function defined on $\mathbb{R}^n$. Then, $p^* \in \mathbb{R}^n$ is an equilibrium price vector if and only if it is a minimizer of L (see [1, 22]). Moreover, M$^\natural$-concavity of f_j implies that the Lyapunov function is a polyhedral L$^\natural$-convex function in the sense of Murota and Shioura [A3], i.e., for every $p, q \in \mathbb{R}^n$ and every $\lambda \in \mathbb{R}_+$, it holds that

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

We then explain how to obtain an ε-approximate equilibrium price vector. Denote $\varepsilon' = \varepsilon/n$, and define function $\hat{L} : \mathbb{R}^n \to \mathbb{R}$ by

$$\hat{L}(p) = L(\varepsilon' p) \qquad (p \in \mathbb{R}^n)$$

and let $\hat{L}_{\mathbb{Z}} : \mathbb{Z}^n \to \mathbb{R}$ be the restriction of $\hat{L}$ on $\mathbb{Z}^n$. Then, $\hat{L}$ is also a polyhedral L$^\natural$-convex function and $\hat{L}_{\mathbb{Z}}$ is an L$^\natural$-convex function on $\mathbb{Z}^n$.

By a proximity theorem of polyhedral L$^\natural$-convex functions in [A3], we can show that for every minimizer q of $\hat{L}_{\mathbb{Z}}$, there exists a minimizer q^* of $\hat{L}$ such that $\|q^* - q\|_\infty < n$. This fact can be rewritten as follows in terms of the original Lyapunov function as follows: for every $q \in \arg\min\{L(\varepsilon p') \mid q' \in \mathbb{Z}^n\}$, there exists a minimizer p^* of L such that $\|p^* - \varepsilon' q\|_\infty < \varepsilon' n = \varepsilon$. This implies that for every $q \in \arg\min\{L(\varepsilon q') \mid q' \in \mathbb{Z}^n\}$, the vector $\varepsilon' q$ is an ε-approximate equilibrium price vector. Since $\min\{L(\varepsilon q') \mid q' \in \mathbb{Z}^n\}$ is essentially equivalent to the minimization of the L$^\natural$-convex function $\hat{L}_{\mathbb{Z}}$, its optimal solution can be obtained by the following variant of the ascending auction, where each component of p is incremented by ε' instead of 1; its validity follows from the previous discussion.

Algorithm ε'-Ascend
Step 0: Set $p := p^\circ$, where $p^\circ \in \varepsilon' \cdot \mathbb{Z}^n$ is a lower bound of some $p^* \in \arg\min L$ (e.g., $p^\circ = \mathbf{0}$).
Step1: Find $X \subseteq N$ that minimizes $L(p + \varepsilon' \chi_X)$.
Step2: If $L(p + \varepsilon' \chi_X) = L(p)$, then output p and stop.
Step3: Set $p := p + \varepsilon' \chi_X$ and go to Step 1.

References

[A1] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. J. ACM 48 (2011) 761–777.

[A2] K. Murota. Recent developments in discrete convex analysis. in: W. J. Cook, et al. (eds.) Research Trends in Combinatorial Optimization, 219–260. Springer, 2009.

[A3] K. Murota and A. Shioura. Extension of M-convexity and L-convexity to polyhedral convex functions. Adv. in Appl. Math. 25 (2000) 352–427.