MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

# Finding a Stable Allocation in Polymatroid Intersection

Satoru IWATA and Yu YOKOI

METR 2017–02 January 2017

# Finding a Stable Allocation in Polymatroid Intersection [*]

Satoru Iwata [†]  Yu Yokoi [†]

January 2017

**Abstract**

The stable matching (or stable marriage) model of Gale and Shapley (1962) has been generalized in various directions such as matroid kernels due to Fleiner (2001) and stable allocations in bipartite networks due to Baïou and Balinski (2002). Unifying these generalizations, we introduce the concept of stable allocations in polymatroid intersection.

Our framework includes both integer- and real-variable versions. The integer-variable version corresponds to a special case of the discrete-concave function model due to Eguchi, Fujishige, and Tamura (2003), who established the existence of a stable allocation by showing that a simple extension of the deferred acceptance algorithm of Gale and Shapley finds a stable allocation in pseudopolynomial time. It has been open to develop a polynomial-time algorithm even for our special case.

In this paper, we present the first strongly polynomial algorithm for finding a stable allocation in polymatroid intersection. To achieve this, we utilize the augmenting path technique for polymatroid intersection. In each iteration, the algorithm searches for an augmenting path by simulating a chain of proposes and rejects in the deferred acceptance algorithm. The running time of our algorithm is $O(n^3\gamma)$, where $n$ and $\gamma$ respectively denote the cardinality of the ground set and the time for computing the saturation and exchange capacities. Moreover, we show that the output of our algorithm is optimal for one side, where this optimality is a generalization of the man-optimality in the stable marriage model.

[*]An earlier version of this paper appeared in Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadelphia, 2016, pp. 1034–1047.

[†]Department of Mathematical Informatics, University of Tokyo, Tokyo 113-8656, Japan. E-mail: {iwata, yu_yokoi}@mist.i.u-tokyo.ac.jp

## 1 Introduction

Since the famous min-max theorem of König [18] in 1931, the bipartite matching has served as a prototype problem in combinatorial optimization. It has been generalized in various directions such as the matroid intersection and transportation problems. In 1970, Edmonds [5] introduced the framework of polymatroid intersection, which unifies these two generalizations as well as various other efficiently solvable combinatorial optimization problems [11, 25].

The primary purpose of this paper is to shed new light from the viewpoint of polymatroids on another fundamental problem on bipartite graphs, i.e., the stable matching (or stable marriage) problem provided by Gale and Shapley [15] in 1962. We introduce the concept of stable allocation in polymatroid intersection and design a strongly polynomial algorithm to find it.

**Problem Description** A pair $(E, f)$ of a finite set $E$ and a set function $f : 2^E \to \mathbf{R}$ is called a polymatroid if it satisfies

- $f(\emptyset) = 0$,
- $A \subseteq B \subseteq E \implies f(A) \leq f(B)$, and
- $\forall A, B \subseteq E : f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$.

We call $f$ the *rank function* of the polymatroid $(E, f)$. Examples include matroid rank functions, coverage functions, and entropy functions [10]. We associate a polytope

$$\mathbf{P}(f) = \{\, x \in \mathbf{R}^E \mid x \geq \mathbf{0},\ \forall A \subseteq E : x(A) \leq f(A) \,\},$$

where $x(A) = \sum_{e \in A} x(e)$ for each $A \subseteq E$ and $x = (x(e) \mid e \in E) \in \mathbf{R}^E$. A vector $x \in \mathbf{R}^E$ is called *independent* if $x \in \mathbf{P}(f)$. For an independent vector $x \in \mathbf{P}(f)$, a subset $A \subseteq E$ is said to be tight if $x(A) = f(A)$. If both $A, B \subseteq E$ are tight at $x \in \mathbf{P}(f)$, then so are $A \cup B$ and $A \cap B$. For $x \in \mathbf{P}(f)$, we denote by $\mathrm{sat}_f(x)$ the unique maximal tight set. Then, we can observe

$$\mathrm{sat}_f(x) = \{\, u \in E \mid \forall \alpha > 0 : x + \alpha\chi_u \notin \mathbf{P}(f) \,\},$$

where $\chi_u$ is the characteristic vector of $u \in E$. For each $u \in \mathrm{sat}_f(x)$, we denote by $\mathrm{dep}_f(x, u)$ the unique minimal tight set that contains $u$. Then we have

$$\mathrm{dep}_f(x, u) = \{\, v \in E \mid \exists \alpha > 0 : x + \alpha(\chi_u - \chi_v) \in \mathbf{P}(f) \,\}.$$

If $u \in E \setminus \mathrm{sat}_f(x)$, then $\mathrm{dep}_f(x, u)$ is defined to be empty. The functions $\mathrm{sat}_f(\cdot)$ and $\mathrm{dep}_f(\cdot, \cdot)$ are called the *saturation function* and the *dependence function*, respectively [11]. If $f$ is the rank function of a matroid and $x$ is the characteristic vector of an independent set $I$, then $\mathrm{sat}_f(x)$ and $\mathrm{dep}_f(x, u)$ correspond to the closure of $I$ and the unique circuit in $I \cup \{u\}$, respectively.

A triple $(E, f, \succ)$ is called a (*totally*) *ordered polymatroid* if $(E, f)$ is a polymatroid and $\succ$ is a total order on $E$, where $a \succ e$ means $a \in E$ is preferred to $e \in E$. We denote $a \succeq e$ to mean $a \succ e$ or $a = e$. For a subset $A \subseteq E$, we denote $A \succeq e$ to mean that every $a \in A$ satisfies $a \succeq e$. Let $(E, h, \succ_{\mathrm{H}})$ and $(E, f, \succ_{\mathrm{F}})$ be two totally ordered polymatroids on the same ground set $E$. The concept of stable allocations is defined as follows.

**Definition 1.1.** A *stable allocation* for a pair of totally ordered polymatroids $(E, h, \succ_{\mathrm{H}})$ and $(E, f, \succ_{\mathrm{F}})$ is a common independent vector $x \in \mathbf{P}(h) \cap \mathbf{P}(f)$ such that, for every $e \in E$, we have $[e \in \mathrm{sat}_h(x),\ \mathrm{dep}_h(x, e) \succeq_{\mathrm{H}} e]$ or $[e \in \mathrm{sat}_f(x),\ \mathrm{dep}_f(x, e) \succeq_{\mathrm{F}} e]$. ■

This model can represent, for example, a labor allocation model as follows. We have two disjoint agent sets $S$ and $T$, which correspond to workers and firms, respectively. Let $E$ be the set of worker-firm pairs, i.e., $E = S \times T$, and define its subsets $E_s = \{\, (s, t) \mid t \in T \,\}$

for each $s \in S$ and $E_t = \{ (s,t) \mid s \in S \}$ for each $t \in T$. A labor allocation is a vector $x = (x(s,t) \mid (s,t) \in E) \in \mathbf{R}^E$, where $x(s,t)$ means the amount of contracted labor time of $s$ at $t$. For an allocation $x$, we write $x_s = x|_{E_s} = (x(s,t) \mid t \in T)$ for each $s \in S$ and $x_t = x|_{E_t}$ for each $t \in T$.

The profile of each $s \in S$ is given as an ordered polymatroid $(E_s, h_s, \succ_s)$. The worker $s$ wishes to have as large allocation as possible in $\mathbf{P}(h_s) \subseteq \mathbf{R}^{E_s}$ with the priority given by $\succ_s$. Note that $\{E_s\}_{s\in S}$ is a partition of $E$. Define $(E, h, \succ_{\mathrm{H}})$ as the direct sum of $\{(E_s, h_s, \succ_s)\}_{s\in S}$, i.e., define $h : 2^E \to \mathbf{R}$ by $h(A) = \sum_{s\in S} h_s(A \cap E_s)$ $(A \subseteq E)$, and let $\succ_{\mathrm{H}}$ be an arbitrary total order on $E$ which is consistent with $\{\succ_s\}_{s\in S}$. Then, $(E, h, \succ_{\mathrm{H}})$ forms an ordered polymatroid. Similarly, profiles of firms are given as ordered polymatroids $\{(E_t, f_t, \succ_t)\}_{t\in T}$, and an ordered polymatroid $(E, f, \succ_{\mathrm{F}})$ is defined on the same ground set $E$.

An allocation $x \in \mathbf{R}^E$ must be feasible for every agent, i.e., $x_s \in \mathbf{P}(h_s)$ $(\forall s \in S)$ and $x_t \in \mathbf{P}(f_t)$ $(\forall t \in T)$. This means $x \in \mathbf{P}(h) \cap \mathbf{P}(f)$. In addition, $x$ should be stable in the following sense. If $(s,t) \notin \mathrm{sat}_{h_s}(x_s)$ or $(s,t) \succ_s (s,t')$ for some $(s,t') \in \mathrm{dep}_{h_s}(x_s,(s,t))$, then $s$ has incentive to increase $x(s,t)$, possibly at the expense of $x(s,t')$. Similarly, if $(s,t) \notin \mathrm{sat}_{f_t}(x_t)$ or $(s,t) \succ_s (s',t)$ for some $(s',t) \in \mathrm{dep}_{f_t}(x_t,(s,t))$, then $t$ has incentive to increase $x(s,t)$, possibly at the expense of $x(s',t)$. If both $s$ and $t$ have incentives to increase $x(s,t)$, there is no direct way to prevent them from doing so. Definition 1.1 requires $x$ not to admit such a pair $(s,t) \in E$.

Note that the existence of a stable allocation appears to be unclear from the definition. By using the general framework of Alkan and Gale [1], however, it is shown in [27] that a stable allocation does exist in any setting of polymatroid intersection. This proof of existence does not tell how to find such a solution efficiently.

**Our Contribution** In this paper, we present the first strongly polynomial algorithm for finding a stable allocation in the general setting of polymatroid intersection. The algorithm combines the augmenting path technique for polymatroid intersection [9, 23] and the deferred acceptance procedure for stable matching [15]. The correctness of our algorithm provides an alternative proof for the existence of a stable allocation. If $h$ and $f$ are integer-valued functions, then the algorithm finds an integral stable allocation.

The running time of this algorithm is $O(|E|^3\gamma)$, where $\gamma$ denotes the time for computing the saturation and exchange capacities on the given polymatroids. Assuming an oracle for evaluating the rank function, these capacities can be computed in strongly polynomial time via submodular function minimization [16, 17, 19, 22, 24]. Most concrete examples of polymatroids, however, admit more direct ways to design such procedures.

Furthermore, we show that our algorithm finds the $\succ_{\mathrm{H}}$-*optimal* stable allocation. Similarly to the stable matching model of Gale and Shapley [15], an instance of our model has multiple stable allocations in general. We prove that the output of our algorithm is $\succ_{\mathrm{H}}$-optimal, i.e., it is $\succ_{\mathrm{H}}$-*preferable* to any other stable allocation. Here, we say that $x \in \mathbf{R}^E$ is $\succ_{\mathrm{H}}$-preferable to $y \in \mathbf{R}^E$ if we have $\sum\{ x(e) \mid e \succeq_{\mathrm{H}} a \} \geq \sum\{ y(e) \mid e \succeq_{\mathrm{H}} a \}$ for every $a \in E$. For an instance of the stable marriage problem, the $\succ_{\mathrm{H}}$-optimality coincides with the well-known notion of "man-optimality" if $\succ_{\mathrm{H}}$ is consistent with preferences of all men.

**Related Stable Matching Models** Our framework includes two major generalizations of the stable matching model: stable allocations in bipartite networks due to Baïou and Balinski [2] and matroid kernels in matroid intersection due to Fleiner [7, 8]. On the other hand, it can be regarded as a special case of the discrete-concave function model of Eguchi, Fujishige, and Tamura [6]. We now describe these three related works.

The stable allocation model of Baïou and Balinski [2] is a special case of the above labor allocation model, which is representable by network flow as follows. Let $G = (V, E)$ be a bipartite graph with edge set $E$ and vertex set $V$ partitioned into $S$ and $T$. Nonnegative capacity functions $b : V \to \mathbf{R}_+$ and $c : E \to \mathbf{R}_+$ are associated with $G$. In addition, each $v \in S \cup T$ has a preference $\succ_v$ on $\delta v$, where $\delta v$ denotes the set of edges incident to $v$. A flow $x : E \to \mathbf{R}_+$ is said to be feasible if $x(e) \leq c(e)$ holds for every $e \in E$ and $\sum_{e \in \delta v} x(e) \leq b(v)$ holds for every $v \in S \cup T$. Define set functions $h$ and $f$ on $E$ by

$$\begin{aligned} h(A) &= \textstyle\sum_{s \in S} \min \left\{ b(s), \sum_{e \in A \cap \delta s} c(e) \right\} \quad (A \subseteq E), \\ f(A) &= \textstyle\sum_{t \in T} \min \left\{ b(t), \sum_{e \in A \cap \delta t} c(e) \right\} \quad (A \subseteq E), \end{aligned}$$

respectively. The set of the feasible flows then coincides with the intersection $\mathbf{P}(h) \cap \mathbf{P}(f)$. Let $\succ_{\mathrm{H}}$ and $\succ_{\mathrm{F}}$ be arbitrary total orders on $E$ consistent with $\{\succ_s\}_{s \in S}$ and $\{\succ_t\}_{t \in T}$, respectively. Stable allocations in the bipartite network $G$ are exactly the same as stable allocations for the pair of totally ordered polymatroids $(E, h, \succ_{\mathrm{H}})$ and $(E, f, \succ_{\mathrm{F}})$.

Baïou and Balinski [2] presented two algorithms for finding a stable allocation in $G$. The first is a natural extension of the deferred acceptance algorithm of Gale and Shapley [15] and runs in pseudopolynomial time if all the capacities are integers. However, for instances with real capacities, this algorithm needs exponential time in the worst-case, as shown by Dean, Goemans, and Immorlica [3]. In contrast, the second algorithm finds a stable allocation in $O(|V| \cdot |E|)$ time using augmenting paths. Dean and Munshi [4] improved the latter algorithm to run in $O(|E| \log |V|)$ time.

We now describe the matroid kernel model of Fleiner [7, 8]. Consider a pair of matroids $(E, \mathcal{I}_{\mathrm{H}})$ and $(E, \mathcal{I}_{\mathrm{F}})$ on the same ground set $E$ with independent set families $\mathcal{I}_{\mathrm{H}}$ and $\mathcal{I}_{\mathrm{F}}$. Assume that total orders $\succ_{\mathrm{H}}$ and $\succ_{\mathrm{F}}$ on $E$ are attached to them. Let $\rho_{\mathrm{H}}$ and $\rho_{\mathrm{F}}$ be the rank functions of $(E, \mathcal{I}_{\mathrm{H}})$ and $(E, \mathcal{I}_{\mathrm{F}})$, respectively. Then, the common independent sets of $(E, \mathcal{I}_{\mathrm{H}})$ and $(E, \mathcal{I}_{\mathrm{F}})$ correspond to the integer vectors in $\mathbf{P}(\rho_{\mathrm{H}}) \cap \mathbf{P}(\rho_{\mathrm{F}})$. A matroid kernel introduced by Fleiner [7, 8] coincides with a subset $K \subseteq E$ whose characteristic vector is a stable allocation for $(E, \rho_{\mathrm{H}}, \succ_{\mathrm{H}})$ and $(E, \rho_{\mathrm{F}}, \succ_{\mathrm{F}})$. Fleiner [7, 8] presented a strongly polynomial algorithm for finding a matroid kernel, extending the deferred acceptance algorithm of Gale and Shapley [15].

The discrete-concave function model, investigated in [6, 12, 13, 21], is a two-sided matching model in which preferences of agents are represented by value functions on integer vectors. This framework was originated by Eguchi, Fujishige, and Tamura [6]. In their model, each side has a value function whose effective domain is the set of integer points of a polymatroid and function values satisfy M$^\natural$-concavity, a kind of concavity for discrete functions [20]. Our framework can be regarded as a special form of this model in which each function is linear in its effective domain as follows. For an ordered polymatroid $(E, h, \succ_{\mathrm{H}})$ with an integer-valued rank function, take an arbitrary positive weight $w_{\mathrm{H}} : E \to \mathbf{R}_+$ such that $a \succ_{\mathrm{H}} e$ implies $w_{\mathrm{H}}(a) > w_{\mathrm{H}}(e)$ for any $a, e \in E$. Define a value function $U_{\mathrm{H}} : \mathbf{Z}^E \to \mathbf{R} \cup \{-\infty\}$ by

$$U_{\mathrm{H}}(x) = \begin{cases} \langle w_{\mathrm{H}}, x \rangle & \text{if } x \in \mathbf{Z}^E \cap \mathbf{P}(h), \\ -\infty & \text{if } x \in \mathbf{Z}^E \setminus \mathbf{P}(h), \end{cases}$$

where $\langle w_{\mathrm{H}}, x \rangle := \sum_{e \in E} w_{\mathrm{H}}(e) x(e)$. Similarly, define a value function $U_{\mathrm{F}} : \mathbf{Z}^E \to \mathbf{R} \cup \{-\infty\}$ from an ordered polymatroid $(E, f, \succ_{\mathrm{F}})$. Then, an integer allocation $x \in \mathbf{Z}^E$ is stable for $(E, h, \succ_{\mathrm{H}})$ and $(E, f, \succ_{\mathrm{F}})$ if and only if it is stable with respect to $U_{\mathrm{H}}$ and $U_{\mathrm{F}}$ in the sense of the discrete-concave function model.

Eguchi et al. [6] established the existence of a stable allocation by showing that a simple extension of the deferred acceptance algorithm of Gale and Shapley finds a stable allocation in pseudopolynomial time. They asked if one can develop a weakly polynomial algorithm, which has remained open for more than a decade. When applied to our framework, their algorithm still requires pseudopolynomial time. Indeed, there is a series of examples in which the algorithm requires the numbers of iterations proportional to the width of the effective domain.

**Related Polymatroid Algorithms** We make comparison of our algorithm with previously known algorithmic results on polymatroids. The concept of polymatroids was introduced by Edmonds [5] as a polyhedral generalization of matroids. He showed that the linear optimization over $\mathbf{P}(f)$ is solved by the greedy algorithm. Let $w : E \to \mathbf{R}_+$ be an arbitrary weight function that takes distinct positive values, and consider a total order $\succ_w$ on $E$ such that $a \succ_w e$ means $w(a) > w(e)$. The correctness of the greedy algorithm implies that $x \in \mathbf{P}(f)$ maximizes $\langle w, x\rangle$ if and only if $e \in \mathrm{sat}_f(x)$ and $\mathrm{dep}_f(x, e) \succeq_w e$ hold for every $e \in E$. This condition means that $x$ is a stable allocation for $(E, f, \succ_w)$ and $(E, f, \succ_w)$.

Given a nonnegative vector $z \in \mathbf{R}^E$, one can think of maximizing $\langle w, x\rangle$ subject to $x \in \mathbf{P}(f)$ and $x \leq z$. A feasible solution $x$ is optimal if and only if $e \in \mathrm{sat}_f(x)$ and $\mathrm{dep}_f(x, e) \succeq_w e$ hold for every $e \in E$ with $x(e) < z(e)$. This optimality condition requires $x$ to be a stable allocation in the polymatroid intersection of $(E, f, \succ_w)$ and $(E, z, \succ_w)$, where $z$ is regarded as a rank function defined by $z(A) = \sum_{e\in A} z(e)$. The feasible region of this problem is in fact identical to the associated polytope $\mathbf{P}(f^z)$ with a polymatroid $(E, f^z)$ defined by $f^z(A) := \min\{f(Y) + z(A \setminus Y) \mid Y \subseteq A\}$. A standard method for solving this optimization problem is to apply the polymatroid greedy algorithm to $\mathbf{P}(f^z)$. Our stable allocation algorithm when applied to this setting coincides with this standard method, which runs in $O(|E|\gamma)$ time, where $\gamma$ denotes the time for computing the saturation and exchange capacities on the given polymatroids.

Edmonds [5] also showed a min-max theorem on the polymatroid intersection problem, i.e., the problem of maximizing $x(E)$ among common independent vectors of two polymatroids on $E$. Schönsleben [23] invented a technique of using lexicographic shortest augmenting paths to develop an algorithm for solving the polymatroid intersection problem in $O(|E|^5\gamma)$ time. Tardos, Tovey, and Trick [26] improved this algorithm to run in $O(|E|^4\gamma)$ time. Currently the best known running time bound for the polymatroid intersection problem is $O(|E|^3\gamma)$ due to Fujishige and Zhang [14].

The weighted version of this problem, i.e., the weighted polymatroid intersection problem asks for a common independent vector of two polymatroids maximizing a linear function. This problem is also solvable in strongly polynomial time, but the current best running time bound is as high as $O(|E|^6\gamma \log |E|)$. As a special case with extremely distinct weights, we can think of maximizing the components lexicographically with respect to a specified total order on $E$. An optimal solution of this problem coincides with a stable allocation for the same total order in both polymatroids. Our stable allocation algorithm when applied to this setting determines each component greedily in the specified total order, which requires only $O(|E|\gamma)$ time.

**Organization** The rest of this paper is organized as follows. Section 2 provides preliminaries on polymatroids by recapitulating properties of saturation and exchange capacities. In Section 3, we describe our algorithm and demonstrate it on a small example. Section 4 provides invariants which are maintained in the algorithm and play a key role in our analysis. The correctness and complexity of the algorithm are shown in Sections 5 and 6, respectively. Finally, we show the $\succ_{\mathrm{H}}$-optimality of the output in Section 7.

## 2 Saturation and Exchange Capacities

In this section, we describe fundamental properties of saturation and exchange capacities. For each $u \in E$, we denote by $\chi_u$ its characteristic vector, i.e., $\chi_u(u) = 1$ and $\chi_u(e) = 0$ for $e \in E \setminus \{u\}$.

Let $(E, f)$ be a polymatroid. For an independent vector $x \in \mathbf{P}(f)$ and an element $u \in E$, define the *saturation capacity* $\hat{c}_f(x, u)$ by

$$\hat{c}_f(x,u) = \max\{\, \alpha \in \mathbf{R} \mid x + \alpha\chi_u \in \mathbf{P}(f)\,\}\,.$$

For any $\alpha$ with $0 \le \alpha \le \hat{c}_f(x,u)$, we have $x + \alpha\chi_u \in \mathbf{P}(f)$. The saturation capacity can also be expressed as

$$\hat{c}_f(x,u) = \min\{\, f(A) - x(A) \mid u \in A \subseteq E\,\}\,. \tag{1}$$

For $x \in \mathbf{P}(f)$ and $u, v \in E$, define the *polymatroid exchange capacity* $\bar{c}_f(x,u,v)$ by

$$\bar{c}_f(x,u,v) = \max\{\, \alpha \in \mathbf{R} \mid x + \alpha(\chi_u - \chi_v) \in \mathbf{P}(f)\,\}\,,$$

where we let $\bar{c}_f(x,u,u) = \infty > 0$ for every $u \in E$. For any $\alpha$ with $0 \le \alpha \le \bar{c}_f(x,u,v)$, we have $x + \alpha(\chi_u - \chi_v) \in \mathbf{P}(f)$. For distinct $u, v \in E$, it is known that $\bar{c}_f(x,u,v)$ is also written as

$$\bar{c}_f(x,u,v) = \min\{\tilde{c}_f(x,u,v), x(v)\}$$

where $\tilde{c}_f(x,u,v)$ is defined for $x \in \mathbf{P}(f)$ and distinct $u, v \in E$ by

$$\tilde{c}_f(x,u,v) = \min\{\, f(A) - x(A) \mid u \in A \subseteq E,\ v \notin A\,\}\,. \tag{2}$$

It can be easily shown that $\bar{c}_f(x,u,v) = \tilde{c}_f(x,u,v)$ holds if $u \in \mathrm{sat}_f(x)$.

The saturation and dependence functions are now expressed as

$$\begin{aligned} \mathrm{sat}_f(x) &= \{\, u \in E \mid \hat{c}_f(x,u) = 0\,\} \qquad (x \in \mathbf{P}(f)),\\ \mathrm{dep}_f(x,u) &= \{\, v \in E \mid \bar{c}_f(x,u,v) > 0\,\} \quad (x \in \mathbf{P}(f),\ u \in \mathrm{sat}_f(x)),\end{aligned}$$

where $\mathrm{dep}_f(x,u)$ is defined to be empty for $u \in E \setminus \mathrm{sat}_f(x)$.

Note that, for $x \in \mathbf{P}(f)$ and $u, v \in \mathrm{sat}_f(x)$, the condition $v \in \mathrm{dep}_f(x,u)$ implies $x(v) > 0$. Also, observe that

$$\bar{c}_f(x,u,v) > 0 \iff x(v) > 0 \text{ and } [u \notin \mathrm{sat}_f(x) \text{ or } v \in \mathrm{dep}_f(x,u)]$$

holds for any $x \in \mathbf{P}(f)$ and $u, v \in E$. This observation implies the following lemma.

**Lemma 2.1 (Transitivity of Dependence).** For $x \in \mathbf{P}(f)$ and $s, t, u \in E$, if $\bar{c}_f(x,s,t) > 0$ and $\bar{c}_f(x,t,u) > 0$, then $\bar{c}_f(x,s,u) > 0$.

*Proof.* If two or three of $s, t, u$ are the same element, the claim is obvious. We assume that they are all distinct.

Since $\bar{c}_f(x,t,u) > 0$ implies $x(u) > 0$, it suffices to show $u \in \mathrm{dep}_f(x,s)$ assuming $s \in \mathrm{sat}_f(x)$. Note that $s \in \mathrm{sat}_f(x)$ and $\bar{c}_f(x,s,t) > 0$ imply $t \in \mathrm{dep}_f(x,s) \subseteq \mathrm{sat}_f(x)$. Suppose, to the contrary, that we have $u \notin \mathrm{dep}_f(s)$. Then, there is $A \subseteq E$ such that $x(A) = f(A)$, $s \in A$, and $u \notin A$. If $t \notin A$, then $A$ satisfies $s \in A \not\ni t$, which implies $\bar{c}_f(x,s,t) = 0$ by $s \in \mathrm{sat}_f(x)$, a contradiction. If $t \in A$, then $A$ satisfies $t \in A \not\ni u$ which implies $\bar{c}_f(x,t,u) = 0$ by $t \in \mathrm{sat}_f(x)$, a contradiction. □

## 2.1 Moving in Polymatroids

Here we show how saturation and exchange capacities change when we move an independent vector $x$ in the polyhedron $\mathbf{P}(f)$.

**Lemma 2.2 (Single Exchange).** For any $x \in \mathbf{P}(f)$ and distinct $u, v \in E$, let $y \in \mathbf{R}^E$ be the vector defined by $y := x + \alpha(\chi_u - \chi_v)$ with $0 \le \alpha \le \bar{c}_f(x,u,v)$. Then, we have $y \in \mathbf{P}(f)$ and the following (a1)–(a5).

**(a1)** $\bar{c}_f(y,u,v) = \bar{c}_f(x,u,v) - \alpha$.

**(a2)** (i) For $s \in E \setminus \{v\}$ with $\bar{c}_f(x,s,v) = 0$, every $t \in E \setminus \{s\}$ satisfies $\bar{c}_f(y,s,t) = \bar{c}_f(x,s,t)$.
(ii) For $t \in E \setminus \{u\}$ with $\bar{c}_f(x,u,t) = 0$, every $s \in E \setminus \{t\}$ satisfies $\bar{c}_f(y,s,t) = \bar{c}_f(x,s,t)$.

**(a3)** For any $s,t \in E \setminus \{u,v\}$, we have
$\bar{c}_f(y,s,v) \geq \min\{\bar{c}_f(x,s,v), \bar{c}_f(y,u,v)\}$ and $\bar{c}_f(y,u,t) \geq \min\{\bar{c}_f(x,u,t), \bar{c}_f(y,u,v)\}$.

**(a4)** If $u \in \mathrm{sat}_f(x)$, then $\mathrm{sat}_f(x) = \mathrm{sat}_f(y)$ and $\hat{c}_f(y,s) = \hat{c}_f(x,s)$ for every $s \in E \setminus \mathrm{sat}_f(x)$.

**(a5)** If $u \notin \mathrm{sat}_f(x)$, then $\hat{c}_f(y,u) \geq \min\{\hat{c}_f(x,u), \bar{c}_f(y,u,v)\}$ and $s \in \mathrm{sat}_f(y)$ for any $s \in \mathrm{sat}_f(x)$ with $\bar{c}_f(x,s,v) = 0$.

*Proof.* It suffices to consider the case that $\bar{c}_f(x,u,v) > 0$, which means that we can assume $x(v) > 0$ and $[u \notin \mathrm{sat}_f(x)$ or $v \in \mathrm{dep}_f(x,u)]$.

**(a1):** Every $A \subseteq E$ with $u \in A \not\ni v$ satisfies $y(A) = x(A) + \alpha$, and hence $f(A) - y(A) = f(A) - x(A) - \alpha$. Then, by (2), $\tilde{c}_f(y,u,v) = \tilde{c}_f(x,u,v) - \alpha$. Also, clearly $y(v) = x(v) - \alpha$. Then, $\bar{c}_f(y,u,v) = \min\{\tilde{c}_f(y,u,v), y(v)\} = \min\{\tilde{c}_f(x,u,v) - \alpha, x(v) - \alpha\} = \bar{c}_f(x,u,v) - \alpha$.

**(a2)-(i):** Since $x(v) > 0$, the condition $\bar{c}_f(x,s,v) = 0$ implies $s \in \mathrm{sat}_f(x)$ and $v \notin \mathrm{dep}_f(x,s)$. By Lemma 2.1, $\bar{c}_f(x,u,v) > 0$ and $\bar{c}_f(x,s,v) = 0$ lead to $\bar{c}_f(x,s,u) = 0$, which implies $u \notin \mathrm{dep}_f(x,s)$. Thus, $C := \mathrm{dep}_f(x,s)$ satisfies $u,v \notin C$, $s \in C$, and $x(C) = y(C) = f(C)$. Take any $t \in E \setminus \{s\}$ and recall (2). Note that now $\bar{c}_f(x,s,t) = \tilde{c}_f(x,s,t)$ because $s \in \mathrm{sat}_f(x)$. For every $A \subseteq E$ with $s \in A$, $t \notin A$, by $x(C) = f(C)$ and $x \in \mathbf{P}(f)$,

$$\begin{aligned} f(A) - x(A) &= f(A) + f(C) - x(C) - x(A) \\ &\geq f(A \cup C) + f(A \cap C) - x(A \cup C) - x(A \cap C) \\ &\geq f(A \cap C) - x(A \cap C). \end{aligned}$$

Also, $A \cap C$ satisfies $s \in A \cap C \not\ni t$. Hence $\bar{c}_f(x,s,t) = \min\{\, f(A) - x(A) \mid A \subseteq C,\ s \in A \not\ni t\,\}$. Similarly, since $y(C) = f(C)$ and $y \in \mathbf{P}(f)$, the above inequality also holds with $x$ replaced by $y$, and hence $\bar{c}_f(y,s,t) = \min\{\, f(A) - y(A) \mid A \subseteq C,\ s \in A \not\ni t\,\}$. Because $A \subseteq C \subseteq E \setminus \{u,v\}$ implies $x(A) = y(A)$, the claim follows.

**(a2)-(ii):** By $\bar{c}_f(x,u,v) > 0$ and $\bar{c}_f(x,u,t) = 0$, $t \in E \setminus \{u\}$ satisfies $t \neq v$, and hence $y(t) = x(t)$. Then, it suffices to show $\tilde{c}_f(y,s,t) = \tilde{c}_f(x,s,t)$. Assume $x(t) > 0$, since otherwise the claim is obvious. Then, $\bar{c}_f(x,u,t) = 0$ implies $u \in \mathrm{sat}_f(x)$ and $t \notin \mathrm{dep}_f(x,u)$. Also, then $\bar{c}_f(x,u,v) > 0$ implies $v \in \mathrm{dep}_f(x,u)$. Thus, $C := \mathrm{dep}_f(x,u)$ satisfies $u,v \in C$, $t \notin C$, and $x(C) = y(C) = f(C)$. Take any $s \in E \setminus \{t\}$ and recall (2). For every $A \subseteq E$ with $s \in A$, $t \notin A$, by $x(C) = f(C)$ and $x \in \mathbf{P}(f)$,

$$\begin{aligned} f(A) - x(A) &= f(A) + f(C) - x(C) - x(A) \\ &\geq f(A \cup C) + f(A \cap C) - x(A \cup C) - x(A \cap C) \\ &\geq f(A \cup C) - x(A \cup C). \end{aligned}$$

Also, $s \in A \cup C \not\ni t$. Hence $\tilde{c}_f(x,s,t) = \min\{\, f(A) - x(A) \mid A \supseteq C,\ s \in A \not\ni t\,\}$. Similarly, by $y(C) = f(C)$ and $y \in \mathbf{P}(f)$, we have $\tilde{c}_f(y,s,t) = \min\{\, f(A) - y(A) \mid A \supseteq C,\ s \in A \not\ni t\,\}$. Because $u,v \in C \subseteq A$ implies $x(A) = y(A)$, the claim follows.

**(a3):** For the first inequality, it suffices to show that $\bar{c}_f(y,s,v) < \bar{c}_f(y,u,v)$ implies $\bar{c}_f(y,s,v) \geq \bar{c}_f(x,s,v)$. By (2), $\bar{c}_f(y,s,v) < \bar{c}_f(y,u,v)$ means

$$\min\{\, f(A) - y(A) \mid A \subseteq E,\ s \in A \not\ni v\,\} < \min\{\, f(A) - y(A) \mid A \subseteq E,\ u \in A \not\ni v\,\}.$$

Then, a minimizer of the left-hand side, say $A^*$, satisfies $s \in A^* \not\ni u,v$. This implies $\bar{c}_f(y,s,v) = f(A^*) - y(A^*) = f(A^*) - x(A^*) \geq \min\{\, f(A) - x(A) \mid A \subseteq E,\ s \in A \not\ni v\,\} = \bar{c}_f(x,s,v)$.

To show the second inequality similarly, assume $\bar{c}_f(y, u, t) < \bar{c}_f(y, u, v)$. Then

$$\min\{\, f(A) - y(A) \mid A \subseteq E,\ u \in A \not\ni t \,\} < \min\{\, f(A) - y(A) \mid A \subseteq E,\ u \in A \not\ni v \,\},$$

and a minimizer $A^*$ of the left-hand side satisfies $u, v \in A^*$ and $t \notin A^*$. This implies $\bar{c}_f(y, u, t) = f(A^*) - y(A^*) = f(A^*) - x(A^*) \geq \min\{\, f(A) - x(A) \mid A \subseteq E,\ u \in A \not\ni t \,\} = \bar{c}_f(x, u, t)$.

**(a4):** As $u \in \mathrm{sat}_f(x)$, the subset $C := \mathrm{sat}_f(x)$ satisfies $u, v \in \mathrm{dep}_f(x, u) \subseteq C$ by $\bar{c}_f(x, u, v) > 0$. Then, $y(C') = x(C')$ for every $C' \supseteq C$, and hence $\mathrm{sat}_f(y) = C = \mathrm{sat}_f(x)$.

Take any $s \in E \backslash \mathrm{sat}_f(x)$ and recall (1). Since $x(C) = f(C)$, the inequality in the proof of (a2) holds for any $A \subseteq E$ with $s \in A$. Hence, we have $\hat{c}_f(x, s) = \min\{\, f(A) - x(A) \mid A \supseteq C, s \in A \,\}$. Similarly, $\hat{c}_f(y, s) = \min\{\, f(A) - y(A) \mid A \supseteq C, s \in A \,\}$. Because $u, v \in C \subseteq A$ implies $x(A) = y(A)$, the claim follows.

**(a5):** When $u \notin \mathrm{sat}_f(x)$, every $s \in \mathrm{sat}_f(x)$ satisfies $u \notin \mathrm{dep}_f(x, s)$. Also, $\bar{c}_f(x, s, v) = 0$ implies $v \notin \mathrm{dep}_f(x, s)$, and hence $y(\mathrm{dep}_f(x, s)) = x(\mathrm{dep}_f(x, s)) = f(\mathrm{dep}_f(x, s))$. Thus, we have $\{\, s \in \mathrm{sat}_f(x) \mid \bar{c}_f(x, s, v) = 0 \,\} \subseteq \mathrm{sat}_f(y)$.

For the inequality, it suffices to show that $\hat{c}_f(y, u) < \bar{c}_f(y, u, v)$ implies $\hat{c}_f(y, u) \geq \hat{c}_f(x, u)$. By (1), $\hat{c}_f(y, u) < \bar{c}_f(y, u, v)$ means

$$\min\{\, f(A) - y(A) \mid A \subseteq E,\ u \in A \,\} < \min\{\, f(A) - y(A) \mid A \subseteq E,\ u \in A \not\ni v \,\}.$$

Then, a minimizer $A^* \subseteq E$ of the left-hand satisfies $u, v \in A^*$. This implies $\hat{c}_f(y, u) = f(A^*) - y(A^*) = f(A^*) - x(A^*) \geq \min\{\, f(A) - x(A) \mid A \subseteq E,\ u \in A \,\} = \hat{c}_f(x, u)$. □

For any positive integer $d \in \mathbf{Z}_{>0}$, we write $[d] := \{1, 2, \ldots, d\}$. For two nonnegative integers $k, l \in \mathbf{Z}_+$ with $k \leq l$, we write the integer interval by $[k, l] := \{k, k+1, k+2, \ldots, l\}$.

**Lemma 2.3 (Multiple Exchange).** For $x \in \mathbf{P}(f)$, let $u_i, v_i$ $(i = 1, 2, \ldots, d)$ be $2d$ distinct elements of $E$ with

$$\begin{aligned} &\bar{c}_f(x, u_i, v_i) > 0 \quad (i \in [d]), \\ &\bar{c}_f(x, u_i, v_j) = 0 \quad (i, j \in [d] \text{ with } i < j). \end{aligned}$$

For any $\alpha \geq 0$ satisfying $\alpha \leq \bar{c}_f(x, u_i, v_i)$ for every $i \in [d]$, define $y \in \mathbf{R}^E$ by

$$y := x + \alpha \textstyle\sum_{i=1}^{d} (\chi_{u_i} - \chi_{v_i}).$$

Then, we have $y \in \mathbf{P}(f)$ and the following (b1)–(b6).

**(b1)** $\bar{c}_f(y, u_i, v_i) = \bar{c}_f(x, u_i, v_i) - \alpha$ for every $i \in [d]$.

**(b2)** For any $s, t \in E$, if we have $\bar{c}_f(x, s, t) \neq \bar{c}_f(y, s, t)$, then there is a pair of indices $(i, j) \in [d]^2$ such that $i \leq j$, $\bar{c}_f(x, s, v_j) > 0$, and $\bar{c}_f(x, u_i, t) > 0$.

**(b3)** For every $s, t \in E$ and $i \in [d]$, we have the following.
If $\bar{c}_f(x, s, v_j) = 0$ $(\forall j > i)$, then $\bar{c}_f(y, s, v_i) \geq \min\{\bar{c}_f(x, s, v_i), \bar{c}_f(y, u_i, v_i)\}$.
If $\bar{c}_f(x, u_j, t) = 0$ $(\forall j < i)$, then $\bar{c}_f(y, u_i, t) \geq \min\{\bar{c}_f(x, u_i, t), \bar{c}_f(y, u_i, v_i)\}$.

**(b4)** $\{u_1, u_2, \ldots, u_{d-1}\} \subseteq \mathrm{sat}_f(x)$.

**(b5)** If $u_d \in \mathrm{sat}_f(x)$, then $\mathrm{sat}_f(x) = \mathrm{sat}_f(y)$ and $\hat{c}_f(y, s) = \hat{c}_f(x, s)$ for every $s \in E \setminus \mathrm{sat}_f(x)$.

**(b6)** If $u_d \notin \mathrm{sat}_f(x)$, then $\hat{c}_f(y, u_d) \geq \min\{\hat{c}_f(x, u_d), \bar{c}_f(y, u_d, v_d)\}$ and $s \in \mathrm{sat}_f(y)$ for any $s \in \mathrm{sat}_f(x)$ with $\bar{c}_f(x, s, v_d) = 0$.

*Proof.* Define $y_0 := x$ and $y_l := y_{l-1} + \alpha(\chi_{u_l} - \chi_{v_l})$ for $l = 1, 2, \ldots, d$. Note that $y_d = y$. We first show that for any $l = 1, 2, \ldots, d$, a vector $y_l$ satisfies the following conditions:

$$\bar{c}_f(y_l, u_i, v_i) = \bar{c}_f(x, u_i, v_i) - \alpha \qquad (i \in [l]), \tag{3}$$
$$\bar{c}_f(y_l, u_i, v_i) = \bar{c}_f(x, u_i, v_i) \qquad (i \in [l+1, d]), \tag{4}$$
$$\bar{c}_f(y_l, u_i, v_j) = \bar{c}_f(x, u_i, v_j) = 0 \qquad (i, j \in [d] \text{ with } i < j), \tag{5}$$
$$\bar{c}_f(y_l, s, v_i) = \bar{c}_f(x, s, v_i) \qquad (i \in [l+1, d],\ s \in E\backslash\{v_i\}), \tag{6}$$
$$\bar{c}_f(y_l, u_i, t) = \bar{c}_f(y_{l-1}, u_i, t) \qquad (i \in [l-1],\ t \in E\backslash\{u_i\}). \tag{7}$$

We use induction on $l$. Assume (3)–(6) hold for $l - 1$; we will prove them for $l$.

Since $y_l = y_{l-1} + \alpha(\chi_{u_l} - \chi_{v_l})$, we have $\bar{c}_f(y_l, u_l, v_l) = \bar{c}_f(y_{l-1}, u_l, v_l) - \alpha = \bar{c}_f(x, u_l, v_l) - \alpha$ by Lemma 2.2 (a1) and (4) for $y_{l-1}$. Hence, (3) for $y_l$ holds.

Note that $y_{l-1}$ satisfies $\bar{c}_f(y_{l-1}, u_i, v_l) = 0$ for $i < l$ and $\bar{c}_f(y_{l-1}, u_l, v_j) = 0$ for $j > l$ by (5). Then, Lemma 2.2 (a2) imply that we have $\bar{c}_f(y_l, u_i, v_j) \neq \bar{c}_f(y_{l-1}, u_i, v_j)$ only when $j \le l \le i$. Thus, (4) and (5) hold for $y_l$.

By (5) for $y_{l-1}$, every $i \in [d]$ with $i > l$ satisfies $\bar{c}_f(y_{l-1}, u_l, v_i) = 0$. By Lemma 2.2 (a2)-(ii), then $\bar{c}_f(y_l, s, v_i) = \bar{c}_f(y_{l-1}, s, v_i)$ for every $s \in E \setminus \{v_i\}$. Note that $i > l$ implies $i \in [l' + 1, d]$ for every $l' \ge l$. Hence (6) holds by induction.

By (5) for $y_{l-1}$, every $i \in [d]$ with $i < l$ satisfies $\bar{c}_f(y_{l-1}, u_i, v_l) = 0$. By Lemma 2.2 (a2)-(i), then $\bar{c}_f(y_l, u_i, t) = \bar{c}_f(y_{l-1}, u_i, t)$ for every $t \in E \setminus \{u_i\}$, and hence (7) holds.

**(b1):** This is already shown by (3) for $y_d = y$.

**(b2):** For $l = 1, 2, \ldots, d$, we show the following claim: For $s, t \in E$ with $0 = \bar{c}_f(x, s, t) < \bar{c}_f(y_l, s, t)$, there is $(i, j)$ such that $1 \le i \le j \le l$ and $\bar{c}_f(x, s, v_j) > 0$ and $\bar{c}_f(x, u_i, t) > 0$. Assume that the claim holds for $y_{l-1}$; we will prove it for $y_l$.

In the case $\bar{c}_f(x, s, t) < \bar{c}_f(y_{l-1}, s, t)$, the claim immediately follows from induction. Hence, suppose $0 = \bar{c}_f(x, s, t) = \bar{c}_f(y_{l-1}, s, t) < \bar{c}_f(y_l, s, t)$. Then, by Lemma 2.2 (a2), we have $\bar{c}_f(y_{l-1}, s, v_l) > 0$ and $\bar{c}_f(y_{l-1}, u_l, t) > 0$. The former means $\bar{c}_f(x, s, v_l) > 0$ by (6). The latter implies $\bar{c}_f(x, u_l, t) > 0$ or $\bar{c}_f(y_{l-1}, u_l, t) > \bar{c}_f(x, u_l, t) = 0$, which yields $\bar{c}_f(x, u_i, t) > 0$ for some $i \le l - 1$ by the inductive hypothesis for $y_{l-1}$. In both cases, we obtain $\bar{c}_f(x, s, v_l) > 0$ and $\bar{c}_f(x, u_i, t) > 0$ for some $i \le l$.

**(b3):** Fix $s^*, t^* \in E$ and $i^* \in [d]$. Suppose $s^* \neq v_{i^*}$ and $t^* \neq u_{i^*}$, since otherwise the claim is obvious. By $y_{i^*} = y_{i^*-1} + \alpha(\chi_{u_{i^*}} - \chi_{v_{i^*}})$ and Lemma 2.2 (a3),

$$\bar{c}_f(y_{i^*}, s^*, v_{i^*}) \ge \min\{\bar{c}_f(y_{i^*-1}, s^*, v_{i^*}), \bar{c}_f(y_{i^*}, u_{i^*}, v_{i^*})\}, \tag{8}$$
$$\bar{c}_f(y_{i^*}, u_{i^*}, t^*) \ge \min\{\bar{c}_f(y_{i^*-1}, u_{i^*}, t^*), \bar{c}_f(y_{i^*}, u_{i^*}, v_{i^*})\}. \tag{9}$$

For the first claim, assume $\bar{c}_f(x, s^*, v_j) = 0$ ($\forall j > i^*$). Since $\bar{c}_f(y_{j-1}, s^*, v_j) = \bar{c}_f(x, s^*, v_j)$ by (6), we obtain $\bar{c}_f(y_{j-1}, s^*, v_j) = 0$ ($\forall j > i^*$). As $y_j = y_{j-1} + \alpha(\chi_{u_j} - \chi_{v_j})$, apply Lemma 2.2 (a2)-(i) with $u, v, s, t$ replaced by $u_j, v_j, s^*, v_{i^*}$, respectively. Then $\bar{c}_f(y_j, s^*, v_{i^*}) = \bar{c}_f(y_{j-1}, s^*, v_{i^*})$ for every $j > i^*$, and hence $\bar{c}_f(y_{i^*}, s^*, v_{i^*}) = \bar{c}_f(y, s^*, v_{i^*})$. Also, we have $\bar{c}_f(y_{i^*-1}, s^*, v_{i^*}) = \bar{c}_f(x, s^*, v_{i^*})$ by (6) for $l = i^* - 1$ and $\bar{c}_f(y_{i^*}, u_{i^*}, v_{i^*}) = \bar{c}_f(y, u_{i^*}, v_{i^*})$ by (3). Substituting these three into (8) gives $\bar{c}_f(y, s^*, v_{i^*}) \ge \min\{\bar{c}_f(x, s^*, v_{i^*}), \bar{c}_f(y, u_{i^*}, v_{i^*})\}$.

For the second claim, assume $\bar{c}_f(x, u_j, t^*) = 0$ ($\forall j < i^*$). Then (a2)-(ii) implies $\bar{c}_f(y_j, u_{i^*}, t^*) = \bar{c}_f(y_{j-1}, u_{i^*}, t^*)$ for every $j < i^*$, and we obtain $\bar{c}_f(y_{i^*-1}, u_{i^*}, t^*) = \bar{c}_f(x, u_{i^*}, t^*)$. Note that every $l \in [i^* + 1, d]$ satisfies $i^* \in [l - 1]$. Then repeated application of (7) yields $\bar{c}_f(y_{i^*}, u_{i^*}, t^*) = \bar{c}_f(y, u_{i^*}, t^*)$. Substituting these two and $\bar{c}_f(y_{i^*}, u_{i^*}, v_{i^*}) = \bar{c}_f(y, u_{i^*}, v_{i^*})$ into (9), we obtain $\bar{c}_f(y, u_{i^*}, t^*) \ge \min\{\bar{c}_f(x, u_{i^*}, t^*), \bar{c}_f(y, u_{i^*}, v_{i^*})\}$.

**(b4):** Every $l \in [d-1]$ satisfies $\bar{c}_f(x, u_l, v_d) = 0$ by $l < d$, and hence $u_l \in \mathrm{sat}_f(x)$.

**(b5):** If $u_d \in \mathrm{sat}_f(x)$, as we have (b4), $u_l \in \mathrm{sat}_f(x)$ holds for every $l \in [d]$. Then, by repeating application of Lemma 2.2 (a4) $d$ times, we obtain $\mathrm{sat}_f(y_l) = \mathrm{sat}_f(x)$ and $\hat{c}_f(y_l, s) = \hat{c}_f(x, s)$ $(\forall s \in E \setminus \mathrm{sat}_f(x))$ for each $y_l$, and so for $y = y_d$.

**(b6):** As shown in the proof of (b4), we have $\mathrm{sat}_f(y_{d-1}) = \mathrm{sat}_f(x)$ and $\hat{c}_f(y_{d-1}, s) = \hat{c}_f(x, s)$ $(\forall s \in E \setminus \mathrm{sat}_f(x))$. Also, $\bar{c}_f(y_{d-1}, s, v_d) = \bar{c}_f(x, s, v_d)$ $(\forall s \in E \setminus \{v_d\})$ by (6). If $u_d \notin \mathrm{sat}_f(x)$, by applying Lemma 2.2 (a5) for $y_{d-1}$, we obtain the claim. □

**Lemma 2.4** (**Simple Augmentation**)**.** For any $x \in \mathbf{P}(f)$ and $u \in E$, define $y \in \mathbf{R}^E$ by $y := x + \alpha\chi_u$ with $0 \leq \alpha \leq \hat{c}_f(x, u)$. Then, we have $y \in \mathbf{P}(f)$ and the following (c1)–(c3).

**(c1)** $\hat{c}_f(y, u) = \hat{c}_f(x, u) - \alpha$.

**(c2)** For every $s \in \mathrm{sat}_f(x)$ and $t \in E \setminus \{s\}$, we have $\bar{c}_f(y, s, t) = \bar{c}_f(x, s, t)$.

**(c3)** Every $s \in \mathrm{sat}_f(x)$ satisfies $s \in \mathrm{sat}_f(y)$ and $\mathrm{dep}_f(y, s) = \mathrm{dep}_f(x, s)$.

*Proof.* We need only consider the case that $\hat{c}_f(x, u) > 0$, i.e., $u \notin \mathrm{sat}_f(x)$. By the definition, $y \in \mathbf{P}(f)$ and condition (c1) are obvious. For (c2), let $s \in \mathrm{sat}_f(x)$ and $C := \mathrm{sat}_f(x)$. Then $s \in C$, $u \notin C$, and $x(C) = y(C) = f(C)$. Take any $t \in E \setminus \{s\}$. Then for any $A \subseteq E$ with $s \in A$ and $t \notin A$, we have $s \in A \cap C$ and $t \notin A \cap C$ and $x(A \cap C) = y(A \cap C)$. Hence, as is the case in Lemma 2.2 (a1), we obtain $\bar{c}_f(y, s, t) = \bar{c}_f(x, s, t)$. The statement (c3) follows from $u \notin \mathrm{sat}_f(x)$, which implies $u \notin \mathrm{dep}_f(x, s)$ for any $s \in \mathrm{sat}_f(x)$, and hence $y(A) = x(A)$ for every $A \subseteq \mathrm{dep}_f(x, s)$. □

## 2.2 Preferences on Polymatroids

We now introduce a partial order on vectors, which is induced from a total order on the ground set. This partial order is regarded as a preference order in our model.

Let $\succ$ be a total order on $E$. For any element $a \in E$, we denote $E_{\succeq a} := \{\, e \in E \mid e \succeq a \,\}$. For two vectors $x, y \in \mathbf{R}^E$, we say that $x$ is $\succ$*-preferable* to $y$ if

$$\forall a \in E: \; x(E_{\succeq a}) \geq y(E_{\succeq a}).$$

A vector $x$ is $\succ$*-optimal* in a set $K \subseteq \mathbf{R}^E$ if $x \in K$ and $x$ is $\succ$-preferable to every $y \in K$.

**Lemma 2.5.** Let $(E, f, \succ)$ be an ordered polymatroid. For $x, y \in \mathbf{P}(f)$ if

$$\forall e \in E: \quad x(e) \geq y(e) \quad \text{or} \quad [e \in \mathrm{sat}_f(x),\ \mathrm{dep}_f(y, e) \succeq e],$$

then $x$ is $\succ$-preferable to $y$.

*Proof.* Take an arbitrary $a \in E$. We show $x(E_{\succeq a}) \geq y(E_{\succeq a})$. Define

$$C := \bigcup \{\, \mathrm{dep}_f(x, e) \mid e \in E_{\succeq a},\ [e \in \mathrm{sat}_f(x),\ \mathrm{dep}_f(x, e) \succeq e] \,\}.$$

Then $C \subseteq E_{\succeq a}$ and every $e \in E_{\succeq a} \setminus C$ satisfies $x(e) \geq y(e)$. Also, since $C$ is a union of tight sets, it is also tight. As $y \in \mathbf{P}(f)$, this implies $x(C) = f(C) \geq y(C)$. Then, we have $x(E_{\succeq a}) = x(C) + x(E_{\succeq a} \setminus C) \geq y(C) + y(E_{\succeq a} \setminus C) = y(E_{\succeq a})$. □

**Lemma 2.6.** Let $(E, f, \succ)$ be an ordered polymatroid. For $x, y \in \mathbf{P}(f)$ and $a \in E$, if we have

$$\forall e \in E_{\succeq a}: \;\; x(e) \geq y(e) \;\text{ or }\; [e \in \mathrm{sat}_f(x),\ \mathrm{dep}_f(x, e) \succeq e],$$

then we have

$$\forall e \in E_{\succeq a}: \;\; y(e) \geq x(e) \;\text{ or }\; \neg[e \in \mathrm{sat}_f(y),\ \mathrm{dep}_f(y, e) \succeq e].$$

*Proof.* Assume that each $e \in E_{\succeq a}$ satisfies $x(e) \geq y(e)$ or $[e \in \mathrm{sat}_f(x),\ \mathrm{dep}_f(x,e) \succeq e]$. Also assume that $e' \in E_{\succeq a}$ satisfies $[e' \in \mathrm{sat}_f(y),\ \mathrm{dep}_f(y,e') \succeq e']$. We now show $y(e') \geq x(e')$. Let $C_y := \mathrm{dep}_f(y,e')$ and

$$C_x := \bigcup \{\ \mathrm{dep}_f(x,e) \mid e \in C_y \setminus \{e'\},\ [e \in \mathrm{sat}_f(x),\ \mathrm{dep}_f(x,e) \succeq e]\ \}$$

and $D := (C_y \setminus C_x) \setminus \{e'\}$. Then, $\{\{e'\}, C_y \cap C_x, D\}$ is a partition of $C_y$. By definition, $y(C_y) = f(C_y)$ and $x(C_x) = f(C_x)$, and hence the submodularity of $f$ implies

$$y(C_y) - y(C_y \cap C_x) \geq f(C_y) - f(C_y \cap C_x) \geq f(C_y \cup C_x) - f(C_x) \geq x(C_y \cup C_x) - x(C_x).$$

As we have $C_y = \mathrm{dep}_f(y,e') \succeq e' \in E_{\succeq a}$, we obtain $C_y \subseteq E_{\succeq a}$, and hence every $e \in D = (C_y \setminus C_x) \setminus \{e'\}$ satisfies $x(e) \geq y(e)$, which implies $x(D) \geq y(D)$. Then, we have $y(e') = y(C_y) - y(C_y \cap C_x) - y(D) \geq x(C_y \cup C_x) - x(C_x) - x(D) = x(e')$. $\square$

## 3 Algorithm

In this section, we present an algorithm for finding a stable allocation for $(E, h, \succ_{\mathrm{H}})$ and $(E, f, \succ_{\mathrm{F}})$. The algorithm adopts the augmenting path technique for polymatroid intersection [23]. Each iteration searches for an augmenting path by simulating a chain of proposes and rejects in the deferred acceptance algorithm [15].

We first introduce *pointer functions* $\mathrm{next}_h$ and $\mathrm{next}_f$ with reference to $(E, h, \succ_{\mathrm{H}})$ and $(E, f, \succ_{\mathrm{F}})$, respectively. In our algorithm, they play fundamental roles of suggesting which element to propose or reject at the next step.

For $D \subseteq E$, $x \in \mathbf{P}(h)$, and $v \in E$, define $\mathrm{next}_h(D,x,v) \in E$ by

$$\mathrm{next}_h(D,x,v) := \max_{\succ_{\mathrm{H}}} \{\, u \in D \setminus \{v\} \mid \bar{c}_h(x,u,v) > 0 \,\}.$$

If the set in the right-hand side is empty, then $\mathrm{next}_h(D,x,v)$ is undefined. Note that $\mathrm{next}_h(D,x,v)$ is the best element w.r.t. $\succ_{\mathrm{H}}$ in $D$ to increase at an exchange with $v$.

For $x \in \mathbf{P}(f)$ and $u \in E$, define $\mathrm{next}_f(x,u) \in E$ by

$$\mathrm{next}_f(x,u) := \min_{\succ_{\mathrm{F}}} \{\, v \in E \mid \bar{c}_f(x,u,v) > 0 \,\}.$$

If $u \in \mathrm{sat}_f(x)$, $\mathrm{next}_f(x,u)$ represents the best element w.r.t. $\succ_{\mathrm{F}}$ to decrease at an exchange with $u$. Note that $\mathrm{next}_f(x,u) = u$ means that every $v \in E$ with $u \succ_{\mathrm{F}} v$ satisfies $\bar{c}_f(x,u,v) = 0$.

For a common independent vector $x \in \mathbf{P}(h) \cap \mathbf{P}(f)$, we introduce capacities of sequences. Let $P = \{u_0, v_1, u_1, \ldots, v_k, u_k\}$ be a sequence of $2k+1$ distinct elements of $E$. We define the *path-capacity* of $P$ by

$$c(x,P) = \min \left\{ \hat{c}_h(x,u_0),\ \min_{i:1 \leq i \leq k} \bar{c}_f(x,u_{i-1},v_i),\ \min_{i:1 \leq i \leq k} \bar{c}_h(x,u_i,v_i),\ \hat{c}_f(x,u_k) \right\}.$$

Let $Q = \{u_0, v_1, u_1, \ldots, u_{k-1}, v_k\}$ be a sequence of $2k$ distinct elements of $E$. We define the *cycle-capacity* of $Q$ by

$$c(x,Q) = \min \left\{ \min_{i:1 \leq i \leq k} \bar{c}_f(x,u_{i-1},v_i),\ \min_{i:1 \leq i \leq k} \bar{c}_h(x,u_i,v_i) \right\},$$

where we regard $u_k := u_0$.

The algorithm keeps a vector $x \in \mathbf{P}(h) \cap \mathbf{P}(f)$ and updates it repeatedly. The algorithm also maintains two disjoint subsets $D, R \subseteq E$.

We now introduce three procedures which will be used in the algorithm. The procedure Augment is applied to odd-length sequences. For a sequence $P = \{u_0, v_1, \ldots, v_{k-1}, u_{k-1}\}$ ($P$ can be $\{u_0\}$), the procedure Augment($P$) updates $(x, D, R)$ by

$$\begin{aligned} &x \leftarrow x + c(x,P)\left(\chi_{u_0} + \textstyle\sum_{j=1}^{k-1}(\chi_{u_j} - \chi_{v_j})\right),\\ &D \leftarrow D \setminus \{v_1, v_2, \ldots, v_{k-1}\},\\ &R \leftarrow R \cup \{v_1, v_2, \ldots, v_{k-1}\}. \end{aligned}$$

The procedure Cycle is applied to even-length sequences. For a sequence $Q = \{u_l, v_{l+1}, \ldots, u_{k-1}, v_k\}$ with $l < k$, the procedure Cycle($Q$) updates $(x, D, R)$ by

$$\begin{aligned} &x \leftarrow x + c(x,Q)\textstyle\sum_{j=l+1}^{k}(\chi_{u_{j-1}} - \chi_{v_j}),\\ &D \leftarrow D \setminus \{v_{l+1}, v_{l+2}, \ldots, v_k\},\\ &R \leftarrow R \cup \{v_{l+1}, v_{l+2}, \ldots, v_k\}. \end{aligned}$$

The procedure Self-loop is applied to an element $e$ of $D$ and moves $e$ from $D$ to $R$, i.e, $D \leftarrow D \setminus \{e\}$ and $R \leftarrow R \cup \{e\}$, without changing $x$.

We are now ready to describe the algorithm for finding a stable allocation.

**Algorithm 1** Find a stable allocation

**Input:** $(E, h, \succ_{\mathrm{H}})$ and $(E, f, \succ_{\mathrm{F}})$;
**Output:** stable allocation;

```
1: $D \leftarrow \emptyset$, $R \leftarrow \emptyset$, $x \leftarrow \mathbf{0}$;
2: while $D \cup R \neq E$ do
3:   $e^* \leftarrow \max_{\succ_{\mathrm{H}}} E \setminus (D \cup R)$;
4:   $D \leftarrow D \cup \{e^*\}$;
5:   while $e^* \in D \setminus \mathrm{sat}_h(x)$ do
6:     $u_0 \leftarrow e^*$;
7:     for $k = 1, 2, \ldots$ do
8:       if $u_{k-1} \notin \mathrm{sat}_f(x)$ then Augment($\{u_0, v_1, \ldots, u_{k-1}\}$) and break;
9:       $v_k \leftarrow \mathrm{next}_f(x, u_{k-1})$;
10:      if $v_k = u_{k-1}$ then Self-loop($u_{k-1}$) and break;
11:      if $v_k = v_l\ (\exists l < k)$ then Cycle($\{u_l, v_{l+1} \ldots, u_{k-1}, v_k\}$) and break;
12:      $u_k \leftarrow \mathrm{next}_h(D, x, v_k)$;
13:      if $u_k = u_l\ (\exists l < k)$ then Cycle($\{u_l, v_{l+1} \ldots, u_{k-1}, v_k\}$) and break;
14:    end for
15:  end while
16: end while
17: return $x$.
```

We now describe how our algorithm works on an example.

**Example 3.1.** Consider total orders $\succ_{\mathrm{H}}$ and $\succ_{\mathrm{F}}$ on $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ given by

$$\begin{aligned} &e_1 \succ_{\mathrm{H}} e_2 \succ_{\mathrm{H}} e_3 \succ_{\mathrm{H}} e_4 \succ_{\mathrm{H}} e_5 \succ_{\mathrm{H}} e_6,\\ &e_2 \succ_{\mathrm{F}} e_3 \succ_{\mathrm{F}} e_5 \succ_{\mathrm{F}} e_6 \succ_{\mathrm{F}} e_1 \succ_{\mathrm{F}} e_4. \end{aligned}$$

Let $G_{\mathrm{H}}$ and $G_{\mathrm{F}}$ be bipartite graphs with vertex sets $E \cup V_{\mathrm{H}}$ and $E \cup V_{\mathrm{F}}$ depicted in Figure 1, where vertex capacity functions $b_{\mathrm{H}} : V_{\mathrm{H}} \to \mathbf{R}$ and $b_{\mathrm{F}} : V_{\mathrm{F}} \to \mathbf{R}$ are also given. Define rank

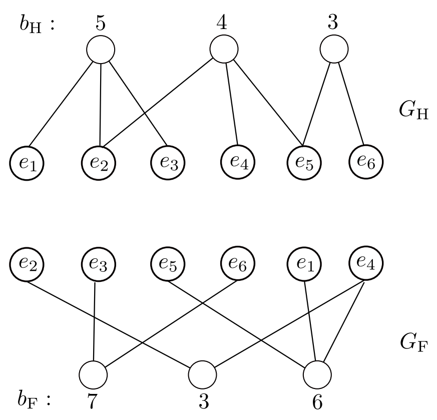
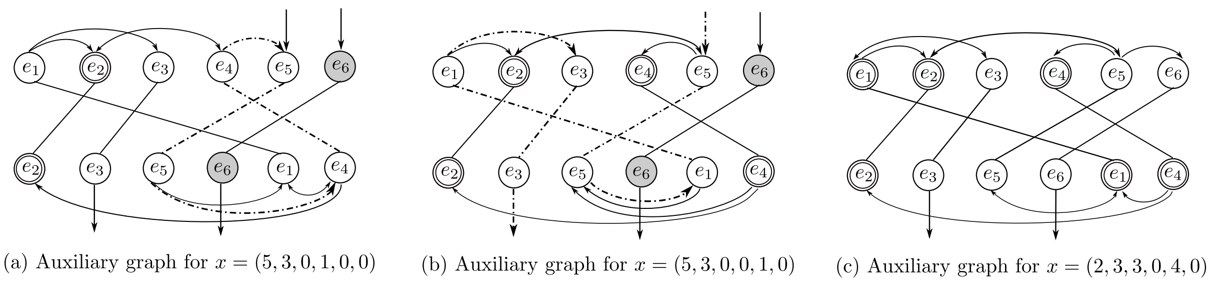


Figure 1: Bipartite graphs inducing $h$ and $f$.



(a) Auxiliary graph for $x = (5, 3, 0, 1, 0, 0)$ (b) Auxiliary graph for $x = (5, 3, 0, 0, 1, 0)$ (c) Auxiliary graph for $x = (2, 3, 3, 0, 4, 0)$

Figure 2: Auxiliary graphs for $x$ at some points of the algorithm. Each graph has two lines of nodes. The upper line consists of the elements of $E$ arranged in descending order w.r.t. $\succ_H$, and so does the lower line w.r.t. $\succ_F$. In the upper line, an arc from $e$ to $e'$ means $\bar{c}_h(x, e', e) > 0$ and an arc from the outside to $e$ means $\hat{c}_h(x, e) > 0$. In the lower line, an arc from $e$ to $e'$ means $\bar{c}_f(x, e, e') > 0$ and an arc from $e$ to the outside means $\hat{c}_f(x, e) > 0$. We omit some exchangeability arcs incident to unsaturated elements or implied by the transitivity. Elements in $D \subseteq E$ and $R \subseteq E$ are represented by single and double circles, respectively. Other elements are colored gray.

functions $h, f : 2^E \to \mathbf{R}$ by

$$h(A) = \sum \{\, b_H(v) \mid v \in \Gamma_H(A)\,\} \quad (A \subseteq E),$$
$$f(A) = \sum \{\, b_F(v) \mid v \in \Gamma_F(A)\,\} \quad (A \subseteq E),$$

where $\Gamma_H(A)$ and $\Gamma_F(A)$ denote the sets of nodes adjacent to $A$ in $G_H$ and in $G_F$, respectively.

We now apply the algorithm to $(E, h, \succ_H)$ and $(E, f, \succ_F)$. Just after Line 4 with $e^* = e_5$, we have $x = (5, 3, 0, 1, 0, 0)$ and its auxiliary graph is depicted in Figure 2 (a). Then, the algorithm searches an augmenting path and finds a cycle $(e_5, e_4)$ whose cycle-capacity is 1. Hence, the procedure $\mathsf{Cycle}(\{e_5, e_4\})$ updates $x \leftarrow (5, 3, 0, 0, 1, 0)$. The updated auxiliary graph is depicted in Figure 2 (b). In the next search, the algorithm finds an augmenting path $(e_5, e_1, e_3)$ with path-capacity 3, and calls $\mathsf{Augment}(\{e_5, e_1, e_3\})$. Thus, $x$ is updated as $x \leftarrow (2, 3, 3, 0, 4, 0)$. Then, we see $e_5, e_6 \in \mathrm{sat}_h(x)$, and the algorithm terminates with returning this $x$, whose auxiliary graph is depicted in Figure 2 (c). In this graph, for any $e \in D$ in the upper line, every arc entering $e$ comes from left, which means $[e \in \mathrm{sat}_h(x),\ \mathrm{dep}_h(x, e) \succeq_H e]$. Also, for any $e \in R$, every arc leaving $e$ in the lower line goes to left, i.e., $[e \in \mathrm{sat}_f(x),\ \mathrm{dep}_f(x, e) \succeq_F e]$. Since $D \cup R = E$, we may conclude that the output $x = (2, 3, 3, 0, 4, 0)$ is indeed a stable allocation. ■

## 4 Invariants

The main part of our analysis, provided in Section 5, is to show that the following invariants are maintained for $(x, D, R)$ throughout the algorithm.

**(P)** $\{D, R\}$ is a partition of $E_{\succeq_{\mathrm{H}} e^*}$ and $x(e) = 0$ for every $e \in E \backslash (D \cup R)$, where $e^* := \min_{\succ_{\mathrm{H}}} D$.

**(H)** $x \in \mathbf{P}(h)$ holds and every $e \in D \setminus \{e^*\}$ satisfies $[e \in \mathrm{sat}_h(x),\ \mathrm{dep}_h(x, e) \succeq_{\mathrm{H}} e]$, where $e^* := \min_{\succ_{\mathrm{H}}} D$. Also, for each $t \in E$, the inequality $\bar{c}_h(x, e^*, t) > 0$ implies $t \succeq_{\mathrm{H}} e^*$.

**(F)** $x \in \mathbf{P}(f)$ holds and every $e \in R$ satisfies $[e \in \mathrm{sat}_f(x),\ \mathrm{dep}_f(x, e) \succeq_{\mathrm{F}} e]$.

As will be shown in Section 5, if these conditions hold at the end of the algorithm, then the output is a stable allocation.

In this section, we prepare three lemmas related to these invariants. They will be used to analyze the algorithm later in Sections 5–7.

**Lemma 4.1 (Multiple Exchange in $\mathbf{P}(h)$).** For an independent vector $x \in \mathbf{P}(h)$ and $D \subseteq E$, let $u_i, v_i$ $(i = 1, 2, \ldots, d)$ be $2d$ distinct elements of $E$ such that

**(h0)** $\mathrm{next}_h(D, x, v_i) = u_i$

holds for every $i \in [d]$. For any $\alpha \geq 0$ satisfying $\alpha \leq \bar{c}_h(x, u_i, v_i)$ for every $i \in [d]$, define $y \in \mathbf{R}^E$ by $y := x + \alpha \sum_{i=1}^{d} (\chi_{u_i} - \chi_{v_i})$. Then, we have $y \in \mathbf{P}(h)$ and the following (h1)–(h3), where $E' := E \setminus \{u_1, v_1, \ldots, u_d, v_d\}$ and $D' := D \setminus \{v_1, v_2, \ldots, v_d\}$.

**(h1)** If (H) holds for $x$ and $D$, then the same statement holds with $x$ and $D$ replaced by $y$ and $D'$, respectively.

**(h2)** For $t \in E \setminus \{u_1, u_2, \ldots, u_d\}$, if $\mathrm{next}_h(D, x, t)$ is defined, then $\mathrm{next}_h(D', y, t)$ is undefined or it is defined and satisfies $\mathrm{next}_h(D, x, t) \succeq_{\mathrm{H}} \mathrm{next}_h(D', y, t)$. If $\mathrm{next}_h(D, x, t)$ is undefined, then $\mathrm{next}_h(D', y, t)$ is also undefined.

**(h3)** For $i \in [d]$, we have $\bar{c}_h(y, u_i, v_i) = \bar{c}_h(x, u_i, v_i) - \alpha$. For $s, t \in E'$ with $\mathrm{next}_h(D, x, t) = s$, we have $\bar{c}_h(y, s, t) = \bar{c}_h(x, s, t)$. For $t \in E'$ and $i \in [d]$ with $\mathrm{next}_h(D, x, t) = u_i$, we have $\bar{c}_h(y, u_i, t) \geq \min\{\bar{c}_h(x, u_i, t), \bar{c}_h(y, u_i, v_i)\}$.

*Proof.* Without loss of generality, we can assume $u_1 \succ_{\mathrm{H}} u_2 \succ_{\mathrm{H}} \ldots \succ_{\mathrm{H}} u_d$. Then, the definition of $\mathrm{next}_h$ and the assumption that (h0) holds for every $i \in [d]$ imply

$$u_i \in D \setminus \{v_1, v_2, \ldots, v_d\} \qquad (i \in [d]), \tag{10}$$

$$\bar{c}_h(x, u_i, v_i) > 0 \qquad (i \in [d]), \tag{11}$$

$$s \succ_{\mathrm{H}} u_i \implies \bar{c}_h(x, s, v_i) = 0 \quad (i \in [d],\ s \in D \setminus \{v_i\}). \tag{12}$$

In particular, since $i < j$ implies $u_i \succ_{\mathrm{H}} u_j$ for every $i, j \in [d]$, the condition (12) yields

$$\bar{c}_h(x, u_i, v_j) = 0 \quad (i, j \in [d] \text{ with } i < j). \tag{13}$$

By (11) and (13), we can apply Lemma 2.3 on multiple exchange to see that $y \in \mathbf{P}(h)$ holds.

**(h1):** Suppose that (H) holds for $x$ and $D$. This is equivalent to the combination of the following two conditions.

(H1) $x \in \mathbf{P}(h)$ and $D \setminus \{\min_{\succ_{\mathrm{H}}} D\} \subseteq \mathrm{sat}_h(x)$.

(H2) For any $s \in D$ and $t \in E$, if $\bar{c}_h(x, s, t) > 0$, then $t \succeq_{\mathrm{H}} s$.

We show that (H1) and (H2) hold with $x$ and $D$ replaced by $y$ and $D'$. We already have $y \in \mathbf{P}(h)$. Fix $e^* := \min_{\succ_{\mathrm{H}}} D$, and note that $D' \subseteq D$ implies $D' \setminus \{\min_{\succ_{\mathrm{H}}} D'\} \subseteq D' \setminus \{e^*\}$.

To obtain (H1) for $y$ and $D'$, it suffices to show $D' \setminus \{e^*\} \subseteq \mathrm{sat}_h(y)$. Recall that we have (10), (11), and (13). If $u_d \neq e^*$, then $u_d \in D \setminus \{e^*\} \subseteq \mathrm{sat}_h(x)$ and Lemma 2.3 (b5) implies $\mathrm{sat}_h(y) = \mathrm{sat}_h(x)$, and hence $D' \setminus \{e^*\} \subseteq D \setminus \{e^*\} \subseteq \mathrm{sat}_h(x) = \mathrm{sat}_h(y)$. Also, if $u_d = e^*$, then $\mathrm{next}_h(D, x, v_d) = u_d = e^* = \min_{\succ_{\mathrm{H}}} D$. This implies that, for every $s \in D \setminus \{e^*, v_d\}$, we have $\bar{c}_h(x, s, v_d) = 0$, and hence $s \in \mathrm{sat}_h(y)$ by (b6). Thus, $D' \setminus \{e^*\} \subseteq D \setminus \{e^*, v_d\} \subseteq \mathrm{sat}_h(y)$.

We show (H2) for $y$ and $D'$. Suppose, to the contrary, that there are $s \in D'$ and $t \in E$ such that $\bar{c}_h(y, s, t) > 0$ and $s \succ_{\mathrm{H}} t$. Then $s \in D' \subseteq D$ and $s \succ_{\mathrm{H}} t$ imply $\bar{c}_h(x, s, t) = 0$, because (H2) holds for $x$ and $D$. As we have $\bar{c}_h(x, s, t) = 0$ and $\bar{c}_h(y, s, t) > 0$, Lemma 2.3 (b2) implies that there are $i, j \in [d]$ such that

$$i \leq j, \quad \bar{c}_h(x, s, v_j) > 0, \quad \bar{c}_h(x, u_i, t) > 0.$$

Since $\mathrm{next}_h(D, x, v_j) = u_j$, conditions $s \in D' \subseteq D \setminus \{v_j\}$ and $\bar{c}_h(x, s, v_j) > 0$ imply $u_j \succeq_{\mathrm{H}} s$. As (H2) holds for $x$ and $D$, conditions $u_i \in D$ and $\bar{c}_h(x, u_i, t) > 0$ imply $t \succeq u_i$. Also, $i \leq j$ implies $u_i \succeq_{\mathrm{H}} u_j$. Thus, we obtain $t \succeq_{\mathrm{H}} u_i \succeq_{\mathrm{H}} u_j \succeq_{\mathrm{H}} s$, which contradicts $s \succ_{\mathrm{H}} t$.

**(h2):** Suppose, to the contrary, the claim fails for some $t \in E \setminus \{u_1, u_2, \ldots, u_d\}$. Then, either of the following holds.

1. Both $s = \mathrm{next}_h(D, x, t)$ and $s' = \mathrm{next}_h(D', y, t)$ are defined and $s' \succ_{\mathrm{H}} s$.
2. $s' = \mathrm{next}_h(D', y, t)$ is defined while $\mathrm{next}_h(D, x, t)$ is undefined.

In Case 1, we have $s \in D \setminus \{t\}$ and $s' \in D' \setminus \{t\} \subseteq D \setminus \{t\}$. By $s' = \mathrm{next}_h(D', y, t)$, the definition of $\mathrm{next}_h$ implies $\bar{c}_h(y, s', t) > 0$. Also, by $s' \succ_{\mathrm{H}} s = \mathrm{next}_h(D, x, t)$ and $s' \in D \setminus \{t\}$, we see that $\bar{c}_h(x, s', t) = 0$. By Lemma 2.3 (b2), then there are $i, j \in [d]$ such that

$$i \leq j, \quad \bar{c}_h(x, s', v_j) > 0, \quad \bar{c}_h(x, u_i, t) > 0.$$

Note that $u_i \in D \setminus \{t\}$ by $t \in E \setminus \{u_1, u_2, \ldots, u_d\}$. Then, $\bar{c}_h(x, u_i, t) > 0$ and $\mathrm{next}_h(D, x, t) = s$ imply $s \succ_{\mathrm{H}} u_i$. Similarly, by $s' \in D' \subseteq D \setminus \{v_j\}$, $\bar{c}_h(x, s', v_j) > 0$, and $\mathrm{next}_h(D, x, v_j) = u_j$, we have $u_j \succeq_{\mathrm{H}} s'$. Also, $i \leq j$ implies $u_i \succeq_{\mathrm{H}} u_j$. Thus, we obtain $s \succeq_{\mathrm{H}} u_i \succeq_{\mathrm{H}} u_j \succeq_{\mathrm{H}} s'$, which contradicts $s' \succ_{\mathrm{H}} s$.

In Case 2, we also have $\bar{c}_h(y, s', t) > 0$ and $\bar{c}_h(x, s', t) = 0$, and hence there are $i, j \in [d]$ such that $i \leq j$, $\bar{c}_h(x, s', v_j) > 0$, and $\bar{c}_h(x, u_i, t) > 0$. Since $u_i \in D \setminus \{t\}$, the condition $\bar{c}_h(x, u_i, t) > 0$ contradicts the fact that $\mathrm{next}_h(D, x, t)$ is undefined.

**(h3):** The first claim of (h3) immediately follows from Lemma 2.3 (b1).

To show the second claim by contradiction, suppose $\bar{c}_h(y, s, t) \neq \bar{c}_h(x, s, t)$ for some $s, t \in E'$ with $\mathrm{next}_h(D, x, t) = s$. By (b2), there exist $i, j \in [d]$ such that $i \leq j$, $\bar{c}_h(x, s, v_j) > 0$, and $\bar{c}_h(x, u_i, t) > 0$. Since $\mathrm{next}_h(D, x, v_j) = u_j$, the condition $\bar{c}_h(x, s, v_j) > 0$ implies $u_j \succeq_{\mathrm{H}} s$, and in particular $u_j \succ_{\mathrm{H}} s$ by $s \in E'$. Also, $\mathrm{next}_h(D, x, t) = s$ and $\bar{c}_h(x, u_i, t) > 0$ imply $s \succ_{\mathrm{H}} u_i$, and hence $u_j \succ_{\mathrm{H}} u_i$, which contradicts $i \leq j$.

We show the last claim of (h3). Let $t \in E'$ and $i \in [d]$ be such that $\mathrm{next}_h(D, x, t) = u_i$. Every $j \in [d]$ with $j < i$ satisfies $u_j \succ_{\mathrm{H}} u_i$ and $u_j \in D \setminus \{v_j\}$, and hence the condition $\mathrm{next}_h(D, x, t) = u_i$ implies $\bar{c}_h(x, u_j, t) = 0$. By Lemma 2.3 (b3), then we obtain $\bar{c}_h(y, u_i, t) \geq \min\{\bar{c}_h(x, u_i, t), \bar{c}_h(y, u_i, v_i)\}$. □

**Lemma 4.2 (Multiple Exchange in $\mathbf{P}(f)$).** For an independent vector $x \in \mathbf{P}(f)$, let $u_{i-1}, v_i$ $(i = 1, 2, \ldots, d)$ be $2d$ distinct elements of $E$ such that

**(f0)** $u_{i-1} \in \mathrm{sat}_f(x), \quad \mathrm{next}_f(x, u_{i-1}) = v_i$

holds for every $i \in [d]$. For any $\alpha \geq 0$ satisfying $\alpha \leq \bar{c}_h(x, u_i, v_i)$ for every $i \in [d]$, define $y \in \mathbf{R}^E$ by $y := x + \alpha \sum_{i=1}^{d} (\chi_{u_{i-1}} - \chi_{v_i})$. Then, we have $y \in \mathbf{P}(f)$ and the following (f1)–(f3), where $E' = E \setminus \{u_0, v_1, u_1, \ldots, u_{d-1}, v_d\}$.

**(f1)** If (F) holds for $x$ and some $R \subseteq E$, then the same statement holds with $x$ and $R$ replaced by $y$ and $R' := R \cup \{v_1, v_2, \ldots, v_d\}$, respectively.

**(f2)** For every $s \in \mathrm{sat}_f(x)$, we have $s \in \mathrm{sat}_f(y)$ and $\mathrm{next}_f(y, s) \succeq_{\mathrm{F}} \mathrm{next}_f(x, s)$.

**(f3)** For $i \in [d]$, we have $\bar{c}_f(y, u_{i-1}, v_i) = \bar{c}_f(x, u_{i-1}, v_i) - \alpha$. For $s, t \in E'$ with $\mathrm{next}_f(x, s) = t$, we have $\bar{c}_f(y, s, t) = \bar{c}_f(x, s, t)$. For $s \in E'$ and $i \in [d]$ with $\mathrm{next}_f(x, s) = v_i$, we have $\bar{c}_f(y, s, v_i) \geq \min\{\bar{c}_f(x, s, v_i), \bar{c}_f(y, u_{i-1}, v_i)\}$.

*Proof.* Without loss of generality, we can assume $v_1 \succ_{\mathrm{F}} v_2 \succ_{\mathrm{F}} \ldots \succ_{\mathrm{F}} v_d$. Then, the definition of $\mathrm{next}_f$ and the assumption that (f0) holds for every $i \in [d]$ imply

$$\bar{c}_f(x, u_i, v_i) > 0 \qquad (i \in [d]), \tag{14}$$

$$v_i \succ_{\mathrm{F}} t \implies \bar{c}_f(x, u_i, t) = 0 \qquad (i \in [d],\ t \in E). \tag{15}$$

In particular, since $i < j$ implies $v_i \succ_{\mathrm{F}} v_j$, the condition (15) yields

$$\bar{c}_f(x, u_i, v_j) = 0 \qquad (i, j \in [d] \text{ with } i < j). \tag{16}$$

By (14),(16), we can apply Lemma 2.3 on multiple exchange. Then, $y \in \mathbf{P}(f)$. Since (f0) says $u_{i-1} \in \mathrm{sat}_f(x)$ for every $i \in [d]$, Lemma 2.3 (b5) implies $\mathrm{sat}_f(y) = \mathrm{sat}_f(x)$. Also, (f0) implies $v_i \in \mathrm{sat}_f(x)$ for every $i \in [d]$, and hence we have

$$\{v_1, v_2, \ldots, v_d\} \subseteq \mathrm{sat}_f(x) = \mathrm{sat}_f(y). \tag{17}$$

We show (f2) first, and then (f1) and (f3).

**(f2):** Since we have $\mathrm{sat}_f(y) = \mathrm{sat}_f(x)$, it suffices to show $\mathrm{next}_f(y, s) \succeq_{\mathrm{F}} \mathrm{next}_f(x, s)$ for every $s \in \mathrm{sat}_f(x)$. Suppose, to the contrary, there is $s \in \mathrm{sat}_f(x)$ such that $t = \mathrm{next}_f(x, s)$, $t' = \mathrm{next}_f(y, s)$ and $t' \prec_{\mathrm{F}} t$. By the definition of $\mathrm{next}_f$, we have $\bar{c}_f(y, s, t') > 0$ and $\bar{c}_h(x, s, t') = 0$. By Lemma 2.3 (b2), then there are $i, j \in [d]$ such that $i \leq j$, $\bar{c}_f(x, s, v_j) > 0$, and $\bar{c}_f(x, u_i, t') > 0$. By $\bar{c}_h(x, u_i, t') > 0$ and $\mathrm{next}_f(x, u_i) = v_i$, we have $t' \succeq_{\mathrm{F}} v_i$. Similarly, $\bar{c}_f(x, s, v_j) > 0$ and $\mathrm{next}_f(x, s) = t$ imply $v_j \succeq_{\mathrm{F}} t$. Also, $i \leq j$ implies $v_i \succeq_{\mathrm{F}} v_j$. Thus, we have $t' \succeq_{\mathrm{F}} v_i \succeq_{\mathrm{F}} v_j \succeq_{\mathrm{F}} t$, a contradiction.

**(f1):** Suppose that (F) holds for $x$ and $R$. By the definition of $\mathrm{next}_f$, for any $s \in \mathrm{sat}_f(x)$, the condition $\mathrm{dep}_f(x, s) \succeq_{\mathrm{F}} s$ is equivalent to $\mathrm{next}_f(x, s) = s$. Then, (F) is rephrased as

$$\forall s \in R: \quad s \in \mathrm{sat}_f(x),\ \mathrm{next}_f(x, s) = s.$$

We show that (F) holds for $y$ and $R'$. By (F) for $x$ and $R$, we have $R \subseteq \mathrm{sat}_f(x)$. Then, (17) implies $R' = R \cup \{v_1, v_2, \ldots, v_d\} \subseteq \mathrm{sat}_f(x) = \mathrm{sat}_f(y)$. By (f2) and the definition of $\mathrm{next}_f$, every $s \in R' \subseteq \mathrm{sat}_f(x)$ satisfies $s \succeq_{\mathrm{F}} \mathrm{next}_f(y, s) \succeq_{\mathrm{F}} \mathrm{next}_f(x, s)$. Hence, it suffices to show $\mathrm{next}_f(x, s) = s$ for every $s \in R' = R \cup \{v_1, v_2, \ldots, v_d\}$. For every $s \in R$, we have $\mathrm{next}_f(x, s) = s$ by (F) for $x$ and $R$. Also, every $v_i$ satisfies $\mathrm{next}_f(x, v_i) = v_i$ as follows. Suppose, to the contrary, we have $\mathrm{next}_f(x, v_i) = t \prec_{\mathrm{F}} v_i$. Then, $\bar{c}_f(x, v_i, t) > 0$. Since we also have $\bar{c}_f(x, u_i, v_i) > 0$ by $\mathrm{next}_f(x, u_i) = v_i$, the transitivity (Lemma 2.1) implies $\bar{c}_f(x, u_i, t) > 0$. This contradicts $\mathrm{next}_f(x, u_i) = v_i \succ_{\mathrm{F}} t$

**(f3):** The first claim of (f3) immediately follows from Lemma 2.3 (b1).

To show the second claim by contradiction, suppose $\bar{c}_f(y,s,t) \neq \bar{c}_f(x,s,t)$ for some $s,t \in E'$ with $\mathrm{next}_f(x,s) = t$. By (b2), there exist $i,j \in [d]$ such that $i \leq j$, $\bar{c}_f(x,s,v_j) > 0$, and $\bar{c}_f(x,u_{i-1},t) > 0$. Since $\mathrm{next}_f(x,u_{i-1}) = v_i$, the condition $\bar{c}_f(x,u_{i-1},t) > 0$ implies $t \succeq_{\mathrm{F}} v_i$, and in particular $t \succ_{\mathrm{F}} v_i$ by $t \in E'$. Also, $\mathrm{next}_f(x,s) = t$ and $\bar{c}_f(x,s,v_j) > 0$ imply $v_j \succ_{\mathrm{F}} t$, and hence $v_j \succ_{\mathrm{F}} v_i$, which contradicts $i \leq j$.

We show the last claim of (f3). Let $s \in E'$ and $i \in [d]$ be such that $\mathrm{next}_f(x,s) = v_i$. Since every $j \in [d]$ with $j > i$ satisfies $v_i \succ_{\mathrm{F}} v_j$, the condition $\mathrm{next}_f(x,s) = v_i$ implies $\bar{c}_f(x,s,v_j) = 0$. By Lemma 2.3 (b3), then we obtain $\bar{c}_f(y,s,v_i) \geq \min\{\bar{c}_f(x,s,v_i), \bar{c}_f(y,u_{i-1},v_i)\}$. □

**Lemma 4.3.** Assume that (P) holds for $x \in \mathbf{P}(h) \cap \mathbf{P}(f)$ and $D,R \subseteq E$. Let $u_0$, $u_i, v_i$ $(i = 1,2,\ldots,d)$ be (not necessarily distinct) elements such that $u_0 \in D$ and (h0) and (f0) hold for every $i \in [d]$. Define $y := x + \alpha \sum_{i=1}^{d} (\chi_{u_{i-1}} - \chi_{v_i})$ for an arbitrary $\alpha \geq 0$. Then (P) also holds with $(x,D,R)$ replaced by $(y,D',R')$, where $D' := D \setminus \{v_1,v_2,\ldots,v_d\}$ and $R' := R \cup \{v_1,v_2,\ldots,v_d\}$.

*Proof.* By (f0), we have $x(v_i) > 0$ for every $i \in [d]$. Then $\{v_1,v_2,\ldots,v_d\} \subseteq D \cup R$ since $x$ satisfies (P). Therefore, $D' \cup R' = D \cup R \cup \{v_1,v_2,\ldots,v_d\} = D \cup R$. By $u_0 \in D$ and (h0), we have $\{u_0,u_1,\ldots,u_{d-1}\} \subseteq D \subseteq D' \cup R'$. Then, every $e' \in E \setminus (D' \cup R') = E \setminus (D \cup R)$ satisfies $y(e') = x(e') = 0$. Thus, (P) holds with $(x,D,R)$ replaced by $(y,D',R')$. □

## 5 Correctness

In this section, we show that the output of the algorithm is indeed a stable allocation. We first provide basic observations from the description of the algorithm. Recall that $[k,l] = \{k,k+1,\ldots,l\}$ for two nonnegative integers $k,l \in \mathbf{Z}_+$ with $k \leq l$.

**Claim 5.1.** Just before Line 11, $v_i \neq u_j$ for every $i \in [k]$ and $j \in [0,k-1]$.

*Proof.* By Line 10, $v_i \neq u_{i-1}$ for every $i \in [k]$. We then show the case that $j < i-1$ or $j > i-1$. Suppose, to the contrary, that $v_i = u_j$ for such $i$ and $j$. The definition of $\mathrm{next}_f$ implies $\bar{c}_f(x,u_j,v_{j+1}) > 0$ and $\bar{c}_f(x,u_{i-1},v_i) = \bar{c}_f(x,u_{i-1},u_j) > 0$. Then, the transitivity of dependence (Lemma 2.1) implies $\bar{c}_f(x,u_{i-1},v_{j+1}) > 0$. As $\mathrm{next}_f(x,u_{i-1}) = v_i$, this implies $v_{j+1} \succeq_{\mathrm{F}} v_i = u_j$, and hence $v_{j+1} \succeq_{\mathrm{F}} u_j$. On the other hand, by Line 10 for $j \in [0,k-1]$, we have $\mathrm{next}_f(x,u_j) = v_{j+1} \neq u_j$, which implies $u_j \succ_{\mathrm{F}} v_{j+1}$. This contradicts $\mathrm{next}_f(x,u_j) \preceq_{\mathrm{F}} u_j$, which must hold by the definition of $\mathrm{next}_f$. □

**Claim 5.2.** Just before Line 12, $e^* \in D \setminus \{v_k\}$ and $\bar{c}_h(x,e^*,v_k) > 0$ hold. Hence, just after Line 12, $\mathrm{next}_h(D,x,v_k)$ satisfies $\mathrm{next}_h(D,x,v_k) \succeq_{\mathrm{H}} e^*$.

*Proof.* Just before Line 12, $x(v_k) > 0$ by $\mathrm{next}_f(x,u_{k-1}) = v_k \neq u_{k-1}$. Also, by Line 5, we have $e^* \in D$ and $e^* \notin \mathrm{sat}_h(x)$, which implies $\bar{c}_f(x,e^*,v_k) > 0$. Also, $e^* = u_0 \neq v_k$ by Claim 5.1. □

**Lemma 5.3.** In the algorithm, the following statements hold.

- When $\mathsf{Self\text{-}loop}(u_{k-1})$ is called, $u_{k-1} \in \mathrm{sat}_f(x)$ and $\mathrm{next}_f(x,u_{k-1}) = u_{k-1}$.
- When $\mathsf{Cycle}(\{u_l,v_{l+1}\ldots,u_{k-1},v_k\})$ is called, the elements of $\{u_l,v_{l+1}\ldots,u_{k-1},v_k\}$ are all distinct and every $i \in [l+1,k]$ satisfies (h0) and (f0), where $u_k = u_l$.
- When $\mathsf{Augment}(\{u_0,v_1,\ldots,v_{k-1},u_{k-1}\})$ is called, the elements of $\{u_0,v_1,\ldots,v_{k-1},u_{k-1}\}$ are all distinct, we have $u_0 \in D \setminus \mathrm{sat}_h(x)$ and $u_{k-1} \notin \mathrm{sat}_f(x)$, and every $i \in [k-1]$ satisfies (h0) and (f0).

*Proof.* By the algorithm, when Cycle is called, we have $u_i \neq u_j$ and $v_i \neq v_j$ for every distinct $i, j \in [l+1, k]$, where $u_k = u_l$. With Lemma 5.1, then all elements in the sequence are distinct. Similarly, when Augment is called, $u_i \neq u_j$ for every distinct $i, j \in [0, k-1]$ and $v_i \neq v_j$ for every distinct $i, j \in [k-1]$, and hence all elements in the sequence are distinct. Other statements immediately follow from the algorithm. □

We prove the correctness of the algorithm by showing that conditions (P), (H), and (F) are maintained throughout the algorithm.

**Observation 5.4.** When the algorithm reaches Line 5 for the first time, (P), (H), and (F) hold.

**Lemma 5.5.** Suppose that conditions (P), (H), and (F) hold just after Line 15. If $D \cup R \neq E$, then the conditions also hold when the algorithm reaches Line 5 the next time. If $D \cup R = E$, then the algorithm halts and the current $x$ is a stable allocation.

*Proof.* After Line 15, we have $e^* \notin D \setminus \mathrm{sat}_h(x)$ by Line 5, and hence $e^* \notin D$ or $e^* \in \mathrm{sat}_h(x)$. Also, in the latter case, $\mathrm{dep}_h(x, e^*) \succeq_{\mathrm{H}} e^*$ because $x(e) = 0$ for every $e \in E$ with $e^* \succ_{\mathrm{H}} e$. Then, just after Line 15, every $e \in D$ satisfies $[e \in \mathrm{sat}_h(x),\ \mathrm{dep}_h(x, e) \succeq_{\mathrm{H}} e]$. This means that (H) holds when the algorithm reaches Line 5 next time. Also, (P) and (F) obviously hold since $x$ and $R$ do not change. If $D \cup R = E$, these conditions mean that $x$ is a stable allocation. □

By Observation 5.4 and Lemma 5.5, what is left is to show that conditions (P), (H), and (F) are maintained whenever the procedures are applied.

**Lemma 5.6.** If (P), (H), and (F) hold when the procedure Self-loop$(u_{k-1})$ is called, then they also hold just after the procedure.

*Proof.* The procedure Self-loop$(u_{k-1})$ does not change the vector $x$ and replaces $(D, R)$ by $(D \setminus \{u_{k-1}\}, R \cup \{u_{k-1}\})$. Clearly the conditions (P) and (H) are maintained. The element $u_{k-1}$, which is added to $R$, satisfies $u_{k-1} \in \mathrm{sat}_f(x)$ by Line 8 and $\mathrm{next}_f(x, u_{k-1}) = u_{k-1}$ by Line 10, which implies $\mathrm{dep}_f(x, u_{k-1}) \succeq_{\mathrm{F}} u_{k-1}$. Thus, (F) is maintained. □

**Lemma 5.7.** If (P), (H), and (F) hold when the procedure Cycle$(\{u_l, v_{l+1}, \dots, u_{k-1}, v_k\})$ is called, then they also hold just after the procedure.

*Proof.* Let $Q$ denote the sequence $\{u_l, v_{l+1}, \dots, u_{k-1}, v_k\}$. The procedure updated the triple $(x, D, R)$ to $(y, D', R')$, where

$$\begin{aligned} y &= x + c(x, Q) \textstyle\sum_{i=l+1}^{k} (\chi_{u_{i-1}} - \chi_{v_i}), \\ D' &= D \setminus \{v_{l+1}, v_{l+2}, \dots, v_k\}, \\ R' &= R \cup \{v_{l+1}, v_{l+2}, \dots, v_k\}. \end{aligned}$$

We now show that conditions (P), (H), and (F) hold with $(x, D, R)$ replaced by $(y, D', R')$. By Lemma 5.3, $u_l, v_{l+1}, \dots, u_{k-1}, v_k$ are distinct and every $i \in [l+1, k]$ satisfies (h0) and (f0), where $u_k = u_l$. Then by Lemma 4.2 (f1), the condition (F) holds with $(x, R)$ replaced by $(y, R')$. Note that $y$ is also written as $y = x + c(x, Q) \sum_{i=l+1}^{k} (\chi_{u_i} - \chi_{v_i})$, and hence Lemma 4.1 (h1) imply that (H) holds with $(x, D)$ replaced by $(y, D')$. By Lemma 4.3, (P) holds with $(x, D, R)$ replaced by $(y, D', R')$. □

**Lemma 5.8.** If (P), (H), and (F) hold when the procedure Augment$(\{u_0, v_1, \dots, v_{k-1}, u_{k-1}\})$ is called, then they also hold just after the procedure.

*Proof.* Let $P$ denote the sequence $\{u_0, v_1, \ldots, v_{k-1}, u_{k-1}\}$. The procedure updated the triple $(x, D, R)$ to $(y, D', R')$, where

$$\begin{aligned}
&y = x + c(x, P)\left(\chi_{u_0} + \textstyle\sum_{i=1}^{k-1}(\chi_{u_i} - \chi_{v_i})\right),\\
&D' = D \setminus \{v_1, v_2, \ldots, v_{k-1}\},\\
&R' = R \cup \{v_1, v_2, \ldots, v_{k-1}\}.
\end{aligned}$$

We now show that conditions (P), (H), and (F) hold with $(x, D, R)$ replaced by $(y, D', R')$. By Lemma 5.3, $u_0, v_1, \ldots, v_{k-1}, u_{k-1}$ are all distinct and there hold $u_0 \in D \backslash \mathrm{sat}_h(x)$, $u_{k-1} \notin \mathrm{sat}_f(x)$, and every $i \in [k-1]$ satisfies (h0) and (f0).

We first show (F). Define a vector

$$x' := x + c(x, P) \textstyle\sum_{i=1}^{k-1}(\chi_{u_{i-1}} - \chi_{v_i}) = y - c(x, P)\chi_{u_{k-1}}.$$

As we have (f*) for each $i \in [k-1]$, Lemma 4.2 (f1) implies that (F) holds for $(x', R')$. As $\{u_0, u_1, \ldots, u_{k-2}\} \subseteq \mathrm{sat}_f(x)$, Lemma 2.3 (b5) and $u_{k-1} \notin \mathrm{sat}_f(x)$ imply $\hat{c}_f(x', u_{k-1}) = \hat{c}_f(x, u_{k-1}) \geq c(x, P)$. Apply Lemma 2.4 to obtain $y = x' + c(x, P)\chi_{u_{k-1}}$. Then Lemma 2.4 (c3) implies that (F) also holds with $(x', R')$ replaced by $(y, R')$.

The condition (H) can be checked similarly. Define a vector

$$x'' := x + c(x, P) \textstyle\sum_{i=1}^{k-1}(\chi_{u_i} - \chi_{v_i}) = y - c(x, P)\chi_{u_0}.$$

As we have (h*) for each $i \in [k-1]$, Lemma 4.1 (h1) implies that (H) holds for $(x'', D')$. As $\{u_1, u_2, \ldots, u_{k-1}\} \subseteq D \setminus \{e^*\} \subseteq \mathrm{sat}_h(x)$, Lemma 2.3 (b5) and $u_0 \notin \mathrm{sat}_h(x)$ imply $\hat{c}_h(x', u_0) = \hat{c}_f(x, u_0) \geq c(x, P)$. Apply Lemma 2.4 to obtain $y = x'' + c(x, P)\chi_{u_0}$. Then Lemma 2.4 (c3) implies that (H) also holds with $(x'', D')$ replaced by $(y, D')$.

As for (P), Lemma 4.3 implies that it holds with $(x, D, R)$ replaced by $(x', D', R')$. Also, since $u_{k-1} \in D \subseteq D' \cup R'$ and $y = x' + c(x, P)\chi_{u_{k-1}}$, it also holds for $(y, D', R')$. □

Observation 5.4 and Lemmas 5.5–5.8 yield the correctness of the algorithm. Also, the definition of the algorithm implies that, if given functions are integer-valued functions, then $x$ is an integer vector at any time of the algorithm. Thus we obtain the following.

**Theorem 5.9.** At the termination of the algorithm, the output $x$ is a stable allocation. If $h$ and $f$ are integer-valued functions, then the output $x$ is an integer vector.

## 6 Complexity

We now consider the time complexity of the algorithm.

First, observe that each element never moves from $D$ to $R$ in the algorithm. Since the procedure Self-loop moves one element from $D$ to $R$, this yields the following fact.

**Lemma 6.1.** The procedure Self-loop is called at most $|E|$ times in the algorithm.

We say that an element $e \in E$ is *$h$-saturated* (resp. *$f$-saturated*) by a procedure, if we have $e \notin \mathrm{sat}_h(x)$ (resp., $e \notin \mathrm{sat}_f(x)$) just before the procedure and $e \in \mathrm{sat}_h(x)$ (resp., $e \in \mathrm{sat}_f(x)$) just after the procedure.

We say that, $\mathrm{next}_h(D, x, e)$ (resp., $\mathrm{next}_f(x, e)$) *increases* if it becomes larger in $\succ_{\mathrm{H}}$ (resp., in $\succ_{\mathrm{F}}$) by some update. Similarly, we say that $\mathrm{next}_h(D, x, e)$ (resp., $\mathrm{next}_f(x, e)$) *decreases* if it becomes smaller in $\succ_{\mathrm{H}}$ (resp., in $\succ_{\mathrm{F}}$). We also say that $\mathrm{next}_h(D, x, e)$ decreases when it turns undefined. Recall that $\mathrm{next}_h(D, x, e)$ may not be defined, while $\mathrm{next}_f(x, e)$ is always defined.

**Lemma 6.2.** In the algorithm, the following statements hold for every $e \in E$.

1. After $e$ is added to $R$, $e$ stays in $R$ and $\mathrm{next}_h(D, x, e)$ never increases. Also, if $\mathrm{next}_h(D, x, e)$ once turns undefined and turns defined again, it becomes smaller compared to the last time it is defined.
2. After $e$ is $f$-saturated, $e$ stays in $\mathrm{sat}_f(x)$ and $\mathrm{next}_f(x, e)$ never decreases.

*Proof.* We show Condition 1. For any $e \in E$, assume $e \in R$ at some point of the algorithm. Since $R$ is monotone increasing, $e \in R$ is maintained thereafter. By the algorithm, $\mathrm{next}_h(D, x, e)$ may change at Line 4 or when some procedure is applied.

At Line 4, $x$ does not change and the added element $e^*$ is smallest in $D \cup \{e^*\}$ w.r.t. $\succ_{\mathrm{H}}$. Then, if $\mathrm{next}_h(D, x, e)$ is defined just before Line 4, it remains the same. Otherwise, $\mathrm{next}_h(D, x, e)$ becomes $e^*$, which is smaller than previous values of $\mathrm{next}_h(D, x, e)$ w.r.t. $\succ_{\mathrm{H}}$.

When Self-loop is applied, $x$ does not change and $D$ becomes a subset of $D$. When Cycle (resp. Augment) is called, by Lemma 5.3, we can apply Lemma 4.1 (h2), and then $e \in R$ implies $e \neq u_i$ for each $i \in [0, k-1]$ (resp. for each $i \in [l, k-1]$). In each case, $\mathrm{next}_h(D, x, e)$ never increases, and it does not turn defined if it is undefined just before.

We now show Condition 2. Since $\mathrm{next}_f(x, e)$ is independent from $D$, it changes only when Cycle or Augment is applied. By Lemma 5.3 and Lemma 4.2 (f2), once $e \in \mathrm{sat}_f(x)$ holds, then $e \in \mathrm{sat}_f(x)$ remains to hold and $\mathrm{next}_f(x, e)$ never decreases until the end of the algorithm. □

**Lemma 6.3.** When Cycle($\{u_l, v_{l+1}, \ldots, u_{k-1}, v_k\}$) is applied in the algorithm, there exists $i \in [l+1, k]$ such that $\mathrm{next}_h(D, x, v_i)$ decreases or $\mathrm{next}_f(x, u_{i-1})$ increases.

*Proof.* Let $Q$ denote the sequence $\{u_l, v_{l+1}, \ldots, u_{k-1}, v_k\}$ and let $y$ denote the vector obtained from $x$ by the procedure. By Lemmas 4.1 (h3) and 4.2 (f3), for every $i \in [l+1, k]$, we have $\bar{c}_h(y, u_i, v_i) = \bar{c}_h(x, u_i, v_i) - c(x, Q)$ and $\bar{c}_f(y, u_{i-1}, v_i) = \bar{c}_f(x, u_{i-1}, v_i) - c(x, Q)$, where $u_k = u_l$. Also, by the definition of $c(x, Q)$, some $i$ satisfies either $\bar{c}_h(y, u_i, v_i) = 0$ or $\bar{c}_f(y, u_{i-1}, v_i) = 0$. By Lemmas 4.1 (h2) and 4.2 (f2), in the former case $\mathrm{next}_h(D, y, v_i)$ decreases, and in the latter case $\mathrm{next}_f(x, u_i)$ increases. □

**Lemma 6.4.** When Augment($\{u_0, v_1, \ldots, v_{k-1}, u_{k-1}\}$) is applied in the algorithm, at least one of the following hold: (i) $u_0 = e^*$ is $h$-saturated, (ii) $u_{k-1}$ is $f$-saturated, (iii) there is $i \in [k-1]$ such that $\mathrm{next}_h(D, x, v_i)$ decreases or $\mathrm{next}_f(x, u_{i-1})$ increases.

*Proof.* Let $P$ denote the sequence $\{u_0, v_1, \ldots, v_{k-1}, u_{k-1}\}$ and let $y$ denote the vector obtained from $x$ by the procedure. As shown in the proof of Lemma 5.8, $y$ is obtained by combining a multiple exchange and a simple augmentation. By Lemmas 4.1 (h3), 4.2 (f3), and 2.4 (c3), then $\hat{c}_h(y, u_0) = \hat{c}_h(x, u_0) - c(x, P)$ and $\hat{c}_f(y, u_{k-1}) = \hat{c}_h(x, u_{k-1}) - c(x, P)$, and every $i \in [k-1]$ satisfies $\bar{c}_h(y, u_i, v_i) = \bar{c}_h(x, u_i, v_i) - c(x, P)$ and $\bar{c}_f(y, u_{i-1}, v_i) = \bar{c}_f(x, u_{i-1}, v_i) - c(x, P)$. By the definition of $c(x, P)$, at least one of their values is 0. The condition $\hat{c}_h(y, u_0) = 0$ implies (i), i.e., $u_0$ is $h$-saturated. Similarly, $\hat{c}_f(y, u_{k-1}) = 0$ implies (ii), i.e., $u_{k-1}$ is $f$-saturated. Otherwise, we have (iii) since, by Lemmas 4.1 (h3) and 4.2 (f3), $\bar{c}_h(y, u_i, v_i) = 0$ implies that $\mathrm{next}_h(D, y, v_i)$ decreases and $\bar{c}_f(y, u_{i-1}, v_i) = 0$ implies that $\mathrm{next}_f(y, u_{i-1})$ increases. □

**Lemma 6.5.** Procedures Cycle and Augment are called at most $2(|E| + |E|^2)$ times in total.

*Proof.* By Lines 3–5 of the algorithm, each element is $h$-saturated at most once in the algorithm, and hence the case (i) of Lemma 6.4 occurs at most $|E|$ times. Also, the case (ii) occurs at most $|E|$ times since $\mathrm{sat}_f(x)$ is monotonically increasing by Lemma 6.2. Note that Cycle and Augment in the case (iii) makes either a decrease of $\mathrm{next}_h(D, x, e)$ with $e$ added to $R$ or an increase of $\mathrm{next}_f(x, e)$ preserving $e \in \mathrm{sat}_f(x)$. By Lemma 6.2, for each $e \in E$, such decreases or increases occur at most $|E|$ times, respectively. Therefore, the total number of applications of procedures Cycle and Augment is at most $2(|E| + |E|^2)$. □

As shown in Lemma 6.2, for each $e \in E$, after the first call of $\mathrm{next}_h(D, x, e)$, the value of $\mathrm{next}_h(D, x, e)$ is monotone nonincreasing in $\succ_{\mathrm{H}}$. Similarly, after the first call of $\mathrm{next}_f(x, e)$, the value of $\mathrm{next}_f(x, e)$ is monotone nondecreasing in $\succ_{\mathrm{F}}$. We exploit this property to compute the functions efficiently. Let the algorithm keep pointers $\mathrm{pt}_h(e)$ and $\mathrm{pt}_f(e)$ for each $e \in E$. Initially, $\mathrm{pt}_h(e)$ and $\mathrm{pt}_f(e)$ are set to be the maximum element in $\succ_{\mathrm{H}}$ and the minimum element in $\succ_{\mathrm{F}}$, respectively, for every $e \in E$. Each time the algorithm needs $\mathrm{next}_h(D, x, v)$, it proceeds as follows. If $\bar{c}_h(x, s, v) > 0$ and $s \in D \setminus \{v\}$ hold for $s = \mathrm{pt}_h(v)$, then $\mathrm{next}_h(D, x, v) = s$. Otherwise, the algorithm decrement $\mathrm{pt}_h(v)$ in $\succ_{\mathrm{H}}$ and iterates. The algorithm eventually achieves $\bar{c}_h(x, s, v) > 0$ and $s \in D \setminus \{v\}$ for $s = \mathrm{pt}_h(v)$, and then $\mathrm{next}_h(D, x, v) = s$. To find $\mathrm{next}_f(x, u)$, the algorithm repeatedly computes $\bar{c}_h(x, u, t)$ for $t = \mathrm{pt}_h(u)$, incrementing $\mathrm{pt}_h(u)$ in $\succ_{\mathrm{F}}$ in the case of $\bar{c}_h(x, u, t) = 0$. With the aid of these pointers, we obtain the following running time bound of our algorithm. Recall that $\gamma$ denotes the time for computing the saturation and exchange capacities on the given polymatroids.

**Theorem 6.6.** The algorithm finds a stable allocation in $O(|E|^3\gamma)$ time.

*Proof.* By Lemmas 6.1 and 6.5, the algorithm calls procedures Self-loop, Augment, and Cycle $O(|E|^2)$ times in total. Each of these procedures requires $O(|E|\gamma)$ time. To compute the pointer functions $\mathrm{next}_f$ and $\mathrm{next}_h$, the algorithm performs additional calls of exchange capacity computation. The total number of such calls during the entire algorithm is $O(|E|^2)$. Thus, the algorithm runs in $O(|E|^3\gamma)$ time. □

## 7 Optimality

For the stable marriage model of Gale and Shapley [15], it is known that the output of the man-oriented deferred acceptance algorithm is optimal for men among all stable matchings. Likewise, our algorithm finds the $\succ_{\mathrm{H}}$-optimal stable allocation, which is what we show in this section.

Recall that a vector $x \in \mathbf{R}^E$ is said to be $\succ$-preferable to $y \in \mathbf{R}^E$ if $x(E_{\succeq a}) \geq y(E_{\succeq a})$ for every $a \in E$. Also, $x$ is called $\succ$-optimal in a set $K \subseteq \mathbf{R}^E$ if $x \in K$ and $x$ is $\succ$-preferable to every $y \in K$. In this section, we show that the output of the algorithm is $\succ_{\mathrm{H}}$-optimal in the set of all the stable allocations. For this purpose, we show that the following condition is also maintained in the algorithm besides (P), (H), and (F).

**(R)** For each $e \in R$, $x(e) \geq z(e)$ holds for every stable allocation $z \in \mathbf{P}(h) \cap \mathbf{P}(f)$.

**Lemma 7.1.** If $(x, D, R)$ satisfies (R) at the end of the algorithm, then $x$ is $\succ_{\mathrm{H}}$-optimal in the set of all the stable allocations.

*Proof.* Take any stable allocation $z$. At the end of the algorithm, we have $D \cup R = E$ and $[e \in \mathrm{sat}_h(x),\ \mathrm{dep}_h(x, e) \succeq_{\mathrm{H}} e]$ holds for every $e \in D$ (as shown in the proof of Lemma 5.5). Then, by (R), every $e \in E$ satisfies $x(e) \geq z(e)$ or $[e \in \mathrm{sat}_h(x),\ \mathrm{dep}_h(x, e) \succeq_{\mathrm{H}} e]$. This implies that $x$ is $\succ_{\mathrm{H}}$-preferable to $z$ by Lemma 2.5. □

At the beginning of the algorithm, $R = \emptyset$ and hence the condition (R) clearly holds. What is left is to show that (R) is maintained throughout the algorithm. To do so, we prepare the following two lemmas.

**Lemma 7.2.** If (P), (H), and (R) hold for $(x, D, R)$, then any stable allocation $z$ satisfies

$$\forall e \in E: \quad z(e) \geq x(e) \quad \text{or} \quad [e \in \mathrm{sat}_f(z),\ \mathrm{dep}_f(z, e) \succeq_{\mathrm{F}} e]. \tag{18}$$

*Proof.* Take an arbitrary stable allocation $z \in \mathbf{P}(h) \cap \mathbf{P}(f)$. Fix $a \in E$ by $a := e^* = \min_{\succ_\mathrm{H}} D$ if $x(e^*) > z(e^*)$ and otherwise $a := \min_{\succ_\mathrm{H}} \{\, e \in E \mid e \succ_\mathrm{H} e^* \,\}$. By (P), then every $e \in E \setminus E_{\succeq_\mathrm{H} a}$ satisfies $z(e) \geq x(e)$. Then, it suffices to show that every $e \in E_{\succeq_\mathrm{H} a}$ satisfies the condition in (18). Since (P), (H), and (R) hold, we have

$$\forall e \in E_{\succeq_\mathrm{H} a}: \quad x(e) \geq z(e) \quad \text{or} \quad [e \in \mathrm{sat}_h(x),\ \mathrm{dep}_h(x, e) \succeq_\mathrm{H} e].$$

By Lemma 2.6 for $x, z \in \mathbf{P}(h)$, this implies

$$\forall e \in E_{\succeq_\mathrm{H} a}: \quad z(e) \geq x(e) \quad \text{or} \quad \neg[e \in \mathrm{sat}_h(z),\ \mathrm{dep}_h(z, e) \succeq_\mathrm{H} e].$$

Since $z$ is a stable allocation, $\neg[e \in \mathrm{sat}_h(z),\ \mathrm{dep}_h(z, e) \succeq_\mathrm{H} e]$ implies $[e \in \mathrm{sat}_f(z),\ \mathrm{dep}_f(z, e) \succeq_\mathrm{F} e]$. Thus, we obtain (18). □

For a sequence $W = \{u_0, v_1, u_1, \ldots, u_{d-1}, v_d, u_d\}$ of (not necessarily distinct) elements of $E$, we define its *walk-capacity* by

$$\check{c}(x, W) = \min \left\{ \hat{c}_h(x, u_0),\ \ \min_{i:1 \leq i \leq d} \bar{c}_f(x, u_{i-1}, v_i),\ \ \min_{i:1 \leq i \leq d} \bar{c}_h(x, u_i, v_i) \right\}.$$

**Lemma 7.3.** Suppose that (P), (H), (F) and (R) hold for $(x, D, R)$. Also, suppose that (not necessarily distinct) elements $u_0$, $u_i, v_i$ $(i = 1, 2, \ldots, d)$ satisfies $u_0 \in D$ and (h0) and (f0) for every $i \in [d]$. Then for any stable allocation $z$, we have $x(v_i) - \check{c}(x, W) \geq z(v_i)$ for each $i \in [d]$.

*Proof.* Take an arbitrary stable allocation $z$. Consider vectors

$$\begin{aligned} x_0 &:= x + \check{c}(x, W)\chi_{u_0}, \\ y_i &:= x + \check{c}(x, W)(\chi_{u_{i-1}} - \chi_{v_i}) \quad (i \in [d]), \\ x_i &:= x + \check{c}(x, W)(\chi_{u_i} - \chi_{v_i}) \quad (i \in [d]). \end{aligned}$$

Also, set $D_i := D \setminus \{v_i\}$ and $R_i := R \cup \{v_i\}$ for $i \in [d]$ and $D_0 := D$, $R_0 := R$. Note that for each $i \in [d]$, the condition $x(v_i) - \check{c}(x, W) \geq z(v_i)$ is equivalent to $y_i(v_i) \geq z(v_i)$. Hence, showing the following three statements completes the proof.

(i) (P), (H), and (R) hold for $(x_0, D_0, R_0)$.

(ii) For each $i \in [d]$, if (P), (H), and (R) hold for $(x_{i-1}, D_{i-1}, R_{i-1})$, then $y_i(v_i) \geq z(v_i)$.

(iii) For each $i \in [d]$, if $y_i(v_i) \geq z(v_i)$, then (P), (H), and (R) hold for $(x_i, D_i, R_i)$.

We first show (i). The conditions (P) and (R) are obvious by definition. Since $\check{c}(x, W) \leq \hat{c}_h(x, u_0)$, we can apply Lemma 2.4 (c3) to see that $e \in \mathrm{sat}_h(x_0)$ and $\mathrm{dep}_h(x_0, e) = \mathrm{dep}_h(x, e)$ for every $e \in D \setminus \{e^*\} \subseteq \mathrm{sat}_h(x)$.

We then show (ii). Since both $x$ and $x_{i-1}$ satisfy (P), (H) and (R), by Lemma 7.2, the condition (18) holds for both $x$ and $x_{i-1}$. Note that $y_i(e) \leq \max\{x(e), x_{i-1}(e)\}$ for every $e \in E$ by definition. Then, the condition $z(e) < y_i(e)$ implies either $z(e) < x(e)$ or $z(e) < x_{i-1}(e)$, each of which implies $[e \in \mathrm{sat}_f(z),\ \mathrm{dep}_f(z, e) \succeq_\mathrm{F} e]$ by (18). Therefore, we have

$$\forall e \in E: \quad z(e) \geq y_i(e) \quad \text{or} \quad [e \in \mathrm{sat}_f(z),\ \mathrm{dep}_f(z, e) \succeq_\mathrm{F} e].$$

By Lemma 2.6, this implies

$$\forall e \in E: \quad y_i(e) \geq z(e) \quad \text{or} \quad \neg[e \in \mathrm{sat}_f(y_i),\ \mathrm{dep}_f(y_i, e) \succeq_\mathrm{F} e]. \tag{19}$$

By $\check{c}(x, W) \leq \bar{c}_f(x, u_{i-1}, v_i)$ and (f0), Lemma 4.2 (f1) implies that the statement (F) holds with $(x, R)$ replaced by $(y_i, R \cup \{v_i\})$, which implies $[v_i \in \mathrm{sat}_f(y_i),\ \mathrm{dep}_f(y_i, v_i) \succeq_{\mathrm{F}} v_i]$. Then, (19) implies $y_i(v_i) \geq z(v_i)$.

Finally, we show (iii). As we have (h0) and (f0), it follows that $u_i, v_i \in D \cup R$, and hence (P) holds. By (h0), we can apply Lemma 4.1 (h1), and hence (H) holds for $(x_i, D_i)$. We now show (R). Since $y_i$ satisfies $y_i(v_i) \geq z(v_i)$, we have $x_i(v_i) = y_i(v_i) \geq z(v_i)$. Also, $x_i(e) \geq x(e)$ for every $e \in E \setminus \{v_i\}$. Thus, (R) holds for $(x_i, R_i)$. $\square$

We now show that each procedure keeps the conditions (P), (H), (F), and (R).

**Lemma 7.4.** If (P), (H), (F), and (R) hold when the procedure Self-loop$(u_{k-1})$ is called, then they also hold just after the procedure.

*Proof.* We only consider (R) because other conditions are already shown in Lemma 5.6. Since the procedure adds $u_{k-1}$ to $R$, it suffices to show $x(u_{k-1}) \geq z(u_{k-1})$. For a stable allocation $z$, we have (18) by Lemma 7.2. Apply Lemma 2.6 to $z, x \in \mathbf{P}(f)$ with respect to $(E, f, \succ_{\mathrm{F}})$. Then, each $e \in E$ satisfies $x(e) \geq z(e)$ or $\neg[e \in \mathrm{sat}_f(x),\ \mathrm{dep}_f(x, e) \succeq_{\mathrm{F}} e]$. Note that $[u_{k-1} \in \mathrm{sat}_f(x),\ \mathrm{dep}_f(x, u_{k-1}) \succeq_{\mathrm{F}} u_{k-1}]$ follows from $\mathrm{next}_f(x, u_{k-1}) = u_{k-1}$, and hence we have $x(u_{k-1}) \geq z(u_{k-1})$. $\square$

**Lemma 7.5.** If (P), (H), (F), and (R) hold when Augment$(\{u_0, v_1, \ldots, v_{k-1}, u_{k-1}\})$ is called, then they also hold just after the procedure.

*Proof.* We only consider (R) because other conditions are already shown in Lemma 5.8. Let $y$ denote the vector obtained from $x$ by the procedure. Since (R) is assumed and $R$ is updated to $R \cup \{v_1, v_2, \ldots, v_{k-1}\}$, it suffices to show $y(v_i) \geq z(v_i)$ for every $i \in [k-1]$, where $z$ is an arbitrary stable allocation.

Let $P$ denote $\{u_0, v_1, \ldots, v_{k-1}, u_{k-1}\}$. By Lemma 5.3, we can apply Lemma 7.3 to $x$ and $P$, and obtain $x(v_i) - \check{c}(x, P) \geq z(v_i)$ for every $i \in [k-1]$. By the definitions of path and walk capacities, we have $c(x, P) \leq \check{c}(x, P)$. Then $y(v_i) = x(v_i) - c(x, P) \geq x(v_i) - \check{c}(x, P) \geq z(v_i)$ for every $i \in [k-1]$. $\square$

**Lemma 7.6.** If (P), (H), (F), and (R) hold when the procedure Cycle$(\{u_l, v_{l+1}, \ldots, u_{k-1}, v_k\})$ is called, then they also hold just after the procedure.

*Proof.* Let $Q$ denote the sequence $\{u_l, v_{l+1}, \ldots, u_{k-1}, v_k\}$ and define $\chi_Q := \sum_{i=l+1}^{k} (\chi_{u_{i-1}} - \chi_{v_i})$. The procedure Cycle for $Q$ updates the vector $x$ to $y := x + c(x, Q)\chi_Q$. Let $W$ denote the sequence $\{u_0, v_1, \ldots, u_{k-1}, v_k, u_k\}$ and set vectors $x_0 := x$ and $x_m := x_{m-1} + \check{c}(x_{m-1}, W)\chi_Q$ for every $m \in \mathbf{Z}_{>0}$. Then

$$x_m = x + \textstyle\sum_{j=0}^{m-1} \check{c}(x_j, W)\chi_Q \qquad (m \in \mathbf{Z}_{>0}).$$

We will show finite convergence of $\{x_m\}_{m \in \mathbf{Z}_{\geq 0}}$ to $y$.

Note that $u_k = u_l$ when Cycle is called, and recall the definitions of capacities:

$$c(x_m, Q) = \min\left\{ \min_{i: l+1 \leq i \leq k} \bar{c}_f(x_m, u_{i-1}, v_i),\ \min_{i: l+1 \leq i \leq k} \bar{c}_h(x_m, u_i, v_i) \right\},$$

$$\check{c}(x_m, W) = \min\left\{ \hat{c}_h(x_m, u_0),\ \min_{i: 1 \leq i \leq k} \bar{c}_f(x_m, u_{i-1}, v_i),\ \min_{i: 1 \leq i \leq k} \bar{c}_h(x_m, u_i, v_i) \right\}.$$

By these definitions, for every $m \in \mathbf{Z}_{\geq 0}$, we have

$$\check{c}(x_m, W) \leq c(x_m, Q). \tag{20}$$

We now show the following conditions for every $m \in \mathbf{Z}_{>0}$. Here, $D'$ and $R'$ are defined as $D' = D \setminus \{v_{l+1}, v_{l+2}, \ldots, v_k\}$ and $R' = R \cup \{v_{l+1}, v_{l+2}, \ldots, v_k\}$, respectively.

(i) $(x_m, D', R')$ satisfies (P), (H), (F), and (R).

(ii) If $x_m \neq y$, then $\check{c}(x_m, W) > 0$, $u_0 \in D'$, and $(x_m, D')$ satisfies (h0), (f0) for every $i \in [k]$. If $x_m = y$, then $\check{c}(x_m, W) = 0$, and hence $x_{m+1} = x_m$.

(iii) $c(x_m, Q) = c(x, Q) - \sum_{j=0}^{m-1} \check{c}(x_j, W)$.

(iv) $\check{c}(x_m, W) = c(x_m, Q)$ or $\check{c}(x_m, W) \geq \check{c}(x_{m-1}, W)$, where we put $\check{c}(x_{-1}, W) = 0$.

We show (i)–(iv) by simultaneous induction on $m \geq 0$.

Consider the case that $m = 0$. Clearly (iii) and (iv) hold. Note that $x_0$ satisfies conditions (i) and (ii) with $D$, $R$ replaced by $D'$, $R'$. By applying Lemmas 4.1 (h1), 4.2 (f1), and 4.3 with $\alpha = 0$, we see that $(x_0, D', R')$ still satisfies (P), (H), (F) and (ii). Also, (R) for $(x_0, D', R')$ follows from Lemma 7.3. Thus, all of (i)–(iv) hold in the case that $m = 0$.

We now turn to the case that $m \geq 1$. We first show (i). By the inductive assumption, $(x_{m-1}, D', R')$ satisfies (P), (H), (F), and (R). By applying Lemma 7.3 to $x_{m-1}$, we obtain $x_{m-1}(v_i) - \check{c}(x_{m-1}, W) \geq z(v_i)$ for any stable allocation $z$ and $i \in [l+1, k]$. Since $x_m(v_i) = x_{m-1}(v_i) - \check{c}(x_{m-1}, W)$, this means that (R) holds for $x_m$ and $R'$. Also, since (20) holds for $x_{m-1}$, we can apply Lemmas 4.1 (h1), 4.2 (f1), and 4.3, which respectively imply (H), (F), and (P) for $(x_m, D', R')$. Thus, (i) holds for $m$. To show conditions (ii)–(iv), let us observe the saturation and exchange capacities of $x_m$.

1. By Lemma 2.3 (b5) and (b6), the saturation capacity $\hat{c}_h(\cdot, u_0)$ satisfies the following.

(1-1) If $u_0 \neq u_k$, then $\hat{c}_h(x_m, u_0) = \hat{c}_h(x_{m-1}, u_0)$.

(1-2) If $u_0 = u_k$, then $\hat{c}_h(x_m, u_0) \geq \min\{\hat{c}_h(x_{m-1}, u_0), \bar{c}_h(x_m, u_k, v_k)\}$.

2. By Lemma 4.2 (f3), exchange capacity $\bar{c}_f$ satisfies the following.

(2-1) If $1 \leq i < l$, then $\bar{c}_f(x_m, u_{i-1}, v_i) = \bar{c}_f(x_{m-1}, u_{i-1}, v_i)$

(2-2) If $v_l \neq v_k$, then $\bar{c}_f(x_m, u_{l-1}, v_l) = \bar{c}_f(x_{m-1}, u_{l-1}, v_l)$.

(2-3) If $v_l = v_k$, then $\bar{c}_f(x_m, u_{l-1}, v_l) = \bar{c}_f(x_m, u_{l-1}, v_k) \geq \min\{\bar{c}_f(x_{m-1}, u_{l-1}, v_l), \bar{c}_f(x_m, u_{k-1}, v_k)\}$.

(2-4) If $l + 1 \leq i \leq k$, then $\bar{c}_f(x_m, u_{i-1}, v_i) = \bar{c}_f(x_{m-1}, u_{i-1}, v_i) - \check{c}(x_{m-1}, W)$.

3. By Lemma 4.1 (h3), exchange capacity $\bar{c}_h$ satisfies the following. Note that $u_k = u_l$.

(3-1) If $1 \leq i < l$, then $\bar{c}_h(x_m, u_i, v_i) = \bar{c}_h(x_{m-1}, u_i, v_i)$

(3-2) If $v_l \neq v_k$, then $\bar{c}_h(x_m, u_l, v_l) = \bar{c}_h(x_m, u_k, v_l) \geq \min\{\bar{c}_h(x_{m-1}, u_{l-1}, v_l), \bar{c}_h(x_m, u_k, v_k)\}$.

(3-3) If $v_l = v_k$, then $\bar{c}_h(x_m, u_l, v_l) = \bar{c}_h(x_m, u_k, v_k)$.

(3-4) If $l + 1 \leq i \leq k$, then $\bar{c}_h(x_m, u_i, v_i) = \bar{c}_h(x_{m-1}, u_i, v_i) - \check{c}(x_{m-1}, W)$.

Using these, we first show (iii). By (2-4) and (3-4), every $i \in [l+1, k]$ satisfies $\bar{c}_f(x_m, u_{i-1}, v_i) = \bar{c}_f(x_{m-1}, u_{i-1}, v_i) - \check{c}(x_{m-1}, W)$ and $\bar{c}_h(x_m, u_i, v_i) = \bar{c}_h(x_{m-1}, u_i, v_i) - \check{c}(x_{m-1}, W)$. Hence,

$$c(x_m, Q) = c(x_{m-1}, Q) - \check{c}(x_{m-1}, W) = c(x, Q) - \textstyle\sum_{j=0}^{m-1} \check{c}(x_j, W),$$

where the last equality is obtained by substituting $c(x_{m-1}, Q) = c(x, Q) - \sum_{j=0}^{m-2} \check{c}(x_j, W)$, which follows from the inductive assumption. Thus, (iii) holds.

We next show (iv). By the definitions of $\check{c}(x_{m-1}, W)$ and $c(x_m, Q)$, we have $\hat{c}_h(x_{m-1}, u_0) \geq \check{c}(x_{m-1}, W)$ and $\bar{c}_h(x_m, u_k, v_k) \geq c(x_m, Q)$. Then, (1-1) and (1-2) above imply

$$\hat{c}_h(x_m, u_0) \geq \min\{\check{c}(x_{m-1}, W), c(x_m, Q)\}. \tag{21}$$

Also, we have $\bar{c}_f(x_{m-1}, u_{i-1}, v_i) \geq \check{c}(x_{m-1}, W)$ for every $i \in [k]$ and $\bar{c}_f(x_m, u_{j-1}, v_j) \geq c(x_m, Q)$ for every $j \in [l+1, k]$. Then, the conditions (2-1)–(2-3) imply

$$\bar{c}_f(x_m, u_{i-1}, v_i) \geq \min\{\check{c}(x_{m-1}, W), c(x_m, Q)\} \quad (i \in [k]). \tag{22}$$

Similarly, the conditions (3-1)–(3-3) imply

$$\bar{c}_h(x_m, u_i, v_i) \geq \min\{\check{c}(x_{m-1}, W), c(x_m, Q)\} \quad (i \in [k]). \tag{23}$$

By (21), (22) and (23), we obtain $\check{c}(x_m, W) \geq \min\{\check{c}(x_{m-1}, W), c(x_m, Q)\}$, i.e., $\check{c}(x_m, W) \geq \check{c}(x_{m-1}, W)$ or $\check{c}(x_m, W) \geq c(x_m, Q)$. As we have (20), this implies (iv).

Finally, we show (ii). We first consider the case that $x_m \neq y$. Since (ii) holds for $m-1$ by the inductive assumption, $x_m \neq y$ implies $x_{m-1} \neq y$ and $\check{c}(x_{m-1}, W) > 0$. Also, $x_m \neq y$ implies $\sum_{j=0}^{m-1} \check{c}(x_j, W) \neq c(x, Q)$, which implies $c(x_m, Q) > 0$ by (iii). Thus, we have $\min\{\check{c}(x_{m-1}, W), c(x_m, Q)\} > 0$. By (23), every $i \in [k]$ satisfies $\bar{c}_h(x_m, u_i, v_i) > 0$. By the definition of $\mathrm{next}_h$ and its monotonicity (Lemma 4.1 (h2)), this implies $\mathrm{next}_f(D', x_m, v_i) = u_i$. Similarly, by (22) and Lemma 4.2 (f2), we obtain $\mathrm{next}_f(x_m, u_{i-1}) = v_i$ for every $i \in [k]$. Thus, we have (h0) and (f0) for every $i \in [k]$. Also, $u_0 \in D'$ is clear and $\check{c}(x_m, W) \geq \min\{\check{c}(x_{m-1}, W), c(x_m, Q)\} > 0$. Thus, all requirements in (ii) hold when $x_m \neq y$. In the case that $x_m = y$, we have $\sum_{j=0}^{m-1} \check{c}(x_j, W) = c(x, Q)$, which implies $c(x_m, Q) = 0$ by (iii). As we have (20), this implies $\check{c}(x_m, W) = 0$. Hence, $x_{m+1} = x_m + \check{c}(x_m, W)\chi_Q = x_m$.

So far, we have shown that $x_m$ satisfies (i)–(iv) for every $m \in \mathbf{Z}_{\geq 0}$. By (i), it completes the proof to show that there is a finite $m^* \in \mathbf{Z}_{\geq 0}$ such that $x_{m^*} = y$. Let $m^* \in \mathbf{Z}_{\geq 0}$ be the minimum number satisfying

$$m^* \cdot \check{c}(x, W) > c(x, Q).$$

Since $\check{c}(x, W) > 0$ by the definition of the algorithm, $m^*$ is finite. To prove $x_{m^*} = y$ by contradiction, suppose that $x_{m^*} \neq y$. By the second claim of (ii), this implies that $x_m \neq y$ for every $m \in [m^*]$. Then, for every $m \in [m^*]$, we have $\sum_{j=0}^{m-1} \check{c}(x_j, W) \neq c(x, Q)$, which implies $\check{c}(x_{m-1}, W) \neq c(x, Q) - \sum_{j=0}^{m-2} \check{c}(x_j, W) = c(x_{m-1}, Q)$, where the last equality follows from (iii). Therefore, $\check{c}(x_m, W) \neq c(x_m, Q)$ for every $m \in [m^* - 1]$. As we have (iv), this implies

$$\check{c}(x_0, W) \leq \check{c}(x_1, W) \leq \check{c}(x_2, W) \leq \cdots \leq \check{c}(x_{m^*-1}, W).$$

Then (iii) for $m^*$ implies

$$c(x_{m^*}, Q) = c(x, Q) - \textstyle\sum_{j=0}^{m^*-1} \check{c}(x_j, W) \leq c(x, Q) - m^* \cdot \check{c}(x_0, W) < 0,$$

which contradicts the nonnegativity of $c(x_{m^*}, Q)$. □

We now obtain the $\succ_{\mathrm{H}}$-optimality of the output of the algorithm.

**Theorem 7.7.** The output of the algorithm is the stable allocation that is $\succ_{\mathrm{H}}$-preferable to any stable allocation $z$. That is, the output is $\succ_{\mathrm{H}}$-optimal in the set of all the stable allocations.

*Proof.* At the beginning of the algorithm, $R = \emptyset$ and hence (R) holds. By Observation 5.4 and Lemmas 7.4–7.6, the output of the algorithm satisfies (P), (H), (F), and (R). Then, Lemma 7.1 implies the $\succ_{\mathrm{H}}$-optimality of the output. □

## Acknowledgments

This work is supported by CREST, Japan Science and Technology Agency. The first author is supported by MEXT Grant-in-Aid for Scientific Research on Innovative Areas (No. 24106005). The second author is supported by JSPS Research Fellowship for Young Scientists.

## References

[1] A. Alkan and D. Gale: Stable schedule matching under revealed preference, *Journal of Economic Theory*, **112** (2003), pp. 289–306.

[2] M. Baïou and M. Balinski: Erratum: The stable allocation (or ordinal transportation) problem, *Mathematics of Operations Research*, **27** (2002), pp. 662–680.

[3] B. C. Dean, M. X. Goemans, and N. Immorlica: Finite termination of "augmenting path" algorithms in the presence of irrational problem data, *Proc. 14th ESA,* Lecture Notes in Computer Science **4168**, Springer-Verlag, Berlin & Heidelberg, 2006, pp. 268–279.

[4] B. C. Dean and S. Munshi: Faster algorithms for stable allocation problems, *Algorithmica*, **58** (2010), pp. 59–81.

[5] J. Edmonds: Submodular functions, matroids, and certain polyhedra, *Combinatorial Structures and Their Applications* (R. Guy, H. Hanani, N. Sauer, and J. Schönheim, eds.), Gordon and Breach, New York, 1970, pp. 69–87: Also in: *Combinatorial Optimization—Eureka, You Shrink!* (M. Jünger, G. Reinelt, and G. Rinaldi, eds.), Lecture Notes in Computer Science **2570**, Springer-Verlag, Berlin, 2003, pp. 11–26.

[6] A. Eguchi, S. Fujishige, and A. Tamura: A generalized Gale-Shapley algorithm for a discrete-concave stable-marriage model, *Proc. 14th ISAAC,* Lecture Notes in Computer Science **2906**, Springer-Verlag, Berlin & Heidelberg, 2003, pp. 495–504.

[7] T. Fleiner: A matroid generalization of the stable matching polytope, *Proc. Eighth IPCO,* Lecture Notes in Computer Science **2081**, Springer-Verlag, Berlin & Heidelberg, 2001, pp. 105–114.

[8] T. Fleiner: A fixed-point approach to stable matchings and some applications, *Mathematics of Operations Research*, **28** (2003), pp. 103–126.

[9] S. Fujishige: Algorithms for solving independent-flow problems, *Journal of the Operations Research Society of Japan*, **21** (1978), pp. 189–204.

[10] S. Fujishige: Polymatroidal dependence structure of a set of random variables, *Information and Control*, **39** (1978), pp. 55–72.

[11] S. Fujishige: *Submodular Functions and Optimization* (2nd ed.), Elsevier, Amsterdam, 2005.

[12] S. Fujishige and A. Tamura: A general two-sided matching market with discrete concave utility functions, *Discrete Applied Mathematics*, **154** (2006), pp. 950–970.

[13] S. Fujishige and A. Tamura: A two-sided discrete-concave market with possibly bounded side payments: An approach by discrete convex analysis, *Mathematics of Operations Research*, **32** (2007), pp. 136–155.

[14] S. Fujishige and X. Zhang: New algorithms for the intersection problem of submodular systems, *Japan Journal of Industrial and Applied Mathematics*, **9** (1992), pp. 369–382.

[15] D. Gale and L. S. Shapley: College admissions and the stability of marriage, *American Mathematical Monthly*, **69** (1962), pp. 9–15.

[16] M. Grötschel, L. Lovász, and A. Schrijver: *Geometric Algorithms and Combinatorial Optimization* (2nd ed.), Springer-Verlag, Berlin, 1993.

[17] S. Iwata, L. Fleischer, and S. Fujishige: A combinatorial strongly polynomial algorithm for minimizing submodular functions, *Journal of the ACM*, **48** (2001), pp. 761–777.

[18] D. Kőnig: Graphok és matrixok (Hungarian; Graphs and matrices), *Matematikai és Fizikai Lapok*, **38** (1931), pp. 116–119.

[19] Y. T. Lee, A. Sidford, and S. C.-W. Wong: A faster cutting plane method and its implications for combinatorial and convex optimization, *Proc. 56th FOCS*, IEEE, 2015, pp. 1049–1065.

[20] K. Murota: *Discrete Convex Analysis*, SIAM, Philadelphia, 2003.

[21] K. Murota and Y. Yokoi: On the lattice structure of stable allocations in two-sided discrete-concave market, *Mathematics of Operations Research*, **40** (2015), pp. 460–473.

[22] J. B. Orlin: A faster strongly polynomial time algorithm for submodular function minimization, *Mathematical Programming*, **118** (2009), pp. 237–251.

[23] P. Schönsleben: *Ganzzahlige Polymatroid Intersektions Algorithmen*, Ph.D. Thesis, ETH Zürich, 1980.

[24] A. Schrijver: A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *Journal of Combinatorial Theory, Series B*, **80** (2000), pp. 346–355.

[25] A. Schrijver: *Combinatorial Optimization: Polyhedra and Efficiency*, Springer-Verlag, Heidelberg, 2003.

[26] É. Tardos, C. A. Tovey, and M. A. Trick: Layered augmenting path algorithms, *Mathematics of Operations Research*, **11** (1986), pp. 362–370.

[27] Y. Yokoi: A generalized polymatroid approach to stable matchings with lower quotas, *Mathematics of Operations Research,* to appear.