Scarf's Algorithm on Arborescence Hypergraphs

Karthekeyan Chandrasekaran
Grainger College of Engineering, University of Illinois, Urbana-Champaign, IL, USA

Yuri Faenza
Columbia University, New York, NY, USA

Chengyue He
Columbia University, New York, NY, USA

Jay Sethuraman
Columbia University, New York, NY, USA

Abstract

Scarf's algorithm – a pivoting procedure that finds a dominating extreme point in a down-monotone polytope – can be used to show the existence of a fractional stable matching in hypergraphs. The problem of finding a fractional stable matching in hypergraphs, however, is PPAD-complete. In this work, we study the behavior of Scarf's algorithm on arborescence hypergraphs, the family of hypergraphs in which hyperedges correspond to the paths of an arborescence. For arborescence hypergraphs, we prove that Scarf's algorithm can be implemented to find an integral stable matching in polynomial time. En route to our result, we uncover novel structural properties of bases and pivots for the more general family of network hypergraphs. Our work provides the first proof of polynomial-time convergence of Scarf's algorithm on hypergraphic stable matching problems, giving hope to the possibility of polynomial-time convergence of Scarf's algorithm for other families of polytopes.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Scarf's algorithm, Arborescence Hypergraphs, Stable Matchings

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.45

Category Track A: Algorithms, Complexity and Games

Related Version *Scarf's Algorithm on Arborescence Hypergraphs*: https://arxiv.org/abs/2412.03397 [11]

Funding *Karthekeyan Chandrasekaran*: Karthekeyan Chandrasekaran is supported in part by NSF grant CCF-2402667.
Yuri Faenza: Yuri Faenza is supported by NSF Grant 2046146 and by a Meta Research Award.
Chengyue He: Chengyue He is supported by NSF Grant 2046146 and by a Meta Research Award.

Acknowledgements The first three authors gratefully acknowledge the organizers of the ICERM Spring 2023 semester program on *Discrete Optimization: Mathematics, Algorithms, and Computation*, where this project was initiated.

1 Introduction

Scarf [32] proved the existence of a core allocation in a large class of cooperative games with non-transferable utility. Key ingredients in this proof include a lemma – *Scarf's lemma* – that asserts the existence of a dominating extreme point in certain polytopes, and a pivoting procedure – *Scarf's algorithm* – to find one. Scarf's results have profoundly influenced subsequent research in combinatorics, theoretical computer science, economics, and game theory: Scarf's lemma has been used to show the existence of fair allocations such as cores and fractional cores [8, 32], strong fractional kernels [2], fractional stable

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).
Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis
Article No. 45; pp. 45:1–45:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

solutions in hypergraphs [1] and fractional stable paths [20]. Although Scarf's results are employed in many applications, it is not known how to efficiently construct these desirable allocations/solutions. It is widely believed that Scarf's algorithm is unlikely to be efficient for arbitrary applications [22]. In this work, we investigate the possibility of efficient convergence of Scarf's algorithm for finding a stable matching in hypergraphs.

The stable matching problem in bipartite graphs (also known as the stable marriage problem) is a classic and well-studied problem. It is well-known that a stable matching in a bipartite graph with preferences always exists [19] and can be found in polynomial time via multiple algorithms, including Gale and Shapley's deferred acceptance algorithm [19], linear programming [4, 30, 31, 35] and other combinatorial algorithms [5, 15]. A stable matching in a bipartite graph can also be found using fixed-point approaches [17], which in general do not translate to polynomial-time algorithms. Scarf's algorithm belongs to the class of fixed point approaches, and it has been recently shown to converge in polynomial time for the stable marriage problem [16]. To the best of our knowledge, this is the first proof of polynomial-time convergence of Scarf's algorithm on any application. This result motivated us to address the possibility of efficient convergence of Scarf's algorithm for hypergraph stable matching problems.

In contrast to graphs, the problem of finding a stable matching in hypergraphs is significantly more challenging. Even in the special case of tripartite 3-regular hypergraphs, it is NP-complete to find a stable matching [24] (this is also known as the *stable family problem* proposed by Knuth [23]). Despite the hardness results of finding (integral) stable matchings in hypergraphs, Aharoni and Fleiner [1] used Scarf's lemma to show that a fractional stable matching exists in every hypergraph. However, this general implication from Scarf's result comes at a computational price: the problem of computing a fractional stable matching on hypergraphs is PPAD-complete [22]. The latter problem remains PPAD-complete even in extremely restrictive cases, for example, even if the hypergraph has node degree/hyperedge size at most 3 [21, 13].

While these results suggest that Scarf's algorithm is unlikely to converge in polynomial-time for arbitrary hypergraphs, investigating the behavior of Scarf's algorithm in special classes of hypergraphs remains an interesting question, also because of real-world applications. For instance, a sequence of works [26, 27, 25] employed Scarf's algorithm as a subroutine to find a fractional stable matching in hypergraphs, then iteratively rounds it to an integral solution which indicates a meaningful allocation. Such allocations provide solutions when complementarities and externalities appear in stable matching problems [29]. One of the most important such problems is the hospital/resident problem with couples (HRC) [10, 26]. HRC can equivalently be formulated as the problem of finding a stable matching in certain hypergraphs. It is shown in [9] that Scarf's algorithm empirically outperforms all other known heuristics when the proportion of couples is high. However, finding a fractional solution of HRC is also known to be PPAD-complete [13]. Such computational gaps between theory and practice regarding Scarf's algorithm are not well understood.

In this paper, we focus on a class of hypergraphs called *arborescence* hypergraphs (in which the hyperedges are paths in an arborescence), and design a polynomial-time algorithm to find a (integer) stable matching in this class. The class of arborescence hypergraph lies between hypergraphs represented by interval matrices [18] and network matrices [33]. In particular, we design a pivoting rule for Scarf's algorithm and show that it terminates in a polynomial number of iterations on the fractional matching polytope of arborescence hypergraphs. The node-hyperedge incidence matrix of an arborescence hypergraph is totally unimodular and consequently, every extreme point of the associated fractional matching polytope is integral.

Thus, every dominating extreme point is also integral. This observation and Aharoni and Fleiner's results [1] imply that the extreme point found by Scarf's algorithm for arborescence hypergraphs corresponds to an *integral* stable matching. Our pivoting rule can be repurposed to apply to the more general family of *network hypergraphs* (of which the incidence matrix is a network matrix[1]). This pivoting rule can also be repurposed to implement the simplex algorithm on a linear program whose constraint matrix is a network matrix, which includes the incidence matrix of digraphs [34, chapter 13.3]; thus, it reduces to a pivoting rule of choosing the leaving arc for the network simplex algorithm proposed by Cunningham [14]. If we do not impose any restrictions on the entering variables, network simplex algorithm under this rule can still perform an exponentially long sequence of degenerate pivots [3]. In contrast, we show that Scarf's algorithm under our pivoting rule always performs non-degenerate pivots. Currently, we are not able to obtain a proof of polynomial convergence of Scarf's algorithm implemented with our pivoting rule for the more general family of network hypergraphs. Nevertheless, understanding Scarf's algorithm on arborescence hypergraphs is a natural first step in this direction.

1.1 Our Contributions

To formally present our result, we define the stable matching problem on hypergraphs as follows.

▶ **Definition 1** (Hypergraphic preference system). *A hypergraph $H = (V, E)$ is defined by a vertex set V and a hyperedge set E, where every hyperedge $e \in E$ is a subset of V. A* hypergraphic preference system *is given by a pair $(H, \succ)$, where $H = (V, E)$ is a hypergraph and $\succ := \{\succ_i : i \in V\}$ is the preference profile, $\succ_i$ being a strict order over $\delta(i) = \{e \in E : i \in e\}$ for each $i \in V$. For $e, e' \in \delta(i)$, we write $e \succeq_i e'$ if either $e \succ_i e'$ or $e = e'$. In addition, we assume that (i) for every $i \in V$, the* singleton hyperedge *$e_i = \{i\} \in \delta(i)$ and for every $e' \in \delta(i)$, $e' \succeq_i e_i$ and (ii) $\{1\}$ is the only hyperedge incident to node 1, that is, $\delta(1) = \{\{1\}\}$*[2]*.*

▶ **Definition 2** (Stable matching). *A* stable matching *for a hypergraphic preference system is a vector $x \in \{0, 1\}^E$ so that for every $e \in E$, there exists a vertex $i \in e$ such that*

$$\sum_{e' \in \delta(i), e' \succeq_i e} x_{e'} = 1. \tag{1}$$

Equation (1) with $e = e_i$ imposes that x is the characteristic vector of a matching. Moreover, for every hyperedge e, there exists a vertex $i \in e$ and a hyperedge e' in the matching (i.e., $x_{e'} = 1$) so that $e' \succeq_i e$. Correspondingly, if a fractional vector $x \in [0, 1]^E$ satisfies (1), then we call such a fractional vector x a fractional stable matching*.*

We remark here that the standard combinatorial definition of stable matching is a hypergraph matching that does not contain blocking hyperedges (see, e.g., [1]), which reduces to the classical stable matching when the hypergraph is a bipartite graph. Definition 2 is equivalent to the combinatorial definition, and helps us define the fractional generalization more naturally.

[1] The node-hyperedge incidence matrix of such a hypergraph is also totally unimodular [34] and consequently there is always an integer stable matching.

[2] The first assumption corresponds, in the bipartite setting, to the usual hypothesis that an agent prefers to be matched rather than being unmatched. The second assumption is without loss of generality and we include it for the sake of simplifying the presentation.

▶ **Definition 3** (Arborescence hypergraph). *An* arborescence *is a directed graph that contains a vertex r (called the* root*) such that every vertex $v \neq r$ has a unique directed path from r. A hypergraph $H = (V, E)$ is an* arborescence hypergraph *if there exists an arborescence $\mathcal{T} = (U, \mathcal{A}_0)$ such that $V = \mathcal{A}_0$ and each hyperedge $e \in E$ is a subset of arcs in $\mathcal{A}_0$ that forms a directed path. Such $\mathcal{T}$ is called a* principal arborescence *of H.*

▶ **Example 4.** We present an example of an arborescence and the corresponding arborescence hypergraph in Figure 1.

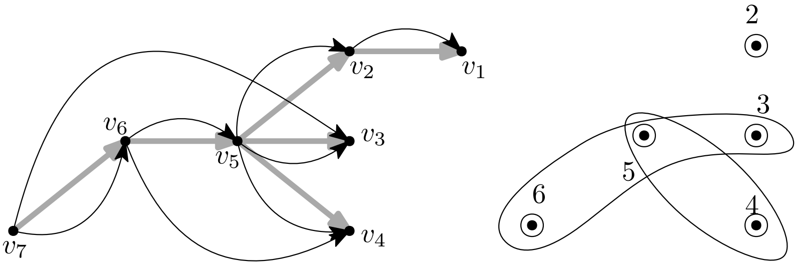

Figure 1 On the left we present an arborescence $\mathcal{T}$ with the root v_7 using gray arcs. On the right we have a hypergraph H with principal arborescence $\mathcal{T}$, $V = [6]$ and 8 hyperedges (6 of them are singletons). Each circle in the hypergraph is a singleton hyperedge. Thus, each node in H corresponds to a gray arc in $\mathcal{T}$, and each hyperedge in H corresponds to a black arc in $\mathcal{T}$.

As our main result, we show that Scarf's algorithm can be implemented to run in polynomial time for every arborescence hypergraphic preference system.

▶ **Theorem 5.** *Let $(H = (V, E), \succ)$ be a hypergraphic preference system where H is an arborescence hypergraph. There exists a pivoting rule such that Scarf's algorithm terminates in at most $|V|$ iterations and outputs a stable matching on $(H = (V, E), \succ)$ in time $O(|V||E|)$.*

A few remarks on this result are in order. To the best of our knowledge, this is the first result showing polynomiality of Scarf's algorithm beyond the case of stable marriage [16]. We note that the pivoting rule used to show polynomiality of Scarf's algorithm in stable marriage [16] was heavily inspired by Gale and Shapley's classical deferred acceptance mechanism, which is a purely combinatorial algorithm. In contrast, the pivoting rule developed in this work does not seem to have a combinatorial counterpart and is substantively different from the one from [16]. In Section 6, we provide evidence suggesting that classical approaches to stable marriage problems, such as linear programming with an exact description of the convex hull of all stable matchings [35], cannot even extend to a subclass of arborescence hypergraphic preference systems.

Secondly, while most of the known results related to polynomiality of stable matching on hypergraphic preference system usually restricts either the degree of nodes [21] or the size of hyperedges [13], we do not assume any condition on them or on the preference lists of agents. Thirdly, a recent work [7] shows that there is a polynomial time algorithm for finding a stable matching on *subtree hypergraphs*. They reduce the problem of finding a stable matching on such instances to finding a *kernel* in the clique-acyclic superorientation of a chordal graph, which can be solved in polynomial time [28]. This reduction indeed implies that there exists a polynomial time algorithm that finds a stable matching in an arboresence hypergraphic system. We remark that the algorithm from [28] is substantially different from Scarf's algorithm and does not generalize to network hypergraphs, while, in our investigation, we uncover novel properties of bases and pivots in network hypergraphs, which may be of independent interest and could lead to polynomial-time convergence proofs for this more general class of hypergraphs.

1.2 Organization of the paper

In Section 2, we give more details on Scarf's results and the origin of arborescence hypergraphs, and formally define notation. Scarf's algorithm can be seen as alternating between two operations, cardinal pivot and ordinal pivot. In Section 3, we discuss the cardinal pivots, while in Section 4, we introduce ordinal pivots. In Section 5, we present the key technical part of the proof of polynomiality of Scarf's algorithm on arborescence hypergraphs. In Section 6, we present an (counter)example that shows non-integrality of the fractional stable matching polytope on interval hypergraphs. Lastly, we give conclusion with open questions in Section 7. Some proofs, generalizations, and extended discussions are omitted and can be found in the full version [11].

2 Preliminaries

While the original idea of Scarf's algorithm [32] applies to more general settings, we review here only its implementation for hypergraphic preference systems, as proposed by Aharoni and Fleiner [1]. In the following, we let $(H = (V, E), \succ)$ be a hypergraphic preference system with $V = [n]$ and $E = \{e_1, \ldots, e_m\}$.

2.1 Scarf's lemma and hypergraphic stable matchings

▶ **Definition 6** (Fractional matching polytope, cardinal basis). *Let $A = (a_{i,j}) \in \{0,1\}^{n\times m}$ be the* incidence matrix *of H such that $a_{i,j} = 1$ iff $e_j \in \delta(i)$. The matrix A is in* standard form *if A has the form $A = (I_n|A')$ where I_n is the $n \times n$ identity matrix*[3]*. The polytope $\mathcal{P} = \{x \in \mathbb{R}^m_{\geq 0} : Ax = \mathbb{1}\}$ is called the* fractional matching polytope*. A set $B \subset [m]$ of n linearly independent columns of A is called a* cardinal basis*. Every feasible cardinal basis corresponds to an* extreme point *$x \in \mathcal{P}$.*

We note that, if x is a fractional stable matching, then $x \in \mathcal{P}$ (i.e., x is a fractional matching). Conversely, given a fractional matching $x \in \mathcal{P}$, one needs more conditions that involve the preference orders to claim that x is a fractional stable matching, which leads to the next definition.

▶ **Definition 7** (Ordinal matrix, ordinal basis, utility vector). *Let $C = (c_{i,j}) \in \mathbb{R}^{n\times m}$ be a matrix. The matrix C is an* ordinal matrix *if it satisfies $c_{i,i} < c_{i,k} < c_{i,j}$ for every distinct $i, j \in [n]$, $k \in [m] \setminus [n]$. Let O be a set of columns of C. For every row $i \in [n]$, the utility of the row i (w.r.t. O) is*

$$u_i^O := \min_{j\in O} c_{i,j}. \tag{2}$$

The set O is called an ordinal basis *of C if $|O| = n$ and for every column $j \in [m]$, there is at least one row $i \in [n]$ such that $u_i^O \geq c_{i,j}$. The associated vector $u^O \in \mathbb{R}^n$ is called the* utility vector *of the ordinal basis O.*

[3] This means $e_i = \{i\}$ for $i \in [n]$, i.e., the first n columns of A correspond to singletons.

▶ **Example 8.** The matrix C below is an ordinal matrix. One can verify that $O_1 = \{\mathit{2}, \mathit{3}, \mathit{5}\}$ is an ordinal basis with $u^{O_1} = (3,0,0)^T$, and $O_2 = \{\mathit{4}, \mathit{5}, \mathit{6}\}$ is an ordinal basis with $u^{O_2} = (1,1,1)^T$. We observe that $O_3 = \{\mathit{1}, \mathit{2}, \mathit{6}\}$ with $u^{O_3} = (0,0,2)^T$ is not an ordinal basis since $c_{i,\mathit{4}} > u_i^{O_3}$ for every $i \in [3]$.

$$C = \begin{array}{c} \begin{array}{cccccc} \mathit{1} & \mathit{2} & \mathit{3} & \mathit{4} & \mathit{5} & \mathit{6} \end{array} \\ \left(\begin{array}{ccc|ccc} 0 & 5 & 4 & 2 & 3 & 1 \\ 5 & 0 & 4 & 1 & 2 & 3 \\ 5 & 4 & 0 & 3 & 1 & 2 \end{array}\right) \end{array}.$$

▶ **Definition 9** (Dominating basis, dominating extreme point)**.** *A feasible cardinal basis B that is also an ordinal basis is called a* dominating basis *for (A, C), and the extreme point of $\mathcal{P}$ corresponding to B is called a* dominating extreme point *for (A, C).*

Scarf's lemma shows the existence of a dominating extreme point under mild conditions on (A, C). In the following, we state a version of it that pertains to our setting, which connects dominating extreme points and fractional stable matchings.

▶ **Theorem 10.** *Let $(H = (V, E), \succ)$ be a hypergraphic preference system. Let $\mathcal{P} = \{x \in \mathbb{R}^m_{\geq 0} : Ax = \mathbb{1}\}$ be the fractional matching polytope of H where A is in standard form. Then, there exists an ordinal matrix $C \in \mathbb{R}^{n \times m}$ such that*

1. *(Scarf's lemma [32]) A dominating extreme point for (A, C) exists.*
2. *([1]) Every dominating extreme point for (A, C) is a fractional stable matching for $(H = (V, E), \succ)$.*

2.2 Scarf's algorithm

Scarf [32] provided a general pivoting algorithm to compute a dominating extreme point. The algorithm involves two operations, namely cardinal pivot and ordinal pivot.

2.2.1 Cardinal Pivot

Let $B = \{j_1, \ldots, j_n\}$ be a feasible cardinal basis and let $j_t \in [m] \setminus B$. A *cardinal pivot* from B with *entering column* j_t returns a feasible cardinal basis $B' := B \cup \{j_t\} \setminus \{j_\ell\}$, where the column $j_\ell = B - B'$ is known as the leaving column. Cardinal pivot is exactly the operation used in the *simplex algorithm* when a column j_t enters the basis B and one wants to choose a leaving column j_ℓ to reach an adjacent basis. When $\mathcal{P}$ is degenerate, in each iteration there may exist multiple candidates to be chosen as a leaving column j_ℓ, thus a pivoting rule under which column is leaving needs to be specified. For a fixed pivoting rule, the choice of leaving column j_ℓ is unique. Unlike the simplex algorithm, the other half of the pivoting procedure and the pivoting rule generating the entering column is obtained via an operation on the ordinal matrix.

2.2.2 Ordinal Pivot

Let O be an ordinal basis and $j_\ell \in O$. An *ordinal pivot* from O with *leaving column* j_ℓ returns an ordinal basis $O' := O \cup \{j^*\} \setminus \{j_\ell\}$, where $j^* \in [m] - O$ is chosen by a series of number comparisons. We give the formal description in Algorithm 1, and provide an example to illustrate this opertaion in Example 11. For correctness of Algorithm 1, we refer to [32].

Algorithm 1 Ordinal Pivot.

```
Let $O$ be an ordinal basis with $u^O$ being its utility vector. Suppose that $j_\ell \in O$ is the leaving column.
$i_\ell \leftarrow$ the unique row $i \in [n]$ for which $u^O_i = c_{i,j_\ell}$.
$j_r \leftarrow \arg\min_{j \in O - j_\ell}\{c_{i_\ell,j}\}$.                                  ▷ Reference column
$i_r \leftarrow$ the unique row $i \in [n]$ for which $u^O_i = c_{i,j_r}$.
$K \leftarrow \{k \in [m] \setminus O : c_{i,k} > u^{O-j_\ell}_i$, for all $i \neq i_r\}$.
$j^* \leftarrow \arg\max_{k \in K} c_{i_r,k}$.                                               ▷ Entering column
$O \leftarrow O \cup \{j^*\} \setminus \{j_\ell\}$.
```

▶ **Example 11.** In this example, we show how an ordinal pivot proceeds. Consider the ordinal matrix C in Example 8, and its submatrix C_O:

$$C = \begin{array}{c} \begin{array}{cccccc} \mathit{1} & \mathit{2} & \mathit{3} & \mathit{4} & \mathit{5} & \mathit{6} \end{array} \\ \begin{pmatrix} 0 & 5 & 4 & \vert & 2 & 3 & 1 \\ 5 & 0 & 4 & \vert & 1 & 2 & 3 \\ 5 & 4 & 0 & \vert & 3 & 1 & 2 \end{pmatrix} \end{array}, \qquad C_O = \begin{array}{c} \begin{array}{ccc} \mathit{2} & \mathit{3} & \mathit{5} \end{array} \\ \begin{pmatrix} 5 & 4 & 3 \\ 0 & 4 & 2 \\ 4 & 0 & 1 \end{pmatrix} \end{array}$$

corresponding to an ordinal basis $O = \{\mathit{2}, \mathit{3}, \mathit{5}\}$ with utility vector $u^O = (3, 0, 0)^T$. If we want to remove column $\mathit{2}$ from O, and perform an ordinal pivot, we let $j_\ell = \mathit{2}$. The row i_ℓ is defined as the row index of the common entry in both the utility vector u^O and the column vector $C_{\mathit{2}} = (5, 0, 4)^T$, thus we have $i_\ell = 2$, since $u^O_2 = c_{2,\mathit{2}} = 0$. In the remaining columns $O - j_\ell = \{\mathit{3}, \mathit{5}\}$, the minimum entry in row $i_\ell = 2$ is $c_{2,\mathit{5}} = 2$, thus we define the reference column $j_r = \mathit{5}$. Similarly, by finding the common entry between u^O and $C_{\mathit{5}} = (3, 2, 1)^T$, we find the row index $i_r = 1$ as $u^O_1 = c_{1,\mathit{5}} = 3$. Now, the goal is to find the columns k in $[\mathit{6}] \setminus O$ such that $c_{i,k} > u^{\{\mathit{3},\mathit{5}\}}_i$ for all $i \neq 1$, i.e., to find columns that dominates the vector $(*, 2, 0)^T$, where $*$ means we do not have any restrictions. One can observe that such columns can be $C_{\mathit{1}} = (0, 5, 5)^T$ or $C_{\mathit{6}} = (1, 3, 2)^T$. Thus, in Algorithm 1, $K = \{\mathit{1}, \mathit{6}\}$. We need to choose one column k from K that maximizes $c_{i_r,k}$, i.e., the row $i_r = 1$ is maximized. Therefore, we have $j^* = \mathit{6}$ as $c_{1,\mathit{6}} > c_{1,\mathit{1}}$. This ordinal pivot returns a new ordinal basis $O' = \{\mathit{3}, \mathit{5}, \mathit{6}\}$ with the new utility vector $u' = (1, 2, 0)^T$.

In contrast to the cardinal pivot, the ordinal pivot reverses the order of entering and leaving. Moreover, the ordinal pivot operation is unique.

▶ **Lemma 12** ([32]). *Let O be an ordinal basis and let $j_\ell \in O$ be an arbitrary column. Then there exists a unique column $j^* \notin O$ such that $O \cup \{j^*\} \setminus \{j_\ell\}$ is an ordinal basis, and such a j^* is determined by Algorithm 1.*

In the following, we will see that an iteration of Scarf's algorithm is the combination of a cardinal pivot and an ordinal pivot (in order). Recall that the simplex algorithm also chooses an entering column/variable and a leaving column/variable in order, but there a pivoting rule $\mathcal{R}$ is usually specified by a pair $(\mathcal{R}^{enter}, \mathcal{R}^{leave})$ [6] and these two rules should cooperate in some way to avoid cycling. In comparison, by Lemma 12, Scarf's algorithm defines a unique rule $\mathcal{R}^{enter}$ when the ordinal matrix C is fixed[4], and we can only control $\mathcal{R}^{leave}$ on cardinal pivots. This observation distinguishes the pivoting rule of Scarf's algorithm from it of the simplex algorithm.

[4] One can of course design a specific ordinal matrix $\bar{C}$ and a cardinal pivoting rule $\mathcal{R}^{leave}$ to let $(\bar{C}, \mathcal{R}^{leave})$ cooperate and see if Scarf's algorithm converges in polynomial time. However, we need to also care about the dominating basis obtained by Scarf's algorithm with $\bar{C}$ – if such $\bar{C}$ does not respect to the preference order given by certain applications, such dominating basis does not have a meaning.

2.2.3 Initialization, Iteration, and Termination

Scarf's algorithm is initialized with a pair (B_0, O_0), where $B_0 = \{1, 2, \ldots, n\}$ and $O_0 = \{2, 3, \ldots, n, n+1\}$. The algorithm iteratively starts with a pair (B, O) where $1 \in B$ is a feasible cardinal basis, $1 \notin O$ is an ordinal basis, and $|B \cap O| = n - 1$. It alternates a cardinal pivot and an ordinal pivot. In particular, it goes through a sequence

$$(B_0, O_0) \to (B_1, O_0) \to (B_1, O_1) \to \cdots, \tag{3}$$

where for every $i \geq 0$, we have that $(B_i, O_i) \to (B_{i+1}, O_i)$ is a cardinal pivot from the feasible cardinal basis B_i with entering column $j_t = O_i - B_i$ using the cardinal pivoting rule, and $(B_{i+1}, O_i) \to (B_{i+1}, O_{i+1})$ is an ordinal pivot from the ordinal basis O_i with leaving column $j_\ell = B_{i+1} - O_i$.

▶ **Definition 13** (Scarf pair, iteration). *A pair (B_i, O_i) in the sequence* (3) *with $B_i \neq O_i$ is called a* Scarf pair. *We denote a consecutive pair of cardinal pivot and ordinal pivot $(B_i, O_i) \to (B_{i+1}, O_i) \to (B_{i+1}, O_{i+1})$ as an* iteration.

The algorithm terminates if either (i) $1 = B_i - O_i$ happens to leave B_i, then $B_{i+1} = O_i$; or (ii) $1 = B_{i+1} - O_i$ happens to enter O_{i+1}, in which case $O_{i+1} = B_{i+1}$. Otherwise, the algorithm starts the next iteration with a new Scarf pair (B_{i+1}, O_{i+1}). Scarf [32] showed that when the polytope $\mathcal{P}$ is non-degenerate, sequence (3) does not cycle. Thus, the sequence stops at a basis $B = O$, both cardinal and ordinal, thus dominating. A dominating extreme point x can be deduced from B.

2.3 Arborescence hypergraphs

To describe our implementation of Scarf's algorithm, it will be useful to describe arborescence hypergraphs through digraphs. Let $\mathcal{D} = (U, \mathcal{A})$ be a directed graph. Let $F \subseteq \mathcal{A}$, and $P = (a_1, a_2, \ldots, a_p)$ be an ordering of $F \subseteq \mathcal{A}$.

- We say F (or P) is a *$\mathcal{D}$-path* (or *path*) if P forms an undirected path in $\mathcal{D}$.
- An arc $a_i \in F$ is *forward* in P if following the order P we visit the tail of a_i before visiting its head, otherwise a_i is *backward* in P.
- We say F (or P) is a *$\mathcal{D}$-directed path* if every arc in P is a forward arc in P.

Recall that an arborescence $\mathcal{T}$ is a directed graph with a root r such that for every vertex $v \neq r$ there is a unique $\mathcal{T}$-directed path from r.

Suppose that an arborescence $\mathcal{T} = (U, \mathcal{A}_0)$ is a subgraph of $\mathcal{D} = (U, \mathcal{A})$. Then, for every $a = (v, v') \in \mathcal{A}$, there exists a unique $\mathcal{T}$-directed path $P(v, v')$ from v to v'. An *arc-path incidence matrix* of $(\mathcal{D}, \mathcal{T})$ is defined as $M = (m_{i,j}) \in \{0,1\}^{\mathcal{A}_0 \times \mathcal{A}}$ such that $m_{i,j} = 1$ iff the j-th arc in $\mathcal{A}$ is (v, v'), and the unique $\mathcal{T}$-directed path $P(v, v')$ from v to v' passes the i-th arc in $\mathcal{A}_0$.

▶ **Definition 14** (Arborescence Hypergraph). *A hypergraph $H = (V, E)$ is an* arborescence hypergraph *if there exists a directed graph $\mathcal{D} = (U, \mathcal{A})$ with an arborescence subgraph $\mathcal{T} = (U, \mathcal{A}_0)$ such that the node-hyperedge incidence matrix of H is the arc-path incidence matrix of $(\mathcal{D}, \mathcal{T})$. If H is an arborescence hypergraph, we say that $\mathcal{D} = (U, \mathcal{A})$ is the principal network of H and $\mathcal{T} = (U, \mathcal{A}_0)$ is the principal arborescence of H.*

A subclass of the arborescence hypergraphs is the *interval hypergraphs* [18].

▶ **Definition 15** (Interval Hypergraph). *Let $H = (V, E)$ be an arborescence hypergraph with principal arborescence $\mathcal{T} = (U, \mathcal{A}_0)$. If $\mathcal{T}$ is a* chain, *then H is called an interval hypergraph.*

Let $H = (V, E)$ be an arborescence hypergraph with principal network $\mathcal{D} = (U, \mathcal{A})$, where $U = \{v_1, v_2, \ldots, v_{n+1}\}$, $\mathcal{A} = \{a_1, \ldots, a_m\}$. Its principal arborescence $\mathcal{T} = (U, \mathcal{A}_0)$ has the form $\mathcal{A}_0 = \{f_1, \ldots, f_n\}$. Notice that $\mathcal{A}_0 \subset \mathcal{A}$ and hence, we assume $a_i = f_i$ for $i \in [n]$. The node-hyperedge incidence matrix $A = (a_{i,j}) \in \{0,1\}^{n\times m}$ of H is equivalently defined as:

- $a_{i,j} = 1$ iff the hyperedge e_j contains node i.
- $a_{i,j} = 1$ iff the $\mathcal{T}$-path formed by a_j passes f_i.

We also assume that $\mathcal{T}$ is *depth-first*, defined as follows. This order is a common topological order on trees. Details can be found in [12].

▶ **Definition 16.** *Let $\mathcal{T} = (U, \mathcal{A}_0)$ be an arborescence with root r. We define a partial order $\geq_{\mathcal{T}}$ over U such that $v \geq_{\mathcal{T}} v'$ if the $\mathcal{T}$-directed path from r to v' passes through v. We say that $\mathcal{T}$ is* depth-first *if*

1. *$U = \{v_1, \ldots, v_{n+1}\}$, and for $i, j \in [n+1]$, $v_i \geq_{\mathcal{T}} v_j$ implies $i \geq j$. In particular, v_{n+1} is the unique root of $\mathcal{T}$.*
2. *$\mathcal{A}_0 = \{f_1, \ldots, f_n\}$, and for every $i \in [n]$, the head of f_i is v_i.*

3 Cardinal pivot on arborescence hypergraphs

In this section, we describe cardinal pivots of Scarf's algorithm as operations on principal network $\mathcal{D} = (U, \mathcal{A})$. In Section 3.1, we first show a structural result that maps the cardinal basis to directed trees on $\mathcal{D}$, then we translate the change of basis operation in cardinal pivot into a cycle elimination operation. In Section 3.2, we define a pivoting rule called *first-forward-leaving rule*, which is interpreted as a choice of arcs that can be removed to form a new tree (basis).

3.1 Cardinal pivot in a digraph

We use the notation in Definition 6, where H is specified as an arborescence hypergraph with principal network $\mathcal{D} = (U, \mathcal{A})$ and principal arborescence $\mathcal{T} = (U, \mathcal{A}_0)$. We first observe that there is a bijection between cardinal bases (not necessarily feasible) of $\mathcal{P}$ and directed trees on $\mathcal{D}$. We note that, the results present in this section can be generalized to the case when A is a network matrix.

▶ **Theorem 17** ([33]). *Consider a subset $B \subset [m]$ of n columns. Denote the corresponding arc set by $\mathcal{A}_B$. Then the following statements are equivalent:*

1. *B is a cardinal basis.*
2. *$\mathcal{T}_B = (U, \mathcal{A}_B)$ is a directed tree.*

Next, we give a combinatorial interpretation of the cardinal pivot. Let B be a feasible cardinal basis with $j_t \in [m] \setminus B$. By Theorem 17, they correspond to a directed tree $\mathcal{T}_B = (U, \mathcal{A}_B)$ and an arc $a_{j_t} \in \mathcal{A}$. Let $a_{j_t} = (v, v')$, then we can find the unique $\mathcal{T}_B$-path denoted by $P_B(v, v')$. The path $P_B(v, v')$ along with the additional arc a_{j_t} form a cycle, from which a leaving column j_ℓ is chosen, since $\mathcal{T}_{B'} = (U, \mathcal{A}_{B'})$ has to be a directed tree where $B' = B \cup \{j_t\} \setminus \{j_\ell\}$. Recall that we also need to take feasibility of B' into consideration, since eliminating an arbitrary arc from the above cycle only guarantees that B' is a cardinal basis. By computing the dynamics of how each variable is changing between a cardinal pivot, we give a graphic characterization of cardinal pivot in Lemma 19. For this, we need the notion of augmenting and descending paths.

▶ **Definition 18** (Augmenting and Descending Paths). *Let $\mathcal{D} = (U, \mathcal{A})$, $x \in \mathbb{R}^{\mathcal{A}}$, and $P = (a_{j_1}, \dots, a_{j_p})$ be a $\mathcal{D}$-path.*

1. *The path P is* x-augmenting *if every forward arc a_{fwd} in P has $x_{a_{fwd}} = 1$ and every backward arc a_{bwd} in P has $x_{a_{bwd}} = 0$.*
2. *The path P is* x-descending *if every forward arc a_{fwd} in P has $x_{a_{fwd}} = 0$ and every backward arc a_{bwd} in P has $x_{a_{bwd}} = 1$.*

▶ **Lemma 19.** *Consider a cardinal pivot from B to B' with j_t as the entering column where $a_{j_t} = (v, v') \in \mathcal{A}$. Let x, x' be the extreme points corresponding to B, B', respectively. Let $P_B(v, v')$ be the unique $\mathcal{T}_B$-path from v to v' (unique by Theorem 17). The following statements are equivalent:*

1. *The cardinal pivot is non-degenerate.*
2. *$x_{j_t} = 0$ and $x'_{j_t} = 1$.*
3. *$P_B(v, v')$ is x-augmenting.*
4. *$P_B(v, v')$ is x'-descending.*

Moreover, if the cardinal pivot is non-degenerate, then $B' = B \cup \{j_t\} \setminus \{j_\ell\}$ is a feasible cardinal basis, where j_ℓ corresponds to an (arbitrary) forward arc in $P_B(v, v')$. If the cardinal pivot is degenerate, then $B' = B \cup \{j_t\} \setminus \{j_\ell\}$ is a feasible cardinal basis, where j_ℓ corresponds to an (arbitrary) forward arc in $P_B(v, v')$ such that $x_{j_\ell} = 0$.

3.2 Cardinal pivoting rule

While Lemma 19 gives a characterization of the candidate leaving columns j_ℓ, often the choice of j_ℓ that satisfies the theorem is not unique. We will use the following *first forward arc leaving (FFL) rule* as the pivoting rule. Roughly speaking, among all the candidate arcs verified by Lemma 19, we want to eliminate the first arc in $P_B(v, v')$.

▶ **Definition 20** (First forward arc leaving rule). *Let B be a feasible cardinal basis. Let $a_{j_t} = (v, v')$ be the arc corresponding to the entering column and $P_B(v, v') = (\bar{a}_1, \cdots, \bar{a}_p)$ be the unique $\mathcal{T}_B$-path from v to v'. If there exists a forward arc $\bar{a}_k$ in $P_B(v, v')$ with $x_{\bar{a}_k} = 0$, choose the one with smallest subscript k as the arc corresponding to the leaving column. If all forward arcs $\bar{a}_{fwd}$ in $P_B(v, v')$ have $x_{\bar{a}_{fwd}} = 1$, then let k be the smallest index such that $\bar{a}_k$ is forward on $P_B(v, v')$, and let $\bar{a}_k$ be the arc corresponding to the leaving column.*

The FFL rule is well-defined, meaning that, for every entering column $j_t \notin B$, the leaving column defined above exists and is unique. We summarize the cardinal pivot operation with FFL rule in Algorithm 2.

4 The design of ordinal matrix and ordinal pivot

Unlike cardinal pivots, an ordinal pivot is uniquely defined by the ordinal matrix. However, we do have some degree of flexibility[5] to design an ordinal matrix C that satisfies Theorem 10. In Section 4.1, we construct a *block partitioned* ordinal matrix C which we use to prove the polynomiality of Scarf's algorithm. Then, we discuss some properties of the ordinal pivots under C in Section 4.2. Throughout this section, let $(H = (V, E), \succ)$ be a hypergraphic preference system with $V = [n]$ and $e_i = \{i\}$ for $i \in [n]$ being the singleton hyperedges[6].

[5] The ordinal matrix that makes Theorem 10 valid may not be uniquely defined. Fixing one of them is in a sense the same as specifying an ordinal pivoting rule.

[6] This section assumes no restriction on H. In particular, H can be any hypergraph.

Algorithm 2 Cardinal Pivot with FFL Rule.

```
Let $B$ be the current feasible cardinal basis, associated with basic feasible solution $x$.
Suppose that $j_t \notin B$ is the entering column.
Find $a_{j_t} = (v, v')$ and the $\mathcal{T}_B$-path $P_B(v, v') = (\bar{a}_1, \dots, \bar{a}_p)$
$I \leftarrow \{i \in [p], \bar{a}_i$ is forward and $x_{\bar{a}_i} = 0\}$
$J \leftarrow \{i \in [p], \bar{a}_i$ is forward$\}$
if $I \neq \emptyset$ then
    $k \leftarrow \min\{i : i \in I\}$
else
    $k \leftarrow \min\{i : i \in J\}$
end if
Let the leaving column $j_\ell$ be such that $a_{j_\ell} = \bar{a}_k$.
$B \leftarrow B \cup \{j_t\} \setminus \{j_\ell\}$
```

4.1 Block structure of the ordinal matrix

In our implementation of Scarf's algorithm, we design a block-partitioned ordinal matrix, defined as follows.

▶ **Definition 21.** *We say that an ordinal matrix C is* block-partitioned *if:*

1. *For each $i \in V$, we define the i-th block $S_i := \{e \in E \setminus \{e_i\} : i = \max_{j \in e} j\}$.*
2. *Let C be a $V \times E$ matrix whose columns are ordered as follows: The columns corresponding to the singleton hyperedges are the first n columns of C. Next, we group the other hyperedges into n blocks $S_1, \dots, S_n$, with the i-th block S_i ordered in decreasing order of preference from left to right according to $\succ_i$. The i-th block appears to the left of the $i+1$-th block for every $i \in [n-1]$.*
3. *Denote by $E = \{e_1, \dots, e_n, e_{n+1}, \dots, e_m\}$, where e_j corresponds to the j-th column of C defined in part 2.*
4. *Assign the entries in C as follows. For each $i \in [n], j \in [m]$ such that $e_j \in \delta(i)$, we set $c_{i,j} = |\delta(i)| - \ell$ if e_j is the ℓ-th best hyperedge with respect to $\succ_i$. The remaining entries in row i are assigned integers that are no less than $|\delta(i)|$ and in decreasing order from left to right.*

▶ **Example 22** (Block-partitioned matrix). Let $H = (V, E)$ be a hypergraph with $V = \{1, 2, 3, 4\}$ in Figure 2 and the following preference list:

$$
\begin{aligned}
&1 : \{1,3\} \succ_1 \{1,3,4\} \succ_1 \{1\},\\
&2 : \{2,3\} \succ_2 \{2\},\\
&3 : \{2,3\} \succ_3 \{1,3\} \succ_3 \{3,4\} \succ_3 \{1,3,4\} \succ_3 \{3\},\\
&4 : \{1,3,4\} \succ_4 \{3,4\} \succ_4 \{4\}.
\end{aligned}
$$

The blocks are $S_1 = \emptyset, S_2 = \emptyset, S_3 = \{\{2,3\}, \{1,3\}\}, S_4 = \{\{1,3,4\}, \{3,4\}\}$. The block-partitioned incidence matrix and ordinal matrix are as follows:

$$
A = \begin{array}{c|cccccccc}
 & \{1\} & \{2\} & \{3\} & \{4\} & \{2,3\} & \{1,3\} & \{1,3,4\} & \{3,4\} \\
\hline
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
3 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
4 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1
\end{array},
$$

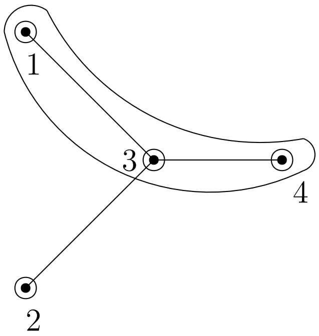

Figure 2 The hypergraph in Example 22. The circles around the nodes are singleton hyperedges. Line segments represent hyperedges with cardinality 2. Other hyperedges (only $\{1,3,4\}$ in this example) are indicated by splinegons.

$$C = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{cccccccc} \{1\} & \{2\} & \{3\} & \{4\} & \{2,3\} & \{1,3\} & \{1,3,4\} & \{3,4\} \end{array} \atop \begin{pmatrix} 0 & 7 & 6 & 5 & 4 & 2 & 1 & 3 \\ 7 & 0 & 6 & 5 & 1 & 4 & 3 & 2 \\ 7 & 6 & 0 & 5 & 4 & 3 & 1 & 2 \\ 7 & 6 & 5 & 0 & 4 & 3 & 2 & 1 \end{pmatrix}.$$

We remark that given a hypergraphic preference system, a block-partitioned matrix is unique and satisfies Theorem 10.

4.2 Controlling node and separator

4.2.1 Controlling node

When implementing Scarf's algorithm, the first row/column of A, C can be seen as a controlling row/column in Scarf's algorithm (see [32]). Recall that we assume $\delta(1) = \{e_1\}$, which means node 1 is incident only to the singleton hyperedge by itself. This assumption is without loss of generality: Indeed, we can reduce an arbitrary hypergraphic preference system to the one that satisfies this assumption, and construct a bijection between the stable matchings in the two instances. We say node 1 is the *controlling node.*

4.2.2 Separator

At each iteration of Scarf's algorithm, we start with a Scarf pair (B, O), where B is a feasible cardinal basis and O is an ordinal basis. Though not explicitly stated, B indicates a fractional matching in the hypergraph. The intuition behind the block structure introduced in the previous section is that we want to separate the nodes that have already been "matched" by B from the singletons. A separator is the node with the largest index that belongs to the former set, and every node behind the separator does not participate in the matching. We define the separator formally below.

▶ **Definition 23.** *Let (B, O) be a Scarf pair. Let $j^{\rightarrow} := \max\{j : j \in O\}$ be the column with the largest index in O. If $j^{\rightarrow}$ belongs to the i-th block S_i, then we say that i is the* separator *of (B, O).*

We show that the separator satisfies the following property, which captures the idea that every node larger than the separator is matched by the singleton:

▶ **Lemma 24.** *Let (B, O) be a Scarf pair and suppose i is the separator of (B, O). Then, for every $i \leq i' \leq n$, we have $i' \in B \cap O$.*

We focus on a type of iteration of Scarf's algorithm that increases the separator, which is triggered by a condition described as follows:

▶ **Lemma 25.** *Let $(B, O) \to (B', O) \to (B', O')$ be an iteration in Scarf's algorithm. Let $j_\ell \in O$ be the leaving column in the ordinal pivot $O \to O'$. Suppose that j_r and j^* are the reference column and the entering column of the ordinal pivot $O \to O'$ respectively. Let i be the separator of (B, O). Denote by $u^O, u^{O'}$ the utility vector of O, O', respectively. If $u^O_i = c_{i,j_\ell}$, then we have*

1. *$j_r = \max\{j : j \in O\}$.*
2. *$u^O_1 = c_{1,j_r}$ and $u^{O'}_1 = c_{1,j^*}$.*
3. *The separator of (B', O') is i' for some $i' > i$.*

5 Polynomiality of Scarf's algorithm

In this section, we show convergence and bound the run-time of our implementation of Scarf's algorithm on arborescence hypergraphic preference systems. We are given a hypergraphic preference system $(H = (V, E), \succ)$ as the input, where H is an arborescence hypergraph and $\delta(1) = \{\{1\}\}$. Next, we find the principal arborescence $\mathcal{T} = (U, \mathcal{A}_0)$ and principal network $\mathcal{D} = (U, \mathcal{A})$ associated with H. Then, we order $U = \{v_1, \ldots, v_{n+1}\}$ and $\mathcal{A}_0 = \{f_1, \ldots, f_n\}$ according to depth-first order on $\mathcal{T}$ (consequently, the root of the arborescence $\mathcal{T}$ is v_{n+1}). Next, we construct the block-partitioned ordinal matrix C with an order of the hyperedges $E = \{e_1, \ldots, e_m\}$, and construct the node-hyperedge incidence matrix A with the rows and columns ordered as above.

We start with an initial Scarf pair (B_0, O_0) where $B_0 = \{1, \ldots, n\}$ and $O_0 = \{2, \ldots, n+1\}$. Every cardinal pivot follows the FFL rule (c.f. Definition 20). Every iteration of the algorithm is unambiguously defined and we obtain a unique sequence of Scarf pairs (3). We will show that the number of iterations, i.e. the length of the sequence (3), is bounded by $n = |V|$. Since each iteration can be implemented in $O(|E|)$ time, the total run-time is $O(|V||E|)$.

We define a notion of a well-structured basis and will inductively show that all cardinal bases visited by the algorithm satisfy it. This structure of the cardinal basis is helpful in bounding the number of iterations of the algorithm.

▶ **Definition 26.** *Let B be a feasible cardinal basis and $\mathcal{T}_B = (U, \mathcal{A}_B)$ be the directed tree corresponding to B. Let x be the extreme point associated with B. Let $i \in [n]$. We say that B is an i-*nice basis *if the following properties hold:*

1. *Let $v \in U \setminus \{v_{n+1}\}$, and $P_B(v_{n+1}, v)$ be the $\mathcal{T}_B$-path from v_{n+1} to v. Then, $P_B(v_{n+1}, v)$ is x-augmenting.*
2. *$\{f_i, f_{i+1}, \ldots, f_n\} \subset \mathcal{A}_B$.*
3. *Let $f \in \{f_i, f_{i+1}, \ldots, f_n\}$. Let $\mathcal{T}^{-f} = (U, \mathcal{A}_0 - \{f\})$ be the subgraph of $\mathcal{T}$ with exactly two connected components (in the undirected sense). Denote by $U = R \cup W$ the partition of vertices into two components such that $v_{n+1} \in R$ and $v_{n+1} \notin W$. If an arc $a \in \mathcal{A}_B$ has its end vertices in R and W, then $a = f$. In other words, the removal of each arc $f \in \{f_i, f_{i+1}, \ldots, f_n\}$ from $\mathcal{T}_B$ partitions U into the same sets as the removal of the same arc from $\mathcal{T}$.*

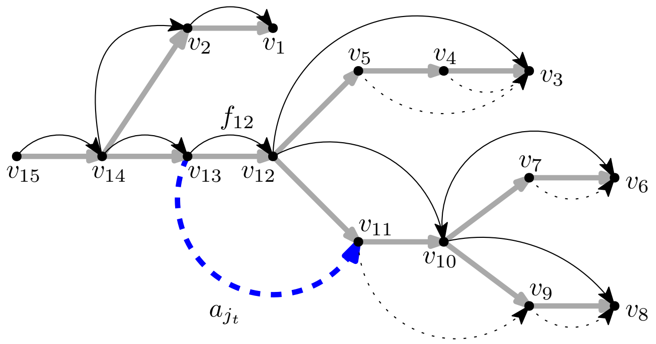

Figure 3 An example of of a feasible cardinal basis B that is a 12-nice basis. The gray arcs form an arborescence $\mathcal{T} = (U, \mathcal{A}_0)$ while the black arcs (both solid and dotted) are the arcs in $\mathcal{T}_B = (U, \mathcal{A}_B)$ (see Theorem 17). The solid (resp., dotted) circular arcs are associated to variables in B with x-value 1 (resp., 0). The blue dashed circular arc $a_{j_t} = (v_{13}, v_{11})$ is not in $\mathcal{T}_B$. Consider a cardinal pivot from B with j_t entering: The $\mathcal{T}_B$-path from v_{13} to v_{11} is $P_B(v_{13}, v_{11}) = ((v_{13}, v_{12}), (v_{12}, v_{10}), (v_{10}, v_8), (v_9, v_8), (v_{11}, v_9))$ (see Lemma 19), where the first three arcs are forward and the last two arcs are backward, and such that $P_B(v_{13}, v_{11})$ is x-augmenting. According to the FFL rule (Definition 20), we should remove the arc f_{12}, as it stands for the first forward arc in $P_B(v_{13}, v_{11})$.

See Figure 3 for an illustration of an i-nice basis. The following lemma is the key observation that allows us to bound the number of iterations in the algorithm. The proof of the third property in the lemma relies on Lemma 25.

▶ **Lemma 27.** *Let (B, O) be a Scarf pair and consider the iteration $(B, O) \to (B', O) \to (B', O')$. Suppose $j_t = O - B$ is the entering column of the cardinal pivot, $j_\ell = B' - B$ is the leaving column of the cardinal pivot, and $j^* = O' - O$ is the entering column of the ordinal pivot. Let $i \in [n]$ be the separator of (B, O). Suppose B is an i-nice basis and $j^* \neq 1$. Then, we have the following:*

1. *Let $a_{j_t} = (v_j, v_k)$. Then, $f_i = (v_j, v_i)$ (recall that f_i is the unique arc entering v_i in $\mathcal{T}$). In other words, a_{j_t} and f_i share the same tail. In addition, f_i is the first arc in $P_0(v_j, v_k)$, where $P_0(v_j, v_k)$ be the $\mathcal{T}$-path from v_j to v_k.*
2. *$a_{j_\ell} = f_i$.*
3. *The separator of (B', O') is i' for some $i' > i$.*
4. *B' is an i'-nice basis.*

We rewrite Scarf's algorithm as a combination of Algorithm 2 (cardinal pivot) and Algorithm 3 (ordinal pivot). To prove Theorem 5, we verify that the initial cardinal basis B_0 an i_0-nice basis, where i_0 is the separator of (B_0, O_0). By inductively applying Lemma 27, we have that every cardinal basis visited by the algorithm is a separator-nice basis, and every ordinal pivot increases the index of the separator. Since the index of the separator is bounded by n, the number of iterations is at most n. At a high level, the intuition is the following: The tree associated with arborescence hypergraphs has a unique source; the potential function we designed takes advantage of this fact, and the vertices of the hypergraph are organized to join the matching one by one according to the depth-first order, which only exists when the tree is an arborescence.

6 Non-integrality of the fractional stable matching polytope

A classical approach to stable matching problem in graphs is to describe the stable matching polytope [35], that is, find an exact description of the convex hull of all characteristic vectors of stable matchings. We construct a hypergraphic preference system $I = (H = (V, E), \succ)$

Algorithm 3 Ordinal Pivot with Separator Change.

```
Let $(O, i)$ be the current ordinal basis with separator $i$, associated with utility vector $u^O$.
$j_\ell \leftarrow i$          ▷ Leaving column is set to correspond to the arc $f_i$ (Lemma 27.2)
$i_\ell \leftarrow i$
$j_r \leftarrow \max O$          ▷ Lemma 25.1
$i_r \leftarrow 1$          ▷ Lemma 25.2
if $\{j > j_r : c_{\bar{\imath},j} > u_{\bar{\imath}}^{O-i}, \forall \bar{\imath} \neq 1\} = \emptyset$ then
    $O' = O \cup \{1\} \setminus \{j_\ell\}$ is a dominating basis for $(A, b, C)$.
else
    $j^* \leftarrow \min\{j > j_r : c_{\bar{\imath},j} > u_{\bar{\imath}}^{O-i}, \forall \bar{\imath} \neq 1\}$          ▷ $c_{1,1} > c_{1,2} > \cdots > c_{1,m}$
    $O \leftarrow O \cup \{j^*\} \setminus \{j_\ell\}$
    Let $j^*$ belong to the block $S_{i'}$ in matrix $C$.
    $i \leftarrow i'$
end if
```

where H is an arborescence hypergraph, on which the above classical approach does not succeed. This result provides evidence that the problem we are interested in is challenging compared to the classical stable matching problems.

We first define

$$P(I) = conv\left(\{x \in \{0,1\}^E : x \text{ is a stable matching}\}\right), \tag{4}$$

and

$$Q(I) = \left\{ x \in \mathbb{R}^E \,\middle|\, \begin{array}{ll} x(\delta(i)) \leq 1, & \forall i \in V, \\ x(e^{\succeq}) \geq 1, & \forall e \in E, \\ 0 \leq x_e \leq 1, & \forall e \in E. \end{array} \right\}, \tag{5}$$

where $\delta(i) = \{e \in E : i \in e\}$ and $e^{\succeq} = \{e' \in E : \exists i \in e, e' \succ_i e\} \cup \{e\}$. We call $P(I)$ the *stable matching polytope* of I and $Q(I)$ the *fractional stable matching polytope* of I. When H is a bipartite graph, it is shown that $P(I) = Q(I)$ (see, e.g., [35]). We show the following.

▶ **Theorem 28.** *There is a hypergraphic preference system $I = (H = (V, E), \succ)$ where H is an arborescence hypergraph, such that the fractional stable matching polytope $Q(I)$ is not integral, thus $P(I) \neq Q(I)$.*

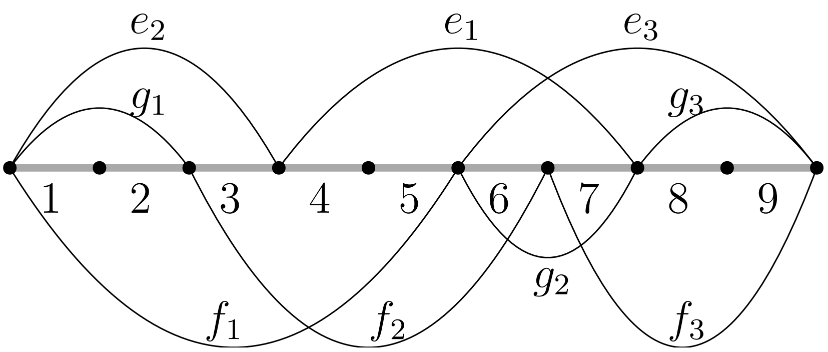

Figure 4 The underlying hypergraph. A node belongs to an edge iff the latter covers the former in this figure.

▶ **Example 29.** Consider the hypergraph in Figure 4: Let $I = (H = (V, E), \succ)$ where $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and $E = \{e_1, e_2, e_3, f_1, f_2, f_3, g_1, g_2, g_3\}$. H is an interval hypergraph. The preference list is as follows:

$$
\begin{aligned}
&1 : e_2 \succ_1 g_1 \succ_1 f_1,\\
&2 : f_1 \succ_2 e_2 \succ_2 g_1,\\
&3 : f_2 \succ_3 f_1 \succ_3 e_2,\\
&4 : e_1 \succ_4 f_2 \succ_4 f_1,\\
&5 : f_1 \succ_5 e_1 \succ_5 f_2,\\
&6 : e_1 \succ_6 f_2 \succ_6 e_3 \succ_6 g_2,\\
&7 : f_3 \succ_7 e_1 \succ_7 e_3 \succ_7 g_2,\\
&8 : e_3 \succ_8 g_3 \succ_8 f_3,\\
&9 : f_3 \succ_9 e_3 \succ_9 g_3.
\end{aligned}
$$

Let x be such that $x(e_1) = x(e_2) = x(e_3) = 0$, $x(f_1) = x(f_2) = x(f_3) = x(g_1) = x(g_2) = x(g_3) = \frac{1}{2}$. Then, x is an extreme point of $Q(I)$, which is non-integral.

7 Conclusion

We showed that Scarf's algorithm converges in polynomial time and returns an integral stable matching on arborescence hypergraphic preference systems. Our result is the first proof of polynomial-time convergence of Scarf's algorithm on hypergraphic stable matching problems. We note that some of our results hold for hypergraphs that are more general than arborescence hypergraphs. It would be interesting to generalize our approach to show polynomial convergence of Scarf's algorithm for more general classes of hypergraphs, such as network hypergraphs. It would also be insightful to interpret our implementation of Scarf's algorithm for arborescence hypergraphs as a purely combinatorial algorithm, if any such interpretation exists.

References

1 Ron Aharoni and Tamás Fleiner. On a lemma of Scarf. *Journal of Combinatorial Theory, Series B*, 87(1):72–80, 2003. doi:10.1016/S0095-8956(02)00028-X.

2 Ron Aharoni and Ron Holzman. Fractional kernels in digraphs. *J. Combinatorial Theory*, 73(1):1–6, 1998. doi:10.1006/JCTB.1997.1731.

3 Ravindra K Ahuja, James B Orlin, Prabha Sharma, and PT Sokkalingam. A network simplex algorithm with o (n) consecutive degenerate pivots. *Operations research letters*, 30(3):141–148, 2002. doi:10.1016/S0167-6377(02)00114-1.

4 Mourad Baïou and Michel Balinski. The stable admissions polytope. *Mathematical programming*, 87:427–439, 2000. doi:10.1007/S101070050004.

5 Mourad Baïou and Michel Balinski. The stable allocation (or ordinal transportation) problem. *Mathematics of Operations Research*, 27(3):485–503, 2002. doi:10.1287/MOOR.27.3.485.310.

6 Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

7 Péter Biró, Gergely Csáji, and Ildikó Schlotter. Stable hypergraph matching in unimodular hypergraphs. *arXiv preprint arXiv:2502.08827*, 2025. doi:10.48550/arXiv.2502.08827.

8 Péter Biró and Tamás Fleiner. Fractional solutions for capacitated ntu-games, with applications to stable matchings. *Discrete Optimization*, 22:241–254, 2016. doi:10.1016/J.DISOPT.2015.02.002.

9 Péter Biró, Tamás Fleiner, and Robert W Irving. Matching couples with scarf's algorithm. *Annals of Mathematics and Artificial Intelligence*, 77:303–316, 2016. doi:10.1007/S10472-015-9491-5.

10 Péter Biró, David F Manlove, and Iain McBride. The hospitals/residents problem with couples: Complexity and integer programming models. In *Experimental Algorithms: 13th International Symposium, SEA 2014, Copenhagen, Denmark, June 29–July 1, 2014. Proceedings 13*, pages 10–21. Springer, 2014. doi:10.1007/978-3-319-07959-2_2.

11 Karthekeyan Chandrasekaran, Yuri Faenza, Chengyue He, and Jay Sethuraman. Scarf's algorithm on arborescence hypergraphs. *arXiv preprint arXiv:2412.03397*, 2024. doi:10.48550/arXiv.2412.03397.

12 Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

13 Gergely Csáji. On the complexity of stable hypergraph matching, stable multicommodity flow and related problems. *Theoretical Computer Science*, 931:1–16, 2022. doi:10.1016/J.TCS.2022.07.025.

14 William H Cunningham. A network simplex method. *Mathematical Programming*, 11:105–116, 1976. doi:10.1007/BF01580379.

15 Piotr Dworczak. Deferred acceptance with compensation chains. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 65–66, 2016. doi:10.1145/2940716.2940727.

16 Yuri Faenza, Chengyue He, and Jay Sethuraman. Scarf's algorithm and stable marriages. *Mathematics of Operations Research*, 2025. Accepted for publication. doi:10.1287/moor.2023.0055.

17 Tamás Fleiner. A fixed-point approach to stable matchings and some applications. *Mathematics of Operations research*, 28(1):103–126, 2003. doi:10.1287/MOOR.28.1.103.14256.

18 Delbert Fulkerson and Oliver Gross. Incidence matrices and interval graphs. *Pacific journal of mathematics*, 15(3):835–855, 1965.

19 David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

20 Penny E Haxell and Gordon T Wilfong. A fractional model of the border gateway protocol (bgp). In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 193–199. Citeseer, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347104.

21 Takashi Ishizuka and Naoyuki Kamiyama. On the complexity of stable fractional hypergraph matching. In *29th International Symposium on Algorithms and Computation (ISAAC 2018)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2018.

22 Shiva Kintali, Laura J Poplawski, Rajmohan Rajaraman, Ravi Sundaram, and Shang-Hua Teng. Reducibility among fractional stability problems. *SIAM Journal on Computing*, 42(6):2063–2113, 2013. doi:10.1137/120874655.

23 Donald Ervin Knuth. Marriages stables. *Technical report*, 1976.

24 Cheng Ng and Daniel S Hirschberg. Three-dimensional stabl matching problems. *SIAM Journal on Discrete Mathematics*, 4(2):245–252, 1991. doi:10.1137/0404023.

25 Hai Nguyen, Thành Nguyen, and Alexander Teytelboym. Stability in matching markets with complex constraints. *Management Science*, 67(12):7438–7454, 2021. doi:10.1287/MNSC.2020.3869.

26 Thanh Nguyen and Rakesh Vohra. Near-feasible stable matchings with couples. *American Economic Review*, 108(11):3154–3169, 2018.

27 Thành Nguyen and Rakesh Vohra. Stable matching with proportionality constraints. *Operations Research*, 67(6):1503–1519, 2019. doi:10.1287/OPRE.2019.1909.

28 Adèle Pass-Lanneau, Ayumi Igarashi, and Frédéric Meunier. Perfect graphs with polynomially computable kernels. *Discrete Applied Mathematics*, 272:69–74, 2020. doi:10.1016/J.DAM.2018.09.027.

29 Alvin E Roth. *Online and matching-based market design*. Cambridge University Press, 2023.

30 Alvin E Roth, Uriel G Rothblum, and John H Vande Vate. Stable matchings, optimal assignments, and linear programming. *Mathematics of operations research*, 18(4):803–828, 1993. doi:10.1287/MOOR.18.4.803.

31 Uriel G Rothblum. Characterization of stable matchings as extreme points of a polytope. *Mathematical Programming*, 54:57–67, 1992. doi:10.1007/BF01586041.

32 Herbert E Scarf. The core of an n person game. *Econometrica: Journal of the Econometric Society*, pages 50–69, 1967.

33 Alexander Schrijver. *Theory of linear and integer programming.* John Wiley & Sons, 1998.

34 Alexander Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

35 Chung-Piaw Teo and Jay Sethuraman. The geometry of fractional stable matchings and its applications. *Mathematics of Operations Research*, 23(4):874–891, 1998. doi:10.1287/MOOR.23.4.874.