**Vol. 34, No. 3 (2024) | DOI: 10.37190/ord240301**

OPEN ACCESS

**Operations Research and Decisions**

www.ord.pwr.edu.pl

OPERATIONS RESEARCH AND DECISIONS QUARTERLY

# On a many-sided matching problem with mixed preferences

Marcin Anholcer[1*] Maciej Bartkowiak[2]

[1] *Institute of Informatics and Quantitative Economics, Poznań University of Economics and Business, Poznań, Poland*

[2] *Doctoral School, Poznań University of Economics and Business, Poznań, Poland*

[*] *Corresponding author, email address: m.anholcer@ue.poznan.pl*

**Abstract**

Motivated by recent results on lexicographic and cyclic preferences, we present new sufficient conditions for the existence of stable matching in many-sided matching problems. Here, our focus shifted towards integrating the two-sided matching problem, characterized by reciprocal preferences, with the many-sided matching problem, which involves cyclic preferences. In particular, we show that one of the configurations presented recently by Zhang and Zhong for three-sided matching problems can be generalized to more dimensions. In our setting, the preferences are cyclic and, in the case of all but two pairs of consecutive sets of agents, also reciprocal, which generalizes the three-set setting of Zhang and Zhong. Our approach can be also applied to generalize the problems with any system of cyclic preferences for which the existence of a stable matching is guaranteed.

**Keywords:** *many-sided matching problem, stable matching, cyclic preferences, lexicographic preferences, mixed preferences, deferred acceptance algorithm*

## 1. Introduction

The two-sided stable matching problem (also known as the stable marriage problem), first introduced and solved by Gale and Shapley [12], has fascinated many researchers ever since. The broader problem of the existence of stable matchings in many-to-many-sided problems with various kinds of preferences and restrictions is only partially explored. Knuth [22] showed that the problem changes significantly when there are more than two parties to match, and Alkan [2] proved that stable matchings in three-sided systems may not exist, in general. Still, it is possible to find stable matchings when we restrict the preferences of the parties (of three or more sets) to lexicographic preferences [9], cyclic preferences [4, 6, 10, 24], or some alternations of lexicographic preferences with additional restrictions [23]. The stable marriage problem also extends by introducing indifference to the preferences of the parties [19, 26]. The problem has been extensively investigated in terms of its computational complexity or using computer simulations of matching instances [7, 16, 18, 20, 26].

Received 9 August 2023, accepted 14 July 2024, published online 17 October 2024
ISSN 2391-6060 (Online)/

The costs of publishing this issue have been co-financed by the program *Development of Academic Journals* of the Polish Ministry of Education and Science under agreement RCN/SP/0241/2021/1

This wide approach to the topic of stable matchings benefits many real-life applications, from medical labour markets [31], through the kidney transplant compatibility problem [5, 15, 21, 30], allocation problems like students-project assignments [1], to assigning a group of students to dormitories [28] and more recently optimization in network services [8] and mobile edge computing [29]. Implementations of the Gale and Shapley algorithm can also be found in online dating [13] or carpooling and ridesharing [11].

Recently, Zhang and Zhong [32] published a paper on two interesting modifications of cyclic preferences, which can give a stable matching for three sets of agents. Soon after that, Arenas and Torres--Martinez [3] showed that one of the proposed families of preference systems does not need to lead to a stable solution (in fact, they indicated a gap in the proof). However, the other proposed system always possesses a stable solution. Although the proof presented in [32] also seems to have some gaps, it can be easily fixed. Moreover, as we observed, the proposed system of preferences can be extended in several ways to an arbitrary number of sets of agents. In this paper, we present a complete proof of the existence of stability in these generalized systems, which will also cover the results of Zhang and Zhong.

The paper is organized as follows. In the next section, we present basic definitions and facts about stable matchings in two-sided and multi-sided systems of preferences, in particular the ones that we are going to use in the remainder of this paper. In Section 3 we present the definitions and basic facts about mixed preference systems (i.e., the systems where cyclic and lexicographic preferences are combined). Section 4 consists of the presentation of a matching algorithm and the proof that it produces a stable matching. Finally, in Section 5, we conclude the paper with some final remarks and a few open problems.

## 2. Preliminaries

Gale and Shapley [12], probably for an intuitive illustration of the problem, introduced stable matchings as stable marriages between a set of men and a set of women. In the case of two sets, to honour the historical idea, we keep the names of the sets and agents to match as $M$, $W$, $m$, and $w$, respectively. For the sake of completeness, we will briefly describe the problem.

Assume that two sets $M$ and $W$ are given, where $|M| = |W| = n$. Each $m \in M$ has a strict and complete preference order $\succ_m$ in pairs $(m, w) \in \{m\} \times W \subset M \times W$ ($(m, w_1) \succ_m (m, w_2)$ means that $m$ prefers $(m, w_1)$ over $(m, w_2)$), which can be equivalently defined as a preference order on the set $W$ (now $w_1 \succ_m w_2$ denotes the fact that $m$ prefers $w_1$ over $w_2$). Similarly, each $w \in W$ has a strict and complete preference order $\succ_w$ in all pairs $(m, w) \in M \times \{w\} \subset M \times W$, which can be equivalently defined as a preference order on the set $M$. Matching is a function $\mu : M \cup W \to M \times W$ such that for every $m \in M$, $\mu(m) = (m, w)$ for some $w \in W$, for every $w \in W$, $\mu(w) = (m, w)$ for some $m \in M$ and $\mu(w) = (m, w) \Leftrightarrow \mu(m) = (m, w)$. Note that from the definition of $\mu$ it follows that the resulting matching $\mu(M \cup W)$ can be denoted by $\mu(M)$ or $\mu(W)$, since $\mu(M \cup W) = \mu(M) = \mu(W)$. Also, for simplicity, when the context is evident, one can use the notation $\mu(w) = m$ and $\mu(m) = w$ instead of $\mu(m) = \mu(w) = (m, w)$, however, in such a case one may use only $\mu(M \cup W)$ to denote the assignment.

Given a matching $\mu$, a blocking pair is a pair $(m, w)$ such that $(m, w) \succ_m \mu(m)$ and $(m, w) \succ_w \mu(w)$ (in the reduced form $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$). Obviously, $(m, w) \notin \mu(M \cup W)$. A matching $\mu$ is stable if there are no blocking pairs. Gale and Shapley proved that, in the presented setting, a stable matching always exists. Their constructive proof used the so-called deferred acceptance algorithm (Algorithm 1), which we present below for the sake of completeness. Here and in the remainder of the paper, $\mu(x) = null$ will denote the fact that agent $x$ is not matched in the current partial matching. Similarly, assignment $\mu(x) \leftarrow null$ means that agent $x$ becomes unmatched.

**Input:** Two sets $M$ and $W$ where $|M| = |W| = n$, and preference lists for all elements from $M \cup W$ on all ordered pairs $(m, w) \in M \times W$ that include them.
**Output:** Stable matching $\mu : M \cup W \to M \times W$ between elements from $M$ and $W$.
Initialize: Mark all elements of $M \cup W$ as available for matching, i.e., set $\mu(x) \leftarrow null$ for every $x \in M \cup W$;
**while** *there exists an available element* $m \in M$ **do**
  Choose the most preferred pair $(m, w) : w \in W$ on $m$'s preference list and cross it out from this list;
  **if** $w$ *is available* **then**
    set $\mu(m) \leftarrow (m, w)$, $\mu(w) \leftarrow (m, w)$, mark $m$ and $w$ as unavailable;
  **end**
  **else if** $w$ *prefers* $(m, w)$ *over her current assignment* $(m', w)$ **then**
    set $\mu(m) \leftarrow (m, w)$, $\mu(w) \leftarrow (m, w)$, $\mu(m') \leftarrow null$, mark $m$ as unavailable, mark $m'$ as available;
  **end**
**end**
**return** $\mu$

**Algorithm 1.** Gale–Shapley deferred acceptance algorithm (GS)

As mentioned in the Introduction, generalized versions of the problem, where the number of sets is greater than two, have been studied. In particular, the three-sided problem (with the widely discussed application to the Kidney Exchange Problem) attracted much interest, but also the problem for an arbitrary number of sets has gained some attention. Let us discuss here this most general version.

This time $k$ sets of agents are given: $X_1, X_2, \ldots, X_k$. For each $x_j \in X_j$ a strict and complete preference order $\succ_{x_j}$ is defined on the set of all $k$-tuples[1] $(x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k) \in \bigotimes_{1 \le i < j} X_i \times \{x_j\} \times \bigotimes_{j < i \le k} X_i \subset \bigotimes_i X_i$ containing $x_j$. Just like in the case of two sets, one can define this relation on the set of $(k-1)$-tuples from $\bigotimes_{1 \le i < j} X_i \times \bigotimes_{j < i \le k} X_i$, but this would make the further considerations hard to follow, so we will use this form only if it significantly simplifies the notation (in particular in the three-sided illustrative examples).

Matching is a function $\mu : \bigcup_{i=1}^{k} X_i \to \bigotimes_{i=1}^{k} X_i$ such that $\forall x \in \bigcup_{i=1}^{k} X_i : x \in \mu(x)$ (which means that every $x \in \bigcup_{i=1}^{k} X_i$ belongs to the unique tuple $\mu(x)$ assigned to it, that is, if $x \in X_j$ for some $j$, then $\mu(x) = (x_1, \ldots, x_{j-1}, x, x_{j+1}, \ldots, x_k)$ for some $(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_k) \in \bigotimes_{1 \le i < j} X_i \times \bigotimes_{j < i \le k} X_i$). Also, here one can skip $x_j$ and define the function on $(k-1)$-tuples as $\mu(x_j) = (x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_k)$, its restriction to a specific dimension $i$ by $\mu_{X_i}(x_j) = x_i$, its restriction to two dimensions $i$ and $h$ by

[1] The symbol $\bigotimes$ denotes the Cartesian product of a number of sets: $\bigotimes_{i=1}^{k} X_i = X_1 \times X_2 \times \cdots \times X_k$.

$\mu_{X_iX_h}(x_j) = (x_i, x_h)$ etc. Finally, instead of $\mu(\bigcup_{i=1}^{k} X_i)$ one can use $\mu(X_j)$ for any $j = 1, \ldots, n$ to denote the final assignment, that is, a set of ordered $k$-tuples (this can be in particular $\mu(X_1)$).

A (strongly)[2] blocking $k$-tuple is a $k$-tuple $\xi = (x_1, \ldots, x_k)$ such that $\xi \succ_{x_j} \mu(x_j)$ for all $j = 1, \ldots, k$. Note that by definition $\xi \notin \mu(X_1)$. For a (weakly) stable matching, a blocking $k$-tuple does not exist. In 1976, Knuth [22] asked whether a stable matching always exists if $k \geq 3$ and the answer turned out to be negative, as the following example shows. For the sake of clarity, we use the notation commonly used in the literature: $M = \{m_1, m_2\}$ (men) $W = \{w_1, w_2\}$ (women) and $C = \{c_1, c_2\}$ (children or cats). In fact, consider the following preferences:

$$(w_1, c_1) \succ_{m_1} (w_1, c_2) \succ_{m_1} (w_2, c_2) \succ_{m_1} (w_2, c_1)$$
$$(w_2, c_1) \succ_{m_2} (w_1, c_1) \succ_{m_2} (w_1, c_2) \succ_{m_2} (w_2, c_2)$$
$$(m_2, c_1) \succ_{w_1} (m_2, c_2) \succ_{w_1} (m_1, c_1) \succ_{w_1} (m_1, c_2)$$
$$(m_2, c_1) \succ_{w_2} (m_2, c_2) \succ_{w_2} (m_1, c_2) \succ_{w_2} (m_1, c_1)$$
$$(m_1, w_1) \succ_{c_1} (m_1, w_2) \succ_{c_1} (m_2, w_2) \succ_{c_1} (m_2, w_1)$$
$$(m_1, w_2) \succ_{c_2} (m_1, w_1) \succ_{c_2} (m_2, w_1) \succ_{c_2} (m_2, w_2)$$

There are four possible matchings. Sample blocking triples for each of them are:

$$\{(m_1, w_1, c_1), (m_2, w_2, c_2)\} \to (m_2, w_1, c_2)$$
$$\{(m_1, w_1, c_2), (m_2, w_2, c_1)\} \to (m_1, w_1, c_1)$$
$$\{(m_1, w_2, c_1), (m_2, w_1, c_2)\} \to (m_1, w_2, c_2)$$
$$\{(m_1, w_2, c_2), (m_2, w_1, c_1)\} \to (m_2, w_2, c_1)$$

Note also that the case where $k = 1$ has been considered in the literature (see, e.g., [16]). It is the so-called roommate problem, where an even cardinality set $X$ of students is given, and every student $s \in X$ has a strict and complete preference order $\succ_s$ in the set $\{\{s, t\} : t \in X \setminus \{s\}\}$. If it is clear from the context, for the sake of simplicity one can write $x \succ_s y$ instead of $\{s, x\} \succ_s \{s, y\}$. The goal is to pair up the students so that the resulting matching is stable. The notions of blocking pair and stable matching are defined in an obvious way, analogously to the case of $k = 2$. To be more specific, the matching in this case is a function $\mu : X \to \{\{s, t\} : s \in X \wedge t \in X \wedge s \neq t\}$ such that for every $s, t \in X$, $\mu(s) = \{s, t\} \Leftrightarrow \mu(t) = \{s, t\}$. A pair $\{s, t\}$ blocks $\mu$ if $\{s, t\} \succ_s \mu(s)$ and $\{s, t\} \succ_t \mu(t)$. Matching $\mu$ is stable if there are no blocking pairs.

[2]One can also consider weakly blocking tuples corresponding with strongly stable matchings. The distinction becomes meaningful if one considers preferences with ties, which we do not consider in the present paper. See, e.g., [17] for more details.

Also in this case a stable matching does not need to exist, as the following example shows (the set of students is $X = \{a, b, c, d\}$). Assume that the preferences are defined as follows:

$$b \succ_a c \succ_a d$$

$$c \succ_b a \succ_b d$$

$$a \succ_c b \succ_c d$$

$$a \succ_d b \succ_d c$$

Then the blocking pairs for all three possible matchings are:

$$\{\{a, b\}, \{c, d\}\} \rightarrow \{b, c\}$$

$$\{\{a, c\}, \{b, d\}\} \rightarrow \{a, b\}$$

$$\{\{a, d\}, \{b, c\}\} \rightarrow \{a, c\}$$

The case of $k \geq 3$ turned out to be problematic even in the case of $k = 3$ of cyclic preferences, where the members of $X_1$ have preferences in $X_2$, the members of $X_2$ in $X_3$ and the members of $X_3$ in $X_1$. To be more specific, for example, for $X_1$ and $X_2$ this means that every $x_1 \in X_1$ has a strict and complete preference order $\succ^{\star}_{x_1}$ on $X_2$ such that for any triples $(x_1, x_2^1, x_3^1), (x_1, x_2^2, x_3^2) \in X_1 \times X_2 \times X_3$, $(x_1, x_2^1, x_3^1) \succ_{x_1} (x_1, x_2^2, x_3^2) \Leftrightarrow x_2^1 \succ^{\star}_{x_1} x_2^2$. In this case, Boros et al. [6] proved that when the number of elements in each set is $n \leq k$, then there is a stable matching. For $k = 3$ Eriksson et al. [10] proved stability assuming that $n \leq 4$ and Hofbauer [14] further pushed the result in the case of $n \leq k+1$. Later, Pashkovich and Poirrie [27] proved $n \leq 5$ for $k = 3$. Finally, Lam and Plaxton [24] showed that in general, stable matching may not exist when $k \geq 3$ and the minimum size of a counterexample for $k = 3$ was recently reduced to $n = 20$ by Lerner [25].

So far, the only successful approach in terms of stable matchings for $k \geq 3$ is when the preferences of agents are defined lexicographically, in a predefined order of precedence for the remaining sets. Danilov [9] proved that when that is the case, a stable matching always exists.

Among numerous papers, where sufficient conditions for the existence of a stable matching have been presented, one can find the one of Zhang and Zhong [32]. The authors consider two systems of preferences defined on three sets and claim that the presented algorithms produce stable matchings. Unfortunately soon after, Arenas and Torres-Martinez [3] showed that the second algorithm does not actually need to result in a stable matching. It cannot, since the system of preferences presented by Zhang and Zhong is less restrictive than cyclic preferences, for which stable matching does not always exist, as shown by Lam and Plaxton [24].

On the other hand, the algorithm presented for the other system of preferences always produces a stable matching. This encouraged us to consider more general systems with $k > 3$ sets. We show that a generalization of the method of Zhang and Zhong (using ideas similar to those presented in [9]) can be applied to solve the instances of these generalized problems.

We briefly summarize the contents of the selected papers in Table 1.

**Table 1.** Summary of MSSM variants literature ($k$ sets of $n$ agent)

| Matching variant | $k = 1$ | $k = 2$ | $k = 3$ | $k \geq 3$ | Possible applications |
|---|---|---|---|---|---|
| $k$-Sided stable matchings general considerations | | [12, 17] [19, 26] [28] | | [2, 22] | students admissions [12] medical labour markets [31] other [1, 11, 13] |
| Lexicographic preferences | | | [9] [23] | [9] | |
| Cyclic preferences | | | [4] [10] ($n \leq 4$) [25] [27] ($n \leq 5$) | [6] ($n \leq k$) [14] ($n \leq k+1$) [24] | kidney exchange programs [5, 15, 21, 30] network services [8, 29] |
| Mixed preferences | | | [3] [32] | | |
| Stable roommates problem | [16] | | | | part of many software packages [16] |

Literature concerning computational complexity: [7], [16], [18], [20], [26].

## 3. Mixed preferences

Let us start with a brief presentation of the model of Zhang and Zhong [32], which we will extend. They consider 3-sided matching problems with sets of agents $U$, $V$ and $W$, and the preferences defined as follows. Every $u \in U$ has preferences first and foremost in $V$, i.e., for each $u \in U$ there exists a strict and complete preference order $\succ_u^V$ on $V$ such that for every $v, v' \in V$ and for every $w \in W$, $(u, v, w) \succ_u (u, v', w) \Leftrightarrow v \succ_u^V v'$. This relation is denoted by $U \longrightarrow V$. In analogy, it is assumed that every $w \in W$ has preferences first and foremost on $U$, which is denoted by $W \longrightarrow U$. Moreover, the members of $W$ have preferences on $V$ and the members of $V$ have preferences on $W$, which could be written as $W \longrightarrow V$ and $V \longrightarrow W$, but for simplicity the authors use the notation $V \longleftrightarrow W$ instead [3]. The authors use two runs of Algorithm 1 to construct a stable matching. The first run matches the members of $V$ and $W$. The second one matches the resulting pairs $x = (v, w) \in X \subseteq V \times W$ with the members of $U$ by defining the preferences as $\succ_u^X = \succ_u^V$ (which is possible because there is a bijection between $V$ and $X$) and $\succ_x^U = \succ_w^U$ (which is possible because there is a bijection between $W$ and $X$).

We generalize the model for an arbitrary number of sets. Similarly as in [32], we will write $X_i \longrightarrow X_j$ if the members of $X_i$ have preferences first and foremost on $X_j$, so that for each $x_i \in X_i$ there is a relation $\succ_{x_i}^j$ such that for any $x_{j_1}, x_{j_2} \in X_j$ and any $k$-tuples[4] $\xi_1 = (x_1, x_2, \ldots, x_i, \ldots, x_{j_1}, \ldots, x_k)$ and $\xi_2 = (x_1, x_2, \ldots, x_i, \ldots, x_{j_2}, \ldots, x_k)$ we have $\xi_1 \succ_{x_i} \xi_2 \iff x_{j_1} \succ_{x_i}^j x_{j_2}$. In addition, we will sometimes refer to this setting by stating that the preferences of $x_i$ are consistent with the preferences in the set $X_j$ given by $x_{j_1} \succ_{x_i}^j x_{j_2}$. If both $X_i \longrightarrow X_j$ and $X_j \longrightarrow X_i$ occur, we write $X_i \longleftrightarrow X_j$. Furthermore, for any index $i$, the case of $X_i \longrightarrow X_j$ can be defined for more than one $j$.

Assume that we construct the matching step by step, by adding the members of the next set $X_j$ to the existing tuples. For any sequence of consecutive (modulo $k$) indices $i_1, i_1 + 1, \ldots, i_2$, the members

[3] Actually, Zhang and Zhong use the notation $U \Rightarrow V$ and $V \Leftrightarrow W$. We use single arrows to avoid confusion, since in our paper the symbols $\Rightarrow$ and $\Leftrightarrow$ are used to denote the implication and equivalence

[4] Without loss of generality, we assume $i < j$. In the opposite case, the tuples are $\xi_1 = (x_1, x_2, \ldots, x_{j_1}, \ldots, x_i, \ldots, x_k)$ and $\xi_2 = (x_1, x_2, \ldots, x_{j_2}, \ldots, x_i, \ldots, x_k)$

$\xi \in \mu(\bigcup_{i=i_1}^{i_2} X_i)$ of a partial matching will also be included in the above notation, where for each tuple $\xi$ we can define its preferences as follows:

- $\mu(\bigcup_{i=i_1}^{i_2} X_i) \longrightarrow X_j$ means that we use the relation $\succ^j_{x_{i_2}}$, i.e., the relation of the tuple is the relation of $x_{i_2}$ on $X_j$,
- $X_j \longrightarrow \mu(\bigcup_{i=i_1}^{i_2} X_i)$ means that we use the relation $\succ^{i_1}_{x_j}$, i.e., the relation of $x_j$ on $X_{i_1}$ is extended to the set of complete tuples in the natural way,
- $\mu(\bigcup_{i=i_1}^{i_2} X_i) \longleftrightarrow X_j$ means that we use the relations $\succ^j_{x_{i_2}}$ and $\succ^{i_2}_{x_j}$ similarly as in the first two cases,
- $X_j \longleftrightarrow \mu(\bigcup_{i=i_1}^{i_2} X_i)$ means that we use the relations $\succ^{i_1}_{x_j}$ and $\succ^j_{x_{i_1}}$ similarly as in the first two cases.

Using the above notation, the two systems of preferences proposed by Zhang and Zhong [32] can be written as

1 $X_1 \longleftrightarrow X_2 \longrightarrow X_3 \longrightarrow X_1$.
2 $X_1 \longrightarrow X_2 \longrightarrow X_3 \longrightarrow X_1 \times X_2$.

As already mentioned above, the second system does not guarantee the existence of a stable matching [3]. We focus on the idea given by the first system of preferences and extend it to $k$ sets $X_i$ $(i \in \{1, 2, \dots, k\})$ such that

$$X_1 \longleftrightarrow X_2 \longleftrightarrow \cdots \longleftrightarrow X_j \longrightarrow X_{j+1} \longleftrightarrow \cdots \longleftrightarrow X_k \longrightarrow X_1$$

Now any blocking $k$-tuple $\xi = (x_1, x_2, \dots, x_k)$ existing in this system for a matching $\mu$ must have all the following properties:

- $x_{i+1} \succ^{i+1}_{x_i} \mu_{X_{i+1}}(x_i)$ where $i \in \{1, \dots, k-1\}$,
- $x_{i-1} \succ^{i-1}_{x_i} \mu_{X_{i-1}}(x_i)$ where $i \in \{2, \dots, j\} \cup \{j+2, \dots, k\}$,
- $x_1 \succ^1_{x_k} \mu_{X_1}(x_k)$.

For a better contrast between the mixed, lexicographic and cyclic preferences, let us consider two examples.

**Example 1.** Consider the following 5-sided matching problem. Let $X_i = \{x_i^{(1)}, x_i^{(2)}\}$, where $i \in \{0, 1, 2, 3, 4\}$. The preferences of $x_i^{(q)}, q = 1, 2$, given by the relation $\succ_{x_i^{(q)}}$, for $i \in \{1, 2, 3\}$ are as follows (all sums and differences in the indices are considered modulo 5):

$$\begin{aligned}
&(x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
\succ_{x_i^{(q)}} &(x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
\succ_{x_i^{(q)}} &(x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
\succ_{x_i^{(q)}} &(x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}}
\end{aligned}$$

$$
\begin{aligned}
&\succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(2)})
\end{aligned}
$$

For $i \in \{0, 4\}$ in turn, let the preferences be

$$
\begin{aligned}
&(x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_{i-1}^{(2)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_{i-1}^{(1)}, x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(2)}).
\end{aligned}
$$

One can easily see that for $i \in \{0,\, 1,\, 2,\, 3, 4\}$, the preferences of $x_i^{(q)}, q = 1, 2$, are consistent with the preferences on the set $X_{i+1}$ given by $x_{i+1}^{(1)} \succ_{x_i^{(q)}}^{i+1} x_{i+1}^{(2)}$ and for $i \in \{1, 2, 3\}$ – with the preferences on the set $X_{i-1}$ given by $x_{i-1}^{(1)} \succ_{x_i^{(q)}}^{i-1} x_{i-1}^{(2)}$. However, they are not consistent with any preference order in any other set. This implies that the preferences are not in accordance with the definition of lexicographic order by Danilov [9] but would be in accordance with the mixed preferences system $X_0 \longleftrightarrow X_1 \longleftrightarrow X_2 \longleftrightarrow X_3 \longrightarrow X_4 \longrightarrow X_0$.

**Example 2.** Consider the following 4-sided matching problem. Let $X_i = \{x_i^{(1)}, x_i^{(2)}\}$, where $i \in \{0,\, 1,\, 2,\, 3\}$. The preferences of $x_i^{(q)}, q = 1, 2$, given by the relation $\succ_{x_i^{(q)}}$, for $i \in \{0,\, 1,\, 2,\, 3\}$ are as follows (all sums and differences in the indices are considered modulo 4):

$$
\begin{aligned}
&(x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
&\succ_{x_i^{(q)}} (x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(2)}).
\end{aligned}
$$

An analysis similar to the one in Example 1 shows that the system is consistent with the cyclic preferences defined, e.g., by Boros et al. [6], which can be written in the form $X_0 \longrightarrow X_1 \longrightarrow X_2 \longrightarrow$

$X_3 \longrightarrow X_0$. However, reciprocal relations between sets do not occur, so the presented system is not in accordance with our definition of mixed preference system.

# 4. Stable matching algorithm

Now we are going to present an algorithm (Algorithm 3) that produces stable matching in an arbitrary $k$-set system with mixed preferences defined as above. It uses as a subroutine the deferred acceptance algorithm (Algorithm 1). We start by rephrasing the latter one (Algorithm 2), so that it is consistent with our notation and the fact that sometimes the members of the matched pairs are the tuples belonging to a partial matching. In the following, given two tuples $x$ and $y$, $(x, y)$ denotes their concatenation. A singleton is treated as a one-element tuple.

The input of Algorithm 2 is defined according to the notation introduced in Section 3 as follows. Two sets of tuples are given, $X = \{\xi = (x_1, \ldots, x_j) : x_i \in X_i, i = 1, \ldots, j\}$ and $Y = \{\xi = (x_{j+1}, \ldots, x_{j+k}) : x_i \in X_i, i = j+1, \ldots, j+k\}$ where $|X| = |Y| = n$, and $X_i, i = 1, \ldots, j+k$ are disjoint sets of agents with preferences satisfying $X_1 \longleftrightarrow X_2 \longleftrightarrow \cdots \longleftrightarrow X_{j+k}$. Each of $X$ and $Y$ are either a set of singletons or the image of a partial stable matching of the members of $X_1, \ldots, X_j$ or $X_{j+1}, \ldots, X_{j+k}$, respectively. Every tuple $\xi$ from $X$ has preferences defined on $Y$ consistent with the preferences of $x_j$ on $Y$, which are in turn consistent with the preferences of $x_j$ on $X_{j+1}$. Similarly, every tuple $\xi$ from $Y$ has preferences defined on $X$ consistent with the preferences of $x_{j+1}$ on $X$, which are in turn consistent with the preferences of $x_{j+1}$ on $X_j$. Such a system of two sets of tuples with appropriately defined preferences will be denoted by $X \longleftrightarrow Y$.

**Input:** System $X \longleftrightarrow Y$ of two sets of tuples with preferences defined as above.
**Output:** Stable matching $\mu_{X\cup Y}$ of the members of $X$ and $Y$, being also a stable matching of the members of $X_1, X_2, \ldots, X_{j+k}$.
Initialize: Mark all elements of $X \cup Y$ as available for matching, i.e., set $\mu(x) \leftarrow null$ for every $x \in X \cup Y$;
**while** *there exists available element* $x \in X$ **do**
  Choose the most preferred tuple $(x, y) : y \in Y$ on $y$'s preference list and cross it out from this list;
  **if** $y$ *is available* **then**
    set $\mu_{X\cup Y}(x) \leftarrow (x, y)$, $\mu_{X\cup Y}(y) \leftarrow (x, y)$, mark $x$ and $y$ as unavailable;
  **end**
  **else if** $y$ *prefers* $(x, y)$ *over its current assignment* $(x', y)$ **then**
    set $\mu(x) \leftarrow (x, y)$, $\mu(y) \leftarrow (x, y)$, $\mu(x') \leftarrow null$, mark $x$ as unavailable, mark $x'$ as available;
  **end**
**end**
**return** $\mu_{X\cup Y}$

**Algorithm 2.** Rephrased Gale-Shapley Deferred Acceptance Algorithm

As one can easily see, Algorithm 2 is indeed a generalization of Algorithm 1, since the latter one is actually a simplified form of Algorithm 2, where both input sets of tuples are sets of singletons and the notation has been adjusted accordingly. In particular, in the case where both $X$ and $Y$ consist of singletons, $X \longleftrightarrow Y$ denotes the system of two sets $X$ and $Y$ of agents with mutual strict and complete preferences.

**Input:** k sets $X_i$ where $i \in \{1, 2, \ldots, k\}$, $|X_i| = n$. Preference lists according to the scheme:
$X_1 \longleftrightarrow X_2 \longleftrightarrow \cdots \longleftrightarrow X_j \longrightarrow X_{j+1} \longleftrightarrow \cdots \longleftrightarrow X_k \longrightarrow X_1$

**Output:** Stable matching $\mu$ of the members of $\bigcup_{i=1}^{k} X_i$

Initialize: Mark all elements of $\bigcup_{i=1}^{k} X_i$ as available for matching, i.e., set $\mu(x) \leftarrow null$ for every $x \in \bigcup_{i=1}^{k} X_i$;
$\mu_1 \leftarrow$ **Algorithm 2** $(X_1 \longleftrightarrow X_2)$; // Run GSX algorithm on $X_1$ and $X_2$
**for** $i = 2$ *to* $j - 1$ **do**
  $\mu_i \leftarrow$ **Algorithm 2** $(\mu_{i-1}(X_1) \longleftrightarrow X_{i+1})$; // Run GSX algorithm on $\mu_{i-1}(X_1)$ and $X_{i+1}$
**end**
$\mu_{j+1} \leftarrow$ **Algorithm 2** $(X_{j+1} \longleftrightarrow X_{j+2})$; // Run GSX algorithm on $X_{j+1}$ and $X_{j+2}$
**for** $i = j + 2$ *to* $k - 1$ **do**
  $\mu_i \leftarrow$ **Algorithm 2** $(\mu_{i-1}(X_{j+1}) \longleftrightarrow X_{i+1})$; // Run GSX algorithm on $\mu_{i-1}(X_{j+1})$ and $X_{i+1}$
**end**
$\mu \leftarrow$ **Algorithm 2** $(\mu_{j-1}(X_1) \longleftrightarrow \mu_{k-1}(X_{j+1}))$; // Run GSX algorithm on $\mu_{k-1}(X_{j+1})$ and $\mu_{j-1}(X_1)$
**return** $\mu$; // Return the matching $\mu$

**Algorithm 3.** Mixed preferences matching algorithm

**Theorem 3.** Matching $\mu$ generated by the Algorithm 3 is stable.

**Proof.** The truth of this statement is based primarily on the stability of the two-sided matchings obtained by the rephrased Gale-Shapley algorithm (Algorithm 2).

First, we will go through the steps of Algorithm 3 and prove that each partial matching is stable.

The first step of either of the two parts of the algorithm employs the GS algorithm to form a stable matching between the sets $X_1$ and $X_2$ ($X_{j+1}$ and $X_{j+2}$, respectively).

Now, assume that a partial matching $\mu_{j-1}$ is combined with the set $X_{j+1}$. Assume that a tuple $\xi = (m, x)$ is blocking, where $m = (x_1, \ldots, x_j) \in X_1 \times \cdots \times X_j$ and $x \in X_{j+1}$. By the conditions listed in the end of Section 3 this would mean in particular that $x_{j+1} \succ_{x_j}^{j+1} \mu_{X_{j+1}}(x_j)$ and $x_j \succ_{x_{j+1}}^{j} \mu_{X_j}(x_{j+1})$, but this is impossible because it would mean the existence of a blocking pair in the partial matching $\mu_i$, resulting in a set of pairs $(m, x)$. And this is impossible, since the two-sided matching produced by Algorithm 2 is always stable. Thus, the stability of two-sided matching implies the stability of the resulting partial matching of tuples. Finally, the fact that every partial matching obtained by Algorithm 2 is stable follows by induction.

Let us switch to the last part of the algorithm, where the (stable) partial matchings $\mu_{j-1}$ and $\mu_{k-1}$ are combined. A reasoning similar to that above shows that the existence of a blocking tuple would be equivalent to the existence of a blocking pair $(m_{j-1}, m_{k-1})$ in the resulting two-sided matching, where $m_{j-1} \in \mu_{j-1}(X_1)$ and $m_{k-1} \in \mu_{k-1}(X_{j+1})$. This is obviously impossible and we are done. □

We end this section with a brief discussion of the complexity of the presented algorithm. Already Gale and Shapley ([12]) showed that Algorithm 1 (and thus Algorithm 2) needs $(n-1)^2 + 1$ iterations in the worst case. Since Algorithm 3 runs Algorithm 2 as a subroutine exactly $k$ times, we can conclude this section with the following observation.

**Fact 4.** The complexity of Algorithm 3 is $O(n^2 k)$.

# 5. Conclusions

Many authors investigating the topic of stable assignments endeavor to find necessary and sufficient conditions for the existence of stable matching. So far, such conditions have not been found in the general case of $k \geq 3$. In this paper, we managed to prove that stable matching always exists for quite restrictive, mixed preferences settings.

It may be noticed here that for any system $X_1 \to X_2 \to \cdots \to X_s \to X_1$ of cyclic preferences which admits stable matching (like those of [6] or [14], see discussion in the Introduction), it is possible to extend our results in an easy way: one can substitute any set $X_i$ in this system with an $X_k^1 \longleftrightarrow X_k^2 \longleftrightarrow \cdots \longleftrightarrow X_k^j \longrightarrow X_k^{j+1} \longleftrightarrow \cdots \longleftrightarrow X_k^k \longrightarrow X_k^1$ batch and the obvious modification of Algorithm 3 will still result with a stable matching. Let us present this idea in the following example.

**Example 5.** Consider the following 4-sided matching problem. Let $X_i = \{x_i^{(1)}, x_i^{(2)}\}$, where $i \in \{0,\ 1,\ 2,\ 3\}$. The preferences of $x_i^{(q)}, q = 1, 2$, given by the relation $\succ_{x_i^{(q)}}$, for $i \in \{0,\ 1,\ 2,\ 3\}$ are as follows (all the sums and differences in the indices are considered modulo 4). For $i = 1$, let:

$$
\begin{aligned}
&(x_0^{(1)}, x_1^{(q)}, x_2^{(1)}, x_3^{(1)}) \succ_{x_1^{(q)}} (x_0^{(1)}, x_1^{(q)}, x_2^{(1)}, x_3^{(2)}) \succ_{x_i^{(q)}} \\
\succ_{x_1^{(q)}} &(x_0^{(1)}, x_1^{(q)}, x_2^{(2)}, x_3^{(1)}) \succ_{x_1^{(q)}} (x_0^{(1)}, x_1^{(q)}, x_2^{(2)}, x_3^{(2)}) \succ_{x_i^{(q)}} \\
\succ_{x_1^{(q)}} &(x_0^{(2)}, x_1^{(q)}, x_2^{(1)}, x_3^{(1)}) \succ_{x_1^{(q)}} (x_0^{(2)}, x_1^{(q)}, x_2^{(1)}, x_3^{(2)}) \succ_{x_i^{(q)}} \\
\succ_{x_1^{(q)}} &(x_0^{(2)}, x_1^{(q)}, x_2^{(2)}, x_3^{(1)}) \succ_{x_1^{(q)}} (x_0^{(2)}, x_1^{(q)}, x_2^{(2)}, x_3^{(2)}),
\end{aligned}
$$

and for $i \neq 1$ let:

$$
\begin{aligned}
&(x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
\succ_{x_i^{(q)}} &(x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_i^{(q)}, x_{i+1}^{(1)}, x_{i+2}^{(1)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
\succ_{x_i^{(q)}} &(x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(2)}) \succ_{x_i^{(q)}} \\
\succ_{x_i^{(q)}} &(x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(2)}, x_{i+3}^{(1)}) \succ_{x_i^{(q)}} (x_i^{(q)}, x_{i+1}^{(2)}, x_{i+2}^{(1)}, x_{i+3}^{(2)})
\end{aligned}
$$

A brief analysis shows that the system is of the form $X_0 \longleftrightarrow X_1 \longrightarrow X_2 \longrightarrow X_3 \longrightarrow X_0$. Like in the original version of Algorithm 3, we start by matching the elements of the sets that are in reciprocal relations, that is, the elements of $X_0 \cup X_1$ using Algorithm 2. After partial matching $\mu_{X_0 \cup X_1}$, we can consider the preferences of the members of $\mu_{X_0 \cup X_1}(X_0 \cup X_1) = \{y^{(q)} = (x_0^{(q)}, x_1^{(q)}) : q = 1, 2\}$ given by $\succ'_{y^{(q)}}$ based on the preferences of $x_1^{(q)}$ on members of $X_2 \cup X_3$ as follows ($q = 1, 2$):

$$
(y^{(q)}, x_2^{(1)}, x_3^{(1)}) \succ'_{y^{(q)}} (y^{(q)}, x_2^{(1)}, x_3^{(2)}) \succ'_{y^{(q)}} (y^{(q)}, x_2^{(2)}, x_3^{(1)}) \succ'_{y^{(q)}} (y^{(q)}, x_2^{(2)}, x_3^{(2)})
$$

The preferences of members of $X_3$ in $\mu(X_0 \cup X_1)$ are based on the preferences of $x_3^{(q)}$ on $X_0$, so the preferences on the tuples are natural restrictions of the original preferences, as listed:

$$
(y^{(1)}, x_2^{(1)}, x_3^{(q)}) \succ_{x_3^{(q)}} (y^{(1)}, x_2^{(2)}, x_3^{(q)}) \succ_{x_3^{(q)}} (y^{(2)}, x_2^{(2)}, x_3^{(q)}) \succ_{x_3^{(q)}} (y^{(2)}, x_2^{(1)}, x_3^{(q)})
$$

Similarly for the members of $X_2$:

$$(y^{(1)}, x_2^{(q)}, x_3^{(1)}) \succ_{x_2^{(q)}} (y^{(2)}, x_2^{(q)}, x_3^{(1)}) \succ_{x_2^{(q)}} (y^{(2)}, x_2^{(q)}, x_3^{(2)}) \succ_{x_2^{(q)}} (y^{(1)}, x_2^{(q)}, x_3^{(2)})$$

Thus, the system can be described as $\mu_{X_0 \cup X_1}(X_0 \cup X_1) \longrightarrow X_2 \longrightarrow X_3 \longrightarrow \mu_{X_0 \cup X_1}(X_0 \cup X_1)$. Moreover, there are exactly two agents in each set. We know from literature (see, e.g. [6], [14], [10], [27]) that stable matching exists in such a system. In the current example, this can be the matching that results with the two triples $(y^{(1)}, x_2^{(1)}, x_3^{(1)})$ and $(y^{(2)}, x_2^{(2)}, x_3^{(2)})$. Recalling the fact that $y^{(q)} = (x_0^{(q)}, x_1^{(q)}) : q = 1, 2$, we obtain the corresponding stable match in the original system, represented by the quadruples $(x_0^{(1)}, x_1^{(1)}, x_2^{(1)}, x_3^{(1)})$ and $(x_0^{(2)}, x_1^{(2)}, x_2^{(2)}, x_3^{(2)})$.

Note that the method described here, similarly as in the case of most of known positive results, uses iteratively the deferred acceptance algorithm. We are convinced that much more can not be proved using this approach, and therefore some new ideas will be necessary to significantly strengthen the existing positive results. On the other hand, there are only a few counterexamples that show that in some circumstances the stable matchings do not need to exist. However, this is far from formulating any reasonable necessary conditions of their existence. Needless to say that the existing sufficient and necessary conditions are far from each other. This motivates us to rewrite the well-known open problem once again.

**Problem 6.** Find a sufficient and necessary condition for the existence of stable matchings in the setting with $k \neq 2$ sets.

Another interesting question is what happens when the preferences are not strict or even incomplete. The problems of these kinds were analyzed from several perspectives, and we are curious what happens when one considers the mixed preference systems.

**Problem 7.** Does stable matching always exist in the mixed preference systems analogous to the one described in Section 3 if the preference relations do not need to be antisymmetric or total? If not, then what additional assumptions must be imposed?

## Acknowledgement

The authors are grateful to two anonymous reviewers for their valuable comments and suggestions made on the previous draft of this manuscript.

## References

[1] Abraham, D. J., Irving, R. W., and Manlove, D. F. Two algorithms for the Student-Project Allocation problem. *Journal of Discrete Algorithms 5*, 1 (2007), 73–90.

[2] Alkan, A. Nonexistence of stable threesome matchings. *Mathematical Social Sciences 16*, 2 (1988), 207–209.

[3] Arenas, J., and Torres-Martínez, J. P. Reconsidering the existence of stable solutions in three-sided matching problems with mixed preferences. *Journal of Combinatorial Optimization 45*, 2 (2023), 62.

[4] Biró, P., and McDermid, E. Three-sided stable matchings with cyclic preferences. *Algorithmica 58*, 1 (2010), 5–18.

[5] Biró, P., van de Klundert, J., Manlove, D., Pettersson, W., Andersson, T., Burnapp, L., Chromy, P., Delgado, P., Dworczak, P., Haase, B., Hemke, A., Johnson, R., Klimentova, X., Kuypers, D., Nanni Costa, A., Smeulders, B., Spieksma, F., Valentín, M. O., and Viana, A. Modelling and optimisation in European kidney exchange programmes. *European Journal of Operational Research 291*, 2 (2021), 447–456.

[6] Boros, E., Gurvich, V., Jaslar, S., and Krasner, D. Stable matchings in three-sided systems with cyclic preferences. *Discrete Mathematics 289*, 1 (2004), 1–10.

[7] Cseh, Á., Escamocher, G., Genç, B., and Quesada, L. A collection of constraint programming models for the three-dimensional stable matching problem with cyclic preferences. *Constraints 27*, 3 (2022), 249–283.

[8] Cui, L., and Jia, W. Cyclic stable matching for three-sided networking services. *Computer Networks 57*, 1 (2013), 351–363.

[9] Danilov, V. I. Existence of stable matchings in some three-sided systems. *Mathematical Social Sciences 46*, 2 (2003), 145–148.

[10] Eriksson, K., Sjöstrand, J., and Strimling, P. Three-dimensional stable matching with cyclic preferences. *Mathematical Social Sciences 52*, 1 (2006), 77–87.

[11] Fajardo-Delgado, D., Hernández-Bernal, C., Sánchez-Cervantes, M. G., Trejo-Sánchez, J. A., Espinosa-Curiel, I. E., and Molinar-Solis, J. E. Stable matching of users in a ridesharing model. *Applied Sciences 12*, 15 (2022), 7797.

[12] Gale, D., and Shapley, L. S. College admissions and the stability of marriage. *The American Mathematical Monthly 69*, 1 (1962), 9–15.

[13] Hitsch, G. J., Hortaçsu, A., and Ariely, D. Matching and sorting in online dating. *American Economic Review 100*, 1 (2010), 130–163.

[14] Hofbauer, J. d-dimensional stable matching with cyclic preferences. *Mathematical Social Sciences 82*, (2016), 72–76.

[15] Huang, C.-C. Circular stable matching and 3-way kidney transplant. *Algorithmica 58*, 1 (2010), 137–150.

[16] Irving, R. W. An efficient algorithm for the "stable roommates" problem. *Journal of Algorithms 6*, 4 (1985), 577–595.

[17] Irving, R. W. Stable marriage and indifference. *Discrete Applied Mathematics 48*, 3 (1994), 261–272.

[18] Irving, R. W., and Leather, P. The complexity of counting stable marriages. *SIAM Journal on Computing 15*, 3 (1986), 655–667.

[19] Kamiyama, N. A new approach to the Pareto stable matching problem. *Mathematics of Operations Research 39*, 3 (2014), 851–862.

[20] Kato, A. Complexity of the sex-equal stable marriage problem. *Japan Journal of Industrial and Applied Mathematics 10*, 1 (1993), 1–19.

[21] Klimentova, X., Biró, P., Viana, A., Costa, V., and Pedroso, J. P. Novel integer programming models for the stable kidney exchange problem. *European Journal of Operational Research 307*, 3 (2023), 1391–1407.

[22] Knuth, D. E. *Stable marriages and their relations to other combinatorial problems: An introduction to the mathematical analysis of algorithms* (Aisenstadt Chair Collection). Presses de l'Université de Montréal, 1976 (in French).

[23] Lahiri, S. Three-sided matchings and separable preferences. In *Contributions to Game Theory and Management, Vol. 2. Collected papers presented on the Second International Conference Game Theory and Management, June 26-27, 2008* (St. Petersburg, Russia, 2009), L. A. Petrosjan and N. A. Zenkevich, Eds., Graduate School of Management SPbU, pp. 251–259.

[24] Lam, C.-K., and Plaxton, C. G. On the existence of three-dimensional stable matchings with cyclic preferences. *Theory of Computing Systems 66*, 3 (2022), 679–695.

[25] Lerner, E. A counterexample of size 20 for the problem of finding a 3-dimensional stable matching with cyclic preferences. *Discrete Applied Mathematics 333* (2023), 1–12.

[26] Manlove, D. F. The structure of stable marriage with indifference. *Discrete Applied Mathematics 122*, 1 (2002), 167–181.

[27] Pashkovich, K., and Poirrier, L. Three-dimensional stable matching with cyclic preferences, 2018. Working paper version available from arXiv: https://doi.org/10.48550/arXiv.1807.05638.

[28] Perach, N., and Anily, S. Stable matching of student-groups to dormitories. *European Journal of Operational Research 302*, 1 (2022), 50–61.

[29] Ren, M., Yang, L., Jiang, H., Chen, J., and Zhou, Y. *Energy-delay tradeoff in helper-assisted NOMA-MEC systems: A four-sided matching algorithm*, 2023. Working paper version available from arXiv: https://doi.org/10.48550/arXiv.2301.10624.

[30] Roth, A. E., Sönmez, T., and Ünver, M. U. Pairwise kidney exchange. *Journal of Economic Theory 125*, 2 (2005), 151–188.

[31] Ünver, M. U. On the survival of some unstable two-sided matching mechanisms. *International Journal of Game Theory 33*, 2 (2005), 239–254.

[32] Zhang, F., and Zhong, L. Three-sided matching problem with mixed preferences. *Journal of Combinatorial Optimization 42* (2021), 928–936.