Check for updates

Three-Dimensional Matching Instances Are Rich in Stable Matchings

Guillaume Escamocher(✉) and Barry O'Sullivan

Insight Centre for Data Analytics, Department of Computer Science, University College Cork, Cork, Ireland
guillaume.escamocher@insight-centre.org,
barry.osullivan@insight-centre.org

Abstract. Extensive studies have been carried out on the Stable Matching problem, but they mostly consider cases where the agents to match belong to either one or two sets. Little work has been done on the three-set extension, despite the many applications in which three-dimensional stable matching (3DSM) can be used. In this paper we study the Cyclic 3DSM problem, a variant of 3DSM where agents in each set only rank the agents from one other set, in a cyclical manner. The question of whether every Cyclic 3DSM instance admits a stable matching has remained open for many years. We give the exact number of stable matchings for the class of Cyclic 3DSM instances where all agents in the same set share the same master preference list. This number is exponential in the size of the instances. We also show through empirical experiments that this particular class contains the most constrained Cyclic 3DSM instances, the ones with the fewest stable matchings. This would suggest that not only do all Cyclic 3DSM instances have at least one stable matching, but they each have an exponential number of them.

1 Introduction

1.1 Different Kinds of Stable Matchings Problems

Stable matching is the problem of establishing groups of agents according to their preferences, such that there is no incentive for the agents to change their groups. It has a plethora of applications; the two most commonly mentioned are the assignment of students to universities and of residents to hospitals [11].

Most of the research done on stable matching focuses on the cases where the agents belong to either one set (the stable roommates problem) or two (the stable marriage problem). Far fewer studies have looked at the three-dimensional version, where every agent in each set has a preference order over *couples* of agents from the two other sets, even though it is naturally present in many situations. It can be used for example to build market strategies that link suppliers, firms and buyers [15], or in computer networking systems to match data sources, servers and end users [4]. Even some applications like kidney exchange, which

© Springer International Publishing AG, part of Springer Nature 2018
W.-J. van Hoeve (Ed.): CPAIOR 2018, LNCS 10848, pp. 182–197, 2018.
https://doi.org/10.1007/978-3-319-93031-2_13

is traditionally associated with the stable roommates problem, can be easily represented in a three-dimensional form [2].

One of the possible reasons for this lack of interest is that, while it is well-known that every two-dimensional matching instance admits at least one stable matching [7], some three-dimensional matching instances do not [1]. In fact, determining whether a given three-dimensional matching instance has a stable matching is NP-Complete [12,16], even when each agent's preference order is required to be consistent, or when ties are allowed in the rankings [8].

Due to the hardness of the general problem, other restrictions on the preferences have been proposed. With lexicographically acyclic preferences, then there is always a stable matching, which can easily be found in quadratic time [5]. If the preferences are lexicographically cyclic, then the complexity of determining whether a given instance admits a stable matching is still open. For the latter kind of preferences, some instances with no stable matching have been found [3].

Most of the work in this paper is about the cyclic Three-Dimensional Stable Matching Problem. In this version, agents from the first set only rank agents from the second set, agents from the second set only rank agents from the third set, and agents from the third set only rank agents from the first set. Hardness results are also known for this variant: imposing a stronger form of stability [9], or allowing incomplete preference lists [2] both make it NP-Complete to determine whether an instance admits a stable matching. However, the standard problem with complete preference lists is still open. It has actually been around for decades and is considered "hard and outstanding" [17]. Few results about it have been found since its formulation. To date, it is only known that there always exists a stable matching for instances with at most 3 agents in each set [3], a result that has been subsequently improved to include instances with sets of size 4 [6].

1.2 Master Preference Lists

Whatever the type of matching problem studied, it is generally assumed that the preferences of each agent are independent from the preferences of the other agents in the same set. However, this is often not the case in real-life settings. Indeed, it is not hard to imagine that in many cases hospitals will have close, if not identical, preferences over which residents they want to accept, or that firms will often compete for the same top suppliers. Shared preference lists have also been used to assign university students to dormitory rooms, where the students were ranked according to a combination of academic record and socio-economic characteristics [13].

Imposing a master preference list on all agents within a same set leads to a much more constrained problem. In most cases, the only stable matching is obtained by grouping the best ranked agent of each set together, the second best ones together, and so on. This is true for the two-dimensional matching problem [10]. As we explain in Sect. 3.4, this is also true for many versions of the three-dimensional Stable Matching (3DSM) problem.

We will show in this paper that the cyclic 3DSM problem is singular with regard to the number of stable matchings for instances with master preference lists. Not only is this number more than one, but it is extremely large, exponential in the size of the instances. We will also demonstrate through experiments that cyclic 3DSM instances with master preference lists are the most constrained cyclic 3DSM instances, the ones with the fewest stable matchings. Combining these two results would indicate that it is the natural behavior of *all* cyclic 3DSM instances to have an exponential number of stable matchings, making cyclic 3DSM an attractive problem when looking for tractable three-dimensional matching classes.

We divide the paper in the following manner. In Sect. 2, we recall the standard definitions of stability for the cyclic 3DSM problem, as well as the notion of master preference list. In Sect. 3, we give the exact number of stable matchings for instances with master preference lists. The bulk of Sect. 3 is about the cyclic 3DSM problem, but we also take a look at instances from other matching conventions to see how they compare. We empirically show in Sect. 4 that in cyclic 3DSM, instances with more balanced preferences have more stable matchings, while instances with the most unanimous preferences (master lists) have the fewest stable matchings. We also observe this behavior in other matching problems. Finally, we reflect on these findings in the conclusion.

2 General Definitions

Definition 1. *A* cyclic three dimensional stable matching instance*, or* cyclic 3DSM instance *comprises:*

- *Three* agent sets $A = \{a_1, a_2, \ldots, a_n\}$, $B = \{b_1, b_2, \ldots, b_n\}$ *and* $C = \{c_1, c_2, \ldots, c_n\}$ *each containing* n agents*.*
- *For each agent* $a \in A$*, a strict preference order* $>_a$ *over the agents from the set* B*. For each agent* $b \in B$*, a strict preference order* $>_b$ *over the agents from the set* C*. For each agent* $c \in C$*, a strict preference order* $>_c$ *over the agents from the set* A*.*

The number n of agents in each agent set is the *size* of the instance.

Definition 2. *A* master list cyclic 3DSM instance *is a cyclic 3DSM instance where all agents within a same set have the same preference order.*

All master list instances of a same size are isomorphic, so from now on we will assume that every master list cyclic 3DSM instance of size n satisfies the following condition: for all i and j such that $1 \leq i < j \leq n$, for all agents $a \in A$, $b \in B$ and $c \in C$, we have $b_i >_a b_j$, $c_i >_b c_j$ and $a_i >_c a_j$. In other words, the agents in each agent set are ranked according to the preferences of the previous agent set.

Definition 3. *Let* I *be a cyclic 3DSM instance of size* n*. A* matching *for* I *is a set* $M = \{t_1, t_2, \ldots, t_n\}$ *of* n *triples such that each triple contains exactly one agent from each agent set of* I*, and each agent of* I *is represented exactly once in* M*.*

Definition 4. *Let I be a cyclic 3DSM instance and let M be a matching for I. Let t be a triple containing the three agents $a \in A$, $b \in B$ and $c \in C$. Let $a_M \in A$, $b_M \in B$ and $c_M \in C$ be three agents such that a and b_M are in the same triple of M, b and c_M are in the same triple of M, and c and a_M are in the same triple of M. Then we say that t is a* blocking triple *for M if $b >_a b_M$, $c >_b c_M$ and $a >_c a_M$.*

Note that, from the definition, no two agents in a blocking triple t can be in the same triple t_i in the matching M.

Definition 5. *Let I be a cyclic 3DSM instance and let M be a matching for I. We say that M is a* stable matching *for I if there is no blocking triple for M.*

We present an example of a master list cyclic 3DSM instance and of a matching in Fig. 1. The dots represent the agents and the lines represent the triples in the matching $M = \{\langle a_1, b_3, c_2\rangle, \langle a_2, b_1, c_1\rangle, \langle a_3, b_4, c_4\rangle, \langle a_4, b_2, c_3\rangle\}$. The triple $\langle a_1, b_2, c_1\rangle$ is a blocking triple because a_1 prefers b_2 over the agent it got in M, b_2 prefers c_1 over the agent it got in M, and c_1 prefers a_1 over the agent it got in M. Therefore M is not stable.

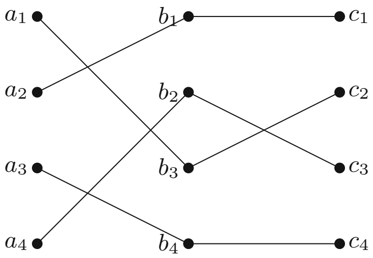

Fig. 1. A matching M for a master list cyclic 3DSM instance of size 4.

3 Stable Matchings for Master List Instances

3.1 Preliminary Notions

In this section we present the main theoretical result of the paper: a function f such that $f(n)$ is the exact number of stable matchings for a master list cyclic 3DSM instance of size n. In the proof, we will consider two kinds of matchings: divisible and indivisible.

Definition 6. *Let I be a master list cyclic 3DSM instance of size n with three agent sets $A = \{a_1, a_2, \ldots, a_n\}$, $B = \{b_1, b_2, \ldots, b_n\}$ and $C = \{c_1, c_2, \ldots, c_n\}$, and let M be a matching for I. We say that M is* divisible *if there exists some p such that $0 < p < n$ and:*

– *for all* i, j *such that* a_i *and* b_j *are in the same triple of* M*, we have* $i \leq p \Leftrightarrow j \leq p$.
– *for all* i, j *such that* a_i *and* c_j *are in the same triple of* M*, we have* $i \leq p \Leftrightarrow j \leq p$.

We also say that p *is a* divider *of* M.

We say that a matching that is not divisible is *indivisible*. Note that a same divisible matching can have several dividers. To illustrate the notion of divisible matching, we present in Fig. 2 two examples of divisible matchings for a master list cyclic 3DSM instance of size 5. The first matching has two dividers, 1 and 4, while the second matching has one divider, 3. The matching from Fig. 1 was an example of an indivisible matching.

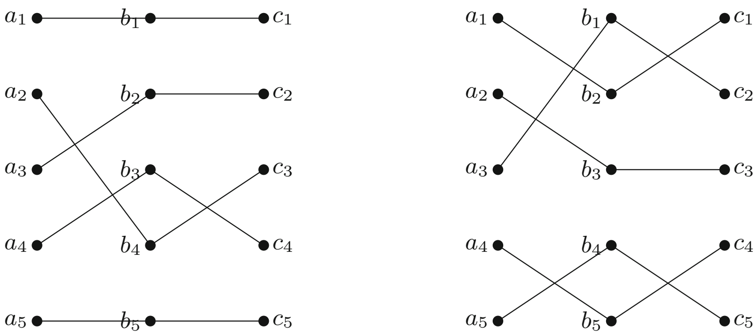

Fig. 2. Two divisible matchings.

3.2 Indivisible Matchings

Before presenting the function f that counts the number of total matchings for a given size n, we look at the function g that counts the number of indivisible matchings. It turns out that this function is very simple: $g(n) = 1$ if $n = 1$ and $g(n) = 3$ otherwise.

Proposition 1. *Let* $n \geq 2$ *be an integer and let* I *be a master list cyclic 3DSM instance of size* n*. Then there are exactly* 3 *indivisible stable matchings for* I.

To prove the proposition, we are going to define a matching $IndMat_n$ for each size n, then show that $IndMat_n$ is stable, and finally show that any indivisible stable matching for a master list cyclic 3DSM instance of size n is either $IndMat_n$ or one of the two matchings that are isomorphic to $IndMat_n$ by rotation of the agent sets.

Definition 7. *Let* I *be a master list cyclic 3DSM instance with three agent sets* $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_n\}$ *and* $C = \{c_1, \ldots, c_n\}$ *of size* $n > 0$*. We call* $IndMat_n$ *the matching* $\{t_1, t_2, \ldots, t_n\}$ *for* I *defined in the following way:*

1. *If* $n = 1$, $IndMat_n = \{\langle a_1, b_1, c_1\rangle\}$. *If* $n = 2$, $IndMat_n = \{\langle a_1, b_2, c_1\rangle, \langle a_2, b_1, c_2\rangle\}$.
   If $n = 3$, $IndMat_n = \{\langle a_1, b_2, c_1\rangle, \langle a_2, b_3, c_3\rangle, \langle a_3, b_1, c_2\rangle\}$.
2. *If* $n > 3$*:* $t_1 = \langle a_1, b_2, c_1\rangle$, $t_2 = \langle a_2, b_3, c_4\rangle$ *and* $t_3 = \langle a_3, b_1, c_2\rangle$.
3. *If* $n > 3$ *and* $n \equiv 1 \mod 3$*:* $t_n = \langle a_n, b_n, c_{n-1}\rangle$.
4. *If* $n > 3$ *and* $n \equiv 2 \mod 3$*:* $t_{n-1} = \langle a_{n-1}, b_n, c_{n-2}\rangle$ *and* $t_n = \langle a_n, b_{n-1}, c_n\rangle$.
5. *If* $n > 3$ *and* $n \equiv 0 \mod 3$*:* $t_{n-2} = \langle a_{n-2}, b_{n-1}, c_{n-3}\rangle$, $t_{n-1} = \langle a_{n-1}, b_n, c_n\rangle$ *and* $t_n = \langle a_n, b_{n-2}, c_{n-1}\rangle$.
6. *If* $i \equiv 1 \mod 3$, $i > 3$ *and* $i \leq n - 3$*:* $t_i = \langle a_i, b_{i+1}, c_{i-1}\rangle$, $t_{i+1} = \langle a_{i+1}, b_{i+2}, c_{i+3}\rangle$ *and* $t_{i+2} = \langle a_{i+2}, b_i, c_{i+1}\rangle$.

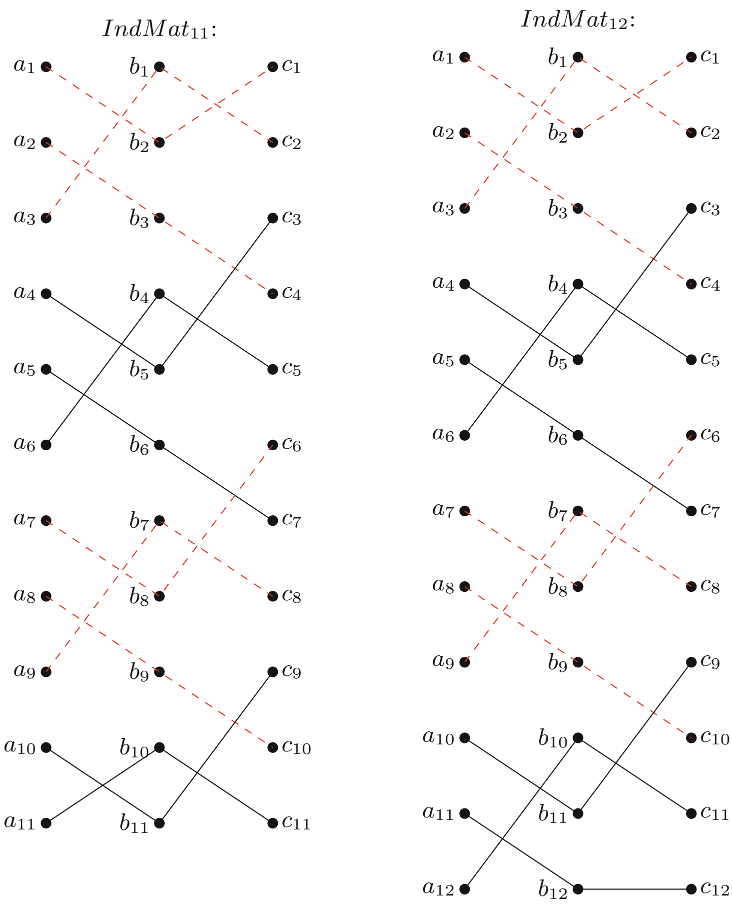

Fig. 3. The matchings $IndMat_{11}$ and $IndMat_{12}$. (Color figure online)

$IndMat_n$ can be seen as a set of $n/3$ gadgets G_i, with each G_i composed of the three triples t_i, t_{i+1} and t_{i+2} for each i such that $i \equiv 1 \mod 3$. All these gadgets are isomorphic by translation, apart from the first one and the last one. Figure 3 shows two examples of $IndMat_n$ matchings, one with $n = 11$ and the other with $n = 12$. Both matchings are almost identical, but they each illustrate a different type of final gadget. To help the reader clearly visualize the structure of the matchings, we alternated the colors and line styles of the gadgets.

Lemma 1. *Let I be a master list cyclic 3DSM instance of size n. Then $IndMat_n$ is a stable matching for I.*

Proof. For each i such that $i \equiv 1 \mod 3$ and $i \leq n$, let G_i be the gadget composed of the three triples t_i, t_{i+1} and t_{i+2}. Let $t = \langle a, b, c \rangle$ be a blocking triple for $IndMat_n$.

Suppose first that a, b and c are from the same gadget G_i. From Definition 7, we have $t_i = \langle a_i, b_{i+1}, c_j \rangle$ with either $j = i - 1$ or $j = i$, $t_{i+1} = \langle a_{i+1}, b_{i+2}, c_k \rangle$ with either $k = i + 2$ or $k = i + 3$, and $t_{i+2} = \langle a_{i+2}, b_i, c_{i+1} \rangle$. Two agents from a same triple in a matching cannot be part of a same blocking triple for this matching, and there are three triples in G_i, therefore at least one agent from each triple is part of t. b_{i+1} cannot be part of t because it got assigned c_i, which is the best ranked agent among the agents of C that are in G_i. Likewise, c_j cannot be part of t because it got assigned a_j, which is the best ranked agent among the agents of A that are in G_i. So $a = a_i$. b_{i+2} cannot be part of t, because it is not as well ranked as b_{i+1}, the agent from B that got assigned to a_i in $IndMat_n$. So $b = b_i$. So c_{i+1} cannot be part of t, because it shares a triple in $IndMat_n$ with b_i. So $c = c_k$. So either $c = c_{i+2}$ or $c = c_{i+3}$. However, neither c_{i+2} nor c_{i+3} is as well ranked as c_{i+1}, the agent from B that got assigned to b_i. So it is not possible to have a blocking triple for $IndMat_n$ with all three agents of the triple in the same gadget G_i.

Suppose now that a, b and c are not all in the same three-triple gadget. Then there must be i and j with $i < j$ such that a is in G_i and b is in G_j, or b is in G_i and c is in G_j, or c is in G_i and a is in G_j. From Definition 7 and by construction of the gadgets, if a is in G_i and b is in G_j then the agent from B that got assigned to a in $IndMat_n$ is better ranked in the preference order of a than b is, so t cannot be a blocking triple. Similarly, if c is in G_i and a is in G_j then the agent from A that got assigned to c in $IndMat_n$ is better ranked in the preference order of c than a is, so t cannot be a blocking triple. So b is in G_i and c is in G_j. The only way for c to be better ranked in the preference order of b than the agent from C that got assigned to b in $IndMat_n$ is if $j = i + 3$, $b = b_{i+2}$ and $c = c_{j-1} = c_{i+2}$. Let r be such that $a = a_r$ and let s be such that b_s got assigned to a_r in $IndMat_n$. If $r \leq i + 2$, then from Definition 7 we have $b_s \geq_a b$ and a prefers the agent of B it got assigned in $IndMat_n$ over b. If $r \geq i + 3$, then from Definition 7 we have $a_{i+1} \geq_c a$ and c prefers the agent of A it got assigned in $IndMat_n$ over a. Either way, t cannot be a blocking triple and we have the result. □

Lemma 2. *Let I be a master list cyclic 3DSM instance of size n and let M be a matching for I. If there are some a_i (respectively b_i, c_i) and b_j (respectively c_j, a_j) in the same triple of M such that $j \geq i + 2$, then M is not stable.*

Proof. We only do the proof for a_i and b_j, as the other two cases are exactly the same after rotation of the agent sets.

Suppose that we have a matching $M = \{t_1, t_2, \ldots, t_n\}$ for I such that $t_i = \langle a_i, b_j, c_k \rangle$ with $j \geq i+2$. The triples $t_{i'} = \langle a_{i'}, b_{j'}, c_{k'} \rangle$ and $t_{i''} = \langle a_{i''}, b_{j''}, c_{k''} \rangle$ will be used in the proof. We distinguish the two cases $k \leq i$ and $k > i$.

- $k \leq i$: from the pigeonhole principle we know that at least one of the $i+1$ agents $\{c_1, c_2, \ldots, c_{i+1}\}$ got assigned an agent $a_{i'} \in A$ such that $i < i'$. Let $k' \leq i+1$ be such that $c_{k'}$ is one such agent. There cannot be a bijection in M among the sets $\{b_1, b_2, \ldots, b_{i+1}\}$ and $\{c_1, c_2, \ldots, c_{i+1}\}$ because b_j is not in the former but got assigned an agent from the latter. So we know that at least one of the agents $b_1, b_2, \ldots, b_{i+1}$ got assigned an agent $c_{k''} \in C$ such that $k'' > i+1$. Let $j'' \leq i+1$ be such that $b_{j''}$ is such an agent. We have $b_{j''} >_{a_i} b_j$ (because $j \geq i+2$), $c_{k'} >_{b_{j''}} c_{k''}$ (because $k' \leq i+1 < k''$) and $a_i >_{c_{k'}} a_{i'}$ (because $i < i'$), so $\langle a_i, b_{j''}, c_{k'} \rangle$ is a blocking triple for M and M is not stable.
- $k > i$: none of the i agents $\{c_1, c_2, \ldots, c_i\}$ got assigned a_i, so from the pigeonhole principle we know that at least one of them got assigned an agent $a_{i'} \in A$ such that $i < i'$. Let $k' \leq i$ be such that $c_{k'}$ is one such agent. Also from the pigeonhole principle, we know that among the $i+1$ agents $\{b_1, b_2, \ldots, b_{i+1}\}$ at least one of them got assigned an agent $c_{k''} \in C$ such that $k'' > i$. Let $j'' \leq i+1$ be such that $b_{j''}$ is such an agent. We have $b_{j''} >_{a_i} b_j$ (because $j \geq i+2$), $c_{k'} >_{b_{j''}} c_{k''}$ (because $k' \leq i < k''$) and $a_i >_{c_{k'}} a_{i'}$ (because $i < i'$), so $\langle a_i, b_{j''}, c_{k'} \rangle$ is a blocking triple for M and M is not stable. □

Lemma 3. *Let I be a master list cyclic 3DSM instance of size n and let M be an indivisible stable matching for I. Then either M is $IndMat_n$, or M is one of the two matchings that are isomorphic to $IndMat_n$ by rotation of the agent sets.*

Proof. Let $M = \{t_1, t_2, \ldots, t_n\}$ be an indivisible matching for I. Without loss of generality, assume that a_i is in t_i for every i. We are going to show by induction that every t_i is equal to the i^{th} triple of $IndMat_n$, modulo rotation of the agent sets.

Case i = 1: if a_1, b_1 and c_1 are in three different triples of M, then they form a blocking triple for M because they each prefer each other over the agent they got assigned in M. If they are in the same triple of M, then $n = 1$ because M is indivisible; since $IndMat_1 = \langle a_1, b_1, c_1 \rangle$ from Definition 7.1, we have the Lemma. So we can assume from now on that $n > 1$ and that exactly two agents among a_1, b_1 and c_1 are in the same triple in M. We will assume that a_1 and c_1 are the ones in the same triple. All three cases are isomorphic by rotation of the agent sets; the case we chose will lead to $IndMat_n$, while the other two would have led to one of the matchings that are isomorphic to $IndMat_n$ by rotation of the agent sets.

We know that $t_1 = \langle a_1, b_j, c_1 \rangle$ with $j \geq 2$. From Lemma 2, we know that $j \leq 2$. So $t_1 = \langle a_1, b_2, c_1 \rangle$.

Inductive step: suppose now that the triples $t_1, t_2, \dots, t_p$ are the same as the first p triples of $IndMat_n$ for some p such that $1 \leq p < n$. We are going to prove that t_{p+1} is the same triple as the $(p+1)^{th}$ triple in $IndMat_n$. With p_{last} the highest $p < n$ such that $p \equiv 1 \mod 3$, there are five possibilities to consider.

- $p \equiv 1 \mod 3$ and $p < p_{last}$: since p_{last} is also congruent to 1 modulo 3, we also have $p \leq n-3$. From Definition 7 we know that the agents that have already been assigned are $a_1, a_2, \dots, a_p$ from A, $b_1, b_2, \dots, b_{p-1}$ and b_{p+1} from B, and $c_1, c_2, \dots, c_p$ from C. So b_p and some c_k are in the same triple, with $k \geq p+1$. From Lemma 2, we know that $k \leq p+1$. So b_p and c_{p+1} are in the same triple of M. a_{p+1} cannot be in this triple, because otherwise either n would be equal to $p+1$ or $p+1$ would be a divider of M. So some a_i is assigned to c_{p+1} with $i \geq p+2$. From Lemma 2, we know that $i \leq p+2$. So a_{p+2} is assigned to c_{p+1} and $t_{p+2} = \langle a_{p+2}, b_p, c_{p+1} \rangle$ (which proves the next bullet point). So $t_{p+1} = \langle a_{p+1}, b_j, c_k \rangle$ for some $j \geq p+2$ and $k \geq p+2$. From Lemma 2, we have $j \leq p+2$ and therefore $b_j = b_{p+2}$. We cannot have $k = p+2$, because otherwise either n would be equal to $p+2$ or $p+2$ would be a divider of M. So $k \geq p+3$. From Lemma 2, $k \leq p+3$ and therefore $t_{p+1} = \langle a_{p+1}, b_{p+2}, c_{p+3} \rangle$, which from Definition 7.2 and 7.6 is the same triple as the $(p+1)^{th}$ triple of $IndMat_n$.
- $p \equiv 2 \mod 3$ and $p < p_{last}$: let $p' = p-1$. So $p' \equiv 1 \mod 3$ and $p' < p_{last}$. So from the proof of the previous bullet point we know that $t_{p'+2} = \langle a_{p'+2}, b_{p'}, c_{p'+1} \rangle$. So $t_{p+1} = \langle a_{p+1}, b_{p-1}, c_p \rangle$, which from Definition 7.2 and 7.6 is the same triple as the $(p+1)^{th}$ triple of $IndMat_n$.
- $p \equiv 0 \mod 3$ and $p < p_{last}$: from Definition 7 we know that the agents that have already been assigned are $a_1, a_2, \dots,\ a_p$ from A, $b_1, b_2, \dots, b_p$ from B, and $c_1, c_2, \dots, c_{p-1}$ and c_{p+1} from C. So c_p is in the same triple as some a_i with $i \geq p+1$. From Lemma 2, we know that $i \leq p+1$. So $i = p+1$ and $t_{p+1} = \langle a_{p+1}, b_j, c_p \rangle$ for some $j \geq p+1$. From Lemma 2, we know that $j \leq p+2$ so either $b_j = b_{p+1}$ or $b_j = b_{p+2}$. If $n = p+1$, then we have $t_{p+1} = \langle a_{p+1}, b_{p+1}, c_p \rangle$ which from Definition 7.3 is equal to the $(p+1)^{th}$ triple of $IndMat_n$. If $n > p+1$, then $b_j = b_{p+2}$, because otherwise $p+1$ would be a divider of M, and $t_{p+1} = \langle a_{p+1}, b_{p+2}, c_p \rangle$. So from Definition 7.5 and 7.6, t_{p+1} is equal to the $(p+1)^{th}$ triple of $IndMat_n$.
- $p = p_{last}$: since $p < n$, either $n = p+1$ or $n = p+2$. If $n = p+1$, then from Definition 7 only the agents $a_{p+1} \in A$, $b_p \in B$ and $c_{p+1} \in C$ have not been assigned. So $t_{p+1} = \langle a_{p+1}, b_p, c_{p+1} \rangle$, which from Definition 7.4 is the same as the $(p+1)^{th}$ tuple of $IndMat_n$. If on the other hand $n = p+2$, then from Definition 7 only the agents a_{p+1} and a_{p+2} in A, b_p and b_{p+2} in B, and c_{p+1} and c_{p+2} in C remain to be assigned. From Lemma 2, c_{p+2} cannot be assigned to b_p, so c_{p+1} is assigned to b_p. a_{p+1} cannot be in the same triple as these two agents, because otherwise $p+1$ would be a divider of M. So a_{p+2} is in the same triple as b_p and c_{p+1} and $t_{p+2} = \langle a_{p+2}, b_p, c_{p+1} \rangle$. Consequently, the three other remaining agents are assigned together in the triple $t_{p+1} = \langle a_{p+1}, b_{p+2}, c_{p+2} \rangle$. This is from Definition 7.5 the same triple as the $(p+1)^{th}$ triple of $IndMat_n$.

– $p = p_{last} + 1$: since $p < n$, $n = p + 1$ and only one agent from each agent set has not been assigned. From Definition 7, we know that these agents are $a_{p+1} \in A$, $b_{p-1} \in B$ and $c_p \in C$. So $t_{p+1} = \langle a_{p+1}, b_{p-1}, c_p \rangle$, which is from Definition 7.5 the same triple as the $(p+1)^{th}$ triple of $IndMat_n$.

We did not consider the case where $p = p_{last} + 2$, because it cannot happen if $p < n$.

We have shown that t_1 is equal to the first triple in $IndMat_n$ and that if $n > 1$ and the first p triples of M are equal to the first triples of $IndMat_n$ for $1 \leq p < n$, then t_{p+1} is equal to the $(p+1)^{th}$ triple of $IndMat_n$. By induction, this completes the proof. □

Lemmas 1 and 3 together prove Proposition 1.

3.3 Main Theorem

Before introducing the Theorem, we need one last Lemma.

Lemma 4. *Let I be a master list cyclic 3DSM instance of size n and let p be an integer such that $1 \leq p < n$. Then the number of stable matchings for I that admit p as their lowest divider is equal to $f(n-p)$ times the number of indivisible stable matchings for a master list cyclic 3DSM instance of size p.*

Proof. Let $M = \{t_1, t_2, \ldots, t_n\}$ be a matching for I such that p is the lowest divider of M and $a_i \in t_i$ for each i. Let $M_1 = \{t_1, t_2, \ldots, t_p\}$ and let $M_2 = \{t_{p+1}, t_{p+2}, \ldots, t_n\}$. Since p is the lowest divider of M, M_1 is indivisible. We show that a triple $\langle a_i, b_j, c_k \rangle$ cannot be a blocking triple for M if it is across the divider p, that is if it fulfills one of the three following conditions: $i \leq p$ and $j > p$, $j \leq p$ and $k > p$, $k \leq p$ and $i > p$. Let t be such a triple. Without loss of generality, assume that $i \leq p$ and $j > p$. Let b_m be the agent of B assigned to a_i in M. Since p is a divider of M, we have $m \leq p < j$. So $b_m >_{a_i} b_j$. So t cannot be a blocking triple for M. So any blocking triple for M is either a blocking triple for M_1 or a blocking triple for M_2. So M is stable if and only if both M_1 and M_2 are stable, and we have the result. □

We now have all the tools we need to state and prove the Theorem:

Theorem 1. *Let f be the function from $\mathbb{N}$ to $\mathbb{N}$ such that $f(1) = 1$, $f(2) = 4$ and for every n such that $n > 2$ we have $f(n) = 2f(n-2) + 2f(n-1)$. Let $n > 0$ be an integer and let I be a master list cyclic 3DSM instance of size n. Then there are exactly $f(n)$ stable matchings for I.*

Proof. For $n = 1$ and $n = 2$ there are 1 and 4 matchings respectively, and they are all trivially stable. Suppose now that $n > 2$. Let g be the function such that for each integer $q \leq n$, $g(q)$ is the number of indivisible matchings for master list cyclic 3DSM instances of size q. For each p such that $1 \leq p < n$, let f_p be the function such that $f_p(n)$ is the number of stable matchings for a master list cyclic 3DSM instance of size n that have p as their lowest divider. We have:

$$f(n) = (\sum_{p=1}^{n-1} f_p(n)) + g(n)$$

From Lemma 4 we have:

$$f(n) = (\sum_{p=1}^{n-1} g(p)f(n-p)) + g(n)$$

From Proposition 1, we know that $g(1) = 1$ and that $g(p) = 3$ for every $p \geq 2$. Therefore we have:

$$\begin{aligned}
f(n) &= f(n-1) + 3f(n-2) + (\sum_{p=3}^{n-1} 3f(n-p)) + 3 \\
&= 2f(n-2) + f(n-1) + f(n-2) + (\sum_{p=2}^{n-2} 3f(n-1-p)) + 3 \\
&= 2f(n-2) + f(n-1) + (\sum_{p=1}^{n-2} g(p)f(n-1-p)) + g(n-1) \\
&= 2f(n-2) + 2f(n-1)
\end{aligned}$$

□

Note that $f(n) > 2f(n-1)$ for all n, so master list cyclic 3DSM instances have a number of stable matchings which is exponential in their size.

3.4 Other Matching Problems

An obvious follow-up to our main theorem would be to determine how the number of stable matchings for master list cyclic 3DSM instances compares to the number of stable matchings for master list instances of matching problems with different rules. We first look at what happens when imposing a stronger form of stability, which is based on the notion of weakly blocking triple [2].

Definition 8. *Let I be a cyclic 3DSM instance and let M be a matching for I. Let t be a triple containing the three agents $a \in A$, $b \in B$ and $c \in C$ such that t does not belong to M. Let $a_M \in A$, $b_M \in B$ and $c_M \in C$ be three agents such that a and b_M are in the same triple of M, b and c_M are in the same triple of M, and c and a_M are in the same triple of M. Then we say that t is a* weakly blocking triple *for M if $b \geq_a b_M$, $c \geq_b c_M$ and $a \geq_c a_M$.*

Informally, a triple t is weakly blocking for some matching M if each agent of t either prefers t over the triple it got assigned to in M, or is indifferent. Note that since we explicitly require t not to belong in M, at least one of the three preferences will be strict.

Definition 9. *Let I be a cyclic 3DSM instance and let M be a matching for I. We say that M is a* strongly stable matching *for I if there is no weakly blocking triple for M.*

Strong stability is more restrictive than standard stability, therefore we can expect a lower number of stable matchings. Indeed, the number of strongly stable matchings is always equal to 1 for master list instances.

Proposition 2. *Let I be a master list cyclic 3DSM instance of size n. Then the number of strong stable matchings for I is equal to 1.*

Proof. Let M be a strongly stable matching for I. Let p be the largest integer such that $0 \leq p \leq n$ and for each $0 < q \leq p$ the triple $\langle a_q, b_q, c_q \rangle$ belongs to M. Suppose that $p < n$. Therefore the triple $t = \langle a_{p+1}, b_{p+1}, c_{p+1} \rangle$ does not belong to M. Let i, j and k be such that a_i is assigned to c_{p+1} in M, b_j is assigned to a_{p+1} in M and c_k is assigned to b_{p+1} in M. We know that for each q such that $1 \leq q \leq p$, a_q, b_q and c_q have been assigned to each other in M. So $i \geq p + 1$, $j \geq p + 1$ and $k \geq p + 1$. So $b_{p+1} \geq_a b_j$, $c_{p+1} \geq_b c_k$ and $a_{p+1} \geq_c a_i$. So from Definition 8, t is a weakly blocking triple for M. Therefore $p = n$ and the only possible strongly stable matching is the matching M_0 which contains the triple $\langle a_p, b_p, c_p \rangle$ for each $1 \leq p \leq n$.

It only remains to prove that M_0 is strongly stable. Let $t = \langle a_i, b_j, c_k \rangle$ be a triple. If $i > j$, then $b_i <_{a_i} b_j$ and a_i strictly prefers b_i, the agent from B it got assigned to in M_0, over b_j, the agent from B it got assigned to in t, which means that t cannot be a weakly blocking triple for M_0. So if t is weakly blocking for M_0, then $i \leq j$. By the same reasoning, if t is weakly blocking for M_0, then $j \leq k$ and $k \leq i$. So if t is weakly blocking for M_0, then $i = j = k$. But in this case, t is in M_0 by construction and therefore cannot be a weakly blocking triple for M_0. So there is no weakly blocking triple for M_0. Therefore M_0 is strongly stable, which completes the proof. □

The same very simple proof can be used to show that for at least two more matching problems, namely lexicographically cyclic 3DSM (defined in [3]) and lexicographically acyclic 3DSM (defined in [5]), master list instances of size n have exactly one stable matching, which is also of the form $\{\langle a_1, b_1, c_1 \rangle, \langle a_2, b_2, c_2 \rangle, \ldots, \langle a_n, b_n, c_n \rangle\}$. This result holds for the extensively studied two-dimensional stable matching (2DSM) too [10]. This indicates that master list cyclic 3DSM instances offer many more stable matchings than their master list counterparts in some others of the most widely used matching problems.

4 Stable Matchings for Instances Without Master Preference Lists

If master list cyclic 3DSM instances have fewer stable matchings than other instances from the same problem, then our main Theorem implies that all cyclic 3DSM instances have a number of stable matchings exponential in their size. This is not a trivial assumption, so we need further study to determine what happens to the number of stable matchings when considering other instances.

In this section, we empirically investigate the evolution of the number of stable matchings when going from a master list cyclic 3DSM instance, which can be seen as an instance with unanimous preferences, to its opposite: a cyclic 3DSM instance with evenly split preferences. We will need a few definitions to formally describe our procedure.

Definition 10. *Let I be a cyclic 3DSM instance of size n. Let g and g' be two agents of I, such that strictly more than $n/2$ agents prefer g over g'. Then we call* adding an ML-step *to I the act of switching g and g' in the preference list of an agent that prefers g over g'.*

Definition 11. *We say that a cyclic 3DSM instance is* perfectly split *if it is not possible to add an ML-step to I.*

Our experiments consist in starting from a master list cyclic 3DSM instance and randomly adding ML-steps until we reach a perfectly split instance. We summarize our results in Figs. 4 and 5. In Fig. 4, we added ML-steps to 1000 starting master list cyclic 3DSM instances of size 8, until getting a perfectly split instance. Note that the number of steps required to arrive to a perfectly split instance is not the same in each of the runs, so the last few data points represent fewer than 1000 instances. This explains why the "minimum" and "maximum" plots seem to converge towards the "average" one at the end. The exact number of instances represented by each data point can be found in Table 1. The numbers of stable matchings were obtained using Cachet [14], an exact SAT model counter.

The figure clearly confirms what we suspected: the cyclic 3DSM instances with the fewest stable matchings are the ones with master preference lists, or are at least very similar to these instances. More precisely, the number of stable matchings seems to initially increase steadily when going away from master list instances, before plateauing when a certain number of ML-steps has been added.

Table 1 contains the exact numbers for Cyclic 3DSM instances. The last line, not represented in Fig. 4, describes the number of stable matchings for 1000 completely random cyclic 3DSM instances, whose construction was not related in any way to master list instances or ML-steps. This serves as a control experiment, to make it clear that our results are not dependent on the particular way that we build our instances.

Figure 5 illustrates the results of the same experiments on two other stable matching variants: 2DSM and cyclic 3DSM with strong stability, both of size 8.

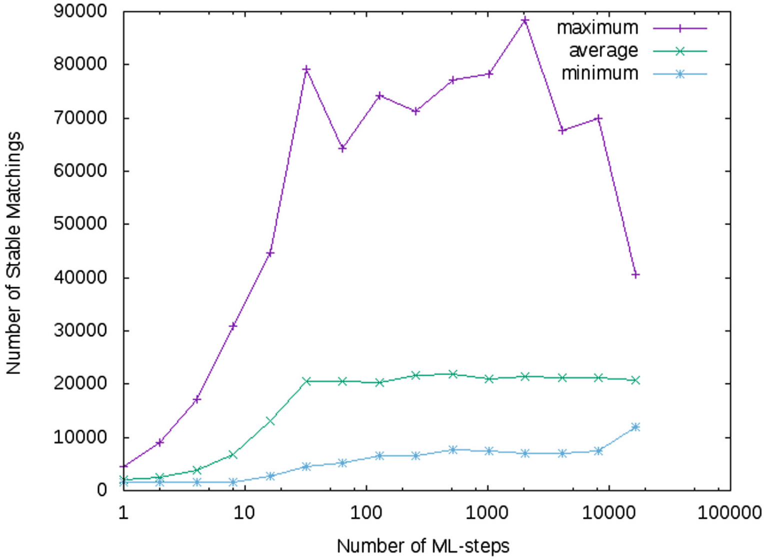

Fig. 4. Number of stable matchings when adding ML-steps to cyclic 3DSM instances.

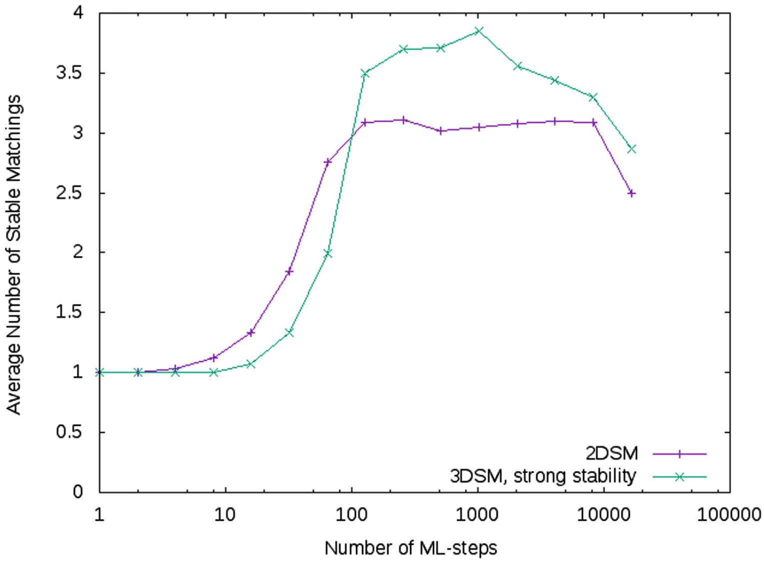

Fig. 5. Number of stable matchings for 2DSM instances and of strongly stable matchings for cyclic 3DSM instances.

Here again, we added ML-steps to 1000 starting master list instances of each problem. The numbers of (strongly) stable matchings are much lower for these two problems, yet we can observe the same behavior of a steady increase followed

Table 1. Number of stable matchings in 3DSM instances.

# ML-steps	# instances	# SM (minimum)	# SM (average)	# SM (maximum)
0	1000	1552	1552	1552
1	1000	1552	2009	4544
2	1000	1508	2561	8917
4	1000	1552	3779	17242
8	1000	1552	6669	30831
16	1000	2681	13095	44766
32	1000	4529	20442	79129
64	1000	5201	20599	64234
128	1000	6615	20216	74233
256	1000	6615	21683	71376
512	1000	7716	21965	77204
1024	993	7515	21084	78257
2048	953	6989	21478	88481
4096	760	7085	21235	67604
8192	301	7515	21201	69996
16384	24	11904	20713	40688
Random	1000	4932	20521	105070

by a plateau. This empirically shows that instances with master preference lists are linked with very high constrainedness in many stable matching problems.

5 Conclusion

We have given the exact number of stable matchings for cyclic 3DSM instances with master preference lists. This number is 1 for many other stable matching problems, but it is exponential in the case of the Cyclic 3DSM problem.

We have also shown through experiments that despite their high number of stable matchings, cyclic 3DSM instances with master preference lists are the most constrained instances of the cyclic 3DSM problem, the ones with the fewest stable matchings, a behavior that mirrors what can be observed in other standard matching problems.

Combining these two results, we propose the following conjecture: each cyclic 3DSM instance has a number of stable matchings exponential in its size. If true, this would make the cyclic 3DSM problem a very interesting object of research when looking for positive and/or tractable three-dimensional matching results.

Acknowledgements. This research has been funded by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

1. Alkan, A.: Nonexistence of stable threesome matchings. Math. Soc. Sci. **16**(2), 207–209 (1988)
2. Biró, P., McDermid, E.: Three-sided stable matchings with cyclic preferences. Algorithmica **58**(1), 5–18 (2010)
3. Boros, E., Gurvich, V., Jaslar, S., Krasner, D.: Stable matchings in three-sided systems with cyclic preferences. Discrete Math. **289**(1–3), 1–10 (2004)
4. Cui, L., Jia, W.: Cyclic stable matching for three-sided networking services. Comput. Netw. **57**(1), 351–363 (2013)
5. Danilov, V.: Existence of stable matchings in some three-sided systems. Math. Soc. Sci. **46**(2), 145–148 (2003)
6. Eriksson, K., Sjöstrand, J., Strimling, P.: Three-dimensional stable matching with cyclic preferences. Math. Soc. Sci. **52**(1), 77–87 (2006)
7. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. Am. Math. Mon. **69**(1), 9–15 (1962)
8. Huang, C.-C.: Two's company, three's a crowd: stable family and threesome roommates problems. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 558–569. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75520-3_50
9. Huang, C.: Circular stable matching and 3-way kidney transplant. Algorithmica **58**(1), 137–150 (2010)
10. Irving, R.W., Manlove, D., Scott, S.: The stable marriage problem with master preference lists. Discrete Appl. Math. **156**(15), 2959–2977 (2008)
11. Manlove, D.F.: Algorithmics of Matching Under Preferences, Series on Theoretical Computer Science, vol. 2. WorldScientific, Singapore (2013)
12. Ng, C., Hirschberg, D.S.: Three-dimensional stable matching problems. SIAM J. Discrete Math. **4**(2), 245–252 (1991)
13. Perach, N., Polak, J., Rothblum, U.G.: A stable matching model with an entrance criterion applied to the assignment of students to dormitories at the Technion. Int. J. Game Theory **36**(3–4), 519–535 (2008)
14. Sang, T., Bacchus, F., Beame, P., Kautz, H.A., Pitassi, T.: Combining component caching and clause learning for effective model counting. In: The Seventh International Conference on Theory and Applications of Satisfiability Testing, SAT 2004, Vancouver, BC, Canada, 10–13 May 2004, Online Proceedings (2004)
15. Stuart Jr., H.W.: The supplier—firm—buyer game and its m-sided generalization. Math. Soc. Sci. **34**(1), 21–27 (1997)
16. Subramanian, A.: A new approach to stable matching problems. SIAM J. Comput. **23**(4), 671–700 (1994)
17. Woeginger, G.J.: Core stability in hedonic coalition formation. In: van Emde Boas, P., Groen, F.C.A., Italiano, G.F., Nawrocki, J., Sack, H. (eds.) SOFSEM 2013. LNCS, vol. 7741, pp. 33–50. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35843-2_4