

A Necessary and Sufficient Condition for the Existence of a Complete Stable Matching

JIMMY J. M. TAN*

*Department of Computer and Information Science, National Chiao Tung University,
Hsinchu, Taiwan, Republic of China*

Received January 15, 1989; revised May 4, 1990

The stable roommates problem is a well-known problem of matching n people into $n/2$ disjoint pairs so that no two unmatched persons both prefer each other to their partners under the matching. We call such a matching “a complete stable matching.” It is known that a complete stable matching may not exist. Irving described an $O(n^2)$ algorithm that would find one complete stable matching if there is one, or would report that none exists. In this paper, we give a necessary and sufficient condition for the existence of a complete stable matching; namely, the non-existence of any odd party, which will be defined subsequently. We define a new structure called a “stable partition,” which generalizes the notion of a complete stable matching, and prove that every instance of the stable roommates problem has at least one such structure. We also show that a stable partition contains all the odd parties, if there are any. Finally we have an $O(n^2)$ algorithm that finds one stable partition which in turn gives all the odd parties. © 1991 Academic Press, Inc.

1. INTRODUCTION

The stable roommates problem is defined as follows. There is a set S of n people. Each person i has a preference list that includes a subset S_i of $S - \{i\}$ and a rank ordering (most preferred first) of the persons in S_i . For person i , the set S_i has the meaning that the only persons he is willing to be matched with are those in S_i . A complete matching M is a partition of the n persons into $n/2$ disjoint pairs of roommates such that for every pair $\{a, b\}$ in M , a is on b 's list and b is on a 's list. A complete matching M is *unstable* if there are two persons who are not matched and both

*This research was supported by National Science Council of Republic of China under Grant NSC 77-0408-E009-18.

prefer each other to their partners under the matching; such a pair is said to *block* the matching M . A complete matching which is not unstable is called *stable*. The problem here is to find a complete stable matching.

Gale and Shapley [1] proposed this problem and gave the following example to show that a complete stable matching may not exist. In this example anyone paired with person 4 will cause instability.

Person	Preference list
1	2 3 4
2	3 1 4
3	1 2 4
4	Arbitrary

Knuth [11] demonstrated that multiple solutions could exist and asked for an efficient algorithm to generate a solution, if one exists. Irving [6] solved this problem by proposing an $O(n^2)$ algorithm that would find one complete stable matching if there is one, or would report that none exists. Recently, Gusfield [4] gave a detailed analysis of the structure of the complete stable matchings for those instances that contain at least one solution. We refer the reader to the recently published book of Gusfield and Irving [5] for a general introduction and comprehensive discussion of this and related problems. However, for a given instance without any complete stable matching, it is not immediately clear why it does not have one except by tracing Irving's algorithm step by step. Therefore, it is desirable to have a "compact" condition that would support the existence or non-existence of a solution.

In this paper, we give a necessary and sufficient condition for the existence of a complete stable matching; namely, the non-existence of any odd party, which will be defined subsequently. We define a new structure called a "stable partition," which generalizes the notion of a complete stable matching and prove that every instance of the stable roommates problem has at least one such structure. We also show that a stable partition contains all the odd parties, if there are any. Finally, we have an $O(n^2)$ algorithm, which is a modified version of Irving's [6], that finds one stable partition which in turn gives all the odd parties.

Remark. One of the main contributions of this paper is to provide an easy way to convince a person (not a computer) who knows only the definition of the problem that a given problem instance has no stable matching. Once a stable partition has been computed, and it has an odd party, we can easily convince the person as follows: show them the stable partition and point out the odd party; then show the person a proof of Theorem 3.3 (it shows that if there is a stable partition with an odd party

then there can be no stable matching) which has been specialized to that particular stable partition. The proof of Theorem 3.3 is short and easy, and so this provides a simple convincing demonstration for the person that no stable matching is possible. In contrast, while the execution trace of Irving's algorithm provides a good machine-verifiable proof that no stable matching exists, it provides a proof for a human only if that person understands Irving's algorithm.

2. DEFINITIONS

An instance of the stable roommates problem is specified by its preference lists. We define a preference relation to be a pair (S, T) , where S is a set of n persons, and T is the table of preference lists of these n people. Throughout this paper, we assume that the table T is *symmetric*, i.e., person a is on b 's list if and only if b is on a 's. If person b is on the preference list of person a , then write $(a|b)$ to denote the entry b in a 's preference list.

Define $r(a, b) = k$, if person b occupies position k in a 's preference list. If $r(a, b) < r(a, c)$, it means that person a prefers b to c .

Let (S, T) be a preference relation, and let A be a subset of S . Denote $|A|$ the cardinality of set A . A cyclic permutation $\Pi(A) = \langle a_1, a_2, a_3, \dots, a_k \rangle$ of the persons in A , where $k = |A|$, is called a *semi-party permutation* if one of the following three conditions holds:

- (i) $|A| \geq 3$, a_{i+1} and a_{i-1} are on a_i 's preference list, and $r(a_i, a_{i+1}) < r(a_i, a_{i-1})$, $i = 1, 2, 3, \dots, k$ (subscripts modulo k);
- (ii) $|A| = 2$, and a_{i-1} is on a_i 's preference list, $i = 1, 2$ (subscript modulo 2);
- (iii) $|A| = 1$.

For example,

Person		Preference list
1	.	2 5
2	.	3 1
3	Arbitrary	4 2 Arbitrary
4	.	5 3
5	.	1 4
6	Arbitrary	7 Arbitrary
7		6

In this example, $\langle 1, 2, 3, 4, 5 \rangle$ and $\langle 6, 7 \rangle$ are semi-party permutations for persons $\{1, 2, 3, 4, 5\}$ and $\{6, 7\}$, respectively.

With a specified semi-party permutation $\Pi(A) = \langle a_1, a_2, \dots, a_k \rangle$ for persons in A , we classify the entries in the preference lists of A into the following categories:

- (I) If $|A| \geq 3$, we say that entry $(a_i|b)$ is
 - (i) a *superior* entry with respect to $\Pi(A)$, if $r(a_i, b) < r(a_i, a_{i-1})$;
 - (ii) an *inferior* entry with respect to $\Pi(A)$, if $r(a_i, a_{i-1}) \leq r(a_i, b)$ (note: the inequality is “ \leq ” not “ $<$ ”);
 - (iii) a *party* entry with respect to $\Pi(A)$, if $b = a_{i+1}$ or $b = a_{i-1}$; for $i = 1, 2, 3, \dots, k$ (subscripts modulo k).
- (II) If $|A| = 2$, i.e., $k = 2$, we say that $(a_i|b)$ is
 - (i) a *superior* entry with respect to $\Pi(A)$, if $r(a_i, b) < r(a_i, a_{i-1})$;
 - (ii) an *inferior* entry with respect to $\Pi(A)$, if $r(a_i, a_{i-1}) < r(a_i, b)$ (note: the inequality is “ $<$ ” not “ \leq ”);
 - (iii) a *party* entry with respect to $\Pi(A)$, if $b = a_{i-1}$; for $i = 1, 2$ (subscripts modulo 2).
- (III) If $|A| = 1$, we say that $(a_i|b)$ is a *superior* entry with respect to $\Pi(A)$ for every person b on a_i 's preference list.

In the above definition, if there is no ambiguity, we will omit the words “with respect to $\Pi(A)$.”

Given a preference relation (S, T) , a *stable partition* Π of (S, T) consists of a partition of the set S ; $S = \cup_{i=1}^m A_i$, $A_i \cap A_j = \emptyset$ if $i \neq j$, and a specified semi-party permutation $\Pi(A_i)$ for each A_i , $i = 1, 2, \dots, m$, such that the following *stable condition* is satisfied:

If $(a|b)$ is a superior entry then $(b|a)$ is an inferior entry.

Remark. Recall that the table T of preference lists is assumed to be symmetric, i.e., a is on b 's list if and only if b is on a 's. If T is not symmetric, then the above stable condition should be modified as follows: If $(a|b)$ is a superior entry then either $(b|a)$ is an inferior entry or a is not on b 's preference list.

In the context of the above definition, the associated semi-party permutation $\Pi(A_i)$ is called a *party permutation* for A_i ; and each A_i is called a *party*. An *odd party* (*even party*, respectively) is a party having odd (even, respectively) cardinality. More precisely, these terms are defined with respect to the given stable partition Π . If there are ambiguities, we will say that A_i is a party in Π , and $(a|b)$ is a superior entry in Π , etc.

A stable partition Π is specified by its party permutations and will be denoted by $\Pi = \{\Pi(A_1), \Pi(A_2), \Pi(A_3), \dots, \Pi(A_m)\}$. Persons a and b are

said to be a *matching pair* (or *matched*) in Π if $\{a, b\}$ forms a two-person party in Π . A subset A of the all-person set S is said to form a *party* (an odd party, respectively), if there exists a stable partition Π such that A is a party (an odd party, respectively) in Π .

We have two remarks about the definition of a stable partition.

Remark 1. $(a|b)$ is a party entry in stable partition Π if and only if $(b|a)$ is.

Remark 2. Let A be a party in stable partition Π :

If $|A| \geq 3$, then any entry on a 's list is either superior or inferior in Π , for each $a \in A$.

If $|A| = 2$, then the party entry $(a|b)$ is neither superior nor inferior in Π , for each $a \in A$.

For a given preference relation, it is not clear whether any stable partition exists. We will show later that such a partition does exist and describe an efficient algorithm to find one. On the other hand, there may exist many distinct stable partitions. However, we will prove that any two stable partitions have exactly the same odd parties (not only having the same persons involved in a corresponding odd party, but also with the same party permutation). Therefore, the existence of an odd party depends on the preference relation, not on a particular stable partition. Moreover, we will show that there exists a complete stable matching if and only if there does not exist any odd party.

To illustrate the above definitions, we give the following example:

Person		Preference list			
1	.	2	.	5	.
2	.	3	.	1	.
3		4		2	
	Superior		Superior		Inferior
4	.	5	.	3	.
5	.	1	.	4	.
6			Superior		
7	.	8	.	10	.
8		9		7	
	Superior		Superior		Inferior
9	.	10	.	8	.
10	.	7	.	9	.
11			12		
	Superior				Inferior
12			11		

A stable partition is shown above, there are four parties; namely $A_1 = \{1, 2, 3, 4, 5\}$, $A_2 = \{6\}$, $A_3 = \{7, 8, 9, 10\}$, $A_4 = \{11, 12\}$, and $\Pi = \{\langle 1, 2, 3, 4, 5 \rangle, \langle 6 \rangle, \langle 7, 8, 9, 10 \rangle, \langle 11, 12 \rangle\}$. To complete this example, we just have to fill in all the other entries and follow the rule that whenever $(a|b)$ is a superior entry, then $(b|a)$ is inferior.

A preference relation may have more than one stable partition. We can identify at least two other stable partitions in the above example:

$$\Pi_1 = \{\langle 1, 2, 3, 4, 5 \rangle, \langle 6 \rangle, \langle 7, 8 \rangle, \langle 9, 10 \rangle, \langle 11, 12 \rangle\}$$

and

$$\Pi_2 = \{\langle 1, 2, 3, 4, 5 \rangle, \langle 6 \rangle, \langle 8, 9 \rangle, \langle 7, 10 \rangle, \langle 11, 12 \rangle\}.$$

3. STABLE PARTITIONS AND STABLE MATCHINGS

The notion of the stable partition generalizes that of the complete stable matching in the following sense.

PROPOSITION 3.1. *A complete stable matching is a stable partition in which every party has cardinality two and vice versa.*

Proof. This is directly from the definitions. $M = \{\{a_i, b_i\} | i = 1 \text{ to } n/2\}$ is a complete stable matching if and only if $\Pi = \{\langle a_i, b_i \rangle | i = 1 \text{ to } n/2\}$ is a stable partition. \square

PROPOSITION 3.2. *A stable partition without any odd party induces a complete stable matching.*

Proof. Suppose that Π is a stable partition without any odd party. Let A be an even party in Π with party permutation $\langle a_1, a_2, a_3, \dots, a_{2k} \rangle$, $k \geq 2$. Then decomposing party A into k matching pairs $\langle a_1, a_2 \rangle, \langle a_3, a_4 \rangle, \dots, \langle a_{2k-1}, a_{2k} \rangle$, we have a new stable partition $\Pi' = (\Pi - \{\langle a_1, a_2, \dots, a_{2k} \rangle\}) \cup \{\langle a_1, a_2 \rangle, \langle a_3, a_4 \rangle, \dots, \langle a_{2k-1}, a_{2k} \rangle\}$. This is because every superior entry in Π' is a superior entry in Π , and every inferior entry in Π , other than the party entries, is an inferior entry in Π' . By continuing to decompose any even party having cardinality four or more, eventually we obtain a stable partition in which every party has cardinality two. \square

The next theorem shows that having an odd party is a sufficient condition for not having a complete stable matching.

THEOREM 3.3. *For any preference relation, if it has a stable partition containing an odd party, then there does not exist any complete stable matching.*

Proof. Suppose not. Let Π be a stable partition which contains at least one odd party and let M be a complete stable matching. Hence M is a stable partition in which every party has cardinality two. Let S_π be the set of persons whose partners in M are superior entries in Π , and let I_π be the set of persons whose partners in M are inferior entries in Π . For the stability of Π , every person in S_π has an M -partner in I_π . So $|S_\pi| \leq |I_\pi|$. Consider a party A in stable partition Π , and let $\langle a_1, a_2, \dots, a_k \rangle$ be the associated party permutation of A . For the stability of M , no two consecutive persons a_i and a_{i+1} (subscripts modulo k) can be in I_π , otherwise a_i and a_{i+1} block the matching M .

Therefore, if A is an odd party in Π , then

$$|A \cap S_\pi| > |A \cap I_\pi|.$$

And if A is an even party in Π , then

$$|A \cap S_\pi| \geq |A \cap I_\pi|.$$

Since stable partition Π contains at least one odd party, we have

$$|S_\pi| = \sum_{\substack{A_i \text{ is a party} \\ \text{in } \Pi}} |A_i \cap S_\pi| > \sum_{\substack{A_i \text{ in a party} \\ \text{in } \Pi}} |A_i \cap I_\pi| = |I_\pi|.$$

This is a contradiction, and the theorem follows. \square

We will prove that the inverse of Theorem 3.3 is also true. Let us point out one use of the above theorem. Suppose that we want to convince a person that a particular roommates instance has no complete stable matching. There are two efficient ways available to do so; one is the execution trace of Irving's algorithm, and the other is the construction of a stable partition containing an odd party. Of course, to find a stable partition one has to, for example, apply the algorithm described later in this paper; once obtained it gives a certificate of non-solvability. A person needs only to know the definition of the roommates problem to follow the proof of Theorem 3.3. So anyone who has the certificate in hand for an instance can use it to prove that no complete stable matching is possible. While the trace of Irving's algorithm is only convincing if one understand the correctness of the whole algorithm, which may not be trivial.

4. IRVING'S ALGORITHM

Given Proposition 3.2 and Theorem 3.3, we now must prove that having an odd party is also a necessary condition for not having a complete stable

matching. Our proof is a constructive one, and is based on a modified version of Irving's algorithm [6], which finds a complete stable matching if one exists, or reports that none exists. We show that there exists at least one stable partition for every preference relation, and it leads to an $O(n^2)$ algorithm to find one. We also obtain an interesting result that every stable partition contains exactly the same odd parties. Combining these facts and Theorem 3.3, we conclude that having an odd party is a characterization for not having a complete stable matching.

Before describing our modifications and extensions, we first summarize Irving's algorithm [6].

IRVING'S ALGORITHM.

Input: A table of preference lists.

Output: Either output a complete stable matching or report that none exists.

The algorithm successively deletes entries from preference lists until either each person has only one entry on his list, or until someone has no entries. In the first case, the entries specify a complete stable matching, and in the second case, there are no solutions. For ease of exposition, we need a definition.

DEFINITION. The current set of preference lists at any point in the algorithm is called a *table*.

Irving's algorithm is divided into two phases:

Phase 1. The first phase of the algorithm is based on a sequence of "proposals" from one person to another. When a person receives a proposal, then

(i) he holds it for consideration if it is the best proposal he has so far received, simultaneously rejecting any poorer proposal already held;

(ii) he rejects it at once if he already holds a better proposal.

Each person proposes to the others in the order in which they appear in his preference list, pausing when a promise of consideration is received; and resuming his sequence of proposals upon any subsequent rejection.

This phase of the algorithm will terminate either

(i) with every person holding a proposal, or

(ii) with one person rejected by everyone.

We note that, in this phase of the algorithm, each person both receives and makes proposals; unlike the classical stable marriage case [1, 12], where only the persons of one sex make proposals and the members of the opposite sex receive proposals.

(a) Original table						(b) Phase 1 table				
Person	Preference list					Person	Preference list			
1	3	2	4	5	6	1	3	2	4	
2	6	3	1	5		2	6	3	1	5
3	5	4	2	1		3	5	4	2	1
4	1	3	6			4	1	3	6	
5	1	2	6	3		5	2	6	3	
6	1	4	5	2		6	4	5	2	

FIGURE 1

It is proved in [1, 6, 12] that if b rejects a in the proposal sequence, then a and b cannot be partners in any complete stable matching. Moreover, at any stage of the proposal process, if a proposes to b , then in any complete stable matching:

- (i) a cannot have a better partner than b ;
- (ii) b cannot have a worse partner than a .

Therefore, the following entries can be deleted from the table when person b holds a proposal from a :

- (i) every entry $(b|x)$, with $r(b, a) < r(b, x)$;
- (ii) every entry $(x|b)$, where entry $(b|x)$ is described in case (i).

The remaining set of lists at the end of phase 1 is called the *Phase 1 table*.

EXAMPLE (Fig. 1). Checking the resulting table at the end of Phase 1,

- (i) if some list is empty then there is no complete stable matching,
- (ii) if every list contains just one person then the lists specify a complete stable matching,
- (iii) if none of (i) and (ii) hold and some of the lists contain more than one person, this brings us to the second phase of the algorithm.

Phase 2. In this phase, entries are also deleted successively from the table, in a very special way, until either each person has only one entry on his list, or until someone has no entries. The basic idea in the second phase of the algorithm is that of a rotation.

DEFINITION. Let T be a table. A *rotation* R exposed in table T , denoted by $R = (a_1, a_2, \dots, a_r)|(b_1, b_2, \dots, b_r)$, is a cyclic sequence a_1, a_2, \dots, a_r of distinct people, where b_i is the first person on a_i 's list in T , $i = 1$ to r , such that the first person b_i on a_i 's list is the second person on a_{i-1} 's, $i = 1$ to r (subscript modulo r).

EXAMPLE. In Fig. 1B, $(1, 5, 2)|(3, 2, 6)$ is an exposed rotation in Phase 1 table.

DEFINITION. A table T is said to be *in the second phase*, or *in Phase 2*, if

(i) the table has been subjected to Phase 1 reduction as described before,

(ii) the table has been subjected to zero or more Phase 2 reductions (so-called rotation eliminations), as described below.

DEFINITION. Let T be a table. A complete stable matching M is said to be *contained in* a table T , if and only if every matched pair in M is in T , i.e., a is on b 's list and b is on a 's list for every pair $\{a, b\}$ in M .

Consider a table T in the second phase of the algorithm; the following crucial result is proved in [6].

THEOREM 4.1. *Let T be a table in phase 2 and let $R = (a_1, a_2, \dots, a_r)|(b_1, b_2, \dots, b_r)$ be a rotation exposed in T . Then*

(i) *in any complete stable matching contained in T , either a_i and b_i are matched for all values of i , or for no value of i ;*

(ii) *if there is a complete stable matching in which a_i and b_i are matched, then there is another in which they are not.*

We need one more definition.

DEFINITION. Let $R = (a_1, a_2, \dots, a_r)|(b_1, b_2, \dots, b_r)$ be a rotation exposed in table T . Rotation R is said to be *eliminated from T* , if the following entries are deleted from T :

(i) every entry $(b_i|x)$ with $r(b_i, a_{i-1}) < r(b_i, x)$, $i = 1$ to r (subscript modulo r);

(ii) every entry $(x|b_i)$, where entry $(b_i|x)$ is described in case (i).

As a consequence of Theorem 4.1, if the original problem instance admits a complete stable matching, then there is one contained in the resulting table after eliminating an exposed rotation. Furthermore, Irving [6] proved algorithmically that, given a table T in Phase 2, if there is a person whose current list has more than one entry, then there is a rotation exposed in T . Since the concept of a rotation also plays a central role later in our algorithm for finding a stable partition, we give a formal statement of this result.

PROPOSITION 4.2 (Irving [6]). *Let T be a table in Phase 2; if there is a person whose current list has more than one entry, then there is a rotation exposed in T .*

In summary, the second phase of Irving's algorithm can be stated as follows

(i) While some person has more than one entry on his list, and no list is empty, find and eliminate an exposed rotation.

(ii) If every person has exactly one entry on his list, then the lists specify a complete stable matching.

(iii) If there is an empty list, then there are no complete stable matchings.

EXAMPLE. Consider the example given in Fig. 1a. The Phase 1 table is shown in Fig. 1b. Eliminate exposed rotations successively in the following order, $R_1 = (1, 5, 2)|(3, 2, 6)$, $R_2 = (3, 6)|(5, 4)$, and $R_3 = (2, 4)|(3, 1)$. Then the resulting table is as follows, and $\{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$ is a complete stable matching:

Person	Preference list
1	2
2	1
3	4
4	3
5	6
6	5

The correctness of Irving's algorithm is proved in [6] and will not be repeated here. The overall time complexity is $O(n^2)$. Our modifications and extensions in the following sections will be self-contained and fully proved.

5. PHASE 1 OF THE STABLE PARTITION ALGORITHM

Given a preference relation (S, T) , we now consider the problem of finding a stable partition. Our algorithm basically utilizes the idea used in Irving's algorithm with some modifications and extensions. We examine closely the processes of the two phases in Irving's algorithm, and investigate the effect of these processes on the structure of the stable partitions. By doing so, we discover new information concerning stable partition and obtain an algorithm for finding one. To locate a stable partition, it suffices to identify all its party entries.

DEFINITION. A stable partition Π is said to be *contained in* a table T , if every party entry in Π is in T .

Our first observation is that no entry removed in Phase 1 of Irving's algorithm can be a party entry in any stable partition. The following proposition justifies this fact.

PROPOSITION 5.1. *Given a preference relation, let T_0 be the initial table, and let T_1 be a table obtained from T_0 by removing zero or more entries which are not the party entries in any stable partition. Suppose that person c is the current first person on both a 's and b 's lists in T_1 , and $r(c, a) < r(c, b)$. Then no entry $(c|x)$, with $r(c, a) < r(c, x)$, can be a party entry in any stable partition.*

Proof. Suppose, on the contrary, that some entry $(c|x)$ with $r(c, q) < r(c, x)$ is a party entry in stable partition Π . Then $(c|a)$ is superior in Π , and so $(a|c)$ is inferior in Π . Therefore, in a 's original list, there should be a party entry before the inferior entry $(a|c)$. This is a contradiction, since in a 's list no entry before $(a|c)$ can be a party entry in any stable partition. Hence the proposition follows. \square

So the first phase of our algorithm for finding a stable partition is simply applying the proposal and rejection process, and removing impossible entries from the table, as stated in Proposition 5.1, until each person either

- (i) holds a proposal; or
- (ii) has an empty list.

PROPOSITION 5.2. *No entry removed in Phase 1 can be a party entry in any stable partition. More precisely, if a person a proposes to c , then all the entries $(c|x)$ and $(x|c)$, with $r(c, a) < r(c, x)$, can be removed from the table; no stable partition is affected.*

Proof. This follows by inductively applying Proposition 5.1 during Phase 1, i.e., when a proposes to c . \square

Similarly as before, the table obtained at the end of this phase is also called *the Phase 1 table*. Before going into details of Phase 2, we summarize some useful properties which are either immediately, or are explicit or implicit in [6].

Let T_0 be the original table of preference lists. We assume that the table T_0 is symmetric, i.e., person a appears on b 's list if and only if b appears on a 's. Let T_2 be a resulting table obtained from T_0 after Phase 1 process and after eliminating zero or more rotations. Recall that such a table T_2 is called a table in Phase 2. We note that the Phase 1 table is also a table in Phase 2.

Property 5.3. Let T_2 be a table in Phase 2. Person a appears on b 's list in T_2 if and only if b appears on a 's list in T_2 .

Property 5.4. Let T_2 be a table in Phase 2. Person a appears first on b 's list in T_2 if and only if b appears last on a 's list in T_2 .

Property 5.5. Let T_0 be the original table, and let T_2 be a table in Phase 2 (obtained from T_0). Suppose that entry $(a|b)$ is originally in table T_0 , but is not in T_2 . Let x and y be the persons who appear last on a 's and b 's lists in T_2 , respectively. Then either b appears after x on a 's list in T_0 , or a appears after y on b 's list in T_0 .

Property 5.5 follows directly from the rules of entry deletion in Irving's algorithm. Let (S, T_0) be a preference relation, and let T be a table obtained from T_0 by deleting some entries. Then (S, T) can be viewed as a new preference relation. We say that Π is a stable partition for instance T , if Π is a stable partition of the preference relation (S, T) . From Property 5.5, the following result follows immediately.

PROPOSITION 5.6. *Let T_0 be the original table, and let T_2 be a table in Phase 2 (obtained from T_0).*

(i) *If Π is a stable partition for instance T_0 and Π is contained in T_2 , then Π is a stable partition for instance T_2 .*

(ii) *Conversely, if Π is a stable partition for instance T_2 , then Π is a stable partition for instance T_0 .*

Proof. (i) By Property 5.3, this is trivial.

(ii) This follows directly from Property 5.5; i.e., if $(a|b)$ is superior in T_0 , then $(b|a)$ must be inferior in T_0 . \square

At the end of Phase 1, let T_1 be the resulting table obtained. Then we have the following result.

PROPOSITION 5.7. *In table T_1 , if every person has zero or one entry on his list, then the lists specify a stable partition.*

Proof. By Properties 5.3 and 5.4, if b is the only person on a 's list in T_1 , then a is also the only person on b 's list in T_1 . So a and b form a pair in table T_1 . Let $\{\{a_i, b_i\} | i = 1 \text{ to } k\}$ be the set of all such pairs in T_1 , and let $\{c_j | j = 1 \text{ to } m\}$ be the set of all persons whose lists are empty in T_1 . Then it is obvious that $\Pi = \{\langle a_i, b_i \rangle | i = 1 \text{ to } k\} \cup \{\langle c_j \rangle | j = 1 \text{ to } m\}$ is a stable partition for instance T_1 . Thus, by Proposition 5.6, Π is also a stable partition for the original instance T_0 . \square

So, checking the Phase 1 table at the end of this phase,

- (i) if every person has zero or one entry on his list, then the lists specify a stable partition,
- (ii) if someone has more than one entry on his list, this brings us to the second phase of the algorithm.

6. PHASE 2 OF THE ALGORITHM

In the second phase of our algorithm, we also use rotation elimination to delete more entries from the table. We have to investigate the effect of the rotation elimination on a stable partition. Let T be a table in Phase 2. Suppose that there is a person whose current list in T has more than one entry; then, by Proposition 4.2, there is a rotation exposed in T . In the following, we have a series of results which reveal the relationship between the structure of an exposed rotation and that of a stable partition. We start our discussion by considering the case that the elimination of a rotation results in some list empty, and study the structure of such a rotation. Let us use the following example to get some intuition (especially, for part (iv) of Lemma 6.1 below).

EXAMPLE (Fig. 2). In Fig. 2a, the current table contains an exposed rotation $R = (1, 2, 3, 4, 5, 6)|(4, 5, 6, 7, 1, 2, 3)$. After eliminating R , the lists of all the persons involved in this rotation become empty. Let us write this exposed rotation as $R = (a_1, a_2, \dots, a_r)|(b_1, b_2, \dots, b_r)$ and point out some special properties about it.

- (i) Each person a_k has only two persons on his list.
- (ii) $A = B$, where $A = \{a_1, a_2, \dots, a_r\}$ and $B = \{b_1, b_2, \dots, b_r\}$.

(a) Person	Preference list	(b) Person	Preference list
1	4 5	4	7 1
2	5 6	7	3 4
3	6 7	3	6 7
4	7 1	6	2 3
5	1 2	2	5 6
6	2 3	5	1 2
7	3 4	1	4 5
The rest	The rest	The rest	The rest

FIGURE 2

(iii) This rotation contains an odd number of persons, i.e., r is odd.

(iv) $\{1, 2, 3, 4, 5, 6, 7\}$ forms an odd party with the cyclic permutation $\langle 4, 7, 3, 6, 2, 5, 1 \rangle$ as shown in Fig. 2b.

Let us explain the sequence of this permutation as follows: In this case $r = 2m - 1$, where $r = 7$ and $m = 4$. Consider Fig. 2a, the cyclic permutation starts from the “middle” person, 4 ($= m$), goes down to the “bottom” person, 7 ($= 2m - 1$), then the pattern follows: 4 and 7, 3 and 6, 2 and 5, and ends up at the “top” person 1.

These are not mere coincidences; in fact, we will see in Theorem 6.2, that in some sense this is the only situation that the elimination of a rotation results in some list empty. We now begin to state and prove the results. We need the following lemma later in Theorem 6.2 and in Theorem 6.5; one may wish to skip the proof temporarily, and come back for it afterwards.

LEMMA 6.1. *Let T be a table in Phase 2, and let $R = (a_1, a_2, \dots, a_r) | (b_1, b_2, \dots, b_r)$ be a rotation exposed in T . Suppose that, for some fixed i and j , $a_i = b_j$, and suppose that the second person on a_i 's list in T is the same as the last person on b_j 's, i.e., $b_{i+1} = a_j$ (subscript modulo r). Then*

- (i) *in table T , each person a_k has only two persons on his current list (i.e., b_k and b_{k+1}), for $k = 1, 2, \dots, r$ (subscript modulo r),*
- (ii) *$A = B$, where $A = \{a_1, a_2, \dots, a_r\}$ and $B = \{b_1, b_2, \dots, b_r\}$,*
- (iii) *A has odd cardinality,*
- (iv) *the cyclic permutation $\Pi(A) = \langle a_m, a_{2m-1}, a_{m-1}, a_{2m-2}, \dots, a_{m-k}, a_{2m-(k+1)}, \dots, a_2, a_{m+1}, a_1 \rangle$ is a semi-party permutation for A , where $|A| = r = 2m - 1$.*

Remark. We will see later that, with the cyclic permutation $\Pi(A)$, A forms an odd party in every stable partition.

Proof. With respect to the given table T and the given rotation $R = (a_1, a_2, \dots, a_r) | (b_1, b_2, \dots, b_r)$, we define property (i, j) to be the following property: $a_i = b_j$ and the second person on a_i 's current list is the same as the last person on b_j 's current list. Suppose that, for some fixed i and j , property (i, j) is true. We claim that property $(i - 1, j - 1)$ is also true (index modulo r).

All the subscripts considered below are modulo r . We first prove that if property (i, j) is true, then so is property $(j - 1, i)$. All we have to show is that (1) $a_{j-1} = b_i$, and (2) $b_j = a_i$ (i.e., the second person on a_{j-1} 's list is the same as the last person on b_i 's list). Because property (i, j) is true, we know that $a_i = b_j$, so (2) holds. By property (i, j) again, the second person

on a_i 's list is the last person on his list (since $a_i = b_j$), so there are only two persons on a_i 's current list. Since $a_i = b_j$, b_j also has only two persons on his current list. Then by the definition of a rotation, b_i and b_{i+1} (a_{j-1} and a_j , respectively) are the only two persons on a_i 's list (b_j 's list, respectively). Thus $b_i = a_{j-1}$ and $b_{i+1} = a_j$. This proves (1). Hence property $(j - 1, i)$ holds. Replacing (i, j) by $(j - 1, i)$, it follows that property $(i - 1, j - 1)$ is true.

Repeating this argument shows that property $(i - k, j - k)$ holds, for $k = 0, 1, 2, \dots, r - 1$ (index modulo r). We now start to prove the results:

(i) Since, for some fixed i and j , property $(i - k, j - k)$ holds for $k = 0, 1, 2, \dots, r - 1$, we know that $a_{i-k} = b_{j-k}$ and the second person on a_{i-k} 's current list is the last person on his list. So a_{i-k} has only two persons on his current list, $k = 0, 1, 2, \dots, r - 1$ (subscript modulo r). This proves (i).

(ii) Again by property $(i - k, j - k)$, we have $a_{i-k} = b_{j-k}$ and $b_{i-k+1} = a_{j-k}$, $k = 0, 1, 2, \dots, r - 1$ (subscript modulo r). Thus $A = B$.

(iii) We now show that the cardinality of A is odd. Since $a_{i-k} = b_{j-k}$ and $b_{i-k+1} = a_{j-k}$, for $k = 0, 1, 2, \dots, r - 1$, substituting $k = i - 1$ into the first equality, and $k = j - 1$ into the second, we have $a_1 = b_{j-i+1}$ and $b_{i-j+2} = a_1$ (subscript modulo r):

$$\begin{aligned} \Rightarrow j - i + 1 &= i - j + 2 \pmod{r}. \\ \Rightarrow 2(j - i) &= 1 \pmod{r}. \end{aligned}$$

Since $2(j - i)$ is even, r must be odd. This proves that $|A| = r$ is an odd number.

We need the following result in part (iv):

$$\begin{aligned} \text{Let } m &= j - i \pmod{r} \quad \text{and} \quad 0 \leq m \leq r - 1 \\ \Rightarrow 2m &= 1 \pmod{r}. \end{aligned}$$

Note that $0 \leq m \leq r - 1$, the last equation has only one solution $2m = 1 + r$. So $r = 2m - 1$.

(iv) We claim that the cyclic permutation

$$\Pi(A) = \langle a_m, b_m, a_{m-1}, b_{m-1}, \dots, a_2, b_2, a_1 \rangle$$

is a semi-party permutation for A . Observe that a_k and b_k have only two persons on their current list; the first person on a_k 's, b_k 's, and a_1 's current lists are b_k , a_{k-1} , and b_1 , respectively, $k = 1, 2, \dots, m$ (note $b_1 = a_m$; this is because $b_{i-k+1} = a_{j-k}$, for $k = 0, 1, 2, \dots, r - 1$, substituting $k = i$, we have $b_1 = a_{j-i} = a_m$). Put all this information together and trace the current table T ; the claim follows.

Notice that $b_{i-k+1} = a_{j-k}$, $k = 0, 1, 2, \dots, r-1$; substituting $k = i-t+1$, we have $b_t = a_{(j-i)-1+t}$, $t = 1, 2, \dots, r$ (subscripts modulo r). Since $m = j-i$ (modulo r), so that $b_t = a_{m-1+t}$, $t = 1, 2, \dots, r$ (subscripts modulo r). Thus

$$\begin{aligned} \Pi(A) &= \langle a_m, b_m, a_{m-1}, b_{m-1}, \dots, a_{m-k}, b_{m-k}, \dots, a_2, b_2, a_1 \rangle \\ &= \langle a_m, a_{2m-1}, a_{m-1}, a_{2m-2}, \dots, a_{m-k}, a_{2m-(k+1)}, \dots, a_2, a_{m+1}, a_1 \rangle \end{aligned}$$

and (iv) follows. \square

We are now ready to state and prove the following theorem as claimed before.

THEOREM 6.2. *Let T be a table in Phase 2, and let $R = (a_1, a_2, \dots, a_r)|(b_1, b_2, \dots, b_r)$ be a rotation exposed in T . Suppose that there is a person c whose current list is not empty, but is so after eliminating R . Then (i), (ii), (iii), and (iv) in Lemma 6.1 hold, and $c \in A = \{a_1, a_2, \dots, a_r\}$.*

Proof. Eliminating of rotation R can be divided into three steps:

- (1) Remove entry $(b_i|x)$ and entry $(x|b_i)$, with $r(b_i, a_{i-1}) < r(b_i, x) < r(b_i, a_i)$, for all such x , $i = 1, 2, \dots, r$ (subscript modulo r).
- (2) Remove entry $(b_i|a_i)$, $i = 1, 2, \dots, r$.
- (3) Remove entry $(a_i|b_i)$, $i = 1, 2, \dots, r$.

No list is caused to be empty after step (1), because x is not the last person on b_i 's list and b_i is not the first person on x 's. Observe that after step (1), a_i and b_i have at least two persons on their lists, $i = 1, 2, \dots, r$, e.g., the entries that form the exposed rotation R . During step (2), we delete the last entry $(b_i|a_i)$ on b_i 's lists, $i = 1, 2, \dots, r$. So, after step (2), each b_i still has a_{i-1} on his list, and thus no list is caused to be empty.

However, there is a person whose list was not empty, but is so after step (3). In step (3), we only delete the first entry $(a_i|b_i)$ on a_i 's list, $i = 1, 2, \dots, r$. Therefore, it must be some person a_i whose list becomes empty, and this person a_i has only one entry left on his list after step (2). Note that after step (1), a_i has at least two persons on his list, but after step (2), a_i has only one entry left. So, it must be that $a_i = b_j$, for some j , and that the last person on b_j 's list in T is the same as the second person on a_i 's list in T . Then the theorem follows from Lemma 6.1. \square

Conversely, suppose that we discover an exposed rotation $R = (a_1, a_2, \dots, a_r)|(b_1, b_2, \dots, b_r)$ satisfying (i) to (iv) as stated in Lemma 6.1. Then the lists of every person involved in this rotation become empty after eliminating R . We will see later that such a rotation gives us an odd party

$A = \{a_1, a_2, \dots, a_r\}$ with the party permutation $\Pi(A)$ as stated in (iv) of Lemma 6.1.

From the above observation, if the elimination of rotation R makes some list empty, it indicates an odd party. We shall backup the algorithm to the table, where R was eliminated, and never permit R to be eliminated again. We then separate the preference lists of persons in $A = \{a_1, a_2, \dots, a_r\}$ from the table, and treat the rest of the table as a smaller instance table (in Phase 2). The rest of the table is unchanged, because any person in A is not on the current lists of the rest of the persons. For convenience, such a rotation is called an *odd rotation* (for persons in A).

EXAMPLE.

Persons	Preference list	Person	Preference list
7	1 6	1	2 7
5	6 4	2	3 1
3	4 2	3	4 2
1	2 7	4	5 3
6	7 5	5	6 4
4	5 3	6	7 5
2	3 1	7	1 6
The rest	The rest	The rest	The rest

In this example $(7, 5, 3, 1, 6, 4, 2)|(1, 6, 4, 2, 7, 5, 3)$ is an exposed rotation satisfying (i) to (iv) as stated in Lemma 6.1. Rearranging the table, we see that $\langle 1, 2, 3, 4, 5, 6, 7 \rangle$ is a semi-party permutation.

The following two results are prepared for the case that the elimination of a rotation makes no list empty, even though they hold for odd rotation.

THEOREM 6.3. *Let T be a table in Phase 2, let*

$$R = (a_1, a_2, \dots, a_r)|(b_1, b_2, \dots, b_r)$$

be a rotation exposed in T , and let Π be a stable partition contained in T . If, for some fixed i , $(a_i|b_i)$ is a party entry in Π , then $(a_j|b_j)$ is also a party entry in Π for every $j = 1, 2, \dots, r$.

Proof. All the subscripts considered in the following are modulo r . By the definition of the stable partition, $(a_i|b_i)$ is a party entry in Π if and only if $(b_i|a_i)$ is a party entry in Π . Since $r(b_i, a_{i-1}) < r(b_i, a_i)$, so $(b_i|a_{i-1})$ is superior in Π , and hence $(a_{i-1}|b_i)$ is inferior in Π . Before the inferior entry $(a_{i-1}|b_i)$ in the current list of a_{i-1} , there must be a party entry in Π . The only entry before $(a_{i-1}|b_i)$ is $(a_{i-1}|b_{i-1})$, so $(a_{i-1}|b_{i-1})$ is

a party entry in Π . Repeating this argument shows that $(a_j|b_j)$ is party entry in Π , for all $j = 1, 2, \dots, r$. This completes the proof. \square

In the context of the above theorem, in any stable partition either $(a_i|b_i)$ and $(b_i|a_i)$ are party entries for all values of i , $1 \leq i \leq r$, or for no value of i . In the following proposition, notice that b_i is the first person on a_i 's list in table T , and b_{i+1} is the second.

LEMMA 6.4. *Let T be a table in Phase 2, and let*

$$R = (a_1, a_2, \dots, a_r)|(b_1, b_2, \dots, b_r)$$

be a rotation exposed in T . Let Π be a stable partition contained in T such that $(a_i|b_i)$ is a party entry for all $i = 1, 2, \dots, r$. Then

(i) *either all the entries $(a_i|b_{i+1})$, $i = 1, 2, \dots, r$, are party entries in Π (subscript modulo r),*

(ii) *or none are; in this case, $A \cap B = \emptyset$, where $A = \{a_1, a_2, \dots, a_r\}$, $B = \{b_1, b_2, \dots, b_r\}$, and $\{a_i, b_i\}$ is a two-person party in Π , for $i = 1, 2, \dots, r$.*

Proof. Suppose that for some fixed i , $(a_i|b_{i+1})$ is not a party entry in Π . We claim that $(a_{i-1}|b_i)$ is not a party entry in Π either. In the following, party entry, superior entry, and inferior entry are all with respect to the given stable partition Π .

Since $r(b_{i+1}, a_i) < r(b_{i+1}, a_{i+1})$, and $(a_{i+1}|b_{i+1})$ is a party entry if and only if $(b_{i+1}|a_{i+1})$ is, hence $(b_{i+1}|a_i)$ is superior, and so $(a_i|b_{i+1})$ is inferior. Observe that person b_{i+1} is the second entry on a_i 's list in T , and $(a_i|b_{i+1})$ is inferior but is not a party entry. Therefore in a_i 's list, only one entry can be a party entry; namely, the first one $(a_i|b_i)$. So $\{a_i, b_i\}$ is a two-person party in Π . Thus $(a_{i-1}|b_i)$ cannot be a party entry in Π , as claimed. Replacing i by $i - 1$, and repeating the same argument, it follows that $(a_j|b_{j+1})$ is not a party entry, and that $\{a_j, b_j\}$ is a two-person in Π , for $j = 1, 2, \dots, r$ (subscript modulo r). It remains to show that, in this case, $A \cap B = \emptyset$. Note that, for any $a_i \in A$ and $b_j \in B$, a_i is matched with the first person on his list, and b_j is matched with the last person on his list. Person a_i and b_j both have at least two persons on their lists, so $a_i \neq b_j$ and hence $A \cap B = \emptyset$. \square

Now we consider the case that the elimination of a rotation makes no list empty. The following theorem is one of our main results.

THEOREM 6.5. *Let T be a table in Phase 2, and let*

$$R = (a_1, a_2, \dots, a_r)|(b_1, b_2, \dots, b_r)$$

be a rotation exposed in T . Suppose that no list, which is non-empty in T , will become empty after eliminating R . If there is a stable partition Π

contained in T , in which $(a_i|b_i)$, $i = 1, 2, \dots, r$, are all party entries, then there is another Π' in which they are not. Moreover, Π and Π' have exactly the same odd parties (not only having the same persons involved in a corresponding odd party, but also with the same party permutation).

Proof. All the subscripts considered in the following are modulo r . By Lemma 6.4, either all the entries $(a_i|b_{i+1})$, $i = 1, 2, \dots, r$, are party entries in Π or none are. Suppose that the latter case holds. Then, again by Lemma 6.4, $\{a_i, b_i\}$ is a two-person party in Π , $i = 1, 2, \dots, r$, and $A \cap B = \emptyset$. Break the pair a_i and b_i , and match a_i with the second person on his list, let $\Pi' = (\Pi - \{(a_i, b_i) \mid i = 1 \text{ to } r\}) \cup \{(a_i, b_{i+1}) \mid i = 1 \text{ to } r\}$. We claim that Π' is a stable partition. (Obviously, Π and Π' have the same odd parties). We need some observations. In stable partition Π , a_i (b_j , respectively) is matched with the first person b_i (the last person a_j , respectively) on his list in table T . While in Π' , a_i is matched with the second person b_{i+1} , and b_j is matched with a_{j-1} whom he prefers to a_j . Therefore, in table T , any instability in Π' must involve some a_i . By Proposition 5.6, if we can prove that Π' is a stable partition for instance T , then Π' is a stable partition for the original instance. Now, in table T , every person a_i has only one superior entry in Π' ; namely, $(a_i|b_i)$. Since b_i is matched with a_{i-1} in Π' and $r(b_i, a_{i-1}) < r(b_i, a_i)$, hence $(b_i|a_i)$ is inferior in Π' , and the claim follows. Note that, in the stable partition Π' , $(a_i|b_i)$ is not a party entry, for $i = 1, 2, \dots, r$.

Now consider the former case that all the entries $(a_i|b_{i+1})$, $i = 1, 2, \dots, r$, are party entries in Π . There are two cases: (i) $A \cap B \neq \emptyset$ or (ii) $A \cap B = \emptyset$, where $A = \{a_1, a_2, \dots, a_r\}$ and $B = \{b_1, b_2, \dots, b_r\}$.

(i) Suppose $A \cap B \neq \emptyset$, say $a_i = b_j$ for some fixed i and j . For person a_i (b_j , respectively), $(a_i|b_i)$ and $(a_i|b_{i+1})$ ($(b_j|a_{j-1})$ and $(b_j|a_j)$, respectively) are party entries in Π . Since $a_i = b_j$, it must be that $(a_i|b_i) \equiv (b_j|a_{j-1})$ and $(a_i|b_{i+1}) \equiv (b_j|a_j)$, i.e., $b_i = a_{j-1}$ and $b_{i+1} = a_j$. So the second person b_{i+1} on a_i 's list in T and the last person a_j on b_j 's are the same. By Lemma 6.1, the list of every person a_i in A will become empty after eliminating rotation R . This is a contradiction, and so $A \cap B = \emptyset$.

(ii) Consider $A \cap B = \emptyset$. We claim that $A \cup B$, with the permutation $\langle a_r, b_r, a_{r-1}, b_{r-1}, \dots, a_2, b_2, a_1, b_1 \rangle$, is an even party in Π . Observe that

1. the persons in A and B are all distinct, $|A| = |B|$, so $|A \cup B|$ is an even number,
2. $(a_i|b_i)$ and $(b_i|a_{i-1})$ are party entries in Π ,
3. $(b_i|a_i)$ and $(a_i|b_{i+1})$ are also party entries in Π ,
4. $r(a_i, b_i) < r(a_i, b_{i+1})$ and $r(b_i, a_{i-1}) < r(b_i, a_i)$, $i = 1, 2, \dots, r$ (subscripts modulo r).

So the claim follows. Now, decomposing the even party $A \cup B$ into pairs, let $\Pi' = (\Pi - \{\langle a_r, b_r, a_{r-1}, b_{r-1}, \dots, a_2, b_2, a_1, b_1 \rangle\}) \cup \{\langle a_i, b_{i+1} \rangle \mid i = 1 \text{ to } r\}$. By the proof of Proposition 3.2, Π' is a stable partition, in which $(a_i|b_i)$ is not a party entry for $i = 1, 2, \dots, r$. Since no odd party is affected, Π' and Π have exactly the same odd parties. This completes the proof of the theorem. \square

Before describing our modified version of Phase 2, we need the following definitions. Assume that the initial table of preference lists is symmetric.

DEFINITION. Let T be the current table at a certain point of the algorithm. The preference list of a person is said to be *inactive*, if

- (i) the list is empty,
- (ii) the list has only one person, or
- (iii) the list is involved in an odd rotation which has been discovered before.

The preference list of a person is *active*, if it is not inactive.

DEFINITION. The *active part* of a table T is the part of the table obtained from T by deleting all the inactive lists.

If there is someone whose preference list is still active, then his list contains at least two persons. Hence by Proposition 4.2, there is a rotation exposed in T . Phase 2 of the algorithm for finding a stable partition can be summarized as follows:

1. While the active part of the table is not empty, find an exposed rotation.

(i) If the elimination of this rotation results in some list empty, then do not eliminate it (i.e., backup to the table where this rotation was eliminated), and declare inactive the lists of all the persons involved in this rotation. (They form an odd party.)

(ii) Otherwise, eliminate this rotation. If some list has only one person left on it due to the elimination, then declare this list inactive.

2. If the active part of the table is empty, stop, the current table specifies a stable partition.

Let us discuss the above algorithm. By Theorem 6.2, going through step (i) of 1 results in the finding of an odd rotation. We will see soon that this indicates an odd party. Consider step (ii) of 1; by Theorem 6.5, if the original instance admits a stable partition Π then there is another Π' contained in the resulting table after eliminating the rotation. Further-

more Π and Π' have exactly the same odd parties. Phase 2 of the algorithm terminates with every list being inactive. In the final table T , there are a collection of persons whose lists have one person left, a collection $\{c_j | j = 1 \text{ to } s\}$ of persons whose lists are empty, and a collection $\{R_i | i = 1 \text{ to } m\}$ of disjoint odd rotations. Any one of these three collections may be empty. Recall that, throughout the algorithm, a is on b 's list if and only if b is on a 's. So those persons whose lists have only one entry appear in pairs, let $\{\{\alpha_i, \beta_i\} | i = 1 \text{ to } k\}$ be the set of all such pairs. For each odd rotation $R = (a_1, a_2, \dots, a_r)|(b_1, b_2, \dots, b_r)$, by Theorem 6.2, there is a cyclic permutation $\Pi(A)$ such that $\Pi(A)$ is a semi-party permutation for $A = \{a_1, a_2, \dots, a_r\}$. Let $\Pi(A_i)$ be that semi-party permutation for the corresponding rotation R_i , $i = 1 \text{ to } m$. Then obviously, $\Pi = \{\langle \alpha_i, \beta_i \rangle | i = 1 \text{ to } k\} \cup \{\langle c_i \rangle | i = 1 \text{ to } s\} \cup \{\Pi(A_i) | i = 1 \text{ to } m\}$ is a stable partition for the final instance T , because there are no other entries in T except the party entries. By Proposition 5.6, Π is also a stable partition for the initial instance T_0 . The correctness of the whole algorithm is thus established. As for the time complexity, since our algorithm is basically Irving's algorithm with some modification and extension, the time bound is exactly the same as that of Irving's algorithm, namely, $O(n^2)$. The analysis is also similar to that of Irving's [6] which will not be repeated here.

Remark 1. Notice that every list which is non-empty at the end of Phase 1, remains non-empty through the whole process of our modified version of Phase 2. Therefore, all the single-person odd parties are found at the end of Phase 1.

Remark 2. By Theorem 6.5, every stable partition contains the same odd parties as Π does, where Π is the stable partition contained in the final table.

To illustrate our algorithm, we give the following example. This example looks complicated, so it is helpful to know the way that we created it. First, we have a stable partition Π in mind, $\Pi = \{\langle 1, 2, 3, 4, 5 \rangle, \langle 6 \rangle, \langle 7, 8, 9, 10 \rangle, \langle 11, 12 \rangle, \langle 13 \rangle, \langle 14 \rangle\}$. After putting down the party entries, whenever we write an entry $(a|b)$ in a superior position, immediately write $(b|a)$ in an inferior position. To make the example more interesting, a rotation $(5, 10, 12)|(11, 1, 7)$ is there. The example is shown in Fig. 3a. The Phase 1 table is displayed in Fig. 3b. After exposed rotation $(5, 10, 12)|(11, 1, 7)$ is eliminated from the Phase 1 table, the resulting table is shown in Fig. 3c. If we try to eliminate exposed rotation $(1, 4, 2, 5, 3)|(2, 5, 3, 1, 4)$, then the lists of all the persons 1, 2, 3, 4, and 5 become empty. So this rotation should not be eliminated. By Theorem 6.2, $\{1, 2, 3, 4, 5\}$ is an odd party with party permutation $\langle 2, 3, 4, 5, 1 \rangle = \langle 1, 2, 3, 4, 5 \rangle$. After eliminating one more rotation $(8, 10)|(9, 7)$, every list is

(a) Original table									(b) Phase I table				
Person	Preference list								Person	Preference list			
1	3	2	5	4	10	6	13	14	1	2	5	4	10
2	5	3	1	6	13	14			2	3	1		
3	4	2	1	6	13	14			3	4	2		
4	5	1	3	6	13	14			4	5	1	3	
5	11	1	4	2	6	13	14		5	11	1	4	
6	1	2	3	4	5	7	11		6	Empty			
7	9	8	10	12	6	13	14		7	8	10	12	
8	10	9	11	7					8	9	11	7	
9	10	8	7						9	10	8		
10	1	7	9	8					10	1	7	9	
11	12	8	5	6	13	14			11	12	8	5	
12	7	11							12	7	11		
13	1	2	3	4	5	7	11		13	Empty			
14	1	2	3	4	5	7	11		14	Empty			

(c)				(d) Final table		
Person	Preference list			Person	Preference list	
1	2	5		1	2	5
2	3	1		2	3	1
3	4	2		3	4	2
4	5	3		4	5	3
5	1	4		5	1	4
6	Empty			6	Empty	
7	8	10		7	8	
8	9	7		8	7	
9	10	8		9	10	
10	7	9		10	9	
11	12			11	12	
12	11			12	11	
13	Empty			13	Empty	
14	Empty			14	Empty	

FIGURE 3

inactive, and the final table is shown in Fig. 3d. We obtain a stable partition $\Pi_1 = \{\langle 1, 2, 3, 4, 5 \rangle, \langle 6 \rangle, \langle 7, 8 \rangle, \langle 9, 10 \rangle, \langle 11, 12 \rangle, \langle 13 \rangle, \langle 14 \rangle\}$.

The following two results should now be clear.

THEOREM 6.6. *For any given preference relation, there exists at least one stable partition. And any two stable partitions have exactly the same odd parties.*

The above theorem follows from the algorithm and Theorem 6.5. And the following theorem follows from the algorithm, Proposition 3.2, and Theorem 3.3.

THEOREM 6.7. *For any instance of the stable roommates problem, there exists a complete stable matching if and only if there does not exist any odd party.*

7. CONCLUSIONS

We conclude this paper by relating our results to two others in [1, 2] concerning the stable marriage problem. At the end, we would like to mention a further research result of ours.

The stable roommates problem is a generalization of the well-known stable marriage problem, which has been discussed in many articles [1–3, 8, 9, 11, 12]. In the stable marriage problem, there are n men and n women; each person ranks **all** the members of the opposite sex in order of preference. A stable marriage is a complete matching of men and women such that no man and woman who are not partners both prefer each other to their actual partners under the matching. Gale and Shapley [1] proved that at least one stable marriage exists for every instance of the stable marriage problem. This result can also be obtained directly from our Theorem 6.7, because an instance of the stable marriage problem can be viewed as an instance of the stable roommates problem. In this particular instance, it is obvious that there does not exist any odd party of cardinality 3 or more, since the preference list of each person contains **only** the member of the opposite sex. There does not exist any single-person odd party either, because the preference list of each person contains **all** the members of the opposite sex. Therefore, by our Theorem 6.7, there exists at least one complete stable matching (i.e., a stable marriage).

In [2], Gale and Sotomayor consider another classical form of the stable marriage problem, namely, the college admission problem. More precisely, they consider the problem of medical school graduates applying to hospitals for serving a year's residency. One of their results is as follows.

Let M be any stable matching, let $S(M)$ be the set of applicants admitted to some hospital and let $n_H(M)$ be the number of applicants admitted to hospital H . Then the set $S(M)$ and numbers $n_H(M)$ are the same for all stable M .

Let us restate this result in our terms. Consider the following form of the stable marriage problem, where the number of males and the number of females may not be equal and where the preference list of each person may include only a proper subset of the members of the opposite sex. Then Gale and Sotomayor's result above can be restated as follows.

Given an instance of the stable marriage problem described above, then the (single-person) odd parties are the same for all stable partitions. Our result, Theorem 6.6, is therefore a generalization of this one.

Given a preference relation (S, T) , since there may not exist any complete stable matching, it is natural to consider the problem of finding a maximum stable matching; i.e., a maximum number of disjoint pairs of persons such that these pairs are stable among themselves. Recently, we obtained an algorithm for solving this problem; the result will appear in a subsequent paper [10].

One final remark. After finishing writing this paper, we were informed of a research report of Irving's [7]. We notice that our results, Lemma 6.1 and Theorem 6.2, are already in [7], though ours extend the idea to the stable partition. In that report, Irving also had the following result: an instance of the stable roommates problem admits a stable matching if and only if the shortlists are non-empty and there is no improper rotation. Translate this result into our terms: an instance of the stable roommates problem admits a complete stable matching if and only if there does not exist any single-person odd party and there is no odd rotation.

ACKNOWLEDGMENT

The author would like to thank the referee for his detailed comments and valuable suggestions.

REFERENCES

1. D. GALE AND L. SHAPLEY, College admissions and the stability of marriage, *Amer. Math. Monthly* **69** (1962), 9–15.
2. D. GALE AND M. SOTOMAYOR, Some remarks on the stable matching problem, *Discrete Appl. Math.* **11** (1985), 223–232.
3. D. GUSFIELD, Three fast algorithms for four problems in stable marriage, *SIAM J. Comput.* **16** (1987), 111–128.
4. D. GUSFIELD, The structure of the stable roommates problem: Efficient representation and enumeration of all stable assignments, *SIAM J. Comput.* **17** (1988), 742–769.
5. D. GUSFIELD AND R. IRVING, "The Stable Marriage Problem: Structure and Algorithms," MIT Press, Boston, MA, 1989.
6. R. IRVING, An efficient algorithm for the stable roommates problems, *J. Algorithms* **6** (1985), 577–595.
7. R. IRVING, "On the Stable Room-Mates Problem," Research Report CSC/86/R5, Department of Computing Science, University of Glasgow, July 1986.
8. R. IRVING AND P. LEATHER, The complexity of counting stable marriages, *SIAM J. Comput.* **15** (1986), 655–667.
9. R. IRVING, P. LEATHER, AND D. GUSFIELD, An efficient algorithm for the "optimal" stable marriage, *J. Assoc. Comput. Mach.* **34** (1987), 532–543.
10. J. J. M. TAN, On a maximum stable matching problem, in preparation.
11. D. E. KNUTH, "Mariages Stables," Presses Univ. Montreal, Montreal, 1976. [French]
12. D. McVITIE AND L. B. WILSON, The stable marriage problem, *Comm. ACM* **14** (1971) 486–490.