CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Finding robust solutions to stable marriage
Authors	Genc, Begum;Siala, Mohamed;Simonin, Gilles;O'Sullivan, Barry
Publication date	2017-08
Original Citation	Genc, B., Siala, M., Simonin, G. and O'Sullivan, Barry (2017) 'Finding Robust Solutions to Stable Marriage ', in: Sierra, C. (ed.) Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17) Melbourne, Australia, 19-25 August. doi: 10.24963/ijcai.2017/88
Type of publication	Conference item
Link to publisher's version	10.24963/ijcai.2017/88
Rights	© 2017 International Joint Conferences on Artificial Intelligence All rights reserved.
Download date	2026-11-14 14:52:32
Item downloaded from	https://hdl.handle.net/10468/5279

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Finding Robust Solutions to Stable Marriage

Begum Genc[1], Mohamed Siala[1], Gilles Simonin[2], Barry O'Sullivan[1]
[1] Insight, Centre for Data Analytics, Department of Computer Science, University College Cork, Ireland
[2] TASC, Institut Mines Telecom Atlantique, LS2N UMR 6004, Nantes, France
{begum.genc, mohamed.siala, barry.osullivan}@insight-centre.org, gilles.simonin@imt-atlantique.fr

Abstract

We study the notion of robustness in stable matching problems. We first define robustness by introducing (a, b)-supermatches. An (a, b)-supermatch is a stable matching in which if a pairs break up it is possible to find another stable matching by changing the partners of those a pairs and at most b other pairs. In this context, we define the most robust stable matching as a $(1, b)$-supermatch where b is minimum. We show that checking whether a given stable matching is a $(1, b)$-supermatch can be done in polynomial time. Next, we use this procedure to design a constraint programming model, a local search approach, and a genetic algorithm to find the most robust stable matching. Our empirical evaluation on large instances show that local search outperforms the other approaches.

1 Introduction

Heraclitus, the Greek philosopher is quoted as saying that "Change is the only constant". Therefore, it is essential to build robust systems that can be repaired by only minor changes in case of an unforeseen event [Sussman, 2007]. Although it is usually difficult to provide robustness to a complex problem, as it may be computationally expensive, a robust solution reduces the cost of future repairs.

This paper focuses on matching problems under preferences, where the aim is to find an assignment between two disjoint sets of agents while respecting an optimality criterion. Each agent has an ordinal preference ranking over agents of the other set. These types of problems have been widely studied by different research communities such as computer scientists and economists over the years; in fact, the 2012 Nobel Prize for Economics was awarded to Shapley and Roth for their work on stable allocations. Some of the variants can be listed as assigning residents to hospitals (HR), matching men and women to find stable marriages (SM) [Gale and Shapley, 1962; Gusfield and Irving, 1989], and finding donors for kidney patients [Roth *et al.*, 2005].

Stable Marriage (SM) [Gale and Shapley, 1962] is the first and the most studied variant of these problems. In SM, the sets of agents correspond to men and women. The goal is to find a matching M between men and women where each person is matched to at most one partner from the opposite sex such that there is no man and woman that prefers each other to their situations in M. Such a matching is called *stable*. We primarily work on the stable marriage problem, but the problem is also meaningful in the context of other matching variants.

We introduce the notion of (a, b)-*supermatches* as a measure of robustness for SM. A stable matching M is called an (a, b)-supermatch if any a agents decide to break their matches in M, thereby breaking a pairs, it is possible to "repair" M (i.e., find another stable matching) by changing the assignments of those a agents and at most b others. This concept is inspired by the notion of (a, b)-supermodels in Boolean satisfiability [Ginsberg *et al.*, 1998] and super solutions in constraint programming [Hebrard *et al.*, 2004; Hebrard, 2007]. When we mention a (or b) as the number of agents, we always refer to the agents from the same set.

In order to give additional insight into the problem we motivate robustness on the Hospital/Residents (HR) problem. The HR problem is a one-to-many generalization of SM. In HR, each hospital has a capacity and a preference list in which they rank the residents. Similarly, each resident has a preference list over the hospitals. A $(1, b)$-supermatch means that if a resident wants to leave his assigned hospital or a hospital does not want to have one of its current residents, it is possible to move that resident to another hospital by also moving at most b other residents to other hospitals. By minimising b, we can ensure that the required number of additional relocations to provide a repair is minimal and therefore the solution is robust. In practice, the most robust matching minimises the cost for recovering from unwanted and unforeseen events.

The first contribution of this paper is a polynomial time procedure to verify whether a given stable matching is a $(1, b)$-supermatch. Next, based on this procedure, we design a constraint programming (CP) model, as well as local search (LS) and genetic algorithm (GA) to find the most robust stable matching based on the polynomial time procedure. Last, we give empirical evidence that the local search algorithm is by far the most efficient approach to tackle this problem.

The structure of the paper is as follows. In Section 2, the notations and the basics of the stable marriage problem are introduced. In Section 3 our polynomial-time method is proposed. Next, we give our meta-heuristic algorithms in Sec-

tion 4. Last, we present our experimental study on random instances in Section 5. The reader is referred to our technical report in [Genc *et al.*, 2017b] for full details including the CP model.

2 Background & Notations

The Stable Marriage problem takes as input a set of men $U = \{m_1, m_2, \ldots, m_{n_1}\}$ and a set of women $W = \{w_1, w_2, \ldots, w_{n_2}\}$ where each person has an ordinal preference list over people of the opposite sex. For the sake of simplicity, we suppose in the rest of the paper that $n_1 = n_2$ (denoted by n), and that each person expresses a complete preference ranking over the set of the opposite sex. We also use m to denote a man in U, and w to denote a woman in W. For a complete background, we refer the reader to [Gusfield and Irving, 1989; Manlove, 2013].

A *matching* M is a one-to-one correspondence between men and women. For each man m, $M(m) = w$ is called the *partner* of m in M. In the latter case, we denote by $M(w) = m$. We shall sometimes abuse notation by considering M as a set of pairs. In that case, a *pair* $\langle m, w\rangle \in M$ iff $M(m) = w$. A pair $\langle m, w\rangle$ is said to be *blocking* a matching M if m prefers w to $M(m)$ and w prefers m to $M(w)$. A *matching* M is called stable if there exists no blocking pair for M. A *pair* $\langle m, w\rangle$ is said to be *stable* if it appears in a stable matching. Also, a pair $\langle m, w\rangle$ is *fixed* if $\langle m, w\rangle$ appears in every stable matching.

The structure that represents all stable matchings forms a *lattice* $\mathscr{M}$. In this lattice, the man-optimal matching is denoted by M_0 and the woman-optimal (man-pessimal) matching is denoted by M_z. A stable matching M_i *dominates* a stable matching M_j, denoted by $M_i \preceq M_j$, if every man prefers his partner in M_i to M_j or is indifferent between them. The size of a lattice can be exponential as the number of all stable matchings can be exponential [Irving and Leather, 1986]. Therefore, making use of this structure is not practical.

Let M be a stable matching. A *rotation* $\rho = (\langle m_{k_0}, w_{k_0}\rangle, \langle m_{k_1}, w_{k_1}\rangle, \ldots, \langle m_{k_{l-1}}, w_{k_{l-1}}\rangle)$ (where $l \in \mathbb{N}^*$) is an ordered list of pairs in M such that changing the partner of each man m_{k_i} from w_{k_i} to $w_{k_{i+1}}$ (the operation +1 is modulo l) leads to a stable matching M/ρ. The latter is said to be obtained after *eliminating* ρ from M. In this case, we say that $\langle m_{k_i}, w_{k_i}\rangle$ is *eliminated* by ρ, $\langle m_{k_i}, w_{k_{i+1}}\rangle$ is *produced* by ρ, and that ρ is *exposed* in M. For each $\langle m, w\rangle$ such that $\langle m, w\rangle \notin M_0$, there exists a unique rotation $\rho_{p_{m,w}}$ that produces $\langle m, w\rangle$. Similarly, if $\langle m, w\rangle \notin M_z$ there is a unique rotation $\rho_{e_{m,w}}$ that eliminates $\langle m, w\rangle$. Note that it is always the case that M (strictly) dominates M/ρ.

There exists a partial order for rotations. A rotation ρ' is said to precede another rotation ρ (denoted by $\rho' \prec\prec \rho$), if ρ' is eliminated in every sequence of eliminations that starts at M_0 and ends at a stable matching in which ρ is exposed [Gusfield and Irving, 1989]. Note that this relation is transitive, that is, $\rho'' \prec\prec \rho' \wedge \rho' \prec\prec \rho \implies \rho'' \prec\prec \rho$. Two rotations ρ and ρ' are said to be *incomparable* if ρ does not precede ρ' and vice versa. The structure that represents all rotations and their partial order is a directed graph called *rotation poset* denoted by $\Pi = (\mathcal{V}, E)$. Each rotation corresponds to a vertex in $\mathcal{V}$ and there exists an edge from ρ to ρ' if ρ precedes ρ'. The number of rotations is bounded by $n(n-1)/2$ and the number of arcs is $O(n^2)$ [Gusfield and Irving, 1989]. It should be noted that the construction of Π can be done in $O(n^2)$.

Predecessors of a rotation ρ in a rotation poset are denoted by $N^-(\rho)$ and successors are denoted by $N^+(\rho)$. Later, we shall need transitivity to complete these lists. Therefore, we denote by $N_t^-(\rho)$ (respectively $N_t^+(\rho)$) the predecessors (respectively successors) of a rotation ρ including transitivity.

A *closed subset* S is a set of rotations such that for any rotation ρ in S, if there exists a rotation ρ' that precedes ρ then ρ' is also in S.

Below is a theorem and a corollary from [Gusfield and Irving, 1989] mainly used in some proofs in the next section. We slightly change few notations.

Theorem 1 (Theorem 2.5.7)**.** *i) There is a one-one correspondence between the closed subsets of* Π *and the stable matchings of* $\mathscr{M}$*. ii) S is the closed subset of rotations of* Π *corresponding to a stable matching* M *if and only if* S *is the (unique) set of rotations on every* M_0*-chain in* $\mathscr{M}$ *ending at* M*. Further,* M *can be generated from* M_0 *by eliminating the rotations in their order along any of these paths, and these are the only ways to generate* M *by rotation eliminations starting from* M_0*. iii) If* S *and* S' *are the unique sets of rotations corresponding to distinct stable matchings* M *and* M'*, then* M *dominates* M' *if and only if* $S \subset S'$*.*

Corollary 1 (Corollary 3.2.1)**.** *Every man-woman pair* $\langle m, w\rangle$ *is in at most one rotation. Hence there are at most* $n(n-1)/2$ *rotations in an instance of the stable marriage problem of size n.*

By Theorem 1, every closed subset in the rotation poset corresponds to a stable matching. We denote by $X(S)$ the set of men that are included in at least one of the rotations in S.

We introduce here a notion that is important to our measure of robustness. Let M_i and M_j be two stable matchings. The *distance* $d(M_i, M_j)$ is the number of men that have different partners in M_i and M_j. A matching M_k is closer to M_i than M_j if $d(M_i, M_k) < d(M_i, M_j)$. Let $a, b \in \mathbb{N}$. A stable matching M is said to be an (a, b)-supermatch if for any set $\Psi \subset M$ of a stable pairs that are not fixed, there exists a stable matching M' such that $M' \cap \Psi = \emptyset$ and $d(M, M') - a \leq b$.

3 Checking $(1, b)$-supermatch in Polynomial Time

A preliminary version of this section appeared in [Genc *et al.*, 2017a]. In this section we first recall the basics, and then show how to find or verify the closest stable matching of a given stable matching with an unwanted couple. Throughout this section, we suppose that M is the given stable matching, S its closed subset, and $\langle m, w\rangle \in M$ is a non-fixed (stable) pair to remove from M.

The closest matching to M that does not include $\langle m, w\rangle$ is a matching M^* in which either $\langle m, w\rangle$ was eliminated or not produced in any sequence of rotation eliminations starting from M_0 leading to M^*. Hence, if $\langle m, w\rangle \notin M_0$ then $\rho_{p_{m,w}}$ exists, and there is a set of stable matchings S_u that dominate

M and do not include $\langle m, w\rangle$. Similarly, if $\langle m, w\rangle \notin M_z$ then $\rho_{e_{m,w}}$ exists, and there is a set of stable matchings S_d, dominated by M and not including $\langle m, w\rangle$.

If $\langle m, w\rangle \notin M_0$, we define a specific set of rotations S^{*m}_{UP} as follows:

$$S^{*m}_{\text{UP}} = S \setminus (\{\rho_{p_{m,w}}\} \cup \{N^+_t(\rho_{p_{m,w}}) \cap S\}). \quad (1)$$

If $\langle m, w\rangle \notin M_z$, we define a specific set of rotations S^{*m}_{DOWN} as follows:

$$S^{*m}_{\text{DOWN}} = S \cup \{\rho_{e_{m,w}}\} \cup \{N^-_t(\rho_{e_{m,w}}) \setminus S\}. \quad (2)$$

Observe first that S^{*m}_{UP} and S^{*m}_{DOWN} are in fact closed subsets since S is a closed subset. Let M^{*m}_{UP} (respectively M^{*m}_{DOWN}) be the stable matching corresponding to S^{*m}_{UP} (respectively S^{*m}_{DOWN}). By construction, we have $M^{*m}_{\text{UP}} \in S_u$ and $M^{*m}_{\text{DOWN}} \in S_d$. We show later that any stable matching $M_i \notin \{M^{*m}_{\text{UP}}, M^{*m}_{\text{DOWN}}\}$ that does not include the pair $\langle m, w\rangle$ cannot be closer to M than M^{*m}_{UP} or M^{*m}_{DOWN}.

Let us illustrate these terms on a sample stable marriage instance. Consider the stable marriage instance specified by the preference lists of 7 men/women in Table 1. For the sake of clarity, we denote each man m_i with i and each woman w_j with j.

Figure 1 represents the rotation poset and all the rotations associated with this sample. In Figure 2, we give the lattice of all stable matchings. There exists two vectors for each stable matching. The first vector represents the set of men and the second vector represents the partner of each man in the matching. Each edge from M to M' on the lattice is labelled with the rotation ρ such that M' is obtained after exposing ρ in M.

As an example, let current stable matching be M_5, and the closed subset associated with it be $S_5 = \{\rho_0, \rho_1, \rho_2\}$. Table 2 illustrates for each man m the matchings S^{*m}_{UP} and S^{*m}_{DOWN} if they exist.

m_0	0 6 5 2 4 1 3
m_1	6 1 4 5 0 2 3
m_2	6 0 3 1 5 4 2
m_3	3 2 0 1 4 6 5
m_4	1 2 0 3 4 5 6
m_5	6 1 0 3 5 4 2
m_6	2 5 0 6 4 3 1

w_0	2 1 6 4 5 3 0
w_1	0 4 3 5 2 6 1
w_2	2 5 0 4 3 1 6
w_3	6 1 2 3 4 0 5
w_4	4 6 0 5 3 1 2
w_5	3 1 2 6 5 4 0
w_6	4 6 2 1 3 0 5

Table 1: Preference lists for men (left) and women (right) for a sample stable marriage instance of size 7.

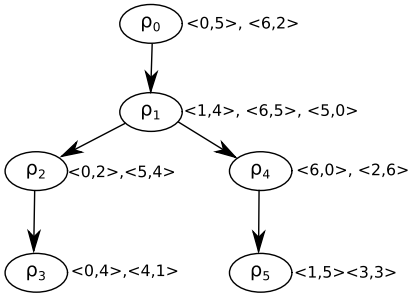


Figure 1: Rotation poset of the instance given in Table 1.

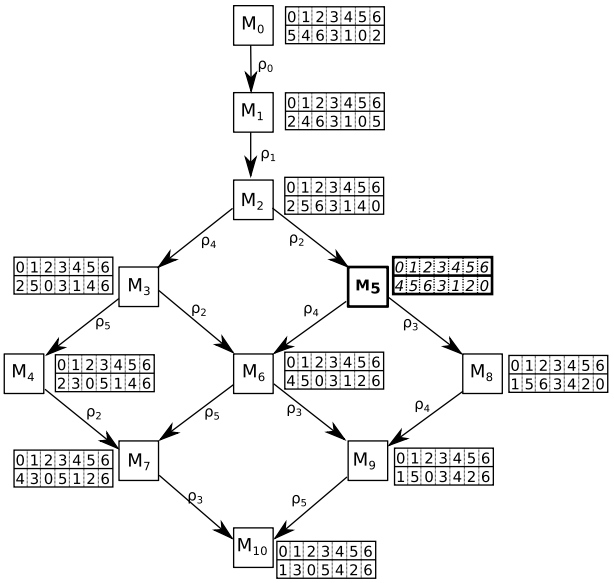


Figure 2: The lattice of all stable matchings corresponding to the instance given in Table 1.

$\langle m, w\rangle$	$\rho_{p_{m,w}}$	$\rho_{e_{m,w}}$	S^{*m}_{UP}	S^{*m}_{DOWN}
$\langle 0, 4\rangle$	ρ_2	ρ_3	$\{\rho_0, \rho_1\}$	$\{\rho_0, \rho_1, \rho_2, \rho_3\}$
$\langle 1, 5\rangle$	ρ_1	ρ_5	$\{\rho_0\}$	$\{\rho_0, \rho_1, \rho_2, \rho_4, \rho_5\}$
$\langle 2, 6\rangle$	-	ρ_4	-	$\{\rho_0, \rho_1, \rho_2, \rho_4\}$
$\langle 3, 3\rangle$	-	ρ_5	-	$\{\rho_0, \rho_1, \rho_2, \rho_4, \rho_5\}$
$\langle 4, 1\rangle$	-	ρ_3	-	$\{\rho_0, \rho_1, \rho_2, \rho_3\}$
$\langle 5, 2\rangle$	ρ_2	-	$\{\rho_0, \rho_1\}$	-
$\langle 6, 0\rangle$	ρ_1	ρ_4	$\{\rho_0\}$	$\{\rho_0, \rho_1, \rho_2, \rho_4\}$

Table 2: The repair closed subsets S^{*m}_{UP} and S^{*m}_{DOWN} for M_5.

We give few lemmas in order to show that the closest stable matching to M when breaking the pair $\langle m, w\rangle$ is either S^{*m}_{UP} or S^{*m}_{DOWN}.

Lemma 1. *Given two incomparable rotations ρ and ρ', $X(\{\rho\}) \cap X(\{\rho'\}) = \emptyset$.*

Proof. By definition of incomparability, if two rotations are incomparable, it means that they modify a set of men who do not require modifications from the other first. Therefore the sets of men are distinct. □

Lemma 2. *Given three stable matchings M_i, M_j and M_k where $M_i \preceq M_j \preceq M_k$, then $d(M_j, M_k) \leq d(M_i, M_k)$ and $d(M_i, M_j) \leq d(M_i, M_k)$.*

Proof. Using the properties of domination and the closed subsets in Theorem 1, we can infer $S_i \subset S_j \subset S_k$.

Assume to the contrary that $d(M_j, M_k) > d(M_i, M_k)$. This situation occurs only if a set of pairs that are present in M_i are eliminated to obtain M_j and then re-matched with the same partners they had in M_i to get M_k. However, this contradicts Corollary 1. For similar reasons, $d(M_i, M_j) < d(M_i, M_k)$. □

The case where stable matchings have the same distance such as $d(M_i, M_j) = d(M_i, M_k)$, if the rotation set in the difference sets $S_j \setminus S_i$, $S_k \setminus S_i$, and $S_k \setminus S_j$ modify the same set of men.

Lemma 3. *If there exists a stable matching M_x that does not contain $\langle m, w\rangle$, dominates M and different from M^{*m}_{UP}, then M_x dominates M^{*m}_{UP}.*

Proof. $M^{*m}_{\text{UP}} \preceq M$ by definition. Suppose by contradiction that there exists an M_x such that $\langle m, w\rangle \notin M_x$ and $M^{*m}_{\text{UP}} \preceq M_x \preceq M$. It implies that $S^{*m}_{\text{UP}} \subset S_x \subset S$. In this case, $(S_x \setminus S^{*m}_{\text{UP}}) \subset \Big\{\{\rho_{p_{m,w}}\} \cup \{N^+_t(\rho_{p_{m,w}}) \cap S\}\Big\}$. However, this set contains $\rho_{p_{m,w}}$ and the rotations preceded by $\rho_{p_{m,w}}$. Adding any rotation from this set to S_x results in a contradiction by either adding $\langle m, w\rangle$ to the matching, thereby not breaking that couple, or because the resulting set is not a closed subset. □

Lemma 4. *If there exists a stable matching M_x that does not contain $\langle m, w\rangle$ dominated by M but different from M^{*m}_{DOWN}, then M^{*m}_{DOWN} dominates M_x.*

Proof. Similar to the proof above, suppose that there exists an M_x such that $\langle m, w\rangle \notin M_x$ and $M \preceq M_x \preceq M^{*m}_{\text{DOWN}}$. We have $S \subset S_x \subset S^{*m}_{\text{DOWN}}$. It implies $(S_x \setminus S) \subset \Big\{\{\rho_{e_{m,w}}\} \cup \{N^-_t(\rho_{e_{m,w}}) \setminus S\}\Big\}$. This set contains the rotation $\rho_{e_{m,w}}$ that eliminates the pair and the rotations preceding $\rho_{e_{m,w}}$. In order to add $\rho_{e_{m,w}}$ all other rotations must be added to form a closed subset. If all rotations are added, $S = S^{*m}_{\text{DOWN}}$ which results in a contradiction. □

Lemma 5. *For any stable matching M_i incomparable with M such that M_i does not contain the pair $\langle m, w\rangle$, M^{*m}_{UP} is closer to M than M_i.*

Proof. Let S_i be the closed subset corresponding to M_i, and S be that corresponding to M.

First, we consider the case in which $S_i \cap S = \emptyset$. If the closed subsets have no rotations in common the rotations in these sets are incomparable. Using Lemma 1, $X(S_i) \cap X(S) = \emptyset$. Therefore, $d(M_i, M) = |X(S_i)| + |X(S)|$, whereas $d(M^{*m}_{\text{UP}}, M) \leq |X(S)|$.

Second, we consider the case in which $S_i \cap S \neq \emptyset$. Let M_c be the closest dominating stable matching of both M_i and M^{*m}_{UP}, along with S_c as its corresponding closed subset. Using Lemma 2 we know that $d(M^{*m}_{\text{UP}}, M_s) \leq d(M_c, M_s)$, where $d(M_c, M_s) = |X(S \setminus S_c)|$.

Using Lemma 1 we know that $X(S_i \setminus S_c) \cap X(S \setminus S_c) = \emptyset$. Therefore, $d(M_i, M) = |X(S_i \setminus S_c)| + |X(S \setminus S_c)|$. By substituting the formula above, $d(M_i, M) \geq |X(S_i \setminus S_c)| + d(M^{*m}_{\text{UP}}, M)$. Using the fact that $|X(S_i \setminus S_c)| > 0$ from the definition of M_i, we can conclude that $d(M_i, M) > d(M^{*m}_{\text{UP}}, M)$. □

The following theorem is a direct consequence of Lemmas 3, 4, 5.

Theorem 2. *The closest stable matching of a stable matching M given the unwanted pair $\langle m, w\rangle$ is either M^{*m}_{UP} or M^{*m}_{DOWN}.*

m	M^{*m}_{UP}	M^{*m}_{DOWN}	d^m_{up}	d^m_{down}	b_m
0	M_2	M_8	2	2	1
1	M_1	M_7	4	4	3
2	-	M_6	∞	2	1
3	-	M_7	∞	4	3
4	-	M_8	∞	2	1
5	M_2	-	2	∞	1
6	M_1	M_6	4	2	1

Table 3: The repair stable matchings M^{*m}_{UP} and M^{*m}_{DOWN} for each man in M_5 and the distances between them.

We show that checking if a stable matching is a $(1, b)$-supermatch can be performed in $O(n \times |\mathcal{V}|)$ time after a $O(n^2 + |\mathcal{V}|^2)$ preprocessing step. First, the pre-processing step consists of building the poset graph ($O(n^2)$), the lists $N^-_t(\rho)$, $N^+_t(\rho)$ for each rotation ρ ($O(|\mathcal{V}|^2)$), and $\rho_{e_{m,w}}$ and $\rho_{p_{m,w}}$ for each pair $\langle m, w\rangle$ whenever applicable ($O(n^2)$). Next, we compute S^{*m}_{UP} and S^{*m}_{DOWN} for each man. Note that S^{*m}_{UP} and S^{*m}_{DOWN} can be constructed in $O(|\mathcal{V}|)$ time (by definition of S^{*m}_{UP} and S^{*m}_{DOWN}) for each man m. Note that $d(M^{*m}_{\text{UP}}, M)$ is equal to the number of men participating in the rotations that are eliminated from S to obtain S^{*m}_{UP}. Similarly, $d(M^{*m}_{\text{DOWN}}, M)$ is equal to the number of men participating in the rotations that are added S to obtain S^{*m}_{DOWN}. Last, if $b < d(M^{*m}_{\text{UP}}, M) - 1$ and $b < d(M^{*m}_{\text{DOWN}}, M) - 1$, we know that it is impossible to repair M when m needs to change his partner with at most b other changes. Otherwise, M is a $(1, b)$-supermatch.

As an illustration, Table 3 shows the stable matchings corresponding to the closed subsets given in Table 2 and the distances between the current stable matching M_5 to each one of them. The distances are denoted as d^m_{up} and d^m_{down} in the table, where for each man m, $d^m_{up} = d(M_5, M^{*m}_{\text{UP}})$ and $d^m_{down} = d(M_5, M^{*m}_{\text{DOWN}})$, respectively. If M^{*m}_{UP} does not exist for a man m, then d^m_{up} is denoted by ∞ (a similar notation is used when M^{*m}_{DOWN} does not exist). Last, b_m represents the repairing cost of each man m and calculated by $b_m = min(d^m_{up}, d^m_{down}) - 1$.

The robustness of a stable matching is measured by the worst case repair cost over all men involved, $b = \max\{b_i \mid i \in \{1...n\}\}$. Hence, M_5 is a (1,3)-supermatch.

4 Metaheuristics

In this section, we describe two models, namely Genetic Algorithm and Local Search using the polynomial time verification algorithm to find (1,b)-supermatches and subsequently the most robust stable matching.

4.1 Genetic Algorithm

Genetic algorithms are being used extensively in optimization problems as an alternative to traditional heuristics. They are compelling robust search and optimization tools, which work on the natural concept of evolution, based on natural genetics and natural selection. Holland introduced the GAs and he has also shown how to apply the process to various computationally difficult problems [Holland, 1975; 1992;

Bäck, 1996]. The model that we propose to find the most robust stable matching is detailed as follows:

Initialization. A number of random stable matchings are generated for constructing the initial population. Recall that each closed subset in the rotation poset corresponds to a stable matching. The random stable matching generation is performed by selecting a random rotation from the rotation poset and adding all of its predecessors to the rotation set to construct a closed subset. Recall that the set of all predecessors of a given node is $N_t^-(\rho)$.

Evaluation. For each stable matching M_i, we denote by b_i its b value. At this step, we compute the value b_i of each stable matching M_i in the population. Then, a fitness value $v_i = b_i$ is assigned to each M_i in the population. Then, the values v_i are normalised in the interval $[0, 1]$.

Evolution. The evolution step consists in selecting stable matchings from the population using the roulette wheel selection method [Goldberg, 1989], then applying crossover and mutation on the selected matchings.

- **Crossover.** Given two stable matchings and their corresponding closed subsets S_1 and S_2, one random rotation is selected from each subset. Let ρ_1 and ρ_2 denote the randomly selected rotations. If $\rho_1 \notin S_2$, then ρ_1 and all its predecessors that are not included in S_2 are added to S_2 to form a new closed subset. This new closed subset corresponds to one of the children stable matchings produced by crossover. Similarly, the same process is repeated for S_1 and another stable matching is obtained. If the rotations are already in the closed subsets, no action is taken.
- **Mutation.** Given the closed subset S of a stable matching, a random rotation ρ from $\mathcal{V}$ is selected. If $\rho \notin S$, then ρ and all its required predecessors to form a closed subset are added to S. However, if $\rho \in S$ then ρ and all its successors in S are removed from S. Removing the successors will also yield in a stable matching.

For each created stable matching, the method described in the end of Section 3 is called to compute the b value. Recall that this procedure takes $O(n \times |\mathcal{V}|)$ time after a $O(n^2 + |\mathcal{V}|^2)$ preprocessing step. In order to speed up this process from a practical point of view, in our experiments an extra data structure is being used to memorize the fitness value of already generated stable matchings.

The algorithm repeats evolution phase until the termination criteria met and the phase consists of calling evaluation, crossover and mutation subsequently. In both crossover and mutation, the worst case time complexity for one call is bounded by $O(|\mathcal{V}|)$ since the set of all predecessors (respectively successors) of a given node is $N_t^-(\rho)$ (respectively $N_t^+(\rho)$).

4.2 Local Search

In our local search model, we use the well-known iterated local search with restarts [Stützle, 1998; Lourenço *et al.*, 2010]. The first step of the algorithm is to find a random solution to start with. A random stable matching M is created by using the method defined in Initialization step in Section 4.1. Then, a neighbouring stable matching set is created using the rotation poset.

A *neighbour* in this context is defined as a stable matching that differs only by one rotation from the closed subset of M. Let S denote the closed subset of M and L_s the set of leaf rotations in S. Removing one rotation $\rho_i \in L_s$ from S corresponds to a dominating neighbour of M. Removing each $\rho_i \in L_s$ one at a time corresponds to a different neighbour. Similarly, let N_s denote the set of rotations that are not included in S and either those rotations have an in-degree of 0 or all of their predecessors are in S. In the same manner, adding a rotation from N_s to S at a time corresponds to a neighbour of M.

Figure 3 illustrates a sample rotation poset, where the closed subset is $S = \{\rho_0, \rho_1, \rho_3, \rho_4, \rho_5, \rho_7\}$. Then, identification of the leaf nodes in this set would be as $L_s = \{\rho_1, \rho_4, \rho_7\}$, since none of their children are included in S. Similarly, the neighbour rotations in $N_s = \{\rho_2, \rho_6\}$ since all their parents are included in the closed subset S.

These two sets correspond to rotations, where removal of any rotation in L_s from S at a time does not require removal of any other rotations in order to obtain another closed subset, i.e. a neighbour. Likewise, adding any rotation from N_s to S does not require adding any additional rotations to obtain a closed subset.

At each iteration, if a neighbour has lower b than M, in other words, it is a more robust solution, the search continues by finding the neighbours of the new solution. The best solution is kept as the best solution found so far in the whole search process. There is an iteration limit, that indicates the depth of search for neighbours of a randomly created stable matching. After the iteration limit is met, a new random stable matching is generated and the neighbour search continues with that stable matching. If there is no improvement in the best solution for a predetermined number of iterations (cut-off limit), the search terminates.

Notice that there can be at most $|\mathcal{V}|$ neighbours of a stable matching. Thus, creating neighbours and finding their respective b values is $O(n \times |\mathcal{V}|^2)$.

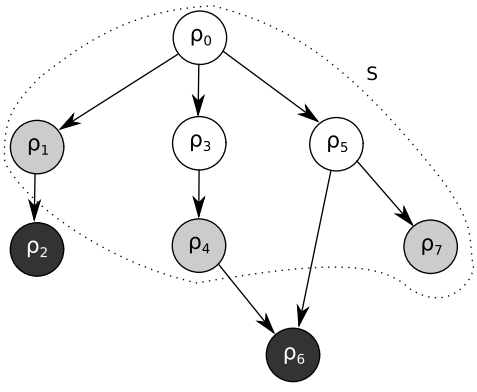


Figure 3: Illustration of the sets L_s (ρ_1, ρ_4, ρ_7) and N_s (ρ_2, ρ_6) on a sample rotation poset for a given closed subset S.

5 Experiments

We experimentally evaluate the three models for finding the most robust stable matching. The CP model is implemented in Choco 4.0.1 [Prud'homme *et al.*, 2016] and the two metaheuristics are implemented in Java. All experiments were performed on DELL M600 with 2.66 Ghz processors under Linux.

We ran each model with 4 different randomization seeds for each instance. The time limit is fixed to 20 minutes for every run. An additional cut-off is used for local search (LS) and genetic algorithm (GA) as follows: if the solution quality does not improve for 10000 iterations, we terminate the search. The genetic algorithm applies a crossover at each iteration unless the roulette wheel selection selects the fittest stable matching from the population. Additionally, the probability of applying mutation on a randomly selected stable matching is fixed as 80%. For local search, we chose to restart the local search with a randomly generated stable matching every 50 iterations. Last, the CP model (CP) uses the weighted degree heuristic [Boussemart *et al.*, 2004] with geometric restarts.

We use two sets of random instances. The first set contains 500 instances. The number of men for this set is in the set $\{300+50*k\}$ where $k \in \{1, 5\}$. The second set contains 600 large instances of size $\{1200+50*k\}$ where $k \in \{1, 6\}$. We generated 100 instances for each size for both benchmarks.

In Figures 4 and 5 we plot the normalized objective value of the best solution found by the search model $h \in \{\mathrm{CP}, \mathrm{GA}, \mathrm{LS}\}$ (x-axis) after a given time (y-axis). Let $h(I)$ be the objective value of the best solution found using model h on instance I and $lb(I)$ (resp. $ub(I)$) the lowest (resp. highest) objective value found by any model on I. The formula below gives a normalized score in the interval $[0, 1]$:

$$score(h, I) = \frac{ub(I) - h(I) + 1}{ub(I) - lb(I) + 1}$$

The value of $score(h, I)$ is equal to 1 if h has found the best solution for this instance among all models, decreases as $h(I)$ gets further from the optimal objective value, and is equal to 0 if and only if h did not find any solution for I.

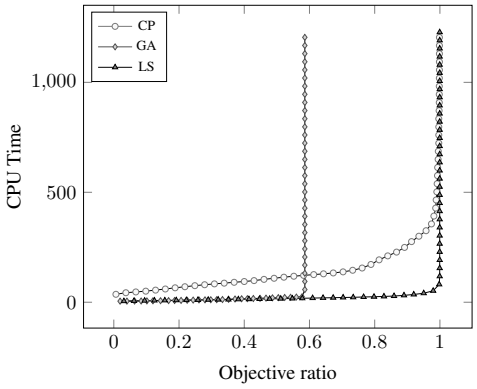


Figure 4: Search Efficiency on the First Set of Instances

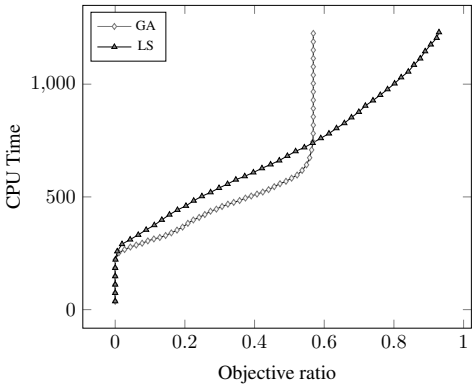


Figure 5: Search Efficiency on the Second Set of Instances

Note that the CP model runs out of memory for large instances. Therefore, we do not plot it in Figure 5.

The outcome from both figures is clear. Local search is efficient both in the quality of the solutions, and in runtime. Indeed, in the first plot, the best solutions are found by LS and CP. However, CP takes much longer time. Note that in Figure 4, CP and LS both find almost always the same objective, and in fact CP claims that the solution is optimal in all instances except one.

The GA model does not seem to be well suited for this problem. In the first data set it does not find the best solutions in all instances. Moreover, it takes much longer time than CP and LS for finding good quality solutions.

The results in Figure 5 are more spectacular for local search. Local search does not always find the best solutions since the normalised objective ratio does not exceed 90%. However, the overall performance is clearly better than GA in both quality and runtime.

6 Conclusions

We studied the notion of robustness in stable matching problems by using the notion of (a, b)-supermatch. We first showed that the problem of finding a stable matching M_i that is closest to a given stable matching M if a pair (man,woman) decides to break up in M can be found in polynomial time. Then, we used essentially this procedure to model the problem of finding the most robust stable matching using a CP formulation, local search, and genetic algorithm. Last, we empirically evaluated these models on randomly generated instances and showed that local search is by far the best model to find robust solutions.

To the best of our knowledge, this notion of robustness in stable matchings has never been proposed before. We hope that the proposed problem will get some attention in the future as it represents a challenge in real-world settings.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

[Bäck, 1996] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK, 1996.

[Boussemart *et al.*, 2004] Frédéric Boussemart, Fred Hemery, Christophe Lecoutre, and Lakhdar Sais. Boosting systematic search by weighting constraints. In *Proceedings of the 16th European Conference on Artificial Intelligence – ECAI*, pages 146–150, 2004.

[Gale and Shapley, 1962] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1):9–15, January 1962.

[Genc *et al.*, 2017a] Begum Genc, Mohamed Siala, Barry O'Sullivan, and Gilles Simonin. Robust stable marriage. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 4925–4926, 2017.

[Genc *et al.*, 2017b] Begum Genc, Mohamed Siala, Barry O'Sullivan, and Gilles Simonin. Technical report on finding robust solutions to stable marriage. *CoRR*, abs/1705.09218, 2017.

[Ginsberg *et al.*, 1998] Matthew L. Ginsberg, Andrew J. Parkes, and Amitabha Roy. Supermodels and robustness. In *In AAAI/IAAI*, pages 334–339, 1998.

[Goldberg, 1989] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[Gusfield and Irving, 1989] Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Cambridge, MA, USA, 1989.

[Hebrard *et al.*, 2004] Emmanuel Hebrard, Brahim Hnich, and Toby Walsh. Super solutions in constraint programming. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, First International Conference, CPAIOR 2004, Nice, France, April 20-22, 2004, Proceedings*, pages 157–172, 2004.

[Hebrard, 2007] Emmanuel Hebrard. *Robust solutions for constraint satisfaction and optimisation under uncertainty*. PhD thesis, University of New South Wales, 2007.

[Holland, 1975] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, USA, 1975.

[Holland, 1992] John Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1992.

[Irving and Leather, 1986] Robert W Irving and Paul Leather. The complexity of counting stable marriages. *SIAM J. Comput.*, 15(3):655–667, Aug 1986.

[Lourenço *et al.*, 2010] Helena R. Lourenço, Olivier C. Martin, and Thomas Stützle. Iterated Local Search: Framework and Applications Handbook of Metaheuristics. volume 146 of *International Series in Operations Research & Management Science*, chapter 12, pages 363–397. Springer US, Boston, MA, 2010.

[Manlove, 2013] D.F. Manlove. *Algorithmics of Matching Under Preferences*. Series on theoretical computer science. World Scientific, 2013.

[Prud'homme *et al.*, 2016] Charles Prud'homme, Jean-Guillaume Fages, and Xavier Lorca. *Choco Solver Documentation*. TASC, INRIA Rennes, LINA CNRS UMR 6241, COSLING S.A.S., 2016.

[Roth *et al.*, 2005] Alvin Roth, Tayfun Sönmez, and Utku Unver. Pairwise kidney exchange. Game theory and information, EconWPA, 2005.

[Stützle, 1998] Thomas Stützle. *Local search algorithms for combinatorial problems: analysis, improvements, and new applications*. PhD thesis, TU Darmstadt, 1998.

[Sussman, 2007] Gerald Jay Sussman. Building robust systems an essay, 2007.