A Constraint Programming Approach to the Stable Marriage Problem*

Ian P. Gent[1], Robert W. Irving[2], David F. Manlove[2], Patrick Prosser[2], and Barbara M. Smith[3]

[1] School of Computer Science, University of St. Andrews, Scotland.
ipg@dcs.st-and.ac.uk
[2] Department of Computing Science, University of Glasgow, Scotland.
rwi/davidm/pat@dcs.gla.ac.uk
[3] School of Computing and Mathematics, University of Huddersfield, England.
b.m.smith@hud.ac.uk

Abstract. The Stable Marriage problem (SM) is an extensively-studied combinatorial problem with many practical applications. In this paper we present two encodings of an instance I of SM as an instance J of a Constraint Satisfaction Problem. We prove that, in a precise sense, establishing arc consistency in J is equivalent to the action of the established Extended Gale/Shapley algorithm for SM on I. As a consequence of this, the man-optimal and woman-optimal stable matchings can be derived immediately. Furthermore we show that, in both encodings, *all* solutions of I may be enumerated in a failure-free manner. Our results indicate the applicability of Constraint Programming to the domain of stable matching problems in general, many of which are NP-hard.

1 Introduction

An instance of the classical Stable Marriage problem (SM) [6] comprises n men and n women, and each person has a preference list in which they rank all members of the opposite sex in strict order. A matching M is a bijection between the men and women. A man m_i and woman w_j form a *blocking pair* for M if m_i prefers w_j to his partner in M and w_j prefers m_i to her partner in M. A matching that admits no blocking pair is said to be *stable*, otherwise the matching is *unstable*. SM arises in important practical applications, such as the annual match of graduating medical students to their first hospital appointments in a number of countries (see e.g. [12]).

Every instance of SM admits at least one stable matching, which can be found in time linear in the size of the problem instance, i.e. $O(n^2)$, using the Gale/Shapley (GS) algorithm [4]. An extended version of the GS algorithm – the Extended Gale/Shapley (EGS) algorithm [6, Section 1.2.4] – avoids some unnecessary steps by deleting from the preference lists certain (man,woman) pairs that cannot belong to a stable matching. The man-oriented version of the EGS

* This work was supported by EPSRC research grant GR/M90641.

Men's lists	Women's lists
1: 1 3 6 2 4 5	1: 1 5 6 3 2 4
2: 4 6 1 2 5 3	2: 2 4 6 1 3 5
3: 1 4 5 3 6 2	3: 4 3 6 2 5 1
4: 6 5 3 4 2 1	4: 1 3 5 4 2 6
5: 2 3 1 4 5 6	5: 3 2 6 1 4 5
6: 3 1 2 6 5 4	6: 5 1 3 6 4 2

(a)

Men's lists	Women's lists
1: 1	1: 1
2: 2	2: 2
3: 4	3: 4 6
4: 6 5 3	4: 3
5: 5 6	5: 6 4 5
6: 3 6 5	6: 5 6 4

(b)

Fig. 1. (a) An SM instance with 6 men and 6 women; (b) the corresponding GS-lists.

algorithm involves a sequence of proposals from the men to women, provisional engagements between men and women, and deletions from the preference lists. At termination, the reduced preference lists are referred to as the *MGS-lists*. A similar proposal sequence from the women to the men (the woman-oriented version) produces the *WGS-lists*, and the intersection of the MGS-lists with the WGS-lists yields the *GS-lists* [6, p.16]. An important property of the GS-lists [6, Theorem 1.2.5] is that, if each man is given his first-choice partner (or equivalently, each woman is given her last-choice partner) in the GS-lists then we obtain a stable matching called the *man-optimal* stable matching. In the man-optimal (or equivalently, *woman-pessimal*) stable matching, each man has the best partner (according to his ranking) that he could obtain, whilst each woman has the worst partner that she need accept, in any stable matching. An analogous procedure, switching the roles of the men and women, gives the *woman-optimal* (or equivalently, *man-pessimal*) stable matching.

An example SM instance I is given in Figure 1, together with the GS-lists for I. (Throughout this paper, a person's preference list is ordered with his/her most-preferred partner leftmost.) There are three stable matchings for this instance: $\{(1,1), (2,2), (3,4), (4,6), (5,5), (6,3)\}$ (the man-optimal stable matching); $\{(1,1), (2,2), (3,4), (4,3), (5,6), (6,5)\}$ (the woman-optimal stable matching); and $\{(1,1), (2,2), (3,4), (4,5), (5,6), (6,3)\}$.

SMI is a generalisation of SM in which the preference lists of those involved can be incomplete. In this case, person p is *acceptable* to person q if p appears on the preference list of q, and *unacceptable* otherwise. A matching M in an instance I of SMI is a one-one correspondence between a subset of the men and a subset of the women, such that $(m, w) \in M$ implies that each of m and w is acceptable to the other. In this setting, a man m and woman w form a blocking pair for M if each is either unmatched in M and finds the other acceptable, or prefers the other to his/her partner in M. As in SM, a matching is stable if it admits no blocking pair. (It follows from this definition that, from the point of view of finding stable matchings, we may assume without loss of generality that p is acceptable to q if and only if q is acceptable to p.) A stable matching in I need not be a complete matching. However, all stable matchings in I involve exactly the same men and women [5]. It is straightforward to modify the Extended Gale/Shapley algorithm to cope with an SMI instance [6, Section 1.4.2]. A pseudocode description of the

```
assign each person to be free;
while some man m is free and m has a nonempty list loop
    w := first woman on m's list;    {m 'proposes' to w}
    if some man p is engaged to w then
        assign p to be free;
    end if;
    assign m and w to be engaged to each other;
    for each successor p of m on w's list loop
        delete the pair {p, w};
    end loop;
end loop;
```

Fig. 2. The man-oriented Extended Gale/Shapley algorithm for SMI.

man-oriented EGS algorithm for SMI is given in Figure 2 (the term *delete the pair* $\{p,w\}$ means that p should be deleted from w's list and vice versa.) The woman-oriented algorithm is analogous. Furthermore, the concept of GS-lists can be extended to SMI, with analogous properties.

The Stable Marriage problem has its roots as a combinatorial problem, but has also been the subject of much interest from the Game Theory and Economics community [13] and the Operations Research community [14]. In this paper we present two encodings of an instance I of SMI (and so of SM) as an instance J of a Constraint Satisfaction Problem (CSP). We show that Arc Consistency (AC) propagation [1] achieves the same results as the EGS algorithm in a certain sense. For the first encoding, we show that the GS-lists for I correspond to the domains remaining after establishing AC in J. The second encoding is more compact; although the arc consistent domains in J are supersets of the GS-lists, we can again obtain from them the man-optimal and woman-optimal stable matchings in I. We also show that, for both encodings, we are guaranteed a failure-free enumeration of all stable matchings in I using AC propagation (combined with a value-ordering heuristic in the case of the first encoding) in J.

Our results show that constraint propagation within a CSP formulation of SM captures the structure produced by the EGS algorithm. We have also demonstrated the applicability of constraint programming to the general domain of stable matching problems. Many variants of SM are NP-hard [11, 10, 8], and the encodings presented here could potentially be extended to these variants, giving a way of dealing with their complexity through existing CSP search algorithms.

The remainder of this paper is organised as follows. In Section 2 we present the first encoding, then prove the consequent relationship between AC propagation and the GS-lists in Section 3; the failure-free enumeration result for this encoding is presented in Section 4. A second encoding, using Boolean variables, is given in Section 5, and in Section 6 we show the relationship between AC propagation in this encoding and the man-optimal and woman-optimal stable matchings, together with the failure-free enumeration result. Section 7 contains some concluding remarks.

2 A first encoding for SM and SMI

In this section we present an encoding of the Stable Marriage problem, and indeed more generally SMI, as a binary constraint satisfaction problem.

Suppose that we are given an SMI instance I involving men $m_1, m_2, \ldots, m_n$ and women $w_1, w_2, \ldots, w_n$ (it is not difficult to extend our encoding to the case that the numbers of men and women are not equal, but for simplicity we assume that they are equal). For any person q in I, $PL(q)$ (respectively $GS(q)$) denotes the set of persons contained in the original preference list (GS-list) of q in I. For the purposes of exposition, we introduce a dummy man m_{n+1} and a dummy woman w_{n+1} into the SMI instance, such that, for each i, m_i (respectively w_i) prefers all women (men) on his (her) preference list (if any) to w_{n+1} (m_{n+1}).

To define an encoding of I as a CSP instance J, we introduce variables $x_1, x_2, \ldots, x_n$ corresponding to the men, and $y_1, y_2, \ldots, y_n$ corresponding to the women. For each i ($1 \leq i \leq n$), we let $dom(x_i)$ denote the values in variable x_i's domain. Initially, $dom(x_i)$ is defined as follows:

$$dom(x_i) = \{j : w_j \in PL(m_i)\} \cup \{n+1\}.$$

For each j ($1 \leq j \leq n$), $dom(y_j)$ is defined similarly. For each i ($1 \leq i \leq n$), let $d_i^m = |dom(x_i)|$ and let $d_i^w = |dom(y_i)|$. Intuitively, for $1 \leq i, j \leq n$, the assignment $x_i = j$ corresponds to the case that man m_i marries woman w_j, and the constraints of our encoding will ensure that $x_i = j$ if and only if $y_j = i$. Similarly, for $1 \leq i \leq n$, the assignment $x_i = n+1$ (respectively $y_i = n+1$) corresponds to the case that m_i (w_i) is unmatched. It should be pointed out that, if the given SMI instance is an SM instance (i.e. every preference list is complete), then no variable will be assigned the value $n+1$ in its domain in any stable matching.

We now define the constraints between the variables to ensure that the solutions to the CSP correspond exactly to the stable marriages in I. Given any i and j ($1 \leq i, j \leq n$), the *stable marriage constraint* x_i/y_j involving x_i and y_j is a set of nogoods which we represent by a $d_i^m \times d_j^w$ conflict matrix C. To make the structure of the conflict matrix clear, we describe it using four possible values for the element $C_{k,l}$ of C, for any k, l ($k \in dom(x_i)$, $l \in dom(y_j)$), as follows. In a conventional conflict matrix, the values I and B are disallowed so would be 0, while the values A and S are allowed and so would be 1.

A: $C_{k,l} =$ A when $k = j$ and $l = i$, which <u>A</u>llows $x_i = j$ (and $y_j = i$). At most one element in C can ever contain the value A.

I: $C_{k,l} =$ I when *either* $k = j$ and $l \neq i$ *or* $l = i$ and $k \neq j$, i.e. the two pairings are <u>I</u>llegal, since *either* $x_i = j$ and $y_j = l \neq i$ *or* $y_j = i$ and $x_i = k \neq j$.

B: $C_{k,l} =$ B when m_i prefers w_j to w_k and w_j prefers m_i to m_l. Any matching corresponding to the assignment $x_i = k$ and $y_j = l$ would admit a <u>B</u>locking pair involving m_i and w_j.

S: $C_{k,l} =$ S for all other entries that are not A, I or B. The simultaneous assignments of $x_i = k$ and $y_j = l$ are <u>S</u>upported.

The size of each conflict matrix is $O(n^2)$ and clearly there are $O(n^2)$ conflict matrices; consequently the overall size of the encoding is $O(n^4)$.

	2	4	6	1	3	5	7
1				I			
3				I			
6				I			
2	I	I	I	A	I	I	I
4				I	B	B	B
5				I	B	B	B
7				I	B	B	B

(a) x_1/y_2

	4	3	6	2	5	1	7
3	I	I	A	I	I	I	I
1			I	B	B	B	B
2			I	B	B	B	B
5			I	B	B	B	B
6			I	B	B	B	B
4			I	B	B	B	B
7			I	B	B	B	B

(b) x_6/y_3

	3	2	6	1	4	5	7
1	I						
4	I						
5	A	I	I	I	I	I	I
3	I	B	B	B	B	B	B
6	I	B	B	B	B	B	B
2	I	B	B	B	B	B	B
7	I	B	B	B	B	B	B

(c) x_3/y_5

Fig. 3. Conflict matrices for stable marriage constraints from the problem in Figure 1

Examples of different types of conflict matrices for stable marriage constraints x_i/y_j are shown in Figure 3 for the SM instance of Figure 1. In all cases, and henceforth in this paper, the values in x_i's (respectively y_j's) domain are listed in order down the rows (along the columns) according to m_i's (w_j's) preference list, and a blank entry represents an S. Another type of conflict matrix can occur in an SMI instance: the value A does not occur in a conflict matrix x_i/y_j if m_i and w_j are unacceptable to each other, and the matrix is then filled with S's.

Figure 3(a) shows the conflict matrix for the stable marriage constraint x_1/y_2. The row and column of I's, representing illegal marriages, intersect at the A entry, and the area to the right of and below A is filled with B's, representing nogood assignments to x_1 and y_2 which would lead to m_1 and w_2 being a blocking pair.

Figure 3(b) shows the conflict matrix for the stable marriage constraint x_6/y_3. Again the area with A at its top left corner is bounded by I's and filled with B's. However, the A is in the top row, since w_3 is at the top of m_6's preference list. Consequently all values in the domain of y_3 to the right of A are unsupported. Similarly, Figure 3(c) shows the conflict matrix for the stable marriage constraint x_3/y_5, where m_3 is at the top of w_5's preference list. The A entry is in the first column and all values in the domain of x_3 below the A are unsupported.

Enforcing AC on the instance of Figure 1 will delete the rows and columns from Figure 3(b) and (c) corresponding to unsupported values. As shown in the next section, these deletions are equivalent to those done by the EGS algorithm.

3 Arc consistency and the GS-lists

In this section we prove that, if I is an SMI instance and J is a CSP instance obtained from I using the encoding of Section 2, AC propagation in J essentially calculates the GS-lists of I.[1] The proof depends on two lemmas. The first shows

[1] Strictly speaking, we prove that, after AC propagation, for any i, j ($1 \leq i, j \leq n$), $w_j \in GS(m_i)$ iff $j \in dom(x_i)$, and similarly $m_i \in GS(w_j)$ iff $i \in dom(y_j)$.

that the domains remaining after AC propagation, apart from the dummy values, are subsets of the GS-lists. We prove this by showing that, when the EGS algorithm removes a value, so does the AC algorithm. The second proves that the GS-lists are subsets of the domains remaining after AC propagation. We do this by showing that the GS-lists correspond to arc consistent domains for the variables in J.

Lemma 1. *For a given variable x_i in J $(1 \leq i \leq n)$, after AC propagation,*

$$\{w_j : j \in dom(x_i) \backslash \{n+1\}\} \subseteq GS(m_i).$$

A similar result holds for each variable y_j $(1 \leq j \leq n)$.

Proof. The GS-lists for I are obtained from the original preference lists in I by deletions carried out by either the man-oriented or woman-oriented EGS algorithms. We show that the corresponding deletions would occur from the relevant variables' domains during AC propagation in J. The proof for deletions resulting from the man-oriented version is presented; the argument for deletions resulting from the woman-oriented version is similar.

We prove the following fact by induction on the number of proposals z during an execution E of the man-oriented EGS algorithm (see Figure 2) on I: for any deletion carried out in the same iteration of the while loop as the zth proposal, the corresponding deletion would be carried out during AC propagation. Clearly the result is true for $z = 0$. Now assume that $z = r > 0$ and the result is true for all $z < r$. Suppose that the rth proposal during E consists of man m_i proposing to woman w_j. At this point of E, we may use the induction hypothesis to deduce that, at some point during AC propagation, the conflict matrix for the stable marriage constraint x_i/y_j has a structure analogous to that of Figure 4(a), since w_j is at the top of m_i's list. Now suppose that in E, during the same iteration of the while loop as the rth proposal, the pair $\{m_k, w_j\}$ is deleted. Then in J, all values in y_j's domain to the right of the entry A (including k and $n+1$) are unsupported, and will be deleted when the constraint is revised during AC propagation. Subsequent revision of the constraint x_k/y_j will remove j from x_k's domain, since k is no longer in y_j's domain and therefore the jth row of the conflict matrix for x_k/y_j contains only I entries. Hence the inductive step is established.

Consequently, any deletion of a value from a preference list by the man-oriented EGS algorithm will be matched by a deletion of a value from the domain of the corresponding CSP variable when AC is enforced. The same is true for the woman-oriented EGS algorithm. The end result is that the domains remaining after AC propagation, omitting the dummy value, are subsets of the GS-lists. □

Lemma 2. *For each i $(1 \leq i \leq n)$, define a domain of values $dom(x_i)$ for the variable x_i as follows: if $GS(m_i) \neq \emptyset$, then $dom(x_i) = \{j : w_j \in GS(m_i)\}$; otherwise $dom(x_i) = \{n+1\}$. The domain for each y_j $(1 \leq j \leq n)$ is defined analogously. Then the domains so defined are arc consistent in J.*

			i			
j	I	I	A	I	I	I
	S	S	I	B	B	B
	S	S	I	B	B	B
	S	S	I	B	B	B
	S	S	I	B	B	B

(a)

					i
j	I	I	I	I	A
	S	S	S	S	I
	S	S	S	S	I
	S	S	S	S	I
	S	S	S	S	I
	S	S	S	S	I

(b)

			i			
	S	S	I	S	S	S
j	I	I	A	I	I	I
	S	S	I	B	B	B
	S	S	I	B	B	B
	S	S	I	B	B	B
	S	S	I	B	B	B

(c)

	S	S	S	S	S
	S	S	S	S	S
	S	S	S	S	S
	S	S	S	S	S
	S	S	S	S	S

(d)

Fig. 4. Four possible types of stable marriage constraints x_i/y_j

Proof. Suppose that the variables x_i $(1 \leq i \leq n)$ and y_j $(1 \leq j \leq n)$ are assigned the domains in the statement of the lemma. To show that these domains are arc consistent, we consider an arbitrary constraint x_i/y_j. There are six cases to consider:

- w_j is at the top of m_i's GS-list. Then m_i is at the bottom of w_j's GS-list. Hence the constraint x_i/y_j has a structure similar to that of Figure 4(b). Every row or column has at least one A or S and the constraint is arc consistent.
- w_j is at the bottom of m_i's GS-list. Then m_i is at the top of w_j's GS-list. Hence the constraint x_i/y_j has a structure similar to that of the transpose of Figure 4(b) and is arc consistent.
- w_j is in m_i's GS-list, but is not at the top or bottom of that list. Then the constraint x_i/y_j has a structure similar to that of Figure 4(c) (i.e. every row or column has at least one A or S), and is again arc consistent.
- $w_j \notin GS(m_i)$, but $w_j \in PL(m_i)$ and $GS(m_i) \neq \emptyset$. Then $m_i \notin GS(w_j)$. The pair $\{m_i, w_j\}$ were deleted from each other's original lists by either the man-oriented EGS algorithm (in which case all successors of m_i on w_j's original list were also deleted) or the woman-oriented EGS algorithm (in which case all successors of w_j on m_i's original list were also deleted). In either case, the constraint x_i/y_j has a structure similar to that of Figure 4(d) and is again arc consistent, since all A,B and I entries have been removed, leaving only S entries.
- $w_j \notin PL(m_i)$, so $w_j \notin GS(m_i)$, but $GS(m_i) \neq \emptyset$. Then it is straightforward to verify that the constraint x_i/y_j has a structure similar to that of Figure 4(d) and is arc consistent.
- $GS(m_i) = \emptyset$. Then the constraint x_i/y_j is a 1×1 conflict matrix with a single entry S and is arc consistent.

Hence no constraint yields an unsupported value for any variable, and the set of domains defined in the lemma is arc consistent. □

The following theorem follows immediately from the above lemmas, and the fact that AC algorithms find the unique maximal set of domains that are arc consistent.

Theorem 3 *Let I be an instance of SMI, and let J be a CSP instance obtained from I by the encoding of Section 2. Then the domains remaining after AC propagation in J are identical (in the sense of Footnote 1) to the GS-lists for I.*

Theorem 3 and the discussion of GS-lists in Section 1 show that we can find a solution to the CSP giving the man-optimal stable matching without search: we assign each x_i variable the most-preferred value[2] in its domain. Assigning the y_j variables in a similar fashion gives the woman-optimal stable matching. In the next section, we go further and show that the CSP yields all stable matchings without having to backtrack due to failure.

4 Failure-free enumeration

In this section we show that, if I is an SM (or more generally SMI) instance and J is a CSP instance obtained from I using the encoding of Section 2, then we may enumerate the solutions of I in a failure-free manner using AC propagation combined with a suitable value-ordering heuristic in J.

Theorem 4 *Let I be an instance of SMI and let J be a CSP instance obtained from I using the encoding of Section 2. Then the following search process enumerates all solutions in I without repetition and without ever failing due to an inconsistency:*

- *AC is established as a preprocessing step, and after each branching decision including the decision to remove a value from a domain;*
- *if all domains are arc consistent and some variable x_i has two or more values in its domain then search proceeds by setting x_i to the most-preferred value j in its domain. On backtracking, the value j is removed from x_i's domain;*
- *when a solution is found, it is reported and backtracking is forced.*

Proof. Let T be the search tree as defined above. We prove by induction on T that each node of T corresponds to a CSP instance J' with arc consistent domains; furthermore J' is equivalent to the GS-lists I' for an SMI instance derived from I, such that any stable matching in I' is also stable in I. Firstly we show that this is true for the root node of T, and then we assume that this is true at any branching node u of T and show that it is true for each of the two children of u.

The root node of T corresponds to the CSP instance J' with arc consistent domains, where J' is obtained from J by AC propagation. By Theorem 3, J' corresponds to the GS-lists in I, which we denote by I'. By standard properties of the GS-lists [6, Theorem 1.2.5], any stable matching in I' is stable in I.

Now suppose that we have reached a branching node u of T. By the induction hypothesis, u corresponds to a CSP instance J' with arc consistent domains, and

[2] Implicitly we assume that variable x_i inherits the corresponding preferences over the values in its domain from the preference list of man m_i.

also J' is equivalent to the GS-lists I' for an SMI instance derived from I such that any stable matching in I' is also stable in I. As u is a branching node of T, there is some i $(1 \leq i \leq n)$ such that variable x_i's domain has size > 1. Hence in T, when branching from node u to its two children v_1 and v_2, two CSP instances J'_1 and J'_2 are derived from J' as follows. In J'_1, x_i is set to the most-preferred value j in its domain and y_j is set to i, and in J'_2, value j is removed from x_i's domain and value i is removed from y_j's domain.

We firstly consider instance J'_1. During arc consistency propagation in J'_1, revision of the constraint x_k/y_j, for any k such that w_j prefers m_k to m_i, forces l to be removed from the domain of x_k, for any l such that m_k prefers w_j to w_l (and similarly k is removed from the domain of y_l). Hence after such revisions, J'_1 corresponds to the SMI instance I'_1 obtained from I' by deleting pairs of the form $\{m_i, w_l\}$ (where $l \neq j$), $\{m_k, w_j\}$ (where $k \neq i$) and $\{m_k, w_l\}$ (where w_j prefers m_k to m_i and m_k prefers w_j to w_l). It is straightforward to verify that any stable matching in I'_1 is also stable in I', which is in turn stable in I by the induction hypothesis. At node v_1, AC is established in J'_1, giving the CSP instance J''_1 which we associate with this node. By Theorem 3, J''_1 corresponds to the GS-lists I''_1 of the SMI instance I'_1. By standard properties of the GS-lists [6, Section 1.2.5], any stable matching in I''_1 is also stable in I'_1, which is in turn stable in I by the preceding argument.

We now consider instance J'_2, which corresponds to the SMI instance I'_2 obtained from I' by deleting the pair $\{m_i, w_j\}$. It is straightforward to verify that any stable matching in I'_2 is also stable in I', which is in turn stable in I by the induction hypothesis. At node v_2, AC is established in J'_2, giving the CSP instance J''_2 which we associate with this node. The remainder of the argument for this case is identical to the corresponding part in the previous paragraph.

Hence the induction step holds, so that the result is true for all nodes of T. Therefore the branching process never fails due to an inconsistency, and it is straightforward to verify that no part of the search space is omitted, so that the search process lists all stable matchings in the SMI instance I. Finally we note that different complete solutions correspond to different stable matchings, so no stable matching is repeated. $\square$

5 A Boolean encoding of SM and SMI

In this section we give a less obvious but more compact encoding of an SMI instance as a CSP instance. As in Section 2, suppose that I is an SMI instance involving men $m_1, m_2, \ldots, m_n$ and women $w_1, w_2, \ldots, w_n$. For each i $(1 \leq i \leq n)$ let l_i^m denote the length of man m_i's preference list, and define l_i^w similarly.

To define an encoding of I as a CSP instance J, we introduce $O(n^2)$ Boolean variables and $O(n^2)$ constraints. For each i, j $(1 \leq i, j \leq n)$, the variables are labelled $x_{i,p}$ for $1 \leq p \leq l_i^m + 1$ and $y_{j,q}$ for $1 \leq q \leq l_j^w + 1$, and take only two values, namely T and F. The interpretation of these variables is:

- $x_{i,p} = T$ iff man m_i is matched to his p^{th} or worse choice woman or is unmatched, for $1 \leq p \leq l_i^m$;

1. $x_{i,1} = T$	$(1 \leq i \leq n)$
2. $y_{j,1} = T$	$(1 \leq j \leq n)$
3. $x_{i,p} = F \rightarrow x_{i,p+1} = F$	$(1 \leq i \leq n, 2 \leq p \leq l_i^m)$
4. $y_{j,q} = F \rightarrow y_{j,q+1} = F$	$(1 \leq j \leq n, 2 \leq q \leq l_j^w)$
5. $x_{i,p} = T$ & $y_{j,q} = F \rightarrow x_{i,p+1} = T$	$(1 \leq i, j \leq n)$ (*)
6. $y_{j,q} = T$ & $x_{i,p} = F \rightarrow y_{j,q+1} = T$	$(1 \leq i, j \leq n)$ (*)
7. $x_{i,p} = T \rightarrow y_{j,q+1} = F$	$(1 \leq i, j \leq n)$ (*)
8. $y_{j,q} = T \rightarrow x_{i,p+1} = F$	$(1 \leq i, j \leq n)$ (*)

Table 1. The constraints in a Boolean encoding of an SMI instance.

- $x_{i,p} = T$ iff man m_i is unmatched, for $p = l_i^m + 1$;
- $y_{j,q} = T$ iff woman w_j is matched to her q^{th} or worse choice man or is unmatched, for $1 \leq q \leq l_j^w$;
- $y_{j,q} = T$ iff woman w_j is unmatched, for $q = l_j^w + 1$.

The constraints are listed in Table 1. For each i and j $(1 \leq i, j \leq n)$, the constraints marked (*) are present if and only if m_i finds w_j acceptable; in this case p is the rank of w_j in m_i's list and q is the rank of m_i in w_j's list.

Constraints 1 and 2 are trivial, since each man and woman is either matched with some partner or is unmatched. Constraints 3 and 4 enforce monotonicity: if a man gets his $p-1^{th}$ or better choice, he certainly gets his p^{th} or better choice. For Constraints 5-8, let i and j be arbitrary $(1 \leq i, j \leq n)$, and suppose that m_i finds w_j acceptable, where p is the rank of w_j in m_i's list and q is the rank of m_i in w_j's list. Constraints 5 and 6 are monogamy constraints; consider Constraint 5 (the explanation of Constraint 6 is similar). If m_i has a partner no better than w_j or is unmatched, and w_j has a partner she prefers to m_i, then m_i cannot be matched to w_j, so m_i has his $(p+1)$th-choice or worse partner, or is unmatched. Constraints 7 and 8 are stability constraints; consider Constraint 7 (the explanation of Constraint 8 is similar). If m_i has a partner no better than w_j or is unmatched, then w_j must have a partner no worse than m_i, for otherwise m_i and w_j would form a blocking pair.

The next section focuses on AC propagation in J.

6 Arc consistency in the Boolean encoding

In this section we consider the effect of AC propagation on a CSP instance J obtained from an SMI instance I by the encoding of Section 5. We show that, using AC propagation in J, we may recover the man-optimal and woman-optimal stable matchings in I, and moreover, we may enumerate *all* stable matchings in I in a failure-free manner.

Imposing AC in J corresponds (in a looser sense than with the first encoding) to the application of the EGS algorithm in I from both the men's and women's sides. Indeed, we can understand the variables in terms of proposals in the EGS algorithm. That is, $x_{i,p}$ being true corresponds to m_i's $p-1^{th}$ choice woman

Men's lists	Women's lists
1: 1	1: 1
2: 2	2: 2
3: 4	3: 4 6
4: 6 5 3	4: 3
5: 5 6	5: 6 4 5
6: 3 6 5	6: 5 6 4

(a)

Men's lists	Women's lists
1: 1	1: 1
2: 2	2: 2
3: 4	3: 4 3 6
4: 6 5 3	4: 3
5: 5 6	5: 6 1 4 5
6: 3 1 2 6 5	6: 5 1 3 6 4

(b)

Fig. 5. (a) The GS-lists for the SM instance of Figure 1, and (b) the possible partners remaining after AC is applied in the Boolean encoding.

rejecting him after a proposal from a man she likes more. Consequently, the maximum value of p for which $x_{i,p}$ is true gives the best choice that will accept m_i, and the lowest value of p such that $x_{i,p+1}$ is false gives the worst choice that he need accept (and the same holds for the $y_{j,q}$ variables). In general, we will prove that, for a given person p in I, AC propagation in J yields a reduced preference list for p which we call the *Extended GS-list* or *XGS-list* – this contains all elements in p's preference list between the first and last entries of his/her GS-list (inclusive). For example, Figure 5(a) repeats the GS-lists from Figure 1, and (b) shows the XGS-lists after AC is enforced. Note that in general, the XGS-lists may include some values not in the GS-lists.

We now describe how we can use AC propagation in order to derive the XGS-lists for I. After we apply AC in J, the monotonicity constraints force the domains for the $x_{i,p}$ variables to follow a simple sequence, for $p = 1$ to $l_i^m + 1$. First, there is a sequence of domains $\{T\}$, then a sequence of domains which remain $\{T, F\}$, and a final sequence of domains $\{F\}$. The first sequence must be non-empty because $x_{i,1} = T$. If the middle sequence is empty then all variables associated with m_i are determined, while if the last sequence is empty it might still happen that m_i fails to find any partner at all. More formally, let π $(1 \leq \pi \leq l_i^m + 1)$ be the largest integer such that $dom(x_{i,\pi}) = \{T\}$, and let π' be the largest integer such that $T \in dom(x_{i,\pi'})$. We will prove that, if $\pi = l_i^m + 1$ then the XGS-list of m_i is empty; otherwise the XGS-list of m_i contains all people on m_i's original preference list between positions π and π' (inclusive). Hence, in the latter case, a man m_i's XGS-list consists of the women at position p in his original list, for each p such that $dom(x_{i,p}) = \{T, F\}$ after AC-propagation, together with the woman in position π in his original list. A similar correspondence exists between the women's XGS-lists and the $y_{j,q}$ variables.

As in Section 3, the proof of this result uses two lemmas. The first shows that the domains remaining after AC propagation correspond to subsets of the XGS-lists, whilst the second shows that the XGS-lists correspond to arc consistent domains.

Lemma 5. *For a given i $(1 \leq i \leq n)$, after AC propagation in J, let p be the largest integer such that $dom(x_{i,p}) = \{T\}$ and let p' be the largest integer such that $T \in dom(x_{i,p'})$. If $p < l_i^m + 1$ then all entries of m_i's preference list between* *positions p and p' belong to the XGS-list of m_i. A similar correspondence holds for the women's lists.*

Proof. The first entry on a man m's XGS-list corresponds to the last woman (if any) to whom m proposed during an execution of the man-oriented EGS-algorithm. Similarly the last entry on a woman w's XGS-list corresponds to the last man (if any) who proposed to w during an execution of the man-oriented EGS-algorithm. A similar correspondence in terms of the woman-oriented EGS-algorithm yields the first entry on a woman's XGS-list and the last entry on a man's XGS-list. We prove that, if a person q is missing from a person p's XGS-list, then AC propagation reduces the domains of the variables relating to person p correspondingly. (We consider only the correspondences involving the man-oriented EGS-algorithm; the gender-reversed argument involving the woman-oriented EGS-algorithm yields the remaining cases.)

It suffices to prove the following result by induction on the number of proposals z during an execution E of the man-oriented EGS algorithm (see Figure 2) on I: if proposal z consists of man m_i proposing to woman w_j, then $x_{i,t} = T$ for $1 \leq t \leq p$ and $y_{j,t} = F$ for $q < t \leq l_j^w + 1$, where p denotes the rank of w_j in m_i's list and q denotes the rank of m_i in w_j's list.

Clearly the result is true for $z = 0$. Now assume that $z = a > 0$ and the result is true for all $z < a$. Suppose that the a^{th} proposal during E consists of man m_i proposing to woman w_j. Suppose that p is the rank of w_j in m_i's list and q is the rank of m_i in w_j's list. Suppose firstly that $p = 1$. Then $x_{i,1} = T$ by Constraint 1, and $y_{j,t} = F$ for $q < t \leq l_j^w + 1$ by Constraints 7 and 4, since $x_{i,p}$'s value has been determined. Now suppose that $p > 1$. Then previously m_i proposed to w_k, his $p - 1^{th}$-choice woman (since m_i proposes in his preference list order, starting with his most-preferred woman). By the induction hypothesis, $x_{i,t} = T$ for $1 \leq t \leq p - 1$. Woman w_k rejected m_i because she received a proposal from some man m_l whom she prefers to m_i. Let r, s be the ranks of m_l, m_i in w_k's list respectively, so that $r < s$. By the induction hypothesis, $y_{k,t} = F$ for $t \geq r + 1$. Thus in particular, $y_{k,s} = F$, so that by Constraint 5, $x_{i,p} = T$, since the values of $x_{i,p-1}$ and $y_{k,s}$ have been determined. Thus by Constraints 7 and 4, $y_{j,t} = F$ for $q < t \leq l_j^w + 1$, since $x_{i,p}$'s value has been determined. This completes the induction step.

Thus the proof of the lemma is established, so that the domains remaining after AC is enforced correspond to subsets of the XGS-lists. □

Lemma 6. *For each i $(1 \leq i \leq n)$, define a domain of values $dom(x_{i,t})$ for the variables $x_{i,t}$ $(1 \leq t \leq l_i^m + 1)$ as follows: if the XGS-list of m_i is empty, $dom(x_{i,t}) = \{T\}$ for $1 \leq t \leq l_i^m + 1$. Otherwise, let p and p' be the ranks (in m_i's preference list) of the first and last women on m_i's XGS-list respectively. $dom(x_{i,t}) = \{T\}$ for $1 \leq t \leq p$, $dom(x_{i,t}) = \{F\}$ for $p' + 1 \leq t \leq l_i^m + 1$ and $dom(x_{i,t}) = \{T, F\}$ for $p < t \leq p'$. The domains for each variable $y_{j,t}$ $(1 \leq j \leq n,\ 1 \leq t \leq l_j^w + 1)$ are defined analogously. Then the domains so defined are arc consistent in J.*

Proof. The proof of this lemma is along similar lines to that of Lemma 2 and involves showing that Constraints 1 to 8 in Table 1 are arc consistent under the assignments defined above; we omit the details for space reasons. □

The following theorem follows immediately from the above lemmas, and the fact that AC algorithms find the unique maximal set of arc consistent domains.

Theorem 7 *Let I be an instance of SMI, and let J be a CSP instance obtained from I by the encoding of Section 5. Then the domains remaining after AC propagation in J are identical (in the sense described before Lemma 5) to the XGS-lists for I.*

Hence Theorem 7 shows that we may find solutions to the CSP giving the man-optimal and woman-optimal stable matchings in I without search.

We remark in passing that the SAT-based technique of unit propagation is strong enough for the same results to hold. This makes no theoretical difference to the cost of establishing AC, although in practice we would expect unit propagation to be cheaper. This observation implies that a SAT solver applying unit propagation exhaustively, e.g. a Davis-Putnam program [2], will perform essentially the same work as an AC-based algorithm.

As before, we show that solutions can be enumerated without failure. The results are better than before in two ways: first, maintenance of AC is much less expensive, and second, there is no need for a specific variable or value ordering.

Theorem 8 *Let I be an instance of SMI and let J be a CSP instance obtained from I using the encoding of Section 5. Then the following search process enumerates all solutions in I without repetition and without ever failing due to an inconsistency:*

- *AC is established as a preprocessing step, and after each branching decision including the decision to remove a value from a domain;*
- *if all domains are arc consistent and some variable v has two values in its domain, then search proceeds by setting v to T, and on backtracking, to F;*
- *when a solution is found, it is reported and backtracking is forced.*

Proof. This result can be proved by an inductive argument similar to that used in the proof of Theorem 4. The full details are omitted here for space reasons, but we indicate below the important points that are specific to this context.

An SMI instance is guaranteed to have a stable matching, though not necessarily a complete one [6, Section 1.4.2] so the initial establishing of AC in J cannot result in failure.

Branching decisions are only made when AC has been established, so Theorem 7 applies at branching points. If all domains are of size 1, we report the solution and terminate. Otherwise, we choose any variable with domain of size 2 and create two branches with the variable set to T and F respectively. If the variable represents a man, setting it to T excludes the man-optimal matching, but the man-pessimal matching remains possible so this branch still contains a solution. Conversely, setting the variable to F excludes the man-pessimal matching but leaves the man-optimal matching, so this branch also contains a solution.

The process of establishing AC never removes values which participate in any solution. As the branching process omits no part of the search space, the search process lists all solutions to the SMI instance. Finally we note that different complete solutions correspond to different stable matchings, so no stable matching is repeated. □

We conclude this section with a remark about the time complexities of AC propagation in both encodings. In general, AC can be established in $O(ed^r)$ time [1], where there are e constraints, each of arity r, and domain size is d. In the encoding of Section 5, $e = O(n^2)$, $d = 2$ and $r \leq 3$. Thus AC can be established in $O(n^2)$ time, which is linear in the size of the input. Hence this encoding of SM achieves the solution in $O(n^2)$ time, which is known to be optimal [9]. We find it remarkable that such a strong result can be obtained without any special-purpose consistency algorithms. Furthermore, this result contrasts with the time complexity of AC propagation in the encoding of Section 2: in this case, $e = O(n^2)$, $d = O(n)$ and $r = 2$, so that AC can be established in $O(n^4)$ time.

7 Conclusion

We have presented two ways of encoding the Stable Marriage problem and its variant SMI as a CSP. The first is a straightforward representation of the problem as a binary CSP. We show that enforcing AC in the CSP gives reduced domains which are equivalent to the GS-lists produced by the Extended Gale-Shapley algorithm, and from which the man-optimal and woman-optimal matchings can be immediately derived. Indeed, we show that all solutions can be found without failure, provided that values are assigned in preference-list order.

Enforcing AC using an algorithm such as AC-3 would be much more time-consuming than the EGS algorithm because of the number and size of the constraints. A constraint propagation algorithm tailored to the stable marriage constraint would do much better, but to get equivalent performance to EGS we should effectively have to embed EGS into our constraint solver.

Nevertheless, the fact that we can solve the CSP without search after AC has been achieved shows that this class of CSP is tractable. Previous tractability results have identified classes of constraint graph (e.g. [3]) or classes of constraint (e.g. [7]) which guarantee tractability. In the binary CSP encoding of SM, it is the combination of the structure of the constraints (a bipartite graph) and their type (the stable-marriage constraint) that ensures that we find solutions efficiently.

The second encoding we present is somewhat more contrived, but allows AC to be established, using a general algorithm, with time complexity equivalent to that of the EGS algorithm. Although the arc consistent domains do not exactly correspond to the GS-lists, we can again find man-optimal and woman-optimal matchings immediately, and all stable matchings without encountering failure during the search. Hence, this encoding yields a CSP-based method for solving SM and SMI which is equivalent in efficiency to EGS.

The practical application of this work is to those variants of SM and SMI which are NP-hard [11, 10, 8], or indeed to any situation in which additional constraints on the problem make the EGS algorithm inapplicable. If we can extend one of the encodings presented here to these variants, we then have tools to solve them, since we have ready-made search algorithms available for CSPs.

This paper provides a partial answer to a more general question: if we have a problem which can be expressed as a CSP, but for which a special-purpose algorithm is available, is it ever sensible to formulate the problem as a CSP? SM shows that it can be: provided that the encoding is carefully done, existing algorithms for simplifying and solving CSPs may give equivalent performance to the special-purpose algorithm, with the benefit of easy extension to variants of the original problem where the special-purpose algorithm might be inapplicable.

References

1. C. Bessière and J-C Régin. Arc consistency for general constraint networks: Preliminary results. In *Proceedings of IJCAI'97*, pages 398–404, 1997.
2. M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5:394–397, 1962.
3. Eugene C. Freuder. A sufficient condition for backtrack-free search. *Journal of the ACM*, 29:24–32, 1982.
4. D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.
5. D. Gale and M. Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11:223–232, 1985.
6. D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. The MIT Press, 1989.
7. P. Jeavons, D. Cohen, and M. Gyssens. A unifying framework for tractable constraints. In *Proceedings CP'95*, volume LNCS 976, pages 276–291. Springer, 1995.
8. D.F. Manlove, R.W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. To appear in *Theoretical Computer Science*.
9. C. Ng and D.S. Hirschberg. Lower bounds for the stable marriage problem and its variants. *SIAM Journal on Computing*, 19:71–77, 1990.
10. C. Ng and D.S. Hirschberg. Three-dimensional stable matching problems. *SIAM Journal on Discrete Mathematics*, 4:245–252, 1991.
11. E. Ronn. NP-complete stable matching problems. *Journal of Algorithms*, 11:285–304, 1990.
12. A.E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 92(6):991–1016, 1984.
13. A.E. Roth and M.A.O. Sotomayor. *Two-sided matching: a study in game-theoretic modeling and analysis*, volume 18 of *Econometric Society Monographs*. Cambridge University Press, 1990.
14. J.E. Vande Vate. Linear programming brings marital bliss. *Operations Research Letters*, 1989.