SAT Encodings of the Stable Marriage Problem with Ties and Incomplete Lists*

Ian P. Gent[†] and Patrick Prosser[‡]

February 5, 2002

Abstract

We encode a new problem into SAT, namely the Stable Marriage problem with Ties and Incomplete lists (SMTI). This is a matching problem of considerable theoretical interest, but also one with real applications: for example the problem of matching medical interns to hospital residencies in Scotland. We show that the problem can be encoded into SAT. We report computational results showing that Chaff can solve reasonably large random instances of SMTI.

Note to SAT 2002 Committee: This paper is linked to a paper submitted to ECAI 2002, on solving the SMTI problem using constraint programming [3]. Our current plan for a long paper to the special issue of AMAI is to combine these two papers and thereby compare the techniques.

1 The SMTI Problem

In the stable marriage problem [4] we have n men and n women. Each man ranks the n women, giving himself a preference list. Similarly each woman ranks the men, giving herself a preference list. The problem is then to marry men and women such

*This work was supported by EPSRC research grant GR/M90641. The authors are grateful for helpful discussions with many colleagues, most especially Rob Irving, Iain McDonald, David Manlove, Barbara Smith and Toby Walsh.

[†]School of Computer Science, University of St. Andrews, Scotland. ipg@dcs.st-and.ac.uk

[‡]Department of Computing Science, University of Glasgow, Scotland. pat@dcs.gla.ac.uk

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

Figure 1: An SMTI instance with 6 men and 6 women. The instance has a largest weakly stable matching of size 6 namely (2,6,3,1,5,4), and a smallest weakly stable matching of size 4, namely (2,-,-,3,5,6). However, it has no strongly or super stable matching.

that they are *stable*. By stable we mean that there is no incentive for individuals to divorce and elope. This problem has a long history, and an optimal algorithm was proposed by Gale and Shapely 40 years ago [1]. The algorithm's complexity is $O(n^2)$, and is linear in the size of the problem.

If men or women find some members of the opposite sex unacceptable, preference lists become incomplete. These problems are classified as stable marriage problems with incomplete lists (SMI) and are again solvable in polynomial time. We might also have ties in the preference lists. That is, a man (or a woman) might be indifferent between a number of his (or her) choices. For example man m_4 might have a preference such that he prefers woman w_9 to w_7, but woman w_7 ties with woman w_2. In the extreme when all potential partners tie with one another, we are asking only for a matching and stability is not an issue. However, when we combine ties with incompleteness, we get the stable marriage problem with ties and incomplete lists (SMTI) and this is NP-Complete [5]. Figure 1 gives an example SMTI instance.

There are three definitions of stability used when there are ties, giving rise to the three versions of the SMTI problem. They define the conditions under which a couple will elope with each other.

- Weak stability: A couple will leave their partners only if each considers the other better than their current partner.
- Strong stability: A couple will elope if one of them (say the man) strictly prefers the other to their current partner, while the other finds him at least as good as her husband. (The intuition is that he might bribe her to leave.)

- Super stability: A couple will elope if each finds the other as good as their current partner. (The intuition here is the application of the principle that the 'grass is greener on the other side.')

We make two simplifying assumptions throughout: first, that there are the same number n of men as of women; and second, that if a woman is acceptable to a given man (i.e. she appears in his preference list), then he is also acceptable to her.

2 A Boolean encoding of SMTI

In this section we give an encoding of the Stable Marriage problem with Ties and Incomplete lists (SMTI). Suppose that I is an SMTI 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_j^w similarly. We assume that each preference list is ordered in order of preference, most preferred first, with some arbitrary order for potential partners who are tied.

To define an encoding of I as a SAT instance J, we introduce $O(n^2)$ Boolean variables and $O(n^2)$ constraints.[1] 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 unmatched or matched to the woman in p^{th} or later position in his preference list, for $1 \leq p \leq l_i^m$;
- $x_{i,p} = T$ iff man m_i is unmatched, for $p = l_i^m + 1$;
- $y_{j,q} = T$ iff woman w_j is unmatched or matched to the man in q^{th} or later position in her preference list, for $1 \leq q \leq l_j^w$;
- $y_{j,q} = T$ iff woman w_j is unmatched, for $q = l_i^m + 1$;

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.

We need some extra notation to express the stability constraints. The woman who is q^{th} in m_i's preference list may be one of a number of equally desirable in

[1] We use the language of constraints rather than clauses for consistency with our planned long version comparing SAT and CSP encodings.

his eyes. We will need to know who is the next woman who is definitely worse. To achieve this we write q^+ for the position in m_i's list of the first woman who is worse than the woman in position q. Where there is no such woman, $q^+ = l_i^m + 1$. In similar vein, we write q^- for the first position in m_i's list representing a woman tied with q in m_i's preference. For example, suppose m_i's list was (3)(762)(41). Then $3^+ = 5$, since the third woman in the list is 6, and the first woman strictly worse than her is woman 4 in position 5 in the list. We have $3^- = 2$, as woman 7 is the first woman of the same quality as 6. Similarly, we have $1^+ = 2$, $1^- = 1$, $5^+ = 7$, and $5^- = 5$. A general point worth noting is that if a woman is tied with no other person in m_i's list, then we have $q^+ = q + 1$, and $q^- = q$.

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 chanelling constraints between the men and women. If m_i has a partner no better than w_j, his p^{th} choice, but better than his $p + 1^{st}$ choice, he must be married to w_j and she married to him. We express that w_j is married to m_i by saying that she is married to somebody no better than him (5a), but no worse either (5b). The remaining constraints are chosen according to the kind of stability being employed.

- Constraint (7we): If m_i does *strictly worse* than w_j, she must be married to someone *at least as good* as m_i.
- Constraint (7st): If m_i does *no better than* w_j, she must be married to someone *at least as good* as m_i.
- Constraint (7su): If m_i does *no better than* w_j, she must be married to someone *strictly better than* m_i, or alternatively m_i and w_j are married to each other. If m_i is not married to w_j then he is married to someone either higher up or lower down his preference list. Constraint (7su(a)) covers the former and constraint (7su(b)) the latter case.

It is interesting to compare our boolean encoding of SMTI with Gent et al's boolean encoding of the SMI problem [2],[2] which formed the starting point for this work. Since ties are not allowed in SMTI, an SMI instance is an SMTI instance

[2] Available from http://www.dcs.st-and.ac.uk/~apes/2001.html

		Monotonicity	
1.		$x_{i,1} = T$	$(1 \leq i \leq n)$
2.		$y_{j,1} = T$	$(1 \leq j \leq n)$
3.		$x_{i,p} = F \rightarrow x_{i,p+1} = F$	$(1 \leq i \leq n, 1 \leq p \leq l_i^m)$
4.		$y_{j,q} = F \rightarrow y_{j,q+1} = F$	$(1 \leq j \leq n, 1 \leq q \leq l_j^w)$
		Chanelling	
5.	*(a)*	$x_{i,p} = T \mathbin{\&} x_{i,p+1} = F \rightarrow y_{j,q} = T$	(*), $1 \leq i, j \leq n$
	(b)	$x_{i,p} = T \mathbin{\&} x_{i,p+1} = F \rightarrow y_{j,q+1} = F$	(*), $1 \leq i, j \leq n$
6.	*(a)*	$y_{j,q} = T \mathbin{\&} y_{j,q+1} = F \rightarrow x_{i,p} = T$	(*), $1 \leq i, j \leq n$
	(b)	$y_{j,q} = T \mathbin{\&} y_{j,q+1} = F \rightarrow x_{i,p+1} = F$	(*), $1 \leq i, j \leq n$
		Weak Stability	
7we.		$x_{i,p^+} = T \rightarrow y_{j,q^+} = F$	(*), $1 \leq i, j \leq n$
8we.		$y_{j,q^+} = T \rightarrow x_{i,p^+} = F$	(*), $1 \leq i, j \leq n$
		Strong Stability	
7st.		$x_{i,p^-} = T \rightarrow y_{j,q^+} = F$	(*), $1 \leq i, j \leq n$
8st.		$y_{j,q^-} = T \rightarrow x_{i,p^+} = F$	(*), $1 \leq i, j \leq n$
		Super Stability	
7su.	*(a)*	$x_{i,p^-} = T \mathbin{\&} x_{i,p} = F \rightarrow y_{j,q^-} = F$	(*), $1 \leq i, j \leq n$
	(b)	$x_{i,p+1} = T \rightarrow y_{j,q^-} = F$	(*), $1 \leq i, j \leq n$
8su.	*(a)*	$y_{j,q^-} = T \mathbin{\&} y_{j,q} = F \rightarrow x_{i,p^-} = F$	(*), $1 \leq i, j \leq n$
	(b)	$y_{j,q+1} = T \rightarrow x_{i,p^-} = F$	(*), $1 \leq i, j \leq n$

Constraints (1) to (6) are used together with the appropriate versions of (7) & (8). See the main text for explanations of the constraints marked (*) and the symbols p, q, p^+, p^-, q^+, and q^-.

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

in which, as we noted earlier, $p^- = p$ and $p^+ = p + 1$. We can use these facts to study the constraints that our SMTI encoding gives rise to when there are no ties.[3] In the case of strong and super stability, we obtain excellent results. It is not even necessary to notice that a particular SMTI instance has no ties to gain the benefit of the theoretical results which apply to the SMI encoding, in particular that search which does unit propagation will never fail [2]. Unfortunately, the results of Gent et al. no longer apply in the case of weak stability, because not all the deductions necessary can be performed by the process of unit propagation.

3 Experimental Results

We have implemented our encoding. We did this in the Constraint Logic Programming language Eclipse, translating SMTI instances according to our encoding and writing to a file in Dimacs format for input to Chaff. We ran experiments on a Pentium 300MHz machine. Our test instances were a class of randomly generated instance of SMTI is represented by a triple $\langle n, p_1, p_2 \rangle$ where n is the number of men and women in the problem, p_1 is the probability of incompleteness and p_2 is the probability of ties. We describe the generation procedure in detail in [3]. Figure 1 in fact showed a randomly generated SMTI with parameters $\langle 6, 0.5, 0.25 \rangle$.

One set of results are shown in Figure 2.[4] These show performance as we vary n and p_2 but fix $p_1 = 0.5$, for the decision question of whether there is a complete stable matching of size n.[5] The results for strong and super stability are entirely novel. These experiments represent the first time, to our knowledge, that any complete algorithm has been reported for the SMTI problem with these forms of stability.

For weak stability, the only precursor is our own encoding using constraint programming [3]. We can thus compare run times with the SAT encoding. In our constraint program (in the language Choco) we compared results with $n = 60, p_1 = 0.5, p_2 = 0.4$. The constraint solution took an average of 0.8s on a Pentium 733, compared with 0.5s in Chaff on a Pentium 300. This is apparently in favour of SAT, but is less so when we include the time taken by the translation program in Eclipse, of about 0.9s. However, there are theoretical reasons for preferring the SAT encoding. The establishment of arc consistency takes worst case time $O(n^4)$ in the constraint encoding, compared to only $O(n^2)$ for unit propagation in our encoding

[3] We omit details in this extended abstract.

[4] Our long paper will contain much more extensive experimental results.

[5] To implement this we simply added unit clauses that each variable $x_{i,l_i^m+1} = F$.

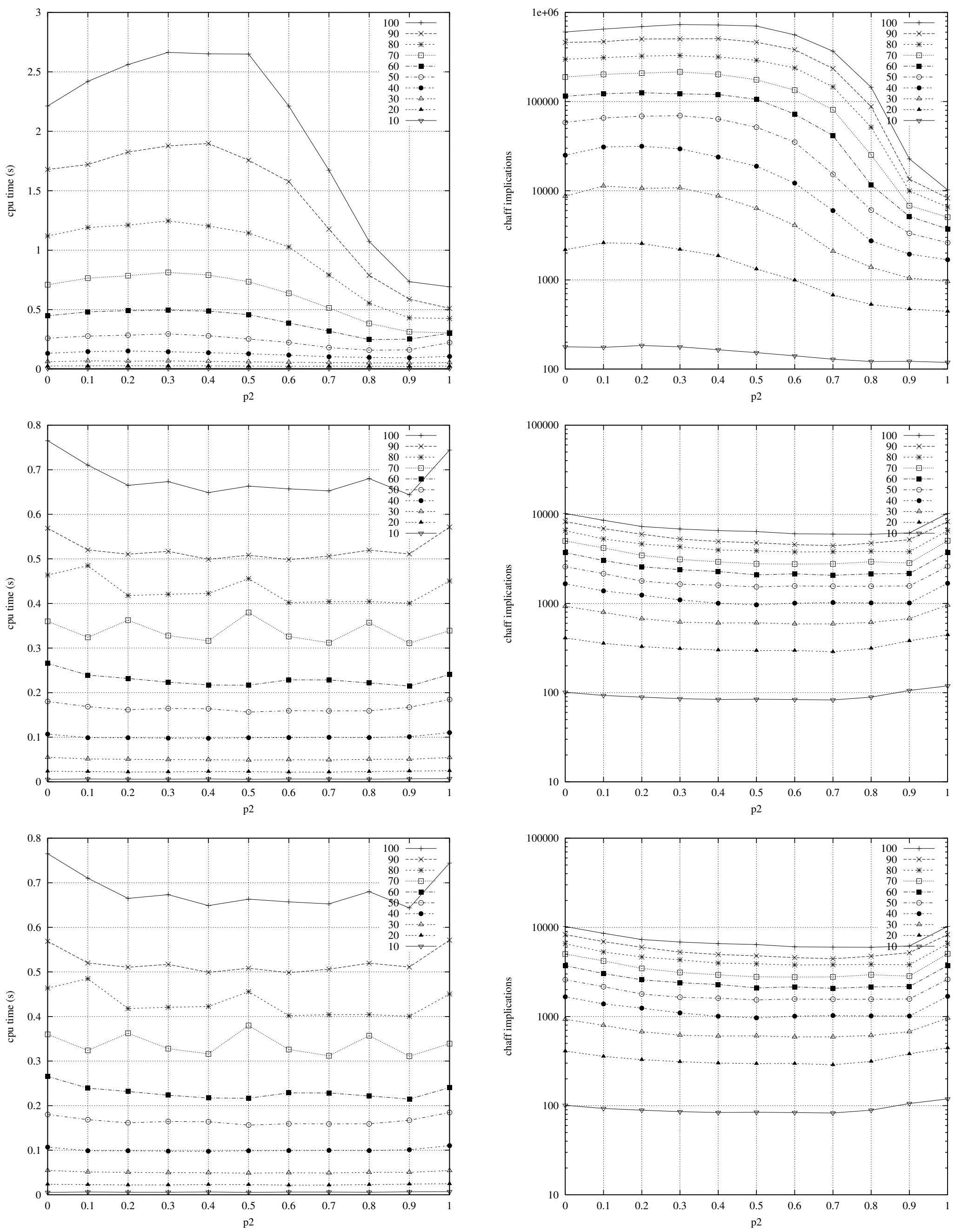

Figure 2: The decision problem: is there a stable matching of size n? Instances with $p_1 = 0.5$ and varying n, p_2. Results are for weak stability (top), strong stability (centre) and super stability (bottom). Left: Chaff's mean cpu time in seconds on a Pentium 300 to solve instances. Right: Chaff's mean number of decisions.

here. This should be seen in improved relative performance of the SAT encoding as we increase n, although this could be offset if more search was necessary.

We conclude that translation into SAT and the use of Chaff represent an effective means of solving SMTI instances, in only a few seconds on instances up to $n = 100$.

4 Further Work

The main item for future work is to allow for the optimisation problem in SMTI: that is, what is the largest (or smallest) stable matching available in a given instance? This is the problem of interest in the Scottish Hospital Residents problem. Unfortunately, unlike constraint solvers, most SAT solvers, and Chaff in particular, are not well suited to this. One possibility would be to use a SAT solver for MAX-SAT, maximising the number of clauses satisfied where certain unit clauses represent that individuals are married.[6] Currently this represents the chief advantage of the constraint encoding, as it is trivial to encode the optimisation problem.

References

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

[2] I.P. Gent, R.W. Irving, D.F. Manlove, P. Prosser, and B.M. Smith. A constraint programming approach to the stable marriage problem. In T. Walsh, editor, *Proceedings of CP-2001*, LNCS, pages 225–239. Springer, 2001.

[3] I.P. Gent and P. Prosser. An empirical study of the stable marriage problem with ties and incomplete lists. Technical Report APES-40-2002, APES Research Group, January 2002. Available from http://www.dcs.st-and.ac.uk/~apes/apesreports.html.

[4] D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. The MIT Press, 1989.

[5] K. Iwama, D.F. Manlove, S. Miyazaki, and Y. Morita. Stable marriage with incomplete lists and ties. In *Proceedings ICALP '99*, pages 443–452, 1999.

[6] We thank Toby Walsh for this suggestion.