Local search algorithms on the Stable Marriage Problem: Experimental Studies [1]

M. Gelain, M. S. Pini, F. Rossi, K. B. Venable [2] and **T. Walsh** [3]

Abstract. The stable marriage problem (SM) has a wide variety of practical applications, ranging from matching resident doctors to hospitals, to matching students to schools, or more generally to any two-sided market. In the classical formulation, n men and n women express their preferences over the members of the other sex. Solving an SM means finding a stable marriage: a matching of men to women with no blocking pair. A blocking pair consists of a man and a woman who are not married to each other but both prefer each other to their partners. It is possible to find a male-optimal (resp., female-optimal) stable marriage in polynomial time. However, it is sometimes desirable to find stable marriages without favoring a group at the expenses of the other one. In this paper we present a local search approach to find stable marriages. Our experiments show that the number of steps grows as little as $O(nlog(n))$. We also show empirically that the proposed algorithm samples very well the set of all stable marriages of a given SM, thus providing a fair and efficient approach to generate stable marriages.

1 Stable marriage problems

A stable marriage (SM) problem [2] consists of matching members of two different sets, usually called men and women. Each person strictly ranks all members of the opposite sex. The goal is to match the men with the women so that there are no two people of opposite sex who would both rather marry each other than their current partners. If there are no such pairs (called blocking pairs) the marriage is *"stable"*. For a given SM instance, let M and M' two stable marriages. M dominates M' iff every man has a partner in M which is at least as good as the one he has in M'. Under the partial order given by this dominance relation, the set of stable marriages forms a distributive lattice [5]. Gale and Shapley give a polynomial time algorithm (GS) ($O(n^2)$) to find the stable marriage at the top (or bottom) of this lattice [1]. The top of such lattice is the male optimal stable marriage, M_m, that is optimal from the men's point of view. This means that there are no other stable marriages in which each man is married with the same woman or with a woman he prefers to the one in M_m. The GS algorithm can also be used to find the female optimal stable marriage (that is the bottom of the stable marriage lattice) just replacing men with women. A common concern with the standard Gale-Shapley algorithm is that it unfairly favors one sex at the expense of the other. This gives rise to the problem of finding "fairer" stable marriages. Previous work on finding fair marriages has focused on algorithms for optimizing an objective function that captures the happiness of both genders [3]. A different approach is to ask if one can define a fair procedure to generate stable marriages. In this respect, it is natural to investigate non-deterministic procedures (such as local search) that can generate a random stable marriage from the lattice with a distribution which is as uniform as possible.

2 Local search on SMs

Local search [4] is one of the fundamental paradigms for solving computationally hard combinatorial problems.

Given a problem instance, the basic idea underlying local search is to start from an initial search position in the space of all solutions (typically a randomly or heuristically generated candidate solution), and to improve iteratively this candidate solution by means of typically minor modifications. At each *search step*, we move to a position selected from a *local neighborhood*, chosen via a heuristic evaluation function. The evaluation function typically maps the current candidate solution to a number such that the global minima correspond to solutions of the given problem instance. The algorithm moves to the neighbor with the smallest value of the evaluation function. This process is iterated until a a solution is found or a predetermined number of steps is reached. To ensure that the search process does not stagnate in unsatisfactory candidate solutions, most local search methods use randomization: at every step, with a certain probability a random move is performed rather than the usual move to the best neighbor.

Given an SM problem P, our local search algorithm starts from a randomly generated marriage M. Then, at each search step, we compute the set BP of blocking pairs in M and the neighborhood, which is the set of all marriages obtained by removing one of the blocking pairs. Consider a blocking pair $bp = (m, w)$ in M, $m' = M(w)$, and $w' = M(m)$; where $M(x)$ is the partner of x in M. Then, removing bp from M (written $M \backslash bp$) means obtaining a marriage M' in which m is married with w and m' is married with w', leaving the other pairs unchanged. Among the neighbors, we move to one with the least number of blocking pairs. To avoid stagnation in a local minimum of the evaluation function, at each search step we perform a random walk with probability p which removes a randomly chosen blocking pair in BP from the cur-

[1] Research partially supported by the Italian MIUR PRIN project 20089M932N: "Innovative and multi-disciplinary approaches for constraint and preference reasoning".

[2] University of Padova, Italy, email:{mgelain,mpini,frossi,kvenable}@math.unipd.it

[3] NICTA and UNSW, Australia, email:Toby.Walsh@nicta.com.au

rent marriage M. In this way we move to a randomly selected marriage in the neighborhood. The algorithm terminates if a stable marriage is found or when a maximal number of search steps is reached. The number of blocking pairs may be very large. Also, the removal of some of them would surely lead to new marriages that will not be chosen by the move. This is the case for the so-called *dominated* blocking pairs. Let (m, w) and (m, w') two blocking pairs. Then (m, w) dominates (from the men's point of view) (m, w') iff m prefers w to w'. We therefore consider only undominated blocking pairs.

Since dominance between blocking pairs is defined from one gender's point of view, to ensure gender neutrality, at the beginning of our algorithm we randomly choose a gender and, at each search step we change the role of the two genders.

3 Experimental results

We tested our algorithms on randomly generated sets of SM instances. We generated stable marriage problems of size n by assigning to each man and to each woman a preference list uniformly chosen from the $n!$ possible total orders of n persons. We studied how fast we converge to a stable marriage, by measuring the ratio between the number of blocking pairs and the size of the problem during the execution. Let us denote by $\langle b \rangle$ the average number of blocking pairs of the marriage found for SMs of size n after t steps. Then the experimental results (Figure 1) show a very good fit with the function $\langle b \rangle = an^2 2^{\frac{-bt}{n}}$, where a and b are constants computed empirically ($a \approx 0.25$ and $b \approx 5.7$). Moreover, the

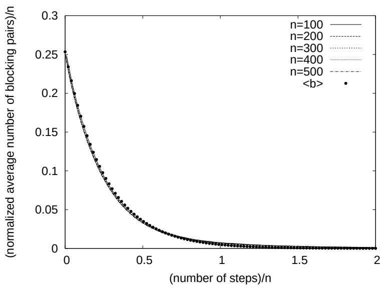

Figure 1. Convergence speed.

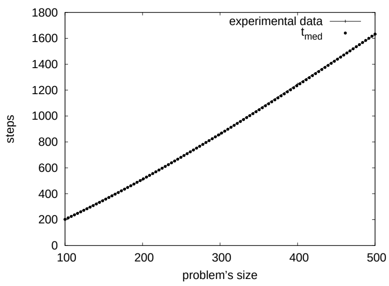

Figure 2. Number of steps needed to find a stable marriage.

average number of blocking pairs, normalized by dividing it by n, decreases during the search process in a way that is independent from the size of the problem. We can use function $\langle b \rangle$ to conjecture the runtime behavior of our local search method. Consider the median number of steps, t_{med}, taken by the algorithm. Assume this occurs when half the problems have one blocking pair left and the other half have zero blocking pairs. Thus, $\langle b \rangle = \frac{1}{2}$. Substituting this value in the equation for $\langle b \rangle$, and grouping constant terms, we get $t_{med} = cn(d + 2log_2(n))$. Hence, we can conclude that t_{med} grows as $O(nlog(n))$. Figure 2 shows how the experimental data fits function t_{med}.

We also evaluated the sampling capability of our algorithm over the lattice of stable marriages of a given SM. To do this, we randomly generated 100 SM instances for each size between 10 and 100, with step 10.

We first measured the distance of the found stable marriages (on average) from the male-optimal marriage. Given an SM P, consider a stable marriage M for P. The distance of M from the top of the lattice, M_m, is the number of arcs from M to M_m in the Hasse diagram of the stable marriage lattice for P. For each SM instance, we compute the average normalized distance from the male-optimal marriage considering 500 runs. Then, we compute the average D_m of these distances over all the 100 problems with the same size. If $D_m = 0$, it means that all the stable marriages returned coincides with the man-optimal marriage. On the other extreme, if $D_m = 1$, it means that all stable marriages returned coincide with the female-optimal one. Figure 3 shows that, for the stable marriages returned, the average distance D_m from the male-optimal stable marriage is around 0.5.

We also consider the entropy, that is, the uncertainty associated with the outcomes of the algorithm. Let $f(M_i)$ the frequency that we find a marriage M_i for an SM instance P. The entropy $E(P)$ for each SM instance P (i.e., for each lattice) of size m is then: $E(P) = -\sum_{i=1 \in \{1..|S|\}} f(M_i) log_2(f(M_i))$, where S is the set of all possible stable marriages of P. In an ideal case, when each node in the stable marriage lattice has a uniform probability of $1/m!$ to be reached, the entropy is $log_2(|S|)$. On the other hand, the worst case is when the same stable marriage is always returned, and the entropy is thus 0. Since we have 100 different problems for each size, we compute the average of the normalized entropies for each class of problems with the same size: $E_n = \frac{1}{100} \sum_{i=1}^{100} E(P_i)/log_2(|S_i|)$, where S_i is the set of stable marriages of P_i. Figure 3 shows that we are not far from the ideal behavior: the normalized entropy E_n starts from a value of 0.85 at size 10, decreasing to above 0.6 as the problem's size grows. Considering E_n and D_m together, it appears that the algorithm samples the stable marriage lattice very well.

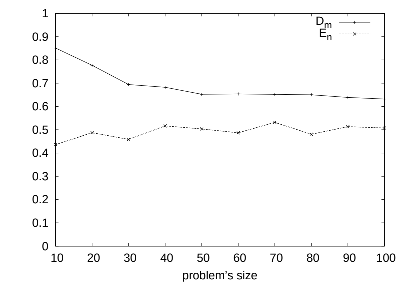

Figure 3. Sampling the SM lattice.

REFERENCES

[1] D. Gale and L. S. Shapley, 'College admissions and the stability of marriage', *The Amer. Math. Monthly*, **69**(1), 9–15, (1962).

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

[3] Dan Gusfield, 'Three fast algorithms for four problems in stable marriage', *SIAM J. Comput.*, **16**(1), 111–128, (1987).

[4] H. H. Hoos and T. Stutzle, *Stochastic Local Search: Foundations and Applications*, Elsevier - Morgan Kaufmann, 2004.

[5] D.E. Knuth, *Marriages Stables*, Les Presses du l'Univ. de Montral, 1976.