On Heuristics for Two-Sided Matching: Revisiting the Stable Marriage Problem as a Multiobjective Problem*

Steven O. Kimbrough
University of Pennsylvania
Philadelphia, PA USA
kimbrough@wharton.upenn.edu

Ann Kuo
University of Pennsylvania
Philadelphia, PA USA
aja@wharton.upenn.edu

ABSTRACT

The stable marriage problem is prototypical of two-sided matching problems, widely encountered in practice, in which agents having preferences, interests and capacities for action of their own are paired up or matched. Standardly, variants of the well-known Gale-Shapley deferred acceptance algorithm (GS/DAA) are used to find stable matches. Using evolutionary computation and an agent-based model heuristics, this paper investigates the stable marriage problem as a multiobjective problem, looking at social welfare and equity or fairness, in addition to stability as important aspects of any proposed match. The paper finds that these heuristics are reliably able to discover matches that are Pareto superior to those found by the GS/DAA procedure. Ramifications of this finding are briefly explored, including the question of whether stability in a matching is often strictly required.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

centralized markets, agent-based models, evolutionary computation, two-sided matching, stable marriage problem, deferred acceptance algorithm

1. INTRODUCTION

In the usual case, markets are *distributed*, with buyers and sellers mostly on their own in finding each other and in negotiating terms of trade. Distributed markets may fail in one way or another, however. A common response is to create a *centralized* market organized by a third party whose responsibility is to set the conditions of trade, for example the price, based on the bids and asks from the buyers and sellers. Many electricity markets are organized in this way. Deregulated electricity producers, for example, offer supply schedules to a third party, often called the *independent systems operator* or ISO, who aggregates the supply schedules, observes the market demand, and sets the price of electricity (for a given period of time).[1]

Quite a number of labor markets are similarly centralized, most famously, markets in which physicians are matched to hospitals for internships [18]. Roughly speaking, the individual doctors submit their rankings of hospitals, the hospitals submit their rankings of doctors, and a third party organization undertakes to match doctors with hospitals. This is an example of a *two-sided matching* problem, which problems are the subject of this paper.

In a two-sided matching problem, we are given two sets ('sides') of individuals and asked to form pairs consisting of one member from each set. Standard examples dealt with widely in practice include pairing men with women, workers with employers, students with schools and so on.

A presumption in matching problems (as distinguished from assignment problems, which are treated in operations research and employ non-strategic decision making) is that both sides consist of agents who have interests of their own and capacities to act on them. Consequently, matches are ordinarily evaluated in terms of *stability.* Matching problems are inherently strategic, or game-theoretic, and stability is the accepted equilibrium concept. A match is said to be stable if there is no pair of matched couples in it containing individuals who would prefer to be matched to each other but are not. (See below for details.) The thought is that if the couple here is unstable with regard to the couple next door, divorce and remarriage will (or at least may) ensue. Requiring matches to be stable in the first place will prevent breakup and reformation among pairs and its attendant costs.

The point of departure for this paper is the observation that two-sided matches can be evaluated—and for many applications should be evaluated—according to several objectives, particularly stability, equity, and social welfare. For present purposes, by stability we mean the count of unstable pairs of matched couples in a match. This should be minimized and at 0 the match is stable.[2] By equity we mean the

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.

Copyright 2010 ACM 978-1-4503-0072-8/10/07 ...$10.00.

[1] There's more to it, but the basic point is correct.

[2] If the count of unstable pairs is 1, there is no guarantee that

sum of the absolute differences in the preference scores of each matched pair. We will be scoring preference on a ranking 1 to n scale (1 = most preferred, n = least preferred), so this too should be minimized. Finally, by social welfare we mean the sum of the agent scores in the match. Again, since scoring is from low to high, this quantity should also be minimized. (To illustrate, if agents i and j are paired in a match, and i's preference score for j is 5 and j's preference score for i is 3, then the matched pair's contribution to social welfare is the sum, 8, and their contribution to equity is the absolute value of the difference of the scores, $|5 - 3| = 2$.)

Given that we would consider designing or even centralizing a matching market (as is widely done in practice), the question arises of how best to provide the market operators and users with match options that map the Pareto frontier (as well as possible) in these three objectives. In what follows we explore two rather different algorithmic approaches to this: an agent-based model that simulates a distributed market and an evolutionary computation approach. We compare them with each other and with what can be produced by the standard approach, the deferred acceptance algorithm of Gale and Shapley [8].

2. RELATED WORK

Two-sided matching problems, and the stable marriage problem in particular, have received exploratory investigation as dual objective problems in [2], [7] and in [20]. Aldershof and Carducci [1] report optimistically on application of a genetic algorithm to two-sided matching problems, but the problems they examine are smaller than the 20×20 problems discussed here, so they do no address scaling issues. In [17] a genetic algorithm is used to find gender-unbiased solutions for the stable marriage problem. [5] has useful and suggestive findings pertaining to multiobjective evolutionary algorithms generally, as do [6] and [4]. [21] presents an innovative use of ant colony optimization for the stable marriage problem. We believe that population-based metaheuristics generally, whether or not they involve evolutionary computation, are very promising for two-sided matching problems, seen as multiobjective problems.

3. ESSENTIAL BACKGROUND

Again, now with more detail, in a two-sided matching problem we are given two sets (sides) of agents, X and Y, and are asked to find a match, μ, consisting of a decision (in or out?) for each pair $(x, y), x \in X, y \in Y$. It is helpful to view a match as represented by a matrix, $\mathbf{M}$, of size $|X| \times |Y|$, based upon arbitrary orderings of X and Y. The element $m_{i,j}$ of $\mathbf{M}$ equals 1, if $x_i \in X$ is matched with $y_j \in Y$; otherwise the element is 0. Thus the element $m_{i,j}$ of $\mathbf{M}$ represents the *pair* (x_i, y_j). Matchings pair up agents from X and Y.

Particular matching problems come with particular requirements on μ (or $\mathbf{M}$) as well as X and Y. For example in the *simple marriage matching* problem (the focus of this paper because it is prototypical for two-sided matching problems), we require that $|X| = |Y| = n$; the number of men equals the number of women. We further require of any (valid) match that each man (or member of X (Y)) be paired (or matched) with exactly one woman (or member of Y (X)), and vice versa. In terms of $\mathbf{M}$, this means that there is one 1 in each row and one 1 in each column. $\mathbf{M}$ is thus a permutation matrix and the number of possible valid matches is $n!$. In *admissions* matching problems, which are used to model, for example, interns applying to hospitals and students applying to schools, one side of the problem, say X, is much larger than the other. There are more doctors than hospitals, more students than schools. Unlike conventional marriage problems, however, one side will have *quotas* larger than 1. Each doctor and each student will have a quota of 1, but each hospital and each school may have a much larger quota and admit many doctors or students. Thus, in a valid match for an admissions problem, each agent on one side (X or Y) is paired to one or more agents in the counterpart side, not to exceed the agent's quota. With students as X, and schools as Y, $\mathbf{M}$ will have one 1 each row, and each column will have a number of 1s not to exceed the quota of the corresponding school.

Many other variations are possible and are met in practice for two-sided matching problems. As Roth notes [18], two-sided matching models are often natural for representing markets, in which agents need to be paired up. Men and women want to find partners, workers want to find employment and employers want to find workers, and so on. Moreover, many of these markets, decentralized or free markets, experience failure and unsatisfactory performance in practice. They experience unraveling, e.g., offers to students are made earlier and earlier; congestion, e.g., offerers have insufficient time to make new offers after candidates have rejected previous offers; and participants engage in disruptive strategic behavior, so that behaving straightforwardly with regard to one's preferences becomes risky, e.g., in scheduling offers and responding to them [3, 18, 19]. In consequence, there is a large and growing number of applications of two-sided matching in which decentralized markets have been replaced by centralized ones, in which a coordinating agency undertakes periodic matching between two sides of a specific market. ([18] lists over 40 labor markets, mostly in medical fields; schools in New York and Boston are using centralized markets to match students to schools; see also [3].)

How do and how should centralized market agencies produce matches? In practice, some form of, variation on, the deferred acceptance algorithm of Gale and Shapley [8] is used to find a *stable* match, which is then used. A match is *unstable* if there is a pair of matched pairs—(x_i, y_j) and (x_k, y_l)—such that x_i prefers to be matched with y_l over being matched with y_j and y_l prefers to be matched with x_i to being matched with x_k. Stable matches are the ones that are not unstable.

The deferred acceptance algorithm (DAA) was first published in a paper by Gale and Shapley [8], although the procedure was discovered and used independently before. Because the algorithm is easily understood and readily available in published works we will, in the interests of space, not repeat it here, except to present it in pseudocode, Figure 1. Instead, we will describe its key properties as we see them for present purposes. First, as proved by Gale and Shapley, under the special assumptions they made (e.g., preference ranking by agents, etc., which for the sake of discussion we retain), the stable marriage problem and the admissions problem (see above) have stable matches and the DAA will find one and will find one quickly ($O(n^2)$). Second, the DAA

if the two pairs rematch by exchanging partners the resulting match would be stable. In fact, the resulting match could have a higher count of unstable pairs [14].

1. Assume: $|X| = |Y| = n$
2. Each $x \in X$ ranks each $y \in Y$, and each $y \in Y$ ranks each $x \in X$.
3. *Matched* $\longleftarrow \emptyset$, *Unmatched* $\longleftarrow \emptyset$.
4. For each y, *string.y* $\longleftarrow$ []
5. Each $x \in X$ proposes to its most-preferred y, appending x to *string.y*.
6. Each y with *length*(*string.y*) > 1 (i.e., with more than one proposal), retains in the string its most preferred member of the string, and removes the rest, adding them to *Unmatched*.
7. Do while *Unmatched* $\neq \emptyset$:
   (a) Each $x \in$ *Unmatched* proposes to its most-preferred y, among the Ys that have not already rejected x, appending x to *string.y*.
   (b) *Unmatched* $\longleftarrow \emptyset$.
   (c) Each y with *length*(*string.y*) > 1 (i.e., with more than one proposal), retains in the string its own most preferred member of the string, and removes the rest, adding them to *Unmatched*.
8. For each x remaining on some *string.y*, (x, y) is added to *Matched*.
9. Stop. Each x is matched to a distinct y, who has x as the sole member of its string. This is recorded in *Matched*.

Figure 1: Pseudocode for the deferred acceptance algorithm (DAA) for the simple marriage matching problem, Xs proposing to Ys, after [8].

is asymmetric. One side proposes, the other disposes. Focusing now on the marriage problem, if the men propose, they obtain a stable match that is male optimal in the sense that no man in this match strictly prefers (does better in) any other stable match. Conversely, the match is female pessimal in the sense that no woman is worse off in any other stable match. And vice versa if the women propose [8, 10, 14].

Although here we consider it only in the context of the marriage problem, this asymmetry is a general characteristic of the DAA in its various forms. It occasions the important question of whether better matches exist and can be found. To this end, we will want to look at stable matches that may be preferable to the matches found by the DAA. As announced above, we want to examine both social welfare and equity. Further, it is natural to raise the question of multiple objectives in the context of 'nearly-stable' matches, by which we mean matches with relatively few unstable pairs. Decision makers, including agents participating in a centralized market, may quite reasonably want to exchange some stability for improvements in, say, social welfare or equity. We note that in many cases it may be practically difficult, or made practically difficult by the operator of the centralized market, for members of matched couples to undertake swaps, regardless of their preferences.

These issues could be neatly resolved by, for any given problem, finding all of the stable solutions and comparing them with respect to equity, social welfare, and whatever other measures of performance are relevant. Predictably, however, this is an intractable problem. Irving and Leather [12] have shown that the maximum number of stable matches for the simple marriage matching problem grows exponentially in n (see also [10, 11]). Further, they provide a lower bound on the maximum by problem size. Remarkably, the lower bound is 104,310,534,400 for a problem as small as $n = 32$ [12]. Further, they establish that the problem of determining the number stable matches is #P-complete. These are, of course, extreme-case results, but very little is known about average cases. So we are left to rely upon heuristics, and we shall for the remainder of this paper.

4. MATCHING WITH AGENTS

We developed an agent-based model, called SimpleMarriageMatching (freely available for research and teaching purposes from the authors), that simulates a distributed market for the simple marriage matching problem. At initialization, each agent is given a preference ranking of the agents in the counterpart (opposite "gender"), and the n "men" (members of Y) are randomly paired with the n "women" (members of X) to create a valid match for the simple marriage matching problem. The program then maintains a valid match throughout its execution. (The results we report here assume the "collective" swapping regime, which we now describe.) In the main loop of the program, agents are put into a random order, then each agent in turn examines the agents of the counterpart set (the women, if the agent is a man, the men if the agent is a woman). If the agent finds a matched pair with the property that the agent prefers the counterpart member of the pair (the woman, if the agent is a man; the man if the agent is a woman) *and* the agent's counterpart member prefers the agent to its own current match, then matching of the two pairs may be swapped. Starting with (x_i, y_j) and (x_k, y_l) we get (x_i, y_l) and (x_k, y_j). The agent identifies all potential swaps in the counterpart set, that is matched couples such that the agent prefers its counterpart to its present match *and* the counterpart prefers the agent to its current match *and* the preferences are positive net of the transaction cost. The agent picks the most attractive of these from its point of view, the swap is made, and the agent's turn is over. (If there are no potential swaps, this also terminates the agent's turn.) Note: the swapping does *not* depend on the preference of the agent's mate for the mate of the counterpart pair, or vice versa. Further, whatever swapping that is done is on the basis of preference net of transaction cost, which is measured in rank units. Thus, for example, if the transaction cost is 2, and x_i is the focal agent, then a swap only occurs if x_i's preference for y_l is more than 2 ranks superior to x_i's preference for its current mate, y_j; and similarly for y_l's preferences.

Table 1 summarizes results from an experiment in which n=20 and the transaction cost was set to 0. There were 100 runs in which the agents were initialized with a random preference regime. For each of these runs, there were 100 replications, with the preference regimes constant, but the initial matches randomly varying and the order of swap consideration randomly changing. Table 2 summarizes results from a similar experiment, with transaction cost set to 1. In both experiments (Tables 1 and 2), every replication of every

Table 1: 20×20, Transaction Cost = 0

	1st Qu	Median	3rd Qu
Init. # unstable pairs	73	83	93
Final # unstable pairs	0	0	0
InitialSocialWelfareSum	395	420	444
Final SocialWelfareSum	161	171	183
Initial Equity	119	133	148
Final Equity	61	69	80
SwapCount	42	55	79
InitialSumXScores	192	210	228
Final SumXScores	74	85	97
InitialSumYScores	192	210	227
Final SumYScores	75	85	97
Number of Runs	100		
Number of Replications	100		
TransactionCost	0		
SwappingRegime	Collective		

run produced a stable match. Neglecting transaction costs (NTC) the median number of unstable pairs in a replication is 3 (Table 2), rather small, and the statistics for social welfare, equity, and agent scores are very similar to those in Table 1, with no transaction cost. A small amount of instability has not lead to significant changes in social welfare, agent welfare, or equity. From a random start, these myoptic and greedy agents manage to improve their positions, and that of their entire society, quite nicely. (We remind the reader that the search space here is huge: $20! \approx 2.4 \times 10^{18}$.)

Table 2: 20×20, Transaction Cost = 1

	1st Qu	Median	3rd Qu
Init. # unstable pairs	59	69	78
Final # unstable pairs	0	0	0
Init. # unstbl pairs NTC	73	83	93
Fin. # unstbl pairs NTC	2	3	4
InitialSocialWelfareSum	395	420	445
Final SocialWelfareSum	165	175	186
Initial Equity	118	132	147
Final Equity	61	70	78
SwapCount	26	31	38
InitialSumXScores	192	210	228
Final SumXScores	77	88	100
InitialSumYScores	192	210	227
Final SumYScores	76	86	98
Number of Runs	100		
Number of Replications	100		
TransactionCost	1		
SwappingRegime	Collective		

At $n = 40$, Tables 3 and 4 tell a story similar to that for $n = 20$. Not every replication, however, leads to a stable match when transaction cost is 0. What stops the search in a replication is the number of swaps (changing partners) reaching `MaximumSwaps` = 100,000. With a transaction cost of 2, Table 4, the swap count is again low and the measures of performance again similar to those with transaction cost 0. Notice that again, the random starts yield poor performance measures, but the agents individually and collectively achieve much improvement.

Table 3: 40×40, Transaction Cost = 0 (†Mean=3, Max=124)

	1st Qu	Median	3rd Qu
Init. # unstable pairs	314	340	367
Final # unstable pairs†	0	0	0
InitialSocialWelfareSum	1570	1638	1709
Final SocialWelfareSum	476	499	529
Initial Equity	492	532	572
Final Equity	200	224	245
SwapCount	1610	5731	22389
InitialSumXScores	769	819	869
Final SumXScores	217	246	280
InitialSumYScores	772	820	869
Final SumYScores	224	253	293
Number of Runs	100		
Number of Replications	100		
TransactionCost	0		
SwappingRegime	Collective		

Table 4: 40×40, Transaction Cost = 2

	1st Qu	Median	3rd Qu
Init. # unstable pairs	258	284	311
Final # unstable pairs	0	0	0
Init. # unstable pairs NTC	314	341	369
Final # unstable pairs NTC	7	9	11
InitialSocialWelfareSum	1569	1641	1711
Final SocialWelfareSum	483	505	529
Initial Equity	493	533	574
Final Equity	198	218	239
SwapCount	89	105	127
InitialSumXScores	771	820	869
Final SumXScores	226	251	279
InitialSumYScores	772	821	870
Final SumYScores	225	251	278
Number of Runs	100		
Number of Replications	100		
TransactionCost	2		
SwappingRegime	Collective		

Finally, when $n = 100$ and transaction cost is 0, the runs do not terminate in acceptable time, so no results are available. The issue of scaling is one we will return to below. Table 5 shows results for when transaction cost is 5. Now each replication results in a stable match, although neglecting transaction costs no replication found a stable match and the median number of unstable pairs has risen to about $\frac{n}{3}$. Even so, the agents make remarkable improvements in the various measures of performance, from their random starts.

5. EVOLUTIONARY MATCHING

We also developed a genetic algorithm based system, called Stable Matching GA, that samples the solution space of the simple marriage matching problem. Solutions are represented as permutations of $1 \ldots n$. After experimentation we settled on a mutation rate of 0.4 per solution, a two-point crossover rate of 0.6, a population size of 50, runs of 2,000 generations, and 100 repetitions or trials per problem. Mutation was effected by randomly choosing alleles at two loci

Table 5: 100×100, Transaction Cost = 5

	1st Qu	Median	3rd Qu
Init. # unstable pairs	1709	1808	1908
Fin. # unstable pairs	0	0	0
Init. # unstbl pairs NTC	2061	2165	2270
Fin. # unstbl pairs NTC	28	32	37
InitialSocialWelfareSum	9830	10104	10376
Final SocialWelfareSum	1984	2048	2113
Initial Equity	3173	3329	3492
Final Equity	878	934	994
SwapCount	655	1049	1901
InitialSumXScores	4859	5050	5242
Final SumXScores	941	1020	1106
InitialSumYScores	4858	5055	5249
Final SumYScores	939	1018	1103
Number of Runs	100		
Number of Replications	100		
TransactionCost	5		
SwappingRegime	Collective		

and swapping them, thereby maintaining a valid solution, as a permutation. We used order crossover, OX, as our two point crossover operator ([16, page 217], [9, page 174]).

The fitness of a given solution is its number of unstable pairs. The lower the fitness value, the fitter a solution is and the higher chance it has for participating in the reproduction process. To a given population, one generation of solutions goes through three processes: fitness evaluation, selection (we used tournament-2), and reproduction (with mutation and crossover as described above). The evolution stops after a specified number of generations are reached.

We ran the GA seeking to minimize the number of unstable pairs in the match solutions. We report here with regard to the *stable* matches found by the GA for these 25 random 20×20 problems, and we compare the GA's solutions with the DAA solutions for these problems. The results may be summarized as in Table 6.

Here, in Table 6, the column labeled Case is for the 25 random 20×20 random problem instances; "D(DorE) 1G.S. Soln" means the count of strictly dominant (better than the DAA (Gale-Shapley) solutions on the two dimensions of fairness and social welfare) or (in parentheses) the count of weakly dominant (at least as good as the DAA solution)—for at least 1 of the solutions (and there may be only one). "D(DorE) 2G.S.Soln" means the count of strictly dominant (better than the DAA on the two dimensions of fairness and social welfare) or (in parentheses) the count of weakly dominant (at least as good as the DAA solutions)—for both DAA solutions (although there may be only one). So the form X(Y)/Z means X solutions strictly dominating, Y solutions weakly dominating, and Z solutions found by the GA overall. (All of these solutions are stable solutions.) Finally, "Found G.S.Soln/#GS" with the form X/Y means of the Y DAA solutions (from Gale-Shapley), X of them were found by the GA.

To summarize, Table 6 shows:

- In 18 of 24 cases, the GA found at least one stable solution that strictly dominates both of the GS/DAA solutions. (In case 20, there is only 1 GS/DAA solution.)

Case Case Case	D(DorE) 1G.S.Soln / TC	D(DorE) 2G.S.Soln / TC	Found G.S.Soln / #GS
1	9(10) / 10	4(5) / 10	2/2
2	6(8) / 8	5(5) / 8	2/2
3	4 (5)/ 6	0(1) / 6	2/2
4	7(8) / 8	3(4) / 8	2/2
5	3(5) / 6	3(3) / 6	2/2
6	4(5) / 6	1(2) / 6	2/2
7	10(11) / 11	9(10) / 11	2/2
8	5(6) / 6	1(2) / 6	2/2
9	4(5) / 6	0(2) / 6	2/2
10	6(7) / 7	2(4) / 7	2/2
11	9(11) / 12	2(4) / 12	2/2
12	4(5) / 6	1(3) / 6	2/2
13	8(9) / 9	4(5) / 9	2/2
14	1(3) / 4	0(1) / 4	2/2
15	4(5) / 6	1(2) / 6	2/2
16	4(5) / 5	1(2) / 5	2/2
17	5 (7)/ 7	3(3) / 7	2/2
18	10(11) / 11	2(4) / 11	2/ 2
19	4(6) / 6	4(4) / 6	2/2
20	0(1) / 1	0(1) / 1	1/1
21	8(10) / 10	7(8) / 10	2/2
22	4(5) / 5	2(3) / 5	2/2
23	1(3) / 3	0(1) / 3	2/2
24	2(3) / 6	0(1) / 6	2/2
25	5(6) / 6	1(2) / 6	2/2

Table 6: 20x20x25 TPXOver Strongly Dominating Solution Counts (m04x06p50t100g2000)

- Excluding case 20 with only 1 GS/DAA solution, in 24 of 24 cases the GA found one or more stable solution that strictly dominates one of the two GS/DSS solutions.

Our study thus shows promising results on using a GA to search for favorable matching schemes. In terms of searching for stable match schemes, the GA found either strictly better or at least equally good solutions compared to the Deferred Acceptance algorithm results with regard to fairness and social welfare. When stability is not the only objective, GAs provide many other dominant solutions. Depending on how different objectives are weighted, sometimes a minimum number of unstable pairs may be a cheap price to pay for the improvement on other objectives (for example, higher individual satisfaction or greater assignment fairness). The key point of our finding is about the GA's capability for providing decision makers with information about the otherwise unseen alternatives.

6. AGENT MODEL COMPARISON

For comparison purposes we report on runs from the agent model on the same 25 20×20 randomly generated problems on which we ran the GA. It is a bit tricky to make the computational efforts of the two programs commensurate. Standardly in GA work comparisons are made on the basis of the number of fitness evaluations, but in the agent model we do not have a fitness function to evaluate. In the two models, however, there is a common basis for comparison:

the number of calculations of whether two matched pairs are mutually unstable. In the GA, this is how fitness is evaluated. The fitness of a solution is the number of (mutually) unstable matched pairs it has. To calculate this number requires $\frac{n(n-1)}{2}$ comparisons of two matched pairs. Thus, the number of comparisons undertaken for a single case (there are 25 in the present context) is $\frac{n(n-1)}{2} \times K \times G \times R$, where K is the population size, G is the number of generations per replication, and R is the number of replications per case.

For the agent model, in each round (tick) during a replication each of the $2n$ agents undertakes a comparison with the n agents of opposite gender, so we have $2n^2$ comparisons per round/tick. Since the number of rounds/ticks varies by replication, we estimated the number of comparisons empirically at about 3500 per replication in the 20×20 cases. In consequence we set the number of replications per case to be 1000, putting us roughly on parity with the GA effort.

Case	D1GS/TC	D2GS/TC
1	8 / 8	3 / 8
2	6 / 8	5 / 8
3	4 / 6	0 / 6
4	7 / 7	3 / 7
5	3 / 6	3 / 6
6	4 / 6	1 / 6
7	10 / 10	9 / 10
8	8 / 8	3 / 8
9	4 / 6	0 / 6
10	6 / 7	2 / 7
11	9 / 12	2 / 12
12	4 / 6	1 / 6
13	8 / 8	4 / 8
14	1 / 4	0 / 4
15	4 / 6	1 / 6
16	4 / 5	1 / 5
17	5 / 6	3 / 6
18	10 / 10	2 / 10
19	4 / 6	4 / 6
20	0 / 1	0 / 1
21	8 / 9	7 / 9
22	4 / 4	2 / 4
23	1 / 3	0 / 3
24	2 / 6	0 / 6
25	5 / 6	1 / 6

Table 7: Agent 20x20x25 SD

Table 7 shows the results. The column labeled "D1GS/TC" has the form X/Y, where X is the number of stable solutions found that strictly dominate one of the GS/DAA solutions and Y is the total number of stable solutions found. Similarly, in the column labeled "D2GS/TC" with form X/Y, X is the number of stable solutions found that strictly dominate both GS/DAA solutions and Y is the number of stable solutions found. Comparing Tables 7 and 6 we find that the agent model and the GA perform about equally as measured by the numbers of dominating solutions found.

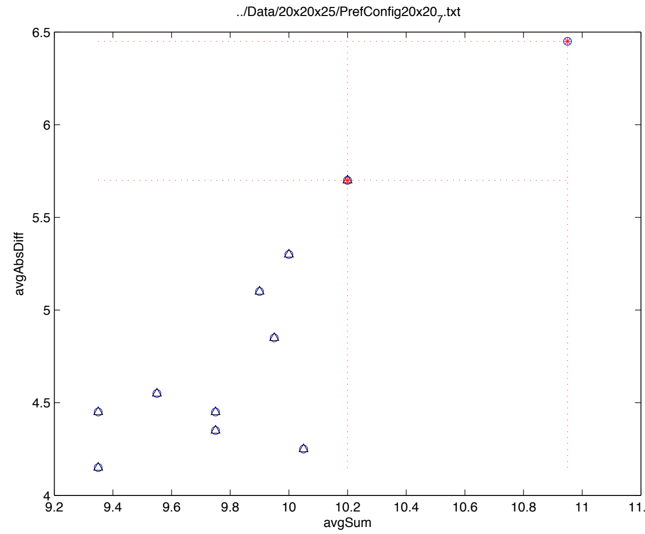

Figure 2: Plot of alternate stable solutions found for test case 7. GS/DAA in red $*$, GA in $\circ$, agent model in $\triangle$.

7. NEARLY-STABLE MATCHES

A stable match is one in which there are no pairs of matched couples that are mutually unstable. A nearly-stable match is one in which there are few pairs of matched couples that a mutually unstable. In a "one-away" match there is only one pair of matched couples that is mutually unstable. We emphasize that in such a one-away match there is no guarantee that swapping the unstable pair of couples will produce a stable match. The new couples resulting from the swap may be unstable with many other couples and unraveling may well be possible [14].

Our GA typically was able to find very many one-away solutions and very many of these were Pareto superior to the GS/DAA solutions. See Table 8 and Figure 3 for a graphical presentation.

8. SUMMARY & DISCUSSION

We find and are reporting in compressed form the following:

1. Typically in simple marriage matching problems there are stable matches that are Pareto superior to the deferred acceptance matches, in regard to equity and social welfare (as we have characterized them).

2. With roughly equal computational effort, both the agent-based model and the genetic algorithm we built find similar numbers and quantities of stable solutions for the simple marriage matching problems we examined.

3. There are typically very many nearly-stable matches that are superior to the deferred acceptance (stable) matches on equity and/or social welfare (or both). We found these solutions with the genetic algorithm.

4. Scale is an important issue. As the size of the problem exceeds 50 or so, the agent-based model becomes generally unable to find any stable solution with zero transaction costs with fewer than millions of swaps

Test Case	D1GS/TC	D2GS/TC
1	121 / 128	31 / 128
2	100 / 143	88 / 143
3	31 / 140	1 / 140
4	78 / 97	49 / 97
5	49 / 96	39 / 96
6	28 / 83	7 / 83
7	113 / 124	75 / 124
8	53 / 67	15 / 67
9	61 / 94	9 / 94
10	78 / 110	14 / 110
11	75 / 154	20 / 154
12	10 / 97	7 / 97
13	82 / 106	18 / 106
14	27 / 65	12 / 65
15	73 / 134	7 / 134
16	56 / 87	14 / 87
17	72 / 87	56 / 87
18	52 / 89	11 / 89
19	43 / 84	27 / 84
20	1 / 38	1 / 38
21	124 / 148	109 / 148
22	64 / 82	35 / 82
23	11 / 61	1 / 61
24	20 / 96	8 / 96
25	76 / 107	8 / 107

Table 8: 20x20x25 TPXOver, 0.8, Mutation, 0.3, One-Away Strongly Dominating Solution Counts

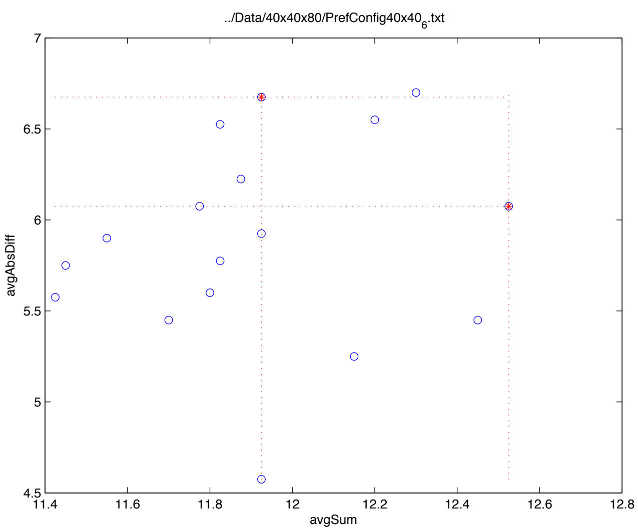

Figure 3: One-away solutions compared to GS/DSS solutions, 40x40, case 6; see Table 9. GS/DAA in red $*$.

Test Case	D1GS/TC	D2GS/TC
1	6 / 8	4 / 8
2	19 / 20	8 / 20
6	13 / 16	6 / 16

Table 9: 40x40 Case1,2,6 GA TPXOver Strongly Dominating Opt Solution Counts

(match alterations). The GA generally continues to perform well, see Table 9, but this needs more extensive testing. We note that these findings apply to the randomly-generated test problems we have used. In these problems, preferences are uncorrelated. Actual applications may be different (think of the marriage market). When preferences are identical there will be only one stable match and generally, the number of stable matches will decline with increasing correlation of preferences. The effects on nearly-stable matches are not known to us.

5. Adding transaction costs to the agent model generally results in better finding of stability, but even costs in the range of $\frac{n}{10}$ are overwhelmed when n is 100 or more. The agents simply do not find stable solutions even in this attenuated sense. The agents do, however, generally improve overall social welfare and equity scores as a side-effect of their swapping.

We see both a practical, applied upshot of these findings and a theoretical one. On the applied side:

1. The case is now quite strong for insisting that two-sided matching problems be viewed as multiobjective, and that policy should look well beyond the GS/DAA-style solutions. Alternate solutions that are Pareto superior do (often) exist and may be found by heuristic methods such as those on display here.

2. The necessity of stability, or equilibrium in a matching, can be questioned, at least in many applications. When the number of unstable pairs is small but non-zero, there will in many cases be no realistic means for the pairs to find each other and unravel the matching. We agree with Kreps's more general comment that

   > Unless a given game has a self-evident way to play, self-evident to the participants, the notion of a Nash equilibrium has no particular claim upon our attention. [15, page 31]

   Matching applications should reconsider their requirements for (exact, full) stability, especially when, as we have seen in an example, there may be very many nearly-stable matches with superior equity and social welfare properties.

3. Our agent-based simulation of a distributed market raises the possibility that centralized markets might become superior to distributed markets by the tactic of simulating them under various conditions, realizable or not, and allocating results based on the simulations. This is an intriguing idea for future consideration.

On the theoretical side, there are two main points arising. First, the agent model is an epistemically very generous model of how real agents cope. That even this model will fail to achieve equilibrium under non-drastic scaling up raises the question concretely of whether real markets can find equilibrium.

Second, looking forward we note that matching problems in general and the simple marriage matching problem in particular may have application broader than heretofore conceived. Consider a model of a market with n buyers of a good such as a house and n sellers of the good. Assume that for each possible pair there is a negotiated price at which they would transact the sale of the house in question. This provides a ranking of the buyers for each seller, since sellers we presume only care about price. On the buyer side, however, price is only one factor among many in determining the value of the property. So buyers have preference rankings on the houses (sellers) that are influenced by, but not determined by, the negotiated prices. At equilibrium, all the houses are sold, and a stable matching is achieved. What we have learned from the agent and evolutionary models described here is that even with the rather heroic epistemic powers of our agents, they will not be able to attain stable matches in any reasonably large market context. Nor is there much assurance that on any one run of the market good equity or social welfare outcomes will be achieved. The upshot is (1) centralized markets, of the sort in which deferred acceptance algorithms are commonly run, may be more widely desirable and (2) heuristic alternatives to deferred acceptance may well be able to offer practical improvements and complements to deferred acceptance in these centralized markets.

9. REFERENCES

[1] B. Aldershof and O. M. Carducci. Stable marriage and genetic algorithms: A fertile union. *Journal of Heuristics*, 5:29–46, 1999.

[2] R. L. Axtell and S. O. Kimbrough. The high cost of stability in two-sided matching: How much social welfare should be sacrificed in the pursuit of stability? In *Proceedings of the 2008 World Congress on Social Simulation (WCSS-08)*, 2008. `http://mann.clermont.cemagref.fr/wcss/`.

[3] L. Bodin and A. Panken. High tech for a higher authority: The place of graduating rabbis from Hebrew Union College–Jewish Institute of Religion. *Interfaces*, 33(3):1–11, May–June 2003.

[4] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer Science+Business Media LLC, New York, NY, second edition, 2007.

[5] D. Dasgupta, G. Hernandez, D. Garrett, P. K. Vejandla, A. Kaushal, R. Yerneni, and J. Simien. A comparison of multiobjective evolutionary algorithms with informed initialization and kuhn-munkres algorithm for the sailor assignment problem. In *GECCO '08: Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*, pages 2129–2134, New York, NY, USA, 2008. ACM.

[6] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, LTD, Chichester, UK, 2001.

[7] T. Fuku, A. Namatame, and T. Kaizouji. Collective efficiency in two-sided matching. In P. Mathieu, B. Beaufils, and O. Brandouy, editors, *Artificial Economics: Agent-Based Methods in Finance, Game Theory and Their Applications*, pages 115–126. Springer-Verlag, Berlin, Germany, 2006.

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

[9] D. E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

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

[11] M. M. Halldórsson, K. Iwama, S. Miyazaki, and H. Yanagisawa. Improved approximation results for the stable marriage problem. *ACM Trans. Algorithms*, 3(3):30, 2007.

[12] R. W. Irving and P. Leather. The complexity of counting stable marriages. *SIAM Journal on Computing*, 15(3):655–667, August 1986.

[13] D. E. Knuth. *Marriages Stables*. Les Presses de l'Université de Montreal, Montreal, Canada, 1976.

[14] D. E. Knuth. *Stable Marriage and Its Relation to Other Combinatorial Problems: An Introduction to the Mathematical Analysis of Algorithms*, volume 10 of *CRM Proceedings & Lecture Notes, Centre de Recherches Mathématiques Université de Montréal*. American Mathematical Society, Providence, RI, 1997. Originally published as [13].

[15] D. M. Kreps. *Game Theory and Economic Modeling*. Clarendon Press, Oxford, England, 1990.

[16] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin, Germany, third edition, 1996.

[17] M. Nakamura, K. Onaga, S. Kyan, and M. Silva. Genetic algorithm for sex-fair stable marriage problem. In *Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on*, volume 1, pages 509–512 vol.1, Apr-3 May 1995.

[18] A. E. Roth. Deferred acceptance algorithms: History, theory, practice, and open questions. *International Journal of Game Theory*, 36:537–569, March 2008.

[19] A. E. Roth and M. A. Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, Cambridge, United Kingdom, 1990.

[20] N. A. Vien and T. C. Chung. Multiobjective fitness functions for stable marriage problem using genetic algrithm. In *SICE-ICASE, 2006. International Joint Conference*, pages 5500–5503, Oct. 2006.

[21] N. A. Vien, N. H. Viet, H. Kim, S. Lee, and T. Chung. Ant colony based algorithm for stable marriage problem. In K. Elleithy, editor, *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, pages 457–461. Springer, Dordrecht, The Netherlands, 2007. DOI: 10.1007/978-1-4020-6264-3.