On the Survival of Some Unstable Two-Sided Matching Mechanisms*

M. Utku Ünver†
Koç University

Abstract

In the 1960s, three types of matching mechanisms were adopted in regional entry-level British medical labor markets to prevent unraveling of contract dates. One of these categories of matching mechanisms failed to prevent unraveling. Roth (1991) showed the instability of that failing category. One of the surviving categories was unstable as well, and Roth concluded that features of the environments of these mechanisms are responsible for their survival. However, Ünver (2001) demonstrated that the successful yet unstable mechanisms performed better in preventing unraveling than the unsuccessful and unstable category in an artificial-adaptive-agent-based economy. In this paper, we conduct a human subject experiment in addition to short- and long-run artificial agent simulations to understand this puzzle. We find that both the unsuccessful and unstable mechanism and the successful and unstable mechanism perform poorly in preventing unraveling in the experiment and in short-run simulations, while long-run simulations support the previous Ünver finding.

Keywords: Matching, experiment, unraveling, hospital-intern markets, genetic algorithm.

J.E.L. Classification Numbers: C92, C78, C68, C63, I11, J44

*I would like to thank an anonymous referee and an associate editor of the journal, Juergen Bracht, John Duffy, Alex Elbittar, John Kagel, Al Roth, and the seminar participants at the Arne Ryde Symposium at Lund, the SAET Conference at Ischia, Bilkent and Boğaziçi Universities for comments. Also many thanks to PEEL staff Jun Han, Charles Kannair, and Scott Kinross for practical help.

†Address: Koç University, Department of Economics, College of Administrative Sciences and Economics, Rumelifeneri Yolu, Sarıyer, 34450, İstanbul, Turkey. E-mail: `uunver@ku.edu.tr`.
Url: `http://home.ku.edu.tr/~uunver`. Phone: +90 (212) 338 1354.

1 Introduction

Occasionally early contracts or unraveling of contract dates result in ex-post inefficiencies in entry-level labor matching markets. Without changing the rules of the market, it may be impossible to eradicate harmful unraveling. The evolution of the entry-level British labor markets for medical interns and hospitals provides a unique natural experiment in reorganization as an attempt to prevent the unraveling of contract dates (Roth 1991). At the beginning of the twentieth century, the British labor market for medical interns and hospitals was decentralized. This organization led to the unraveling of contract dates. Consultants, the supervising physicians and surgeons who manage hospital positions, started offering positions to the intern candidates almost two years in advance of their graduation from medical school. These created inefficiencies: there were often mismatches as a result of losses in planning flexibility and lack of proper information at the time of the early match. Therefore, regional health service officials decided to employ centralized matching clearinghouses. They employed three major types of matching mechanisms in different regions of the country.

A matching mechanism is a systematic procedure that matches interns with consultants according to their revealed ordinal preferences over each other. A matching is "unstable" if there is a consultant-intern pair each of whom prefers the other to her match or if the matching is individually irrational. A mechanism is "unstable" if it produces at least one unstable matching for some preference profile.[1]

Roth (1991) shows that two of these three categories, priority matching mechanisms and linear programming mechanisms, are "unstable," while the third category is a version of the Gale-Shapley (1962) mechanisms. In the field, it appeared that the "unstable" priority matching mechanisms were "unsuccessful" and were abandoned, since interns and consultants continued signing contracts up to two years in advance of graduation of interns.[2]

Although instability appeared as the explanation of the failure in several studies,[3] the "unstable" linear programming mechanisms nevertheless stood as a "successful" category. This category of mechanisms is still being used in Britain. This anomaly presents a challenge to the "stability" hypothesis as an explanation of the field success of matching mechanisms. The linear programming mechanisms have been used in the field for almost 30 years. The markets in which

[1] Entry-level labor markets have been studied theoretically in the two-sided matching framework (See Roth and Sotomayor 1990 for theoretical background and motivation) through the marriage and assignment models. There is a recent interest for analyzing unraveling in game theoretical settings, for example see Roth and Xing 1994, Sönmez 1999, Kesten 2003 and Niederle, Roth, and Ünver 2004. Also, Li and Rosen (1998) initiated a literature that analyzes unraveling in competitive environments.

[2] See Ehlers 2002 for game theoretical properties of British two-sided matching mechanisms under incomplete information.

[3] For example, see Roth 1984, 1991, Roth and Xing 1994, Kagel and Roth 2000.

the linear programming mechanisms were introduced are the two smallest markets in Britain. Consultants and graduating medical students know each other very well. On the other hand, the markets in which priority matching mechanisms were used are much larger. Roth (1991) comments on the survival of the linear programming mechanisms as follows:

> One hypothesis is that the environments in which the [linear programming] markets are conducted differ significantly from other environments: each of these two [linear programming] schemes is for the graduates of a single medical school, on the small end of the markets considered here. Thus, there may be social and other kinds of pressures that make it difficult to circumvent the formal matching.

As an alternative to the Roth hypothesis, we propose a hypothesis related to the mechanism characteristics of the linear programming markets as follows: It is easy to adjust to a linear programming matching system; by manipulating their rank-order lists, agents obtain almost stable outcomes and do not unravel the mechanism. A previous study, Ünver 2001, finds evidence in support of this hypothesis by showing that computational adaptive agents can evolve to adjust themselves to the linear programming mechanisms in the long run.[4] It prevents unraveling through adaptation and manipulative strategic behavior. That study is the only paper in the literature that investigates the dynamics of the linear programming mechanisms. A field study, Mongell and Roth 1991, also finds that this is the case for the unstable matching mechanism that is successfully used to accept members to American college sororities.

Kagel and Roth 2000 reports a laboratory experiment on matching markets. They observe in a controlled environment that a priority matching market causes high levels of early matching (comparable with the decentralized market). However, they observe that a Gale-Shapley mechanism is very successful in decreasing the levels of early matching below those of the decentralized markets.[5]

In this paper, we will test mechanism characteristics under a preference and information structure inspired by the actual British markets. We will use experimental and computational methods in the analysis. If the linear programming mechanism prevents early matching more effectively than the priority matching mechanism, this will support the hypothesis that the halt in unraveling is due to the mechanism itself (i.e., it implements another mechanism that obtains desirable matches for the participants) and not to the non-anonymous character of the small markets in which it has been used.

[4] This means that the artificial agents learn to collectively manipulate the mechanism so as to obtain almost ex-post stable outcomes in a dynamic environment without making early contracts.

[5] There is also an emerging literature on matching market experiments. Besides the Kagel and Roth study, for example Chen and Sönmez (2002, 2003) test different matching mechanisms, and Haruvy, Roth, and Ünver (2001), McKinney, Niederle, and Roth (2002), Niederle and Roth (2003), Niederle, Roth, and Ünver (2004) study timing of transactions in various matching contexts.

In addition to human subject experiments, we will also use a computational learning model to model the behavior of human subjects in the experiment. Static equilibrium analysis gives very limited insight to the dynamics that govern the subject learning observed in the experiment. Simulating a learning model makes it possible to obtain insights about subject learning. We use genetic algorithms for this purpose, since individual learning models used in the literature cannot handle the modeling of the learning of subjects in complex environments with many possible strategies (like the games examined here). The importance of this exercise is that it makes possible the comparison of the results with Ünver 2001. Ünver (2001) used long-run simulation results to draw conclusions about why linear programming mechanisms survived in the field in an environment similar to that used in this study. In this study, we use short-run simulation results to show how subjects learn in the laboratory.

The experimental results show that there is no difference between the linear programming mechanism and the priority mechanism in terms of preventing unraveling. They both perform more poorly than the surviving Gale-Shapley mechanism. Short-run artificial agent simulations support this finding as well. However, long-run simulations yield results consistent with the previous Ünver study, demonstrating that in the long-run linear programming mechanisms perform quite well in preventing unraveling.

2 The British Matching Mechanisms

We have Gale and Shapley's (1962) marriage model in mind as the environment where the British matching mechanisms are used. In this model, a matching market consists of a set of consultants, a disjoint set of interns, each consultant's strict preferences over interns and staying unmatched, and each intern's strict preferences over consultants and staying unmatched. Each consultant and each intern can be matched with a single partner or remain unmatched. A matching is a mapping that (i) matches each intern with a consultant or leaves her unmatched and each consultant with an intern or leaves her unmatched, and (ii) maps an intern to a consultant if and only if it maps the same consultant to the same intern. A matching mechanism is a systematic procedure that selects a matching for each matching market. Although various versions of the mechanisms were used in real life, we chose only one from each category to use in our experiment. Next, we explain these mechanisms.

2.1 Linear Programming Mechanisms

Two regional hospital systems (London, 1973 and Cambridge, 1978), each involving a single medical school and its teaching hospital, developed schemes using the optimal assignment model. The rank-order lists of the interns and the consultants are taken as the input. After weights are assigned to these choices, they are summed for each potential consultant-intern pair $[f, w]$ in a matching and denoted by $\alpha_{f,w}$. That is, $\alpha_{f,w}$ is the sum of f's weight of w and w's weight of f. The resulting weights form the basis for an integer programming assignment problem of matching the interns with the consultants so as to maximize the sum of matches. It is equivalent to the following linear programming problem, since corner solutions are obtained:

$$\max_{\{x_{f,w}\}} \sum_{f,w} \alpha_{f,w} x_{f,w}$$
subject to
$$\text{(i) } \sum_{f} x_{f,w} = 1 \quad \forall\, w$$
$$\text{(ii) } \sum_{w} x_{f,w} = 1 \quad \forall\, f$$
$$\text{(iii) } x_{f,w} \in [0,1] \ \forall\, f, w$$

where $x_{f,w} = 0$ means no match between f and w and $x_{f,w} = 1$ denotes a proposed match. First, optimal matrix of "proposed matches" x is determined by solving the above problem.[6] Pair $[f, w]$ who actually listed each other in their rank-order lists and for whom there is a proposed match (i.e., $x_{f,w} = 1$) are matched to each other in the market. Note that even when $x_{f,w} = 1$ if consultant f did not rank intern w or if intern w did not rank consultant f, a match between f and w is not realized in the version of the mechanism we use in this study.

When intern w lists consultant f in k'th place in her rank-order list and the same consultant ranks the intern in l'th place, such a $[f, w]$ match is called a (k, l) match. In the London linear programming mechanism, choices 1, 2, 3, 4, 5, 6 are given weights of 36, 28, 21, 15, 10, and 6, respectively. Thus, a (1,1) match receives weight $\alpha_{f,w} = 72$, (1,2) and (2,1) matches each receives weight $\alpha_{f,w} = 64$, and so forth. An unlisted choice is given the negative weight of -100. We use the London mechanism and these weights in our study.

The linear programming mechanism is unstable. More specifically, the mechanism allows the existence of a consultant-intern pair each of whom prefers another to her match. Even (1,1) matches are not guaranteed to be realized under the linear programming mechanism. Nevertheless, although it is unstable, the linear programming mechanism has *survived* and is still being used. Based on a Monte Carlo study, we observe that only 73.05% (std. dev. %41.61) of all $(1, 1)$ lists turn into a match under the linear programming mechanism per market.[7]

2.2 Priority Matching Mechanisms

The mechanisms introduced in Newcastle (1967) and in two other regions (Birmingham, 1966 and Edinburgh, 1967) assign each match a priority in terms of stated preference rankings of

[6] Whenever there are multiple candidates for optimal x, one is picked with a uniform random draw.

[7] In a market of 6 workers and 6 firms and when the preferences are generated randomly using the partially correlated preference profile described in a later section.

the consultants and the interns. In the Newcastle mechanism, the priority of a (k, l) match is the product of the intern's ranking of the consultant and the consultant's ranking of the intern i.e. $k \times l$. After priorities are found, the matches are realized starting from the lowest priority number.[8] We use this mechanism in our study.

A priority matching mechanism is unstable. However, possible $(1, 1)$ matches are always realized in this particular mechanism. This mechanism was abandoned after a few years of trial in the field. In Newcastle, 80% of the lists consisted of a single choice. This was an evidence of early agreements. Similar centralized matching procedures failed and could not fix the unraveling problem that first appeared during the decentralized matching era.

2.3 Gale-Shapley Mechanisms

Gale-Shapley mechanisms were adopted in Edinburgh (1969) and in one other market (Cardiff, 1971). The Edinburgh (1969) mechanism is a variant of the consultant-optimal Gale-Shapley mechanism.[9] This mechanism has been used successfully for about 30 years. The current mechanism in some Scottish markets is a variant of the intern-optimal Gale-Shapley mechanism.

The outcome of this mechanism is determined by the following consultant-proposing deferred acceptance algorithm:

At the first step, each consultant proposes to her most preferred intern. Each intern is tentatively matched with the best consultant among the pool of consultants who propose to her, and she rejects all other offers.

At any other step, each consultant who does not have a tentative match from the previous step proposes to the best intern who has not yet rejected her. Each intern is tentatively matched with the best consultant among the owners of the proposals at this step and her tentative match from the previous step, and she rejects all other offers. When no offers are rejected at a step, the algorithm terminates and tentative matches are realized as matches.

In Cardiff, a version of the intern-optimal Gale-Shapley mechanism was used (the outcome of this mechanism is determined by the intern-proposing algorithm, dual of the consultant-proposing algorithm, which is obtained by reversing the roles of the consultants and the interns). Gale and Shapley (1962) showed that deferred acceptance algorithms determine stable matchings. Note that pairs with potential $(1, 1)$ matches are always realized under stable mechanisms. In our study, we use the Edinburgh (1969) mechanism.

[8] In the scheme adopted in Newcastle, in cases of ties in priority numbers, interns' preferences are given higher priority: a (1,4) match is more favorable than a (2,2) match, which is more favorable that a (4,1) match although all have the priority number 4.

[9] Its name reflects the fact that this mechanism finds the best stable matching for consultants.

3 Experimental Design

Following the chronological organization of the British markets, we use two different matching game designs in the experiment: a decentralized game, which is later replaced with a mixed game. The second game is called "mixed" because agents can choose to be matched in a decentralized manner or in a centralized manner, whereas in the first game agents can be matched only in a decentralized manner. Each experimental session consists of 10 consecutive decentralized games followed by 15 consecutive mixed games.

In the experiment, there are two disjoint sets of subjects: firms (consultants) and workers (interns).[10] Each worker seeks a job at one firm and each firm can hire only one worker. Subjects belong to one of two types, high productivity or low productivity. They are also chosen to be firms or workers. The set of firms is denoted as F and the set of workers is denoted as W. The types of agents are common knowledge. At the beginning of each session, 6 subjects are randomly assigned as firms and 6 subjects are randomly assigned as workers. Their types are randomly determined as well: 3 firms and 3 workers are assigned the high productivity type and the remaining 3 firms and 3 workers are assigned the low productivity type. Subjects remain the same type throughout the experiment. The utility function of agent v is denoted by u_v. The utility function of firm f is $u_f : W \cup \{f\} \to \mathbb{R}_+$, and it is defined for worker w' as

$$u_f(w') = \sigma_{w'} + s_{f,w'}$$

where random variable $s_{f,w'}$ is drawn from a discrete cumulative distribution function G with support $[-\$1, \$1]$.

The utility function of worker w is $u_w : F \cup \{w\} \to \mathbb{R}_+$ and is defined for firm f' as

$$u_w(f') = \sigma_{f'} + s_{w,f'}$$

where $s_{w,f'}$ is also distributed with discrete cumulative distribution G. Any agent v remaining unmatched has utility 0 and this is denoted by $u_v(v) = 0$.

For any agent v, the non-random part in her partner's utility function, σ_v, is determined by v's productivity type as follows:

$$\sigma_v = \begin{cases} \$5 & \text{if } v \text{ is type low} \\ \$15 & \text{if } v \text{ is type high} \end{cases}$$

Marginal distribution G is obtained from a *joint discrete uniform distribution* in interval $[-\$1, \$1]$ (with 5 cent intervals) without permitting any ties in the utility values of an agent to sustain strict preferences. The utility profiles of the experiment are fixed throughout the sessions. 25 utility profiles are used, one for each matching market. We determine 15 utility

[10] We refer to the consultants as "firms" and the interns as "workers" in the experiment.

profiles used in the mixed games as follows: We randomly generate a utility profile and use this in the experiment if it involves at least one mismatch between the high types and the low types under the linear programming mechanism and the priority matching mechanism. We want to ensure by doing so that the unstable mechanisms can really produce unstable outcomes when agents reveal their preferences truthfully.[11]

By a Monte-Carlo study, we verify that about 39.70% of the random matchings produced by the priority matching mechanism have at least one mismatched pair. Almost 96.14% of the time we observe a mismatch under the linear programming mechanism.

The preference structure is similar to that of the British markets where there is usually a most preferred central hospital with many residency positions and several less preferable smaller hospitals. Interns participate both with and outside the region, and their academic performance is known by the consultants so that they can be classified as good or bad candidates. Specific features of the interns can cause them to be ranked differently by each consultant.

Sets $\{\sigma_f\}$ and $\{\sigma_w\}$, i.e. the types of agents, are common knowledge. For each worker w, $s_{w,f'}$ is private information to w. For each firm f, $s_{f,w'}$ is private information to f. Thus all subjects know each other's types, σ values, the support for random s values for other agents, and the realization of their own s values.

3.1 The Decentralized Game Design

The decentralized game has three periods: in round -2 (that is, round minus two), each firm has the option to offer a position to at most one worker, and each worker has the option to accept at most one offer. An accepted offer binds parties for an early match. Round -1 (that is, round minus one) is a replay of round -2 among those players who did not make any contracts in round -2. Similarly, round 0 consists of the replay of round -1 among those players who have not made previous contracts. The rounds differ from each other in the sense that matching in round -2 costs \$2, matching in round -1 costs \$1, and matching in round 0 does not cost anything. Thus earlier contracts cost more than later ones.

The structure of the decentralized game is motivated by the decentralized British markets. Costs to the participants when a market unravels can be caused by several factors, including uncertainty about the quality of agents and the loss of planning flexibility. In this experiment, all such costs are represented by the dollar cost of matching early. This is an easy but effective way of incorporating the opportunity costs of unraveling with monetary payoffs. The utility of

[11] The Kagel and Roth (2000) design is somewhat similar to our design, except that the preference profiles were picked by hand to ensure that the priority matching mechanism created at least one mismatched pair, and they consider only the priority and Gale-Shapley treatments. The number of the mismatched pairs in their design is fixed to be higher on average than the number of mismatched pairs in our design under the priority matching mechanism.

being matched with a high type in round 0 is in the interval [\$14, \$16], and the utility of being matched with a low type in round 0 is in the interval [\$4, \$6]. Moreover matching in round -2 with a high-type agent (which brings at least \$12) is better than matching with a low-type agent in round 0 (which brings at most \$6). However, matching to the best high-type choice in round -2 brings (at most \$14) brings at most the same payoff as matching to the worst high-type choice in round 0 (at least \$14). Matching with the best high-type choice in round -1 can bring more or less payoff than matching with the worst high-type choice in round 0.

The net payoff of an agent in the market is the utility she gets from being matched minus the "early" contract cost if any. An unmatched agent gets utility 0, representing the prospect facing the unmatched participants in the observed markets, which involves entering a secondary after-market.

3.2 The Mixed Game Design

The centralized matching markets employed after the decentralized era in Britain motivate the mixed games. A mixed game also has three periods. In rounds -2 and -1, each firm has the option to make an offer to at most one worker of its own choice, and each worker has the option to accept at most one of those offers. An early contract is costly. A centralized market, where the agents submit rank-order lists, replaces round 0. These lists are processed by a centralized mechanism for matches.

In each game early matches are interpreted as a commitment by the parties to list each other in the first place of their rank-order lists of round 0 (i.e., they arrange for a $(1, 1)$ match).[12] Arranging agents are free to fill in the rest of their rank-order lists in any way they wish. The costs of contracts are set as \$2 in round -2, \$1 in round -1 and \$0 in round 0. A cost is charged only if the early contract is successful and is realized as a match in round 0. Under the linear programming mechanism, as noted earlier $(1, 1)$ matches are not always achieved.

In the mixed games with the priority matching mechanism and the Gale-Shapley mechanism, $(1, 1)$ matches are always achieved, so early contracts are binding. Agents who make an early contract are matched with each other and they do not participate in the experiment's centralized matching round.

The information structure is private throughout the play of the games. No other agent can observe an arrangement between a firm and a worker in rounds -2 and -1. All other offers made by firms, rejections of offers, and the submitted rank-order lists are also private.

The net payoff of an agent in the market is the utility she gets from being matched minus the "early" contract cost if any.

[12] Because positions in the UK belong not to the consultants but to the health service, it was necessary for consultants and interns to fill out rank-order lists even if they had reached an early agreement.

3.3 Experimental Sessions

We consider three treatment groups in which mixed games use priority matching, Gale-Shapley, and linear programming mechanisms, respectively. We conduct nine sessions total, three sessions of each treatment. Subjects were recruited among inexperienced undergraduate students from different disciplines at the University of Pittsburgh and sessions were run at the Pittsburgh Experimental Economics Laboratory (PEEL).

Each play of a game is called a *market.* Each experimental session consisted of 10 decentralized markets followed by 15 mixed markets. At the beginning of the experiment, a computer tutorial explained the objectives of the experiment and described the decentralized game to the subjects. We gave written instructions and read them aloud during the tutorial session. After the decentralized games were played, we used transparencies accompanied with written instructions to describe the mixed games to the subjects. We read the instructions aloud. For the priority matching treatment and the Gale-Shapley treatment, the instructions explicitly stated that $(1, 1)$ lists were certain to be realized as matches and therefore they did not need to fill in rank-order lists in round 0 if they made early contracts. For compatibility, we gave the subjects explicit instructions that $(1, 1)$ lists were not guaranteed to be realized as a match under the linear programming mechanism. The rest of the information about the centralized mechanisms was similar to the information given to the interns in the field.[13]

There were 12 subjects in each session: 3 high-type firms, 3 low-type firms, 3 high-type workers, and 3 low-type workers. The subjects retained their identity as firm or worker and their productivity type throughout the session. To keep the identities of the subjects private and to prevent repeated game effects, we generated random ID numbers in each market. At the end of each market, the matches were announced: the players observed the productivity types of the agents involved in the matches and the round that each match occurred. Easy visibility of instabilities is a feature of the British markets, so in the experiment interns and consultants know the outcome of the match of the previous year before participating in the current year's match.

Each subject was paid an \$8 participation fee plus the payoff she achieved in one of the plays of the decentralized game (maximum \$16) and the payoff achieved in one of the plays of the mixed game (maximum \$16). Thus subjects could earn payoffs between \$8 and \$40. The average was close to \$20. The results of these payment games were determined randomly at the end of the sessions.

[13]The intstructions and preference profiles used are given in Appendices B and C, located at `http://home.ku.edu.tr/~uunver/research/survivalappendix.pdf`.

4 Analysis of the Results

4.1 Cost of Unraveling in the Laboratory and in the Simulations

To measure the total level of unraveling of contracts via early matches, we define *cost of unraveling* as the actual total cost of all "successful" early contracts in a matching market: the sum of all \$1 and \$2 fees collected for early matches after each matching game is over.

We report the results in groups of five consecutive markets (i.e., we use average results for each set of 5 markets) called *blocks*. We take the averages of the last five or the first five markets as a single datum to represent each cohort. Therefore we have a low sample size. We use parametric t-tests to analyze the data. Because of the small sample size (3 in most cases), we use $\alpha = 0.1$ as the significance level for the tests. Data are analyzed using "one-sided" paired or two-sample tests unless otherwise noted. The degrees of freedom denoted by d.f. tells whether the test is a two-sample or a paired one. For example, d.f.=2 or 8 refers to a paired test and d.f.=4 or 16 refers to a two-sample test. Our main finding about the costs of unraveling is stated as follows:

Result 1. When the unraveling costs are considered, the Gale-Shapley market performs better in decreasing costs of unraveling of the decentralized markets than both the priority matching and the linear programming markets, which perform equally badly. High types are mostly responsible for early matching in all treatments.

In the remainder of this subsection, we discuss the data supporting Result 1. Figure 1a is a graph of the cost of unraveling versus matching markets in the laboratory. The first 10 markets use the decentralized game design (denoted by "Decentralized" in the graph) and the next 15 markets use one of the mixed games (denoted by "Mixed" in the graph). In these graphs and all of the others, the horizontal axis refers to the market blocks, and the vertical axis refers to the average of the quantity in question. In the last block, the cost in the Gale-Shapley market is significantly less than the cost in the priority matching market and the linear programming market. When compared with the last block of the Gale-Shapley market, for the priority matching market the difference in cost is \$1.77, which has p=0.08920, and for the linear programming market the difference in cost is \$0.97, which has p=0.09466 (both in t-tests with d.f.=4). Note that in the last block, the linear programming market cost and the priority matching market cost are not significantly different from each other, although the linear programming market cost seems less.

At the disaggregated level, the high types are more responsible for early matching than the low types. For the linear programming market, the difference between the unraveling cost for the high types and the low types is \$1.2 per session with p=0.06223; for the priority matching market we observe a \$2.18 difference with p=0.03986 and for the Gale-Shapley market this

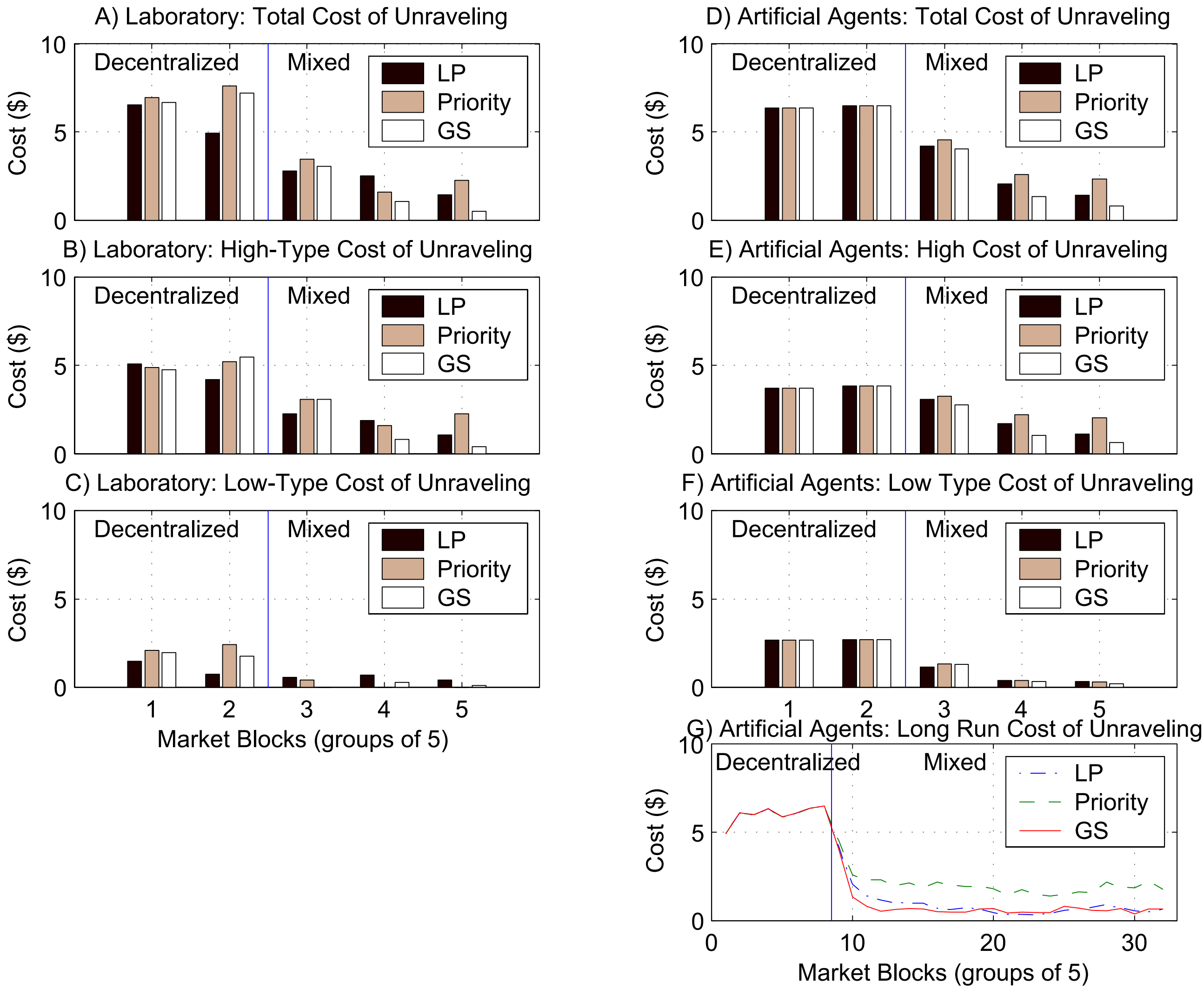

Figure 1: Cost of unraveling in the laboratory (Graphs A-C) and in the short-run (Graphs D-F) and long-run simulations (Graph G). The first two blocks of markets in graphs A-F (10 markets in total) are under the decentralized game and the last three blocks in graphs A-F (15 markets in total) are under the mixed game using one of the mechanisms in the legend of the figure. In the legend, LP refers to the linear programming treatment, priority refers to the priority matching treatment, and GS refers to the Gale-Shapley treatment.

difference is \$1.31 with p=0.09551 (all are obtained as results of t-tests with d.f.=2). For the overall decentralized market the difference is \$3.2 with p=0.0002443 (as the result of a t-test with d.f.=8). Figure 1b and Figure 1c summarize these results. Therefore, the high types are responsible for most of the decrease in early contracts in the transition from the decentralized design to the mixed design.[14]

We try to explain the behavior observed in the experiment with an artificial-agent-based learning model. We use as the learning model a computational algorithm called a *genetic algorithm* (Holland 1975). The method used is similar to the Ünver (2001) study. In this study, we include only some simulation results for comparison. One major difference between the current simulation and the ones in Ünver 2001 is the length and duration of each treatment. Here each artificial treatment lasts as long as an experimental treatment, 25 markets. In Ünver 2001, we were mostly interested in the long-run adjustments.[15]

In terms of the unraveling costs in the artificial agent markets, the deferred acceptance market performs better than the linear programming market, which in turn leads to lower early matching costs than the priority matching markets. This can be observed in Figure 1d. Computationally, the mixed markets cause lower costs than the decentralized markets (mixed markets cause \$2.60 per market on average, and decentralized markets cause on average \$6.42 per market, for a difference of \$3.82). Note that the simulations reflect an average of 30 sessions for each treatment, whereas the experimental results reflect averages of 3 sessions for each treatment. Since the differences in the simulations are mostly significant we do not report their t-test analyses.

4.1.1 Long-Run Simulation Dynamics

In this subsection, we extend the length of the social learning simulations to create a meaningful comparison with the Ünver (2001) study, which predicted that the linear programming mechanisms should be very successful in preventing early contracts contrary to the experimental and short-run simulation findings. Now, we use the simulation results in the long run (i.e., for dura-

[14] If we inspect number of early contracts as an alternative measure of unraveling, the results do not change extensively. The number of contracts made by high types is significantly larger than that of low types in round -1. In round -1, for the linear programming mixed markets the difference is 0.82 with p=0.01797; for the priority mixed markets the difference is 0.96 with p=0.05976; and for the Gale-Shapley mixed markets the difference is 0.54 with p=0.1060 (all results are obtained in t-tests with d.f.=2). In round -2, there is a significant difference between high- and low-type agent contracts only in the mixed Gale-Shapley markets and in the decentralized markets. Low-type agents virtually stop matching early in transition from the last five decentralized markets to the first five mixed markets; this is significant only for the priority matching treatment with p=0.03414 (in a t-test with d.f.=2).

[15] The details of these simulations are given in Appendix A, located at `http://home.ku.edu.tr/~uunver/research/survivalappendix.pdf`.

tions longer than can physically be studied in the laboratory) as in Ünver 2001 methodology. We use the same parameters as the short-run simulations reported above, and the same preference profiles as the experiment. We hold 40 decentralized markets followed by 120 mixed markets. The costs of unraveling are shown in Figure 1g. As can be clearly seen, the Gale-Shapley mechanism and the linear programming mechanism are successful in the long run in preventing costs, while the priority mechanism performs relatively more poorly. In the last 40 matching markets (from 121 to 160), the Gale-Shapley and the linear programming mixed market costs are lower than the priority mixed market costs. We see a -$1.17 difference between the linear programming and the priority matching mixed markets and a -$1.22 difference between the Gale-Shapley and the priority matching mixed markets. The unraveling cost in the deferred acceptance and the linear programming markets are comparable to each other in the last 40 mixed generations, with a non-significant difference of -$0.04533 with p=0.4008 (as the result of a two-sided t-test with d.f.=58). These findings are consistent with the findings of Ünver (2001).

4.2 Other Characteristics of the Experiment

This subsection is related to the experimental results. We analyze whether agents manipulate their rank-order lists in round 0 of the mixed games, how stable the final matching is, and how many agents are unmatched in general. Our main finding in this subsection is stated as follows:

Result 2. Comparable numbers of agents remain unmatched in the mixed markets; however, the linear programming market leads to more mismatches between agent types than the other mixed markets. This is a result of the nature of the linear programming mechanism and its effects are inflated by low agents of the linear programming markets truncating their rank-order-lists more aggressively than the low agents in the other mixed markets.

In the remainder of this subsection, we explain the data supporting Result 2. As seen in Table 1, agents do not usually submit full lists in round 0 of the mixed games. The low-type agents submit longer lists than the high-type agents: there is a 1.0025 agent difference with p=0.01108 in the Gale-Shapley markets, a 0.5333 agent difference with p=0.01181 in the linear programming markets, and a 0.8656 agent difference with p=0.005240 in the priority markets for the last two blocks (using t-tests with d.f.=2).

Table 1. Laboratory Experiment: Average Length of Rank-Order Lists in Round 0 of Mixed Markets

Averages in Last Ten Markets	LP	Priority	GS
For High Types	4.4944 (0.2311) s.e.	4.5642 (0.3054) s.e.	4.8371 (0.3830) s.e.
For Low Types	5.0278 (0.1182) s.e.	5.4278 (0.1251) s.e.	5.8592 (0.3054) s.e.

The high-type agent rank-order lists have comparable lengths under all treatments: we observe an insignificant p=0.4004 for a comparison between the Gale-Shapley market and the priority markets, an insignificant p=0.2826 for a comparison between the Gale-Shapley market and the linear programming market, and again an insignificant p=0.7747 for a comparison between the linear programming market and the priority market, using averages in the last two blocks (using a two-sided test with d.f.=4).

The shorter lists of the high-type agents reflect their effort to avoid mismatches with the low types. A low-type agent has the highest opportunity for a mismatch to a high-type agent under the linear programming scheme by the nature of the linear programming mechanism. By omitting some low-type agents from her rank-order list, a low-type agent increases her chances to be matched with a high-type agent (the linear programming mechanism algorithm avoids leaving agents unmatched as much as possible). This can be the underlying reason that the linear programming low-type agent lists are the shortest among the low types: for the rank-order list length of low-type agents in the linear programming market, we see a difference of -0.4 agents from the priority market with a significant p=0.007900, and a -0.8335 agent difference from the Gale-Shapley market with a significant p=0.0008552 for averages obtained in the last two blocks (using t-tests with d.f.=4).

Below we analyze characteristics of the market matching such as its efficiency for the unmatched agents and its ex-post stability.

The *number of mismatched agents* is defined as the number of agents who are matched with types different from their own. Similarly, we define the *number of unmatched agents* to measure how many agents are not matched in a market. The latter measures inefficiency in the Pareto sense.

As seen in Table 2, the centralized mechanisms decrease the number of unmatched agents below the decentralized market level: for the linear programming treatment we have a highly significant p=0.00003470, for the priority treatment we have a highly significant p=0.00010728, and for the Gale-Shapley treatment we have a highly significant p=0.00002029 using averages of the last two blocks when each design is compared with its decentralized design (using t-tests with d.f.=2). Since the opportunity to submit a rank-order list can be interpreted as each agent making as many as 6 offers, the mixed games have an advantage over the decentralized game where each firm extends at most 1 offer. This is one of the factors driving the high number of early contracts in the decentralized market as well. Although the priority treatment causes slightly more agents to stay unmatched (with an insignificant difference), we observe a comparable number of unmatched agents in each mixed treatment.

Table 2. Laboratory Experiment: Characteristics of the Market Matching

Averages in the Last 10 Markets	Decentralized	LP	Priority	GS
Unmatched Agents	3.6622	0.6	0.9933	0.6815
	(0.5230) s.e.	(0.5292) s.e.	(0.5774) s.e.	(0.4496) s.e.
percentage	30.52%	5%	8.28%	5.68%
Mismatched Agents	0.7778	3.4	1.0667	0.4074
	(0.6815) s.e.	(1.1136) s.e.	(0.5033) s.e.	(0.5251) s.e.
percentage among all matched	9.33%	29.82%	9.76%	3.60%

We also see in Table 2 that the decentralized market does not cause a substantial amount of mismatches between high types and low types. We observe a substantial amount, 3.4 mismatched agents on average, in the last ten linear programming mixed markets, which is much greater than the other treatments: for the linear programming treatment, when compared with the priority treatment we have a significant p=0.01487 (with d.f.=4), when compared with the Gale-Shapley treatment, we have a significant p=0.007618 (with d.f.=4), and when compared with the decentralized design, we have a significant p=0.0001227 (with d.f.=2) for averages in the last two blocks. We think that this is a result of the nature of the linear programming mechanism, where mismatched pairings can bring an optimal matching (especially when the low agents submit truncated lists and the high agents occasionally continue listing the low types). The priority mechanism causes fewer mismatched pairs than the linear programming mechanism. If everyone had played non-strategically and submitted truthful lists, the mixed markets would have led on average to 2 mismatched agents under the priority mechanism and 3.2 mismatched agents under the linear programming treatment. The high types cut down on mismatches by achieving early contracts in the priority treatment. However, even with early contracts the high types cannot cut down on mismatches in the linear programming treatment. We think that this last point is one of the main reasons why the number of early contracts in the linear programming treatment is not greater than it is in the priority treatment. We observe only very few mismatched agents in the Gale-Shapley treatment (see Table 2).[16]

[16] Kagel and Roth's (1990) experiment adopts priority and Gale-Shapley matching mechanisms. There are two differences that we observe between the results of this study for the priority and Gale-Shapley treatments and the Kagel and Roth (2000) study. The first difference concerns the magnitude of unraveling costs under the priority matching treatment and the second concerns the way that the early matching is prevented under the Gale-Shapley treatment. Since our preference profiles have fewer mismatched agents than theirs under the priority treatment, the level of early matching settles to a slightly lower level than in their findings. Moreover, under the Gale-Shapley treatment in our experiment, high-type firms first stop to make early offers as opposed to their findings. Besides these two minor details, findings of this study and the Kagel and Roth study suggest a robust phenomenon in regards to the unraveling potential of the priority matching markets.

5 Conclusions

We found no result that would suggest that the field success of the unstable linear programming mechanisms *in the short run* should be attributed to intrinsic game theoretical properties of the mechanism. Our findings are robust, since we observe that they are consistent with the Kagel and Roth (2000) and the Ünver (2001) findings.

We conclude that Roth's (1991) explanation for the success of the linear programming mechanisms (even in the short run) still needs to be tested in the laboratory. A future research agenda may require an experimental design using the market culture of these specific markets in determining why the linear programming mechanisms have survived. We currently reject the hypothesis that the features of the market related to the mechanism characteristics as the reason for its success.

References

[1] Chen Y and Sönmez T (2002) Improving Efficiency of On-Campus Housing: An Experimental Study. Amer Econ Rev 92: 1669-1686.

[2] Chen Y and Sönmez T (2003) School Choice: An Experimental Study. Working paper.

[3] Ehlers L (2002) In Search of Advice for Physicians in Entry-Level Medical Markets. Working paper.

[4] Gale D and Shapley LS (1962) College Admissions and the Stability of Marriage. Am Math Mo 69: 9-15.

[5] Haruvy E, Roth AE and Ünver MU (2001) The Dynamics of Law Clerk Matching: An Experimental and Computational Investigation of Proposals for Reform of the Market. Working paper.

[6] Holland JH (1975) Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann-Arbor, MI.

[7] Kagel JH and Roth AE (2000) The Dynamics of Reorganization in Matching Markets: A Laboratory Experiment Motivated by a Natural Experiment. Quart J Econ 115: 201-235.

[8] Kesten O (2003) On Three Kinds of Manipulation in Two-Sided Matching Markets. Working paper.

[9] Li H and Rosen S (1998) Unraveling in Matching Markets. Amer Econ Rev 88: 371-87.

[10] McKinney CN, Niederle M, and Roth AE (2002) The Collapse of a Medical Labor Clearinghouse (and Why Such Failures are Rare). Working paper.

[11] Mongell S and Roth AE (1991) Sorority Rush as a Two-Sided Matching Mechanism Amer Econ Rev 81: 441-464.

[12] Niederle M and Roth AE (2003) Market Culture: How Norms Governing Exploding Offers Affect Market Performance. Working paper.

[13] Niederle M, Roth AE and Ünver MU (2004) Unraveling Results from Comparable Supply and Demand. Working paper in preparation.

[14] Roth AE (1991) A Natural Experiment in the Organization of Entry-Level Labor Markets: Regional Markets for New Physicians and Surgeons in the United Kingdom. Amer Econ Rev 81: 415-440.

[15] Roth AE and Sotomayor M (1990) Two-Sided Matching: A Study in Game Theoretic Modeling and Analysis. Cambridge University Press, Cambridge.

[16] Roth AE and Xing X (1994) Jumping the Gun: Imperfections and Institutions Related to the Timing of Market Transactions. Amer Econ Rev 84: 992-1044.

[17] Sönmez T (1999) Can Pre-Arranged Matches Be Avoided in Two-Sided Matching Markets? J Econ Theory 86: 148-156.

[18] Ünver MU (2001) Backward Unraveling over Time: The Evolution of Strategic Behavior in the Entry-Level British Medical Labor Markets. J Econ Dynam Control 25: 1039-1080.