Anarchy, Stability, and Utopia: Creating Better Matchings

Elliot Anshelevich
Dept. of Computer Science
Rensselaer Polytechnic
Institute
Troy, NY 12180
eanshel@cs.rpi.edu

Sanmay Das
Dept. of Computer Science
Rensselaer Polytechnic
Institute
Troy, NY 12180
sanmay@cs.rpi.edu

Yonatan Naamad
Dept. of Computer Science
Rensselaer Polytechnic
Institute
Troy, NY 12180
naamay@cs.rpi.edu

ABSTRACT

We consider the loss in social welfare caused by individual rationality in matching scenarios. We give both theoretical and experimental results comparing stable matchings with socially optimal ones, as well as studying the convergence of various natural algorithms to stable matchings. Our main goal is to design mechanisms in order to incentivize agents to participate in matchings that are socially desirable. We show that theoretically, the loss in social welfare caused by strategic behavior can be substantial. We analyze some natural distributions of utilities that agents receive from matchings, and find that in most cases the stable matching attains close to the optimal social welfare. Furthermore, for certain graph structures, simple greedy algorithms for partner-switching (some without convergence guarantees) converge to stability remarkably quickly in expectation. Even when stable matchings are significantly socially suboptimal, slight changes in incentives can provide good solutions. We derive conditions for the existence of approximately stable matchings that are also close to socially optimal, which demonstrates that adding small switching costs can make socially optimal matchings stable. We also show that introducing heterogeneity in tastes can greatly improve social outcomes.

1. INTRODUCTION

This paper investigates the social quality of stable matchings. The theory of stable matching has received a tremendous amount of attention because of its many applications, including matching graduating medical students to residency programs [21], and matching kidney donors with recipients [4]. Most of the work on stable matching has assumed that the agents being matched have some preference ordering on who they would like to be matched with, without assigning a concrete utility for agent i being matched with agent j [22; 23; 15, inter alia]. This is natural, because stability as a concept does not need the stronger requirement of ascribing utilities to outcomes: it only needs the ranking of matchings from the perspective of every agent.

Matching problems, however, often bring with them outcomes that need to be evaluated in terms of utility. This occurs, for example, in pair programming, a central practice of the software engineering methodology known as Extreme Programming [12]. The utility of a matching is a function of the productivity of a pair of programmers working together. In kidney exchange, as well as many other stable matching scenarios, the goal is not only to form stable matchings, but also to form a matching with high overall utility.

The properties of matching mechanisms determine the utilities received by agents in these situations. A good mechanism for kidney exchange could make donors happier with their decision to donate while arranging the best possible matches for recipients. A good mechanism for pairing programmers would lead to the best possible programming productivity for their employer. Inevitably, there is a tradeoff between stable matchings, which are pairwise (or groupwise) rational, and socially optimal matchings (for our purposes, for the rest of this paper we assume simple additive social utilities, so that the socially optimal matching is the one that maximizes the sum of utilities received by each individual). The central question of mechanism design for matching markets is how to get people into "good" matchings, however "good" is defined. Almost all the work on matching mechanism design has focused on engineering stable matchings. This work has met with significant large-scale success in applications like matching graduating medical students to residency programs, and matching students to public high schools [1, 21]. Some of this work, especially recent work on designing high school student matches, also explicitly seeks to realize the best matchings for one side of the market (in the high school case, the best matchings for students), but the notion of welfare is weak pareto-optimality among the set of stable matches for one side of the market [2].

The focus of this paper is on extending our understanding of matching problems in situations where we are concerned with social welfare in terms of utility, instead of just stability and choice among stable outcomes. Several alternatives may be available in these situations, ranging from purely centralized allocation based on information available to a matchmaker, to purely individual decision-making based on personal preferences. The first set of questions that arises can be divided into three categories: (1) How bad are stable matchings when compared with socially optimal ones? (2) Can agents find stable matchings on their own? What are the outcomes of algorithms they may actually use in practice? (3) How can we incentivize agents to participate in matchings that are socially desirable?

Our Results. We initiate an investigation of the questions described above in the context of two-sided matchings, and give both theoretical and experimental results. Specifically, we study the effects of different network structures and utility distributions on the price of anarchy: the ratio of social utilities achieved by stable and optimal matchings respectively. We find that in most cases the stable matching attains close to the optimal social welfare (generally above 90%). We characterize some situations where the price of anarchy can be more substantial, and then study a potential means of incentivizing good stable matchings in Section 5. We consider *approximate stability*, which corresponds to the addition of a switching cost to the mechanism, so that an agent would have to pay in order to deviate from the current matching. We show both theoretically and experimentally that the addition of a small switching cost can greatly improve the price of anarchy. Finally, in Section 6 we consider several greedy algorithms for partner-switching, and show experimentally that they converge quickly to stability for some simple yet natural distributions of utilities, as well as prove convergence guarantees.

2. MATCHING, STABILITY, AND SOCIAL WELFARE

Matching, the process of agents forming beneficial partnerships, is one of the most fundamental social processes. Examples of matching with self-interested agents range from basic social activities (marriage, mate assignment [9]), to the core of economic activity (matching employees and employers [17]), to recent innovations in health care (matching kidney donors and recipients [4]). The process of matching can be extremely complex, since (1) agents can have complicated preferences, and (2), in most social applications agents are self-interested: they care mostly about their own welfare, and would not obey a centralized matching algorithm unless it was to their benefit.

For this reason, the outcomes of matching processes are usually analyzed in terms of *stability*, the requirement that no collection of agents could form a group together, and become better off than they are currently [22]. For the classic "stable marriage" problem [13], this corresponds to the lack of desire of any pair to drop their current partners and instead match with each other. Stable matching algorithms have been used in many applications including matching medical residents with hospitals [21], students with sororities and schools [1, 19], and online users with servers.

While stable matchings may be natural outcomes, desirable for various reasons, there are few guarantees on the quality and social welfare of stable matchings. Most research on matchings of self-interested agents has focused on (1) defining outcomes with stability as the goal (most of the work on the design of two-sided matching markets attempts to do exactly this by defining problems appropriately [22]), (2) computing stable outcomes and understanding their properties (ranging from the seminal work of Gale and Shapley [13] to algorithms that try and compute "optimal" matches, for example by minimizing the average preference ranking of matched partners [16]), and (3) designing truthful preference-revealing mechanisms (such as in the New York City [2] and Boston public school matches [3]). Questions about the social welfare of stable matchings have been less studied.[1] There has been almost no research on constructing socially desirable stable outcomes, partly because in most situations one cannot instruct self-interested agents on what to do in order to engineer such outcomes, since an agent will only follow instructions if it benefits them personally.

An increasing body of literature in behavioral economics and social science (e.g. [25]), however, suggests that desirable outcomes can be achieved by giving people a little "nudge" in certain directions, perhaps by altering their incentives slightly, while still leaving them with freedom to choose their own actions. Small changes that greatly improve a social system are easy to identify in some situations: for example, making 401(K) plans opt-out rather than opt-in increases participation dramatically. Finding similar changes in matching scenarios is more difficult because of the complexity of a system where any agent's actions can theoretically affect a large number of other agents.

Before addressing the mechanism design question of how to achieve better social outcomes, we first need to address the question of whether or not stable matching can lead to substantial social losses. For this question to make sense, we first need an objective function that measures the quality of a matching. As mentioned in the introduction, one of the reasons why the social quality of stable matchings is usually not addressed is because the agents in question are assumed to have a preference ordering on their possible partners, without a specific utility function that states how good a match would be. While there has been some work on measuring the quality of a matching by, for example, the average preference ranking of matched partners [16], such measures can sometimes be hard to justify. For example, for an agent A, the second choice in its preference order might be a lot worse than its first choice, while for agent B, the second choice might be only a little bit worse. Measures such as the one above would make no such distinction. In this paper, we are specifically concerned with contexts where every agent has a utility function, not just a preference ordering: that is, for every possible partner v, an agent has a value $U(v)$ specifying how happy it would be to be matched with v. We are especially concerned with measuring the quality of a matching in terms of social welfare: the total sum of utilities for all the agents.

We would like to understand the social welfare of stable matching. The tradeoff between stable matchings and socially optimal matchings is quantified by the *price of anarchy*: the ratio between the maximum possible social utility and the utilities of equilibrium outcomes (stable matchings). Understanding the price of anarchy is important, since it acts as a bound on the amount of improvement in stable matchings that better mechanisms could yield.

Price of Anarchy Bounds. The price of anarchy can vary widely depending on the problem instance and the preference structure. As an example, Figure 1 illustrates some cases where the stable matching is highly socially suboptimal (discussed in more detail in the next section). In two of the underlying types of graph structures, the price of anarchy is at most two (and the bound can be tight), while

[1] As mentioned in the introduction, one of the desiderata for matching students with schools or medical students with residencies can be to compute the stable matching that is best (typically) for the students, but this is a different notion of welfare.

in the third the social utility of the stable matching can be arbitrarily bad compared with the socially optimal one. But how bad are stable matchings in expectation?

This question is tackled in detail in Section 4. Empirically, we find that despite the potentially bad worst-case behavior, across many different random distributions of preferences and several graph structures the price of anarchy tends to be lower (stable matchings usually achieve above 90% of the utility of socially optimal matchings). There are also some cases where the price of anarchy is not the right measure – we show a case where tweaking a preference-related parameter increases the price of anarchy significantly, but makes everyone better off in expectation because it raises the value of the optimal social matching. When the price of anarchy is a good measure, how can we incentivize socially good matchings?

Creating Better Stable Matchings. Given the agents' utilities, the social-welfare maximizing matching can be computed by finding a maximum weighted matching on a graph. We cannot just force people to accept such a matching because of individual preferences. But what if we could suggest a good matching, and provide some incentives for agents to go along with those matchings? This is the subject of Section 5. We consider changing incentives to make more socially desirable matchings become stable by adding switching costs into the system. We show both theoretically and empirically that a small amount of incentives can greatly affect the quality of stable matchings.

Convergence to Stability. Another natural question we ask is whether stable matchings will arise in practical situations, where each participant does not want to submit his or her preferences to a centralized matchmaker. Previous work has focused especially on randomized best response dynamics [5, 20]. We know that simple decentralized partner switching algorithms can fail to converge to stable matchings in many situations [5]. However, what happens in cases where the structure of preferences obeys some extra constraints? We explore this question in Section 6.

3. THE MATCHING MODEL

In this paper we are concerned with pairwise matching problems. While we focus on bipartite graphs, (most of) our results also hold for general graphs, and in our experiments we did not find a significant difference between the quality of matchings in bipartite and non-bipartite graphs. We assume that each agent gains some utility from being paired up with another agent. The utility of remaining unmatched is assumed to be 0. We consider each agent as a vertex in a graph G, and only agents u and v with the edge (u, v) being present in G are allowed to be matched with each other. In two-sided matching scenarios, the agents can be separated into two types, one on each side of the graph, and no edges are allowed between agents of the same type.

We consider several different utility structures:

1. **Vertex-labeled graphs:** A vertex-labeled graph is defined as $G = (V, E, w)$ where V is the set of vertices, E is the set of (undirected) edges, and w is a vector of weights corresponding to the vertices. When two vertices u and v are in a matching, the agent corresponding to u receives utility $w(v)$ and the agent corresponding to v receives utility $w(u)$. These graphs correspond to a situation where being paired with agent X will yield the same utility to any agent Y allowed to match with X, independent of the identity of Y.

2. **Symmetric edge-labeled graphs:** A symmetric edge-labeled graph $G = (V, E, w)$ is different in that the weights w correspond to edges rather than vertices. When two vertices u and v are in a matching, the agents corresponding to both u and v receive utility $w(\{u, v\})$. These graphs reflect situations where the utility received by both members of a pair is the same, perhaps determined by their combined output when working together – for example, pair programming may be judged by the productivity of the pair. Markets with these types of utilities are called "correlated two-sided markets" in [5].

3. **Asymmetric edge-labeled graphs:** An asymmetric edge-labeled graph $G = (V, E, w)$ is the same except that edges are now directed, and the utility received by agent u in a matching that includes the pair u, v is given by $w(u, v)$, while the utility received by v is given by $w(v, u)$. This is the most general case, in which each agent receives an unconstrained value from each agent they may possibly be paired with.

We also consider combinations of the above models, such as when agent u's utility for being matched with v has a vertex-labeled component $w(v)$, as well as an edge-labeled component $w(u, v)$. The types of utilities mentioned above arise in many contexts including market sharing games [14] and distributed caching games [18]. In the context of marriage markets, vertex-labeled graphs are equivalent to what Das and Kamenica call *sex-wide homogeneity of preferences*, and edge-labeled graphs are equivalent to what they call *pairwise homogeneity of preferences* [11].

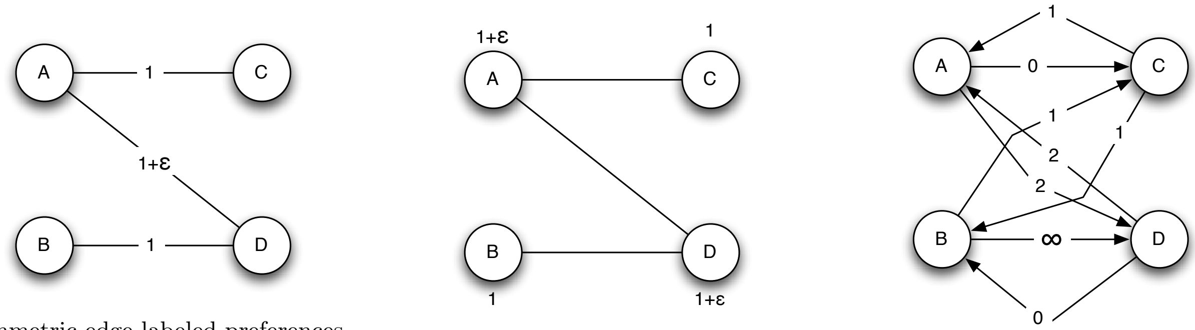
In addition to these, one can also vary the distributions from which actual utility values are sampled. We focus on presenting results from experiments with exponential and uniform distributions. The results we obtained for other distributions, as well as for other graph structures (e.g., non-bipartite, small-world, preferential attachment), were not significantly different.

4. THE PRICE OF ANARCHY

In general, the price of anarchy is the ratio between the social utility of the (worst) equilibrium outcome of a game and the maximum social utility possible in that game. The usual definition relates the largest social welfare achievable to the social welfare of the worst Nash equilibrium. In the context of matching, we have to move from the concept of Nash equilibrium to the concept of stable equilibrium described above, because stable outcomes are determined by the possibility of pairwise deviations rather than individual deviations.

The price of anarchy can vary widely depending on the problem instance and the preference structure. As an example, Figure 1 illustrates some cases where the stable matching is highly socially suboptimal (the price of anarchy is high) in the three different preference settings for two-sided matching described in Section 3. On the positive side, below we present price of anarchy bounds for the three models we consider.

Observation 1. *In symmetric edge-labeled graphs, the*



Symmetric edge-labeled preferences

Vertex-labeled preferences

Asymmetric edge-labeled preferences

Figure 1: Worst-case realizations of the price of anarchy in different models. In each case the socially optimal matching is $\{(A, C), (B, D)\}$ but the only stable matching pairs A and D.

social utility of any stable matching is at least one-half of the social utility of the optimum matching.

In other words, this observation says exactly that the price of anarchy is at most 2. Notice that the socially optimal matching is simply the maximum-weight matching in this model. The above observation is a special case of Theorem 1 (proved in Section 5), but it can also be seen to follow from two facts: (1) Any stable matching can be returned by an algorithm that examines edges greedily by magnitude, adding them to the matching if the vertices involved have not yet been matched (the particular stable matching produced depends on the procedure for breaking ties between equal-weighted edges), and (2) Any greedy solution to the maximum weighted matching problem is within a factor of two of the optimal solution. Note that this argument holds generally, even for non-bipartite graphs. Figure 1(a) provides an example of a graph where this bound is achieved, showing that the bound of 2 on the price of anarchy is tight.

Observation 2. *In vertex labeled graphs the social utility of any stable matching is at least one-half of the social utility of the optimum matching.*

This is a consequence of Theorem 2 (see Section 5 for further discussion). Again, Figure 1(b) provides an example of a graph where this bound is achieved.

Observation 3. *In asymmetric edge-labeled graphs, the social utility of the stable matching can be arbitrarily bad compared with the socially optimal matching.*

Consider the case in Figure 1(c) – the utility received by agent B from being matched with Agent D is arbitrarily high, but the pair is not part of the stable matching, so the loss in utility can be unbounded. Again this argument holds for non-bipartite graphs as well.

These are worst-case constructions. A natural question is what the price of anarchy is like in realistic graphs with different distributions over utilities. We examined several different distributions of utilities within the three models described above, and also considered different graph structures in order to get a sense of the potential practical implications of these price of anarchy results. We used random distributions of the utility values on random bipartite (and later non-bipartite) graphs of the different types described above, and computed both the maximum-weighted stable matching (the socially optimal matching) and a stable matching using the Gale-Shapley algorithm (in all cases considered here, except one described in more detail below, the proposing side does not affect the outcome in expectation because preference distributions are symmetric).

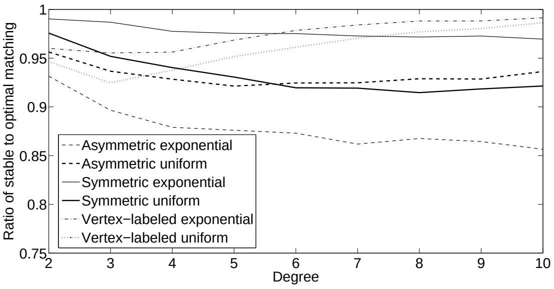


Figure 2: Average ratio of the realized stable matching to the maximum weighted matching in three different preference models when utilities are sampled at random from exponential and uniform distributions with the same mean (0.5: the rate parameter is 2 for the exponential and the support of the uniform is $[0, 1]$). Reported values are averaged over 200 runs. There are 100 agents on each side of the matching market in all cases. The X axis shows the degree of each node. Note that the ratio is very high, almost never dropping below 85%, even in individual runs.

Figure 2 shows that when utilities are randomly distributed according to two common distributions (exponential and uniform, although this result seems to be robust across many different distributions), the social loss due to stability is not particularly high in any of the three models we describe above. This is not surprising for vertex labeled graphs – since any person in the matching will contribute the same to the total utility regardless of whom they are matched with (for example, every perfect matching is socially optimal). As the average degree of each vertex increases, the number of agents getting matched increases, and the ratio quickly reaches 1, because all stable matchings become perfect at some point. However, the result is considerably more surprising for the other two cases, particularly for asymmetric edge-labeled preferences. The only case in which the ratio goes below 0.9 is for exponentially distributed utilities with asymmetric edge-labeled preferences (the ratio stops declin-

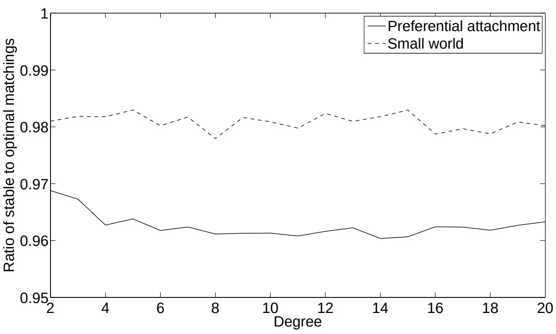


Figure 3: Average ratio of the realized stable matching to the maximum weighted matching with two different non-bipartite graph structures: (1) small world networks and (2) preferential attachment networks of different average degree, both with 100 nodes. Utilities are sampled independently from an exponential distribution with mean 0.5. Results are averaged over 200 runs.

ing significantly beyond degree 10). For asymmetric edge labeled graphs, it makes sense that the ratio declines as the degree of the graph gets larger, because it becomes possible to construct matchings that are socially much better. Our experiments show that the value of the optimal matching grows quickly (since it has more options available), while the value of stable matching grows slowly (since it is hampered by the stability constraint). The actual high percentage is quite surprising given that in theory, the ratio could be arbitrarily bad. The uniform distribution ratios are generally higher than those for the exponential distribution because the uniform distribution enforces a compression in the range of high utilities by capping utilities at 1.

Figure 3 shows that the high ratio is not an accident of using random bipartite graphs. In non-bipartite graphs that are known for their power in modeling social and engineering systems, namely preferential attachment networks [8] and small-world networks on a lattice [26], the results are similar, with the computed stable matching achieving, on average, above 95% of the value of the socially optimal matching. This result also holds in lattice networks and in networks defined in Euclidean space where the utility of a matching for any pair is the inverse of the distance between them.

Thus it appears that in random graphs, stable matchings attain a very high proportion of the maximum social utility. There are however some preference structures for which this does not hold. Consider a case where the utilities received by one side of the market are much higher than utilities received by the other side. In addition, suppose that the side with lower utilities is more powerful, and is therefore able to choose the stable matching optimal for those on that side of the market (these situations could correspond to many in real life – for example, employers are more powerful than employees). This power structure is implemented by running the Gale-Shapley algorithm with the more powerful side being the side that proposes, which results in the best stable matching for the proposing side. In this case the ratio of utilities can be substantially lower, as seen in Figure 4. In other words, if we only care about the welfare of one side of the market, there can exist stable matchings much worse than the optimal ones (although still much better than the theoretical bound of one-half).

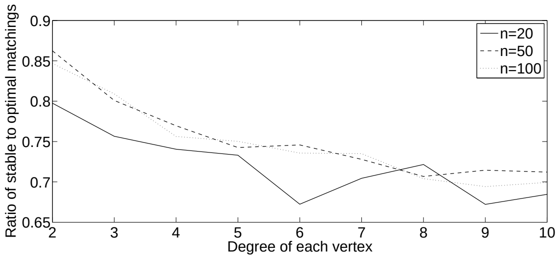


Figure 4: Average ratio of the realized stable matching to the maximum weighted matching when the utilities received by those on the less "powerful" side of the market are 10000 times as high as those received by those on the more powerful side, but the stable matching is the one optimal for the more powerful side. Results are averaged over 200 runs. Utilities are exponentially distributed.

When anarchy is good.

The price of anarchy is not the only important measure. Our experiments so far reveal that the price of anarchy is lower for vertex labeled graphs, especially as the degree grows. This is mostly because any perfect matching is socially optimal. As more and more vertices get included in the matching, we get closer and closer to the socially optimal matching. But this is essentially a case of scarce resources, and no synergies – the average utility received by everyone in a perfect matching is the value of the average vertex – there is no chance to make everyone better off because some pairs work better together or like each other more. If preferences were more heterogeneous, there would be more such synergies that could be exploited. In order to explore this further, we experiment with varying the level of homogeneity in preferences by making preferences a convex combination of vertex-labeled and asymmetric edge-labeled preferences, while holding the average value constant. In this case the value received by u from matching with v is given by $\lambda w(v) + (1 - \lambda)z$ where both $w(v)$ and z are sampled from exponential distributions with mean 0.5, but $w(v)$ is an intrinsic feature of the node v which is the same for any u that is connected to v, while z is idiosyncratic (independently sampled for each u that is connected to v). Then λ represents the degree of homogeneity of preferences. Figure 5 shows that, while the ratio of stable-to-optimal utilities goes up dramatically as preferences approach pure homogeneity, this is accompanied by a decline in average utility received by each individual. This indicates that having some heterogeneity in preferences is a good thing for society: even if it leads to a higher price of anarchy, everyone is better off than they would be in a lower price-of-anarchy society.

5. IMPROVING SOCIAL OUTCOMES

In this section, we consider how to improve the quality of stable matchings. We consider, both theoretically and in simulation, the addition of a switching cost to the mechanism so that an agent would have to pay in order to deviate from the current matching. We find that it is possible to improve the quality of social outcomes substantially by mak-

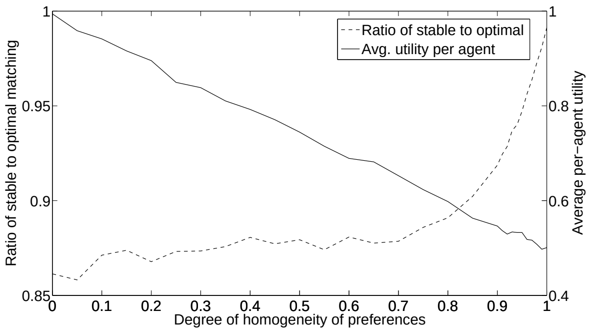


Figure 5: The ratio of the realized stable matching to the maximum weighted matching (going up from left to right, left Y axis) and the average utility received by each agent (going down from left to right, right Y axis) as a function of the degree of homogeneity of preferences (0 being completely heterogeneous, i.e. asymmetric edge-labeled, and 1 being completely homogeneous, i.e. vertex-labeled). The graphs are bipartite, containing 100 nodes on each side, and the degree of each vertex is 10. The average utility of any edge remains 0.5 for each setting. Results are averaged over 200 runs.

ing only small changes to the incentives of the agents, and thus without drastically changing the nature of the matching market. Note that in the cases considered in this section, there is no change in preferences of the sort discussed immediately above, so the price of anarchy is actually a good proxy for social (dis)utility.

5.1 Approximate Stability and Switching Costs

An approximate equilibrium is a solution where no agent gains more than a small factor in utility by deviating. In the case of matching, we consider the following notion of approximately-stable matching.

Definition 1. A matching is called *α-stable* if there does not exist a pair of agents not matched with each other who would both increase their utility by a factor of more than α by switching to each other.

If $\alpha = 1$, then this is exactly a stable matching. An α-stable matching also corresponds to a stable solution if we assume that switching has a cost. In other words, in the presence of switching costs, the set of stable matchings is simply the set of α-stable matchings without switching costs.

In this section we are concerned with understanding how increasing α improves the quality of stable matchings. We are specifically concerned with the *price of stability* [6], which is the ratio of the utility of the *best* stable matching relative to the optimum matching. Much recent work in network design [7] and routing [10, 24] has considered the price of stability in various contexts. The price of stability is especially important from the point of view of a mechanism designer with limited power, since it can compute the best stable solution and suggest it to the agents, who would implement this solution since it is stable. Therefore, the price of stability captures the problem of optimization subject to the stability constraint.

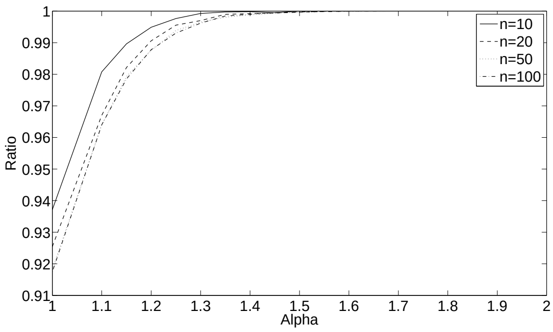


Figure 6: Ratio of the social utilities of best α-stable and socially optimal matchings as a function of α when the matchings are constructed according to our algorithm in symmetric edge-labeled graphs. The dramatic increase between $\alpha = 1$ and $\alpha = 1.1$ shows that introducing even small switching costs has the potential to produce significant social benefits. Preferences were sampled uniformly at random on $[0, 1]$.

Below we present various theoretical bounds, showing that for symmetric edge-labeled graphs, there always exists an α-stable matching with utility of at least $\frac{\alpha}{2}$OPT (where OPT is the value of the optimum matching), and that in vertex-labeled graphs, there always exists an α-stable matching with utility at least $\frac{\alpha}{1+\alpha}$OPT. We provide a constructive algorithm for finding such an α-stable matching. This shows that by increasing α, we can implement much better stable solutions than for $\alpha = 1$, and decrease the price of stability. Our empirical results using this algorithm show an even more dramatic improvement than predicted by the theoretical bounds. For example, Figure 6 shows that for $\alpha = 1.1$ we already obtain a tremendous improvement in the quality of stable matching, essentially obtaining stable matchings that are as good as a matching with maximum utility. This means that adding a switching cost as small as five or ten percent can make an enormous difference in the quality of stable matchings. In many situations, it is reasonable to believe that a central controller can compute a good α-stable matching, assign agents to that matching, and only allow them to deviate on payment of the switching cost.

5.2 Edge-labeled Graphs

For edge-labeled graphs, we prove below that in the presence of switching costs of a factor α, the price of anarchy is at most 2α, but the price of stability is at most $2/\alpha$. This means that as we increase α, there begin to be stable matchings that are worse, but there always exists a stable matching that is close to optimal. For $\alpha = 1$, these bounds coincide, giving us the result that all stable matchings are within a factor of 2 from the maximum weight matching. For $\alpha = 2$, this gives us the easily verifiable fact that the optimum matching is 2-stable.

THEOREM 1. *Let OPT be the value of the socially optimal matching. In any undirected edge-labeled graph, there exists an α-stable matching whose social utility is at least $\frac{\alpha}{2}OPT$. Furthermore, the social utility of any α-stable matching is* *at least* $\frac{1}{2\alpha}OPT$.

Proof. Denote by $w(M)$ the weight of a matching M. First, notice that the socially optimal matching is simply the maximum weight matching in this model, since the social welfare of a matching is exactly twice its weight. Let OPT denote the weight of the maximum weight matching, and prove that the weight of α-stable matchings obeys the lower bounds mentioned in the theorem statement. We first prove that for every $\alpha \geq 1$, every α-Stable Matching in G is of weight *at least* $\frac{OPT}{2\alpha}$.

Let M be an α-stable matching in G, and M^* be a maximum-weight matching in G. Let $e_1 = (u, v)$ be an arbitrary edge in $M^* \setminus M$. Since M is an α-stable matching, there must be either an edge $e_2 = (u, w_1) \in M$ or an edge $e_3 = (v, w_2) \in M$ such that $w(e_1) \leq \alpha w(e_2)$ or $w(e_1) \leq \alpha w(e_3)$ (if neither were true, then u and v could match to each other and gain more than a factor of α in utility). Therefore for every edge e in M^*, either $e \in M$, or there is an edge e' of M sharing a node with e such that $w(e) \leq \alpha w(e')$. Since at most two edges of M^* can share a node with the same edge e' of M (because M^* is a matching), this means that if we sum the above inequalities, we obtain $w(M^*) \leq 2\alpha \cdot w(M)$, as desired.

We now prove that there always exists an α-stable matching M such that $w(M) \geq \frac{\alpha}{2} w(M^*)$ by giving an algorithm for finding such a matching:

```
Set $M = M^*$
Sort the edges of $G$ in order of decreasing weight.
For each edge $e = (v_1, v_2) \in G$ in this order:
  Let $e_1, e_2$ be edges to which $v_1, v_2$ are incident in $M$, respectively (if they exist)
  If $\frac{w(e)}{\alpha}$ is greater than both $w(e_1)$ and $w(e_2)$:
    Remove $e_1$ and $e_2$ from $M$.
    Add $e$ to $M$.
  End If
Loop
```

This algorithm considers all edges in the graph in order of decreasing weight, and if the two nodes in the edge can gain a factor of α utility by deviating to this edge, then we let them. If an edge e_1 does not exist, then for the new edge e to be added to the matching, all we need is that $\frac{w(e)}{\alpha} > w(e_2)$. Call the edge $e = (v_1, v_2)$ in the algorithm as the edge being *currently examined*. To prove correctness, we must show two facts:

(i) The algorithm results in an α-Stable Matching.

(ii) The resulting matching is of weight at least $\frac{w(M^*)\alpha}{2}$.

To begin the proof of (i), notice that M is a matching. This is simply because whenever we add an edge (u, v) to M, we also remove the edges incident to the nodes u and v. Since we start with a matching M^*, we know that M is a matching at every point in the algorithm.

By Lemma 1, we know that if an edge $e = (u, v)$ is in the matching M immediately after it is examined, then it will not be removed from M later. Notice also that if edge $e = (u, v)$ is *not* in the matching M after it is examined, then it will never be added to M later in the course of the algorithm, because the algorithm only adds edges to the matching at the time that it is examining them. Therefore, the final matching M consists exactly of edges that are kept in M at the time the algorithm examines them.

To show that the returned matching is α-stable, suppose to the contrary that there is an instability in the final matching M, i.e., an edge $e_1 = (u, v) \notin M$ such that $w(e_1) > \alpha w(e_2)$ and $w(e_1) > \alpha w(e_3)$, where e_2 and e_3 are the edges of M incident to u and v (which may not exist). Since e_1 is not in the final matching M, it could not have been included in the matching when it was examined. This implies that at this time there was an edge $e' \in M$ incident to (without loss of generality) u, with $w(e_1) \leq \alpha w(e')$. This edge e' cannot still be in the matching M at the end of the algorithm's execution, since otherwise e_1 would not form an instability. Therefore, the algorithm must have removed edge e' at a later point. The only reason why edge e' would be removed is if an edge e'' were added to the matching, with $w(e'') > \alpha w(e') \geq w(e_1)$. Since the algorithm considers the edges in order of decreasing weight, however, this edge e'' could only have been added before the algorithm examined edge e_1, and so we have a contradiction.

We now prove (ii). At each examination in the algorithm, one of two things can occur. The trivial case is that no edge is formed so no change occurs in M. The other case, in which a new edge e is added to the matching, adds an edge of weight $w(e)$ to M while removing at most $2 \cdot \frac{w(e)}{\alpha}$. The ratio of the new edge weight to the old edges weight is therefore $\frac{w(e)}{2 \cdot \frac{w(e)}{\alpha}} = \frac{\alpha}{2}$. By Lemma 1, once an edge is added to the matching M by the algorithm, it is never removed again, so the total weight of the final matching M is at least $\frac{\alpha}{2} w(M^*)$, as desired, completing the proof of Theorem 1. □

Lemma 1. *If an edge $e = (u, v)$ is in the matching M immediately after it is examined, then it will not be removed from M later.*

Proof. Suppose to the contrary that $e = (u, v) \in M$ directly after it is examined, but is no longer in M at a later point. Without loss of generality, assume that e was removed from M because some edge $e' = (u, w)$ was added. For this to occur, it must be that $w(e') > \alpha w(e)$. But since $\alpha \geq 1$, and the algorithm examines the edges in order of decreasing weight, then this addition of edge e' could only have occurred before the algorithm examined e, a contradiction. □

5.3 Vertex Labeled Graphs

For vertex labeled graphs, results similar to Theorem 1 hold: the price of anarchy is at most $1 + \alpha$ and the price of stability is at most $(1 + \alpha)/\alpha$. For $\alpha = 1$ this gives us the observation in Section 4 (notice that while it is easy to show a correspondence between stable matchings for edge-labeled and vertex-labeled graphs, the same does not hold for α-stable matchings).

Theorem 2. *Let OPT be the value of the maximum-weight perfect matching. In any vertex-labeled graph, there exists an α-stable matching whose social utility is at least $\frac{\alpha}{1+\alpha}OPT$. Furthermore, the social utility of any α-stable matching is at least $\frac{1}{1+\alpha}OPT$.*

Proof. For an edge $e = (u, v)$, define $w(e) = w(u) + w(v)$, and denote by $w(M)$ the weight of a matching M. First, notice that the socially optimal matching is simply the maximum weight matching in this model, since the social welfare of a matching is exactly equal to its weight. Therefore, we let OPT denote the weight of the maximum weight matching, and prove that the weight of α-stable matchings obeys the stated lower bounds. We first prove that for every $\alpha \geq 1$, every α-Stable Matching in G is of weight *at least* $\frac{1}{1+\alpha}$OPT.

The proof is similar to the proof of Theorem 1, but some extra details are necessary. Let M be an α-stable matching in G, and M^* be a maximum-weight matching in G. Let $e_1 = (u,v)$ be an arbitrary edge in $M^* \setminus M$. Since M is α-stable, there must be either an edge $e_2 = (u, w_1) \in M$ or an edge $e_3 = (v, w_2) \in M$ such that $w(u) \leq \alpha w(w_2)$ or $w(v) \leq \alpha w(w_1)$ (otherwise u and v could match to each other and gain more than a factor of α in utility). We call this edge a "witness" for e_1, since it prevents e_1 from being an instability for the α-stable matching M. Therefore for every edge e_1 in M^*, either $e_1 \in M$, or there is such a witness edge e of M sharing a node with e_1.

The structure of vertex labeled graphs allows us to obtain better bounds than we could for edge-labeled graphs. We prove that M has high weight by comparing the weight of edges in M^* with the edges that act as their witnesses. As in Theorem 1, if the edge is also in M, then the weight does not change. Consider the case where $e = (u,v) \in M$ acts as a witness for two edges $e_u = (u,v')$ and $e_v = (v,u')$ of M^*. In this case, $w(e_u)+w(e_v) = w(u)+w(v)+w(u')+w(v') \leq w(u)+w(v)+\alpha w(u)+\alpha w(v) = (1+\alpha)w(e)$. If e only acts as a witness for e_u, then we know that $w(e_u) = w(u)+w(v') \leq w(u)+\alpha w(v) \leq \alpha w(e)$. The edge e cannot act as a witness for more than two edges, since M^* is a matching, and so e can only be touching two edges of M^*. Therefore, in the worst case $w(M^*) \leq (1+\alpha)w(M)$, as desired.

To prove the other statement in the theorem, we construct an α-stable matching with weight at least $\frac{\alpha}{1+\alpha}w(M^*)$. We use the same algorithm as in the proof of Theorem 1, but we must sort the edges using a more complicated ordering than simply by the sum of their node weights. Specifically, we define a new notion of edge weight by $\rho(e) = w(u) \cdot w(v)$ for an edge $e = (u,v)$. We then run the algorithm in the proof of Theorem 1, with the weight of an edge e being $\rho(e)$. In the rest of this proof, we use the same notation as in the proof of Theorem 1. We must show that:

(i) This algorithm results in an α-Stable Matching.

(ii) The resulting matching is of weight at least $\frac{w(M^*)\alpha}{1+\alpha}$.

Consider the definition of what it means for a node u to be α-stable in a vertex labeled graph. It states that if $(u,v) \in M$, then there cannot be an edge (u,v') with $w(v') > \alpha w(v)$. This is equivalent to stating that $w(u)w(v') > \alpha w(u)w(v)$, which is the same as saying that $\rho(u,v') > \alpha\rho(u,v)$. Therefore, a vertex labeled graph is α-stable exactly when the same edge labeled graph is α-stable, with edge weights being $\rho(e)$. Since we know that our algorithm produces an α-stable matching for edge labeled graphs with edge weights $\rho(e)$, then it must also produce an α-stable matching for our vertex labeled graph.

We now prove (ii). At each examination in the algorithm, one of two things can occur. The trivial case is that no edge is formed so no change occurs in M. The other case, in which a connection is formed, adds an edge $e = (u,v)$ instead of edges $e_u = (u,v'), e_v = (v,u')$ such that $\rho(e) > \alpha\rho(e_u)$ and $\rho(e) > \alpha\rho(e_v)$. By our definition of ρ, this implies that $w(v) > \alpha w(v')$ and $w(u) > \alpha w(u')$. The ratio of the new edge weight to the old edge weight is $(w(u)+w(v))/(w(u)+w(v)+w(u')+w(v')) \geq 1/(1+\frac{1}{\alpha}) = \frac{\alpha}{1+\alpha}$. By Lemma 1, once an edge is added to the matching M by the algorithm, it is never removed again, so the total weight of the final matching M is at least $\frac{\alpha}{1+\alpha}w(M^*)$. This concludes the proof of Theorem 2. □

6. CONVERGENCE TO STABILITY

While many good algorithms exist for computing stable matchings (Gale-Shapley being the most standard), we would like to consider more natural dynamics for forming stable matchings. Such dynamics are likely to occur in practice if there were no central planner to compute a matching for the agents, and if instead the agents tried to do what was best for themselves in a decentralized manner. In such cases, how likely is it that realistic algorithms yield stable outcomes?

We study the convergence properties of a particular decentralized partner-switching algorithm in which the vertices on a graph are sorted randomly and then the following algorithm is repeated until convergence: for each vertex, in the sorted order, find the best partner that vertex can be matched with. The vertex can be matched with a partner if an edge connects them and the deviation is utility-increasing for both the vertex and its new partner. The best partner is the one of these that yields maximum utility for this vertex. Add this new pair to the matching, removing any pairs that this vertex or its new partner were previously connected to.

This algorithm captures the intuitive notion that, in a society of agents, pairs take turns deviating from the current matching if it is in their interest to do so. We call each iteration through all agents a *phase.* Notice that instead of iterating through all the agents in a fixed order, we could instead pick random agents to deviate at every step, as in [5]. None of our results change significantly in this case.

Theorem 3. *This algorithm converges to a stable matching after at most n phases in vertex-labeled and symmetric edge-labeled graphs.*

Proof. First we show the result for vertex-labeled graphs. Let S be the set of nodes on one side of the matching with maximum weight w (there can be many such nodes, since the weights of nodes may not be distinct). Define v to be the node from S such that after the first phase of the algorithm, v has a partner u with the largest weight $w(u)$.

If out of all the neighbors of v, u has the largest weight, then the matching between v and u will always be stable from this point until the end of the algorithm's execution, since v and u are each others' highest weighted neighbors. This means we can simply think of v and u as removed from the graph, since they will not affect the algorithm in future phases. Otherwise, we can assume that there exists a neighbor $u' \neq u$ of v with $w(u') > w(u)$. When we consider v in phase 1, v would like to connect to u' over u. The only reason why u' would not be matched with v is if it were already matched to a node $v' \in S$. But this cannot be by our choice of node v.

Therefore, we know that during each phase, we can remove a pair of nodes (v,u) and their incident edges from the graph (since this pair will always be stable and matched during the rest of the algorithm). After at most n phases, the resulting matching will be stable (where n can be the size of the smaller side of the matching).

The proof of convergence for symmetric edge-labeled graphs is similar, and is essentially the same as in [5]. Consider an edge (u,v) of maximum weight in the graph. After the first

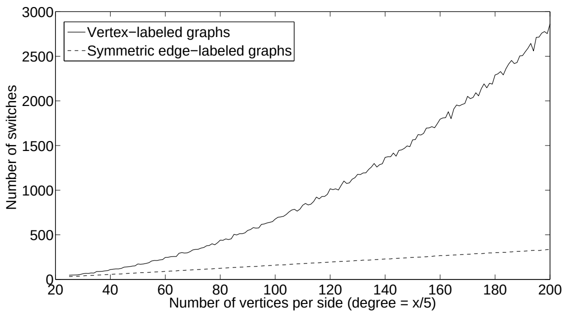


Figure 7: Average number of switches the greedy algorithm makes before the resulting matching is stable for vertex-labeled and symmetric edge-labeled graphs. Note the quadratic growth for vertex-labeled and linear growth for edge-labeled graphs. Utilities are sampled independently from an exponential distribution with mean 0.5. Results are averaged over 200 runs.

phase, u will be matched with v (because u prefers v to all its other neighbors and v prefers u to its other neighbors), and we can remove v and u from the graph. The rest of the argument is the same as above. □

The above theorem says that the simple decentralized algorithm described above converges to a stable matching in time $O(n^2)$, since each phase takes linear time. Notice, however, that if instead of switching to its best partner, the agents simply switched to a random improving partner, the same argument would guarantee convergence to a stable matching in an expected time of $O(n^2 d)$, where d is the maximum degree of the graph.

In practice (see Figure 7), on random utility distributions similar to those described in previous section, the convergence time for vertex-labeled graphs does indeed appear to be quadratic, but it is interesting to see that the convergence time for symmetric edge-labeled graphs seems to be linear. We conjecture that the algorithm converges in expected linear time for these graphs, perhaps because good edges for one node are in expectation also good for the other node in the edge, because of the symmetric preferences.

Asymmetric edge-labeled.

While Theorem 3 guarantees convergence for the vertex-labeled and symmetric edge-labeled utilities, this is not the case for asymmetric edge-labeled graphs. Unfortunately, in this case there are easy examples where this algorithm can cycle. In our experiments, however, for small n (the number of nodes on each side) this algorithm converged to a stable matching on all but a small percentage of cases, showing that the bad scenarios are not "typical." As n gets larger, this algorithm converges more and more rarely (approximately 2% less for every additional node), with convergence essentially non-existent for $n = 70$.

7. DISCUSSION

This paper explores the prices of anarchy and of stability in matching markets. We demonstrate that even though the price of anarchy can theoretically be high, when utilities are randomly sampled, the loss in social welfare from strategic behavior is limited. This result encompasses many different graph and preference structures, and is experimentally robust. While the downside is limited, even this downside can be alleviated: a significant improvement in social welfare can be obtained by suggesting a good matching and requiring agents to pay small switching costs to deviate. We show this theoretically using an algorithm for constructing *approximately stable* matchings, and then demonstrate that the algorithm is effective in experiments. We also show that simple greedy partner switching algorithms can converge quickly to stable matchings in some graph structures. From a practical perspective, future work should include understanding real-world utility distributions and how they affect the social outcomes of matching as compared to random distributions of utilities. From a mechanism design perspective, it would be interesting to explore whether agents would choose to participate in a switching-cost based, designer-suggested matching mechanism.

Acknowledgements

The authors would like to thank an anonymous reviewer for his comments on how to improve the paper.

8. REFERENCES

[1] A. Abdulkadiroglu, P. Pathak, and A. Roth. The New York City High School Match. *American Economic Review*, 95(2):364–367, 2005.

[2] A. Abdulkadiroglu, P. Pathak, and A. Roth. Strategy-proofness versus Efficiency in Matching with Indifferences: Redesigning the NYC High School Match. *American Economic Review*, 2009. To appear.

[3] A. Abdulkadiroglu, P. Pathak, A. Roth, and T. Sonmez. The Boston Public School Match. *American Economic Review Papers and Proceedings*, 95(2):368–371, 2005.

[4] D. Abraham, A. Blum, and T. Sandholm. Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 295–304. ACM Press New York, NY, USA, 2007.

[5] H. Ackermann, P. Goldberg, V. Mirrokni, H. Roglin, and B. Vocking. Uncoordinated two-sided markets. In *Proceedings of the 9th ACM Conference on Electronic Commerce (EC)*, 2008.

[6] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *Proc. FOCS*, pages 295–304, 2004.

[7] E. Anshelevich, A. Dasgupta, E. Tardos, and T. Wexler. Near-optimal network design with selfish agents. In *Proceedings STOC*, pages 511–520. ACM Press New York, NY, USA, 2003.

[8] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.

[9] G. Becker. A Treatise On The Family. *Family Process*, 22(1):127–127, 1983.

[10] G. Christodoulou and E. Koutsoupias. On the Price of Anarchy and Stability of Correlated Equilibria of Linear Congestion Games. *Lecture Notes In Computer Science*, 3669:59, 2005.

[11] S. Das and E. Kamenica. Two-sided bandits and the dating market. In *Proc. IJCAI*, pages 947–952, Edinburgh, UK, August 2005.

[12] M. Dawande, S. Kumar, V. Mookerjee, and C. Sriskandarajah. Maximum Commonality Problems: Applications and Analysis. *Management Science*, 54(1):194, 2008.

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

[14] M. Goemans, L. Li, V. Mirrokni, and M. Thottan. Market sharing games applied to content distribution in ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(5):1020–1033, 2006.

[15] N. Immorlica and M. Mahdian. Marriage, Honesty, and Stability. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005.

[16] R. Irving, P. Leather, and D. Gusfield. An efficient algorithm for the ”optimal” stable marriage. *Journal of the ACM (JACM)*, 34(3):532–543, 1987.

[17] B. Jovanovic. Job Matching and the Theory of Turnover. *The Journal of Political Economy*, 87(5):972, 1979.

[18] V. Mirrokni. *Approximation Algorithms for Distributed and Selfish Agents.* PhD thesis, Massachusetts Institute Of Technology, 2005.

[19] S. Mongell and A. Roth. Sorority Rush as a Two-Sided Matching Mechanism. *American Economic Review*, 81(3):441–464, 1991.

[20] A. Roth and J. Vande Vate. Random Paths to Stability in Two-Sided Matching. *Econometrica*, 58(6):1475–1480, 1990.

[21] A. E. Roth and E. Peranson. The redesign of the matching market for American physicians: Some engineering aspects of economic design. *American Economic Review*, 89(4):748–780, 1999.

[22] A. E. Roth and M. Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis.* Econometric Society Monograph Series. Cambridge University Press, Cambridge, UK, 1990.

[23] A. E. Roth and X. Xing. Jumping the gun: Imperfections and institutions related to the timing of market transactions. *The American Economic Review*, 84(4):992–1044, 1994.

[24] A. Schulz and N. Moses. On the performance of user equilibria in traffic networks. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 86–87, 2003.

[25] R. Thaler and C. Sunstein. *Nudge.* Yale University Press, 2008.

[26] D. Watts and S. Strogatz. Collective dynamics of ’small-world’ networks. *Nature*, 393(6684):440–442, 1998.