Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik

An interactive system for finding stable student-seminar matchings

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Science (B. Sc.)

eingereicht von:	Aaron Oertel
geboren am:	18.02.1997
geboren in:	Dinslaken
Gutachter/innen:	Dr. Alexander van der Grinten Prof. Dr. Henning Meyerhenke

eingereicht am: .... verteidigt am: ....

Contents

1 Abstract

Work in progress

2 Introduction

2.1 Motivation

Many universities require students to enroll in a seminar in order to obtain their degree. Usually the students have a choice between a handful of different seminars, however there are capacity constraints that make it hard to give all students their first choice. Let's consider the following example: 100 students have to be assigned to one of 6 seminars, where each of the seminars has a capacity of 20. The students express their preferences by supplying a strict, but incomplete preference list of the 6 seminars. The goal for the school's administration now is to assign as many students as possible to a seminar of their choice.
What makes this problem harder is that the students preferences aren't necessarily equally distributed. Oftentimes a majority of students prefers one seminar in which case conflicts exist in choosing which students get their first choice. At the same time, it can happen that students go unmatched if their preference lists are short and full of seminars that have reached full capacity already.
The goal of this thesis is to formally model the aforementioned problem, while presenting different algorithms for finding possible matchings and finally to evaluate these algorithms using certain metrics that make sense in the domain of matching, such as stability, rank-maximality, popularity and pareto-optimality. Additionally, an interactive system will be developed, which allows a school's administration to find a student-seminar matching using one of the presented algorithms.

2.2 Formal Definition

The problem of assigning students to seminars can be described as a many-to-one matching, with a set of students $S := \{s_1, s_2, ..., s_n\}$ and a set of seminars $T := \{t_1, t_2, ..., t_m\}$. Every student $s_i \in S$ provides a strict preference order over a subset of T, and every seminar $t_j \in T$ has a capacity of c_j students. The goal is to find a matching $M : S \to T$, that assings students to their preferred seminars while respecting the capacity of the seminars. That means that for every seminar t_j the following is satisfied: $|M(t_j)| \leq c_j$. Using this definition we can describe the problem as a many-to-one matching with one-sided, incomplete preferences.

2.3 Similar Problems

There are many similar problems, which differ in what the preference lists look like and how many entities of the first set are matched to how many entities of the second set. Studying these problems reveals some important insights into matching markets.

2.3.1 Stable Marriage Problem

The stable marriage problem was one of the first matching problems to be researched[1] and consequently motivated a lot more research in the domain of matching.

The problem is stated as follows: A set of m men and n women shall be matched one-to-one, where each men and women provide a complete strict-preference order over the agents of the other set. The deferred acceptance algorithm presented in Gale and Shapley's paper[1] finds a stable, complete matching in polynomial time. Stability is defined as follows: given a women w and any man that she was not matched to m, w does not prefer m more than her current partner, and m does not prefer w more than his current partner.
TODO: present algorithm here?
The algorithm can be executed in two ways:

1. the men have priority, by proposing to women, where the woman has to accept the proposal iff it improves her situation.

2. the women have priority and propose to men. This case is analogue to the first one

It has been shown that all possible executions of the algorithm with men as proposers yield the same stable matching. That matching is men-optimal, which means that every man has the best partner that he can have in any stable matching.[2] Additionally, with men proposing the produced matchings has also been shown to be women-pessimal, meaning that every woman is matched to the worst partner that she can have in any matching.[2]
While the scenario of matching women to men like described (hopefully) doesn't occur in the real world, Gale and Shapley's paper inspired a lot more approaches to other practical problems:

2.3.2 Hospitals/Residents Problem

A few years after the publication of Gale and Shapley's original paper, it was found that the deferred acceptance algorithm was essentially the same algorithm used by the National Resident Matching Program (NRMP) in the United States to match graduating medical students to residency positions in hospitals.[2] As a matter of fact, Gale and Shapley's paper described an algorithm for the so-called "college admissions problem"[1], which is essentially the same problem. Just like in the stable-marriage problem, there is a solution that be hospital-optimal or resident-pessimal.
The problem can be described as finding a one-to-many matching with two-sided incomplete, but strict preferences. Essentially, a hospital can offer multiple spots and both parties can mark entities from the other set as unacceptable by not including them in their preference list.[3]
In reality, the problem is a bit more complex, as it permits couples of residents to submit preferences together. It has been shown that a stable solution does not always exist and that finding one if it exists or showing that it doesn't exist is NP-complete.[4]
The revised algorithm used by the NRMP uses findings about stability and simple matching markets to find a good approximation, while minimizing opportunities for strategic manipulation, which was indeed possible before.[5]

2.3.3 House allocation problem

Many economists and game theorists[6] have studied variants of the house allocation (HA) problem, where a set of indivisible items H needs to be divided among a set A of applicants. Each one of those applicants may have a strict preference order over a subset of H. Formally this means that an instance I of the problem consists of two disjoint sets, where $H := \{h_1, h_2, ..., h_n\}$ is the set of houses and $A := \{a_1, a_2, ..., a_m\}$ is the set of applicants. Each applicant $a_i \in A$ ranks a subset of the houses in H using a preference list. The houses on the other hand do not have any preferences over applicants. A matching or assignment M is a subset of $A \times h$, so that for every applicant a_i, the house $M(a_i)$ is indeed on the applicants preference list.[7]
The house allocation problem is essentially an alias for a matching problem on bipartite graphs with one-sided preferences. There are many applications including matching clients to servers, professors to offices and also students to seminars. For the latter some generalisations have to be made to the problem. Specifically one seminar should now be matched to more than one student, whereas the houses in HA are matched to one and only one applicant. In the literature that variant of the problem is also referred to as the Capacitated House Allocation Problem, denoted by CHA.[8] (TODO)
The next section will explore performance indicators and algorithms for finding such matchings in the context of student-seminar matchings.

2.3.4 Assignment Problem

The problem of matching students to seminars can also be defined as an assignment problem. The goal of the assignment problem is to find a minimum weight perfect matching in a bipartite graph. In this case the goal of the problem is, given the set of students A, seminars B, and a cost function $W : A \times B \to I\!R$, to find a map $M : A \to B$, which minimzes the following objective function: $\sum_{a \in A} W(a, M(a))$.
One of the first algorithms used for solving this problem was the Hungarian algorithm by Munkres, which finds a minimum-weight matching in polynomial time.[9]
Alternatively the problem can be transformed into an instance of the minimum-cost flow problem to determine a minimum weight matching. Section 3.4 (TODO) will further investigate using this algorithm for the problem of student-seminar matchings.

2.4 Outline

Now that the problem has been formalized and similar problems have been presented, I will present several different algorithmic approaches to find possible matchings. To evaluate these matchings I will also present commonly used metrics like:

- Rank-maximality (maximum number of students ranked to their first priority)
- Maximum cardinality
- Pareto optimality
- Popularity

- Profile-based optimality

Using these metrics I will choose one algorithm for implementation and analyze it's space and time complexities as well as evaluating it's performance using the aforementioned metrics against the other approaches. Next, I will describe the interactive web system developed for using the algorithm and lastly extend the problem to a two-sided problem.

3 Optimality criteria for matchings

Looking at the previously presented problems, it is clear that there are different objective functions or optimality criteria for such matchings. For instance, in the stable marriage problem, matchings are primarily judged by the stability characteristic. However such a characteristic does not make sense in the case of the student-seminar problem, since stability assumes both-sided preferences. For that very reason, different criteria have to be used for judging the quality of a matching.
Intuitively, when thinking about matching students to seminars, it would be desirable to match as many students as possible, as well as matching the students to their first choice, etc. To formalize these requirements, a few criteria have been discussed in academia, which will be helpful for comparing different approaches.

3.1 Maximum cardinality

The goal of the maximum cardinality problem is finding a matching M on a graph $G = (V = (X, Y), E)$, so that $|M|$ is maximal.[10] Consequently, maximum cardinality as an optimality criteria means that a matching is ideal if the number of students that are matched is maximized among all possible matchings. It should be noted here, that the student's preferences are not considered when computing the maximum cardinality matching. As a matter of fact, it is possible that multiple matchings (TODO: source) of the same cardinality exist, where one of the matchings could be better in the sense of a different optimality criteria. Therefore, maximum cardinality may be desirable, but should be used in conjunction with a different criteria, as it does not consider student preferences.

3.2 Pareto Optimality

Pareto optimality or Pareto efficiency is a commonly used term in economics to describe the state of resource allocations. Intuitively an allocation, or in our case a matching, is Pareto optimal, iff no improvement can be made to a single individual, without worsening the situation for other individuals. For example a matching, in which two students s_1, s_2 would be better off by swapping their seminars is not Pareto optimal. In order to more formally define Pareto-optimality we must first define student preferences more clearly:
Given two matchings M, M' and a student $s \in S$, the student s prefers M' over M in the following cases:

1. s is matched in M' and unmatched in M, or
2. s is matched in both M' and M, however prefers $M'(a)$ over $M(a)$

Using this defintion we now define Pareto optimality as follows: Given an instance I of a matching problem and its set of possible matchings $\mathcal{M}$, we define a relation $\succ$ on $\mathcal{M}$, where given two matchings $M, M' \in \mathcal{M}$ the following holds true: $M' \succ M$ if no students prefers M to M', but some student prefers M' over M. Consequently a matching $M' \in \mathcal{M}$ is called Pareto-optimal iff there exists no other matching $M \in \mathcal{M}$, such that $M' \succ M$.[8] A Pareto-optimal matching always exists for any instance of a matching problem and can be efficiently computed using one of various algorithms. A simple greedy algorithm uses the random serial-dictatorship mechanism to draw each agent in random order and let's them select their most-preferred, available item from their preference list.[11, 12] However the greedy algorithm does not always compute a Pareto optimal matching of maximum cardinality, which would be desirable in student-seminar matching.[13]

3.3 Popularity

In the context of matching with one-sided preference lists, using an optimality criteria (TODO) like Stability, that is based on the preferences of both parties, doesn't apply. Instead a commonly used criteria is a matching M's Popularity, which indicates that more students prefer that matching M over any other possible matching.[14] Given the definition of the student-seminar matching problem, we can formally define Popularity as follows: Let $P(M', M)$ be the set of students who prefer M' over M. A matching M' is said to be more popular than M, denoted by $M' \succ M$, iff $|P(M', M)| > |P(M, M')|$. That concludes that a matching M' is popular, iff there is no other matching M that is more popular than M', i.e. $M' \succ M$.[15, 16] Due to that definition, this criteria is often also referred to as the majority assignment.[17]
Using this definition, we can see that every popular matching also is a pareto-optimal matching. Given a popular matching M' and any matching M for an instance I of the problem, M' is pareto-optimal if $P(M, M') = 0$ and $P(M', M) \geq 1$, which obviously implies that M' is popular as well.[15]
It is important to note that a popular matching's cardinality could be smaller than the maximum cardinality, meaning that in the case of student-seminar matchings, a group of students could be left unassigned in favor of the majority of the students having a match that they prefer. (TODO: sentence) Additionally, a popular matching, unlike a pareto-optimal matching, does not always exist. To illustrate this, let's consider the following instance: $S = \{s_1, s_2, s_3\}$, $T = \{t_1, t_2, t_3\}$, where each student has the same preferences, being $t_1 < t_2 < t_3$. Given the following matchings, we can confirm that there exists no popular matching for the given instance:

1. $M_1 = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}$
2. $M_2 = \{(s_1, t_3), (s_2, t_1), (s_3, t_2)\}$
3. $M_3 = \{(s_1, t_2), (s_2, t_3), (s_3, t_1)\}$

Obviously M_2 is more popular than M_1, M_3 is more popular than M_2 and M_1 is more popular than M_3.[16]

3.4 Strategy Proofness

TODO: should this be included?

3.5 Profile-based Optimality

Contrary to popularity and Pareto optimality, where the students' satisfaction with a matching is compared, we could also examine the structure of a matching by defining the profile of a matching and comparing it. Intuitively the profile of a matching M is a vector whose ith component indicates the number of students obtaining their ith-choice seminar in M according to their preference list.
Formally, let I be an instance and $\mathcal{M}$ the set of its matchings. Given a matching $M \in \mathcal{M}$ with the set of students S and seminars T, we define the regret $r(M)$ of M as follows: $r(M) = \max\{rank(s_i, t_j) : (s_i, t_j) \in M \, s_i \in S, t_j \in T\}$, where for every match $(s_i, t_j) \in M$, $rank(s_i, t_j)$ is defined as the position of t_j on s_i's preference list. The profile of M is now defined as a vector $\langle p_1, ..., p_r* \rangle$, with $r^* = r(M)$ and for each $k \in [1, r^*]$, the kth component is defined as: $p_k = |\{(s_i, t_j) \in M : rank(s_i, t_j) = k\}|$.[8]
Using the defintion of a matching's profile, it is now possible to define a matching as rank-maximal as follows: A matching M is rank-maximal, if its profile $p(M)$ is lexicographically maximum over all possible matchings in $\mathcal{M}$. That means that the number of students in M who are matched to their first choice is maximum among all $M' \in \mathcal{M}$ and taking that into consideration, the number of students who are matched to their 2nd choice is maximum among all matchings, and so on. TODO: greedy rank maximal as well?

3.6 Application to student-seminar matching

Given the problem description of student-seminar matching, it would be desirable to find a matching that has the following properties:

1. Maximum Cardinality: As few students as possible should be left unmatched.
2. Pareto Optimality: A set of students should not feel the need to swap their match to improve their situation.
3. Popularity: The number of students who are satisfied with their matching should be greater than the number of students dissatisfied.
4. Rank Maximality: As many students as possible should be matched to their first choice or if not possible second choice, and so on.

Using these requirements, the next sections will present algorithms, for finding matchings that have some of those properties. One of these will be implemented to be used by the interactive system to find matchings which will try to optimize some of the mentioned metrics. As we have already seen, there does not always exists a popular matching or a pareto-optimal matching that is also agent complete. Additionally, the fact that students can supply incomplete preference lists, can very well lead to matchings that leave a few students unassigned, even if the sum of the capacity of all seminars is greater than the number of students. Given these constraints and observations, we will see that there is no such thing as an ideal matching for all instances based on the criteria we have presented. However, it will be possible to find matchings that will make the majority of the students satisfied.

4 Algorithmic approaches

In the previous section, we have discussed several optimality criteria that apply to the problem of matching students to seminars. This chapter will present algorithms for computing matchings that fulfill some of those criteria, as well as evaluating them against each other. The goal of this evaluation is choosing the "ideal" algorithm for implementation. We will see that each of the algorithms has some draw-backs, which might make them undesirable for the student-seminar problem.

4.1 Greedy with serial dictatorship

One of the simplest algorithms for the student-seminar matching problem is a greedy approach, that iterates over the set of students and assigns each of the students to their most preferred seminar that still has some capacity left. In contrast to Gale & Shapley's deferred acceptance algorithm for the stable marriage problem, this algorithm does not tentatively match students once they make their selection, but makes a final assignment. Due to that this algorithm finds a matching in $\mathcal{O}(n)$ time with n being the number of students. This mechanism of letting students successively pick their highest available preference in order is known as serial dictatorship.[18] In detail the algorithm looks like this:

Algorithm 1 Greedy serial dictatorship matching

```
Require: set of Students with preferences $S$, set of Seminars $T$
  $M = \emptyset$
  for each $s \in \mathcal{S}$ do
      $t \Leftarrow$ highest ranked, available seminar on s's preference list
      if $t \neq null$ then
          $M = M \cup (s, t)$
      end if
  end for
  return M
```

Even though this algorithm is very simple and fast, it has some desirable properties including one of the optimality criteria defined before:

4.1.1 Properties of the computed matching

Since the order, in which the students get to pick their match is pre-defined, we can easily show that the algorithm always produces a pareto-optimal matching.

Theorem 1. *A greedy algorithm that uses serial dictatorship always produces a pareto-optimal matching.*

Proof. Let M be the matching produced by the algorithm. We assume that there exists a matching N that pareto-dominates M. Now let $s \in S$ be the first student who prefers his match in N over M. Since s prefers $N(s)$ over $M(s)$, the seminar $N(s)$ must have been unavailable when he made his pick. That means that another student $s' \in S$ exists, who picked $N(s)$ before s could. However, we required that s was matched to a better seminar in N, which means that s' gets a worse match in N. This is a contradiction, so N cannot pareto-dominate M. □

TODO: citation

Another important property of the algorithm is, that it is strategy-proof. This means that no student has an incentive to lie about his true preference in order to receive a better match. That holds true, because the first student who gets matched, can only pick his preferred, remaining seminar, the second student can only pick his preferred, remaining seminar and so on.

4.1.2 Drawbacks

When looking at the algorithm it is clear that it has a strict preference order over students, specified by the order in which students are matched in the for loop. Additionally, the algorithm makes no effort to match all students. If it's a student's turn to pick his match, and none of the seminars on his preference lists are free, that student will not be matched at all. This problem gets worse, when we consider that our problem statement allows for incomplete preference lists, which increases the chances of having a high number of unmatched students. To illustrate this let's consider the following example in Table 1: In this example, each seminar only has a capacity of 1 and both students have seminar t_1 as their first preference. If the algorithm first gives s_1 a chance to pick and then s_2, s_1 will be matched to t_1, making t_1 full and not allowing s_2 to be matched. On the other hand, matching s_2 to t_1 first and then matching s_1 to t_1 also yields a pareto optimal matching, however in this case all the students are matched.

Agent	Pref list	Seminar	Capacity
s_1	t_1, t_2	t_1	1
s_2	t_1	t_2	1

Table 1: Instance where Serial Dictatorship admits no max cardinality matching

To address the other problem of preference over students, a simple approach is using the random serial dictatorship mechanism, which instead creates a random order of students as the pick order. This approach is still not fair in the sense, that the first student in that order has a better chance at receiving his top priority seminar, than all other students, however, any student has the chance to be the first one to make a pick.

4.2 Pareto Optimal Maximal Matchings for CHA

4.3 Assignment Problem

4.4 Transformation as an instance of min-weight matching

In order to find a rank-maximal matching, a matching of minimum weight can be computed, because that matching also satisfies the rank-maximality property. Since the problem of matching students to seminars can simply be described as finding a matching on a bipartite graph, we can construct such a graph, given the set of students, preference lists and seminars and apply graph algorithms that find such matchings for us.
Indeed, we can simply transform the student-seminar matching problem into an instance of the minimum-cost flow problem. The goal of this algorithm will be to send $|S|$ units of flow through the network, while minimizing the cost of the flow, which is indicated by a seminar's rank on the student's preference lists.

4.4.1 Input Transformation

The problem of matching students as seminars can be given as a bipartite graph $G = (V = (S, T), E)$, where S is the set of students and T the set of seminars. The set of edges E is defined as follows: $E := \{(s, t) \mid s \in S \wedge t \in T \wedge \text{t is on the prefernce list of s}\}$. Additionally, a weight function $W : E \to \mathbb{N}$ is specified, which maps each edge to the position of the seminar on the student's preference list. In order to transform this bipartite graph into an input for the minimum-cost flow problem, a flow network has to be constructed using the bipartite graph. (TODO: define flow network?)

To transform the bipartite graph into a flow network, we first add a source and sink vertex to the graph. Then, we add weights, capacities and edges from and to the source and sink. Specificially, we create an edge from the source to each of the student vertices with a capacity of 1 and a cost of 0. These edges indicate that a student can only be assigned once. Next, for every student we re-use the edges from the bipartite graph G, where each edge $e \in E$ is assigned a capacity of 1 and a cost of $W(e)$. The capacity, again indicates that a student can only be assigned once and the weight indicates the position of the seminar on the student's preference list. Last, one edge is added from each seminar $s \in S$ to the sink with a capacity of $C(s)$ and a cost of 0.

4.4.2 Drawbacks

The matching computed using this method has the minimum weight property, and therefore is rank-maximal, however a big drawback is that the algorithm is not strategy-proof: Using this algorithm, students are encouraged to provide short preference lists in order to get matched to a higher ranked seminar. The algorithm tries to match every student, which means that students with a list of just one seminar will be prioitized over students, who also prefer that seminar but also supply other preferences. This problem could lead to having all students supply singleton preference lists, which also makes it harder to find perfect matchings. Therefore the algorithm does not encourage the students to supply preference lists that reflect their true preferences, which is a desirable property that has been considered for other matching problems like the hospital-residents-problem.[2]

4.5 Popular Matchings in CHA

5 Methodology

test

6 Implementation

test

7 Evaluation

test

8 Discussion

test

9 Conclusion

References

[1] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

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

[3] A. Roth, "The theory and practice of market design," Nobel Prize in Economics documents 2012-5, Nobel Prize Committee, 2012.

[4] E. Ronn, "Np-complete stable matching problems," *Journal of Algorithms*, vol. 11, no. 2, pp. 285 – 304, 1990.

[5] A. E. Roth and E. Peranson, "The redesign of the matching market for american physicians: Some engineering aspects of economic design," Working Paper 6963, National Bureau of Economic Research, February 1999.

[6] S. P. Fekete, M. Skutella, and G. J. Woeginger, "The complexity of economic equilibria for house allocation markets," *Information Processing Letters*, vol. 88, no. 5, pp. 219 – 223, 2003.

[7] C. Thiam Soon Sng, "Efficient algorithms for bipartite matching problems with preferences," 05 2019.

[8] D. F. Manlove, *Algorithmics of Matching Under Preferences*, vol. 2 of *Series on Theoretical Computer Science.* WorldScientific, 2013.

[9] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[10] D. B. West, *Introduction to Graph Theory (2nd Edition).* Pearson, 2000.

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

[12] A. Abdulkadiroglu and T. Sonmez, "Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems," *Econometrica*, vol. 66, pp. 689–702, May 1998.

[13] D. J. Abraham, K. Cechlárová, D. F. Manlove, and K. Mehlhorn, "Pareto optimality in house allocation problems," in *Proceedings of the 16th International Conference on Algorithms and Computation*, ISAAC'05, (Berlin, Heidelberg), pp. 1163–1175, Springer-Verlag, 2005.

[14] D. Manlove and C. Sng, "Popular matchings in the capacitated house allocation problem," September 2006.

[15] B. Klaus, D. F. Manlove, and F. Rossi, “Matching under preferences,” in *Handbook of Computational Social Choice* (F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, eds.), pp. 333–355, Cambridge ; New York: Cambridge University Press, April 2016.

[16] D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn, “Popular matchings,” in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’05, (Philadelphia, PA, USA), pp. 424–432, Society for Industrial and Applied Mathematics, 2005.

[17] P. Gärdenfors, “Match making: Assignments based on bilateral preferences,” *Behavioral Science*, vol. 20, no. 3, pp. 166–173, 1975.

[18] M. Manea, “Serial dictatorship and pareto optimality,” *Games and Economic Behavior*, vol. 61, no. 2, pp. 316 – 330, 2007.

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Berlin, den May 26, 2019 ....