VIENNA UNIVERSITY OF TECHNOLOGY

105.625 PR ADVANCED ECONOMICS PROJECT

Double Sided Matching

9702170, Karin Leithner
0725439, Florin Bogdan Balint
1025735, Clemens Proyer
1027143, Mattias Haberbusch
1027433, Thomas Solich

April 29, 2016

Contents

Introduction

Double sided matching is about finding a match between two sets of elements. The first algorithm which should solve this problem was described by David Gale and Lloyd S. Shapley in 1962 [1]. The focus will lie on the examination of this algorithm and the creation of a simulation model for it, which can be applied to different economic issues. The selected economic issues will be presented in the following chapters.

Motivation and Problem Description

In many daily live situations there are hidden double sided matching problems. For example students usually have the opportunity to apply to a number of schools. In this case students and schools have to be matched together. Both parties have preferences which need to be taken into consideration. In our everyday life we usually don't have the time to find an appropriate solution.

In some cases these problems even go by without solving them. In other cases however, it is crucial to solve them, e.g. if we consider medical school graduates and open positions in hospitals. Due to multiple reasons (e.g. different preferences, high number of elements in the two sets) this problem cannot be solved intuitively or in linear time.

Aim and Objectives

The aim of this paper is to develop a simulation model prototype which can solve similar problems, like the ones discussed above. With this model we could answer questions like:

- Are there "stable" solutions for this kind of problem?
- How can it be applied to all prospective students and universities in Austria?
- Is there also a solution for the current labour supply and demand in the European Union?

Structure

In the second chapter a theoretical overview is provided regarding the double sided matching problem and what stability means in this context. The focus lies on the Gale and Shapley Algorithm, basic information about the game theory and the relationship between them.

The third chapter describes the prototype of the simulation model, which implements the previously discussed theory into practice. As a simulation environment NetLogo will be used.

The fourth chapter deals with the examination of data from Austrian universities and their integration into the prototype. As a result, a solution for this double sided matching problem will be provided.

Chapter five deals with the labour market in the European Union, trying to match the labour demand and labour supply.

Finally the most important findings and results are summarized and compared in the last chapter. Additionally there is a short outlook into further research.

Theory

In this chapter the necessary theoretical background for the realization of this project is discussed.

The Double Sided Matching Problem

The double sided matching problem describes a problem, which has two sets of elements. These sets are disjoint and may be equal in size, but this is not mandatory. The purpose of the double sided matching algorithm is to find a solution, so every element of the first set has a corresponding counterpart in the other set. This can be best described in an example: Assume that the elements are men and women and that there are only heterosexuals preferences for the individuals. The two sets are now male and female. The double sided matching algorithm shall create couples, which consists of a man and woman.

The solutions of the matchings can have different properties, which might be very important (e.g. stability). The output of the algorithm of men and women is not allowed to be unstable, i.e. there are two men α and β who are assigned to women A and B although β prefers A to B and A prefers α to β. If this does not occur, the assignment is called *stable*. If there is more than one stable solution the *optimal* one can be of particular interest.
Optimality means that a solution has the best outcome for an individual or set. In this case optimality can be achieved from the perspective of the first set of elements or from the second set. There can also be a trade-off from the perspective of the both sets, it this case optimality is achieved by weighting the expected value of both sets. These properties have been described in the paper by Gale and Shapley [1].
Generally in economics this is also known as pareto efficiency [2, p. 46]. Pareto efficiency is a characteristic for a solution where any change which would improve one property on one hand, but will worsen another one on the other hand.

Gale and Shapley Algorithm

In the paper of Gale and Shapley [1] an algorithm was proposed for solving the double sided matching problem. Lets assume there are two sets of elements: one contains men and the other one women. The size of the two sets is equal, the

preferences are heterosexual and there are no polygamies (i.e. one element is only allowed to be matched with one element from the other set). The aim of the algorithm is to find a stable way of marrying men and women. Each men and women has their own preference for their marriage partner.

First Iteration

In the first iteration each man proposes to the woman who he ranked first. Afterwards a woman has a list of men that proposed to her. This list might be zero, if not she rejects every men, except her highest preferred man on this list. If a man is not rejected he and the woman create a temporary couple. All man who are not in a couple are single.

Second Iteration

In the second interaction each man who is single, proposes to his highest ranked woman, who he didn't already proposed to. Each woman now chooses the man which she ranks the highest out of her received proposals. If a man is not rejected he and the woman create a temporary couple (Note: after the second iteration some men, who have been in a couple from the first iteration might be now single).

Further Iterations

The third iteration is the same as the second one. This continues until no men is single any more. If this condition is satisfied a stable solution is found for the double sided matching problem. The stability of the solution of this algorithm has been proven by contradiction [1].

Relevance to Game Theory

General

"Game theory is about what happens when people - or genes, or nations - interact" [3, p. 1]. An important aspect for participating parties is to anticipate how the opposite party will react on certain actions. Mathematics shall help to analyse, understand and estimate outcomes of such games. Depending on the information participants have, they decide how to act basing on rules contained in

their strategy. Game theory is widely applied in economics. Companies use game theory to estimate e.g. reactions of competitors or behaviour of employees. The major advantages of game theory are its precision and its universal applicability to all kind of games. [3, pp. 1-3]

In addition there are some assumptions which are made during preparation and execution by every individual who is interacting with another one:

- Well-specified choices
- Well-defined end-state
- Specified pay-off
- Perfect knowledge
- Rationality

However, these perfect preconditions will never be met in real scenarios. In the context of double sided matching, it is important to have well-specified choices defined. Nevertheless, there might be chances to get into a nearly worst-case scenario instead. But furthermore, as a consequence of these listed assumptions above, there are two different kind of games:

- Static ones
- Dynamic ones

The two items can be combined with the following kind of knowledge

- Incomplete
- Complete (there is no private information)

Depending on the nature of the game, whether it is static or dynamic and the knowledge is complete or incomplete four different combinations can be made. For a better understanding, following examples only consist of two players (every combination can be used for a higher amount of players too):

- Static and (in)complete information
  As of complete information, the players action set can be presented in an N x M matrix. Therefore all information (also private) is known to every player and each player is able to easily eliminate the pay-offs of the other players. The pay-off is the outcome of a chosen action of a player which depends on the state or previous actions of the game. The pay-off can change dynamically while the game is in progress. As of incomplete information players are forced to be uncertain about the other player's actions and pay-offs. Besides that, there are two basic steps

  1. Player 1 and 2 are choosing actions out of a set, which were predefined by them before.
  2. After their actions, they receive their pay-offs.

- Dynamic and complete information
  When basic assumptions are made on complete information, players will create a strategy to ensure to get the highest return of each action. Although the strategy might be changed during the game after each turn. Incomplete information leads to communication of uninformed parties (i.e. closed private information) which might end up to under- or overestimate the others pay-offs. In general there will be a Nash Equilibrium between those parties. This equilibrium is created by players because they choose a strategy which do not depend on the other players. There are three basic steps:

  1. Player 1 chooses an action out of his/her predefined set.
  2. Player 2 observes player 1's action and chooses an action based on his/her observation.
  3. After their actions, they receive their pay-offs.

The previously described theory is based on information of the paper "An introduction to applicable game theory" [4].

Example for the Application of Game Theory

Game theory is also very popular in the field of economics. Many companies primarily apply game theory to estimate the reactions of their competitors. In the last auction for broadband internet frequencies in Germany, all major providers hired game theorists out of two reasons. Firstly, they wanted to win the auction

for the desired frequency and secondly, they also did not want to overpay for the frequency. The bidding behaviour of a provider indicated the interest rate for a frequency. As a consequence, other providers retained their bids for parts of this frequency so the interested provider did not need to overpay for it. The positive result for the providers was, that the 700-MHz frequency parts were sold for the minimum bidding price. [5]

Double Sided Matching Problem and Game Theory

Like described in chapter 2, the goal of the three main problems, which will be simulated in the course of this project, is to bring different parties together. In the original marriage problem discussed by Shapley, the goal was to match men and women so the overall situation is stable. This idea can also be applied to other areas like the labour market or university applications. In each of these three problems the parties need to have a strategy how to react to moves of the counterpart. If, for example, a university offers a place to a student and this is not the student's favourite one, he/she has to decide whether to accept the place (to be on the safe side) or still hope to get accepted from the favourite university (and as a consequence accept to be on the waiting list there). Not only the student needs to have a strategy how to deal with these situations but also universities have to incorporate different reactions of students into their application process (e.g. how many students to invite) [1].
The double sided matching problem is a dynamic game. The players of the two sets have incomplete information. The algorithm uses the information of both sets, therefore it has complete information.

An important aspect of the game or matching is that the rules have to be clear, which was mentioned by Roth and Sotomayor. The way agents are matched to each other influences the analysis of the problem. A possible rule might be that individuals like a student and university are only brought together if both parties agree to the matching. Other norms of a game might contain the way an individual proposes to another one or whether there exists a moderating individual [6, p. 492].

Double Sided Matching Problem and Optimal Stopping

The double sided matching problem also relates to the optimal stopping problem. The optimal stopping problem concerns itself with choosing the optimal time to stop an undertaken action and at the same time to minimize the expected costs and maximize the profit. A concrete example in this case is if a men is trying to find a women to spend the rest of his life with, when should he stop looking?

A few algorithms as a possible solution for the optimal stopping problem have been presented by Christian & Griffiths 2016 [7]. For the previously mentioned example of partner looking, one algorithm proposes to invest 37% of the time in looking and immediately after to stop looking and take the next best offer. 37% is the provably optimal solution [7, p. 2]. This algorithm is recommended in the case of *no-information games* (e.g. in the case of partner searching, it is hard to quantify the information, because it is hard to compare people to each other. There is no concrete criteria, one can only assume that one person is better or more suitable than another person, but not by how much) [7, p. 18].

Another possible solution for the previously mentioned problem is to set a threshold from the beginning and to take the first option that exceeds it. Christian & Griffiths 2016 [7] recommended this approach in the case of *full-information games*. A suitable example for this case is that of a house selling: the value of the house is known and there is also information about the state of the market, which allows to predict a range of offers. In this case a threshold is set and the first offer that exceeds this threshold can be accepted. This example does not include waiting costs. In the case of waiting costs, they also have to be taken into account and it is recommended to decline an offer if the chance of a better offer multiplied by how much better it is compensate the waiting costs.

Finally Christian & Griffiths 2016 [7, pp. 28-30] recommend to always stop looking at a certain point.

Prototype

For the simulation model NetLogo was used as a simulation tool whereas the random data was generated with R. In the following, the data model, the R script, the GUI and code from Netlogo will be described.

Data model

According to the problem description we defined a generic data model for every individual. This data model is valid for the disco example, the university and labour market example, and contains the following attributes:

- id: unique object identifier
- name: object name (e.g. male1 or female1)
- maxMatchesInt: integer value representing the maximum number of matchings for this individual (e.g. in the disco example each person is maximally matched with one other person)
- sideInt: integer helper variable assigning an individual to one of the two participant groups
- partnerList: list of identifiers of the partners, ordered according to the preference of an individual
- rankList: list of ranks (values between 1 to 0) for the partners from the partnerList; the list is ordered from highest to lowest preferences. If the value is equal to 1 this partner is a perfect match.
- hasProposedToList: list of identifiers representing the individuals to which an individual has proposed
- gotProposedByList: list of identifiers representing the individuals from which an individual received proposals. The gotProposedByList is only valid for one iteration.
- tmpMatchList: list containing the identifiers of individuals to whom a temporary match was established

- activeFlag: boolean value stating if an individual is matched or not respectively if the individual gave up (they already proposed to all individuals from the partnerList but only received rejections)

Data generation

In order to generate data for the disco example, the statistical computing language R[1] has been used. This generates a CSV list, which is then read from NetLogo. The script accepts the following parameters:

- seed: integer value, which is used as a seed by the number generator. The default value is 123.
- numberOfMen: integer value representing the number of men which should be created. The default value is 10.
- numberOfWomen: integer value representing the number of women which should be generated. The default value is 10.
- pickyLower: float value between 0 and 1 representing the lower bound which will be used in order to generate a number that represents how picky a person is. The value 0 means accepts every possible match, the value 1 excepts none.
- pickyUpper: float value between 0 and 1 representing the upper bound which will be used in order to generate a number that represents how picky a person is. The value 0 means excepts every possible match, the value 1 excepts none.

The script can be called as follows from the command line:

```
Rscript initDisco.R 123 10 10 0 0
```

The script generates a CSV which looks as follows:

```
"";"id";"name";"maxMatchesInt";"sideInt";"partnerList";"rankList"
```

[1] Available at https://www.r-project.org/, accessed April 28, 2016

```
"1";1;"male1";1;1;"13#18#14#17#16#11#20#19#12#15";"0.96#0.95#0.9#0.68#0.57#0.45#0.33#0
...
"11";11;"female1";1;2;"3#9#5#4#10#8#2#1#6#7";"0.92#0.82#0.7#0.67#0.48#0.41#0.35#0.25#0
```

GUI

One of the main goals in the design phase was to keep the GUI simple and clear. The GUI was structured in two main parts - the control and the information section.

Figure 3.1: The control section for the disco example

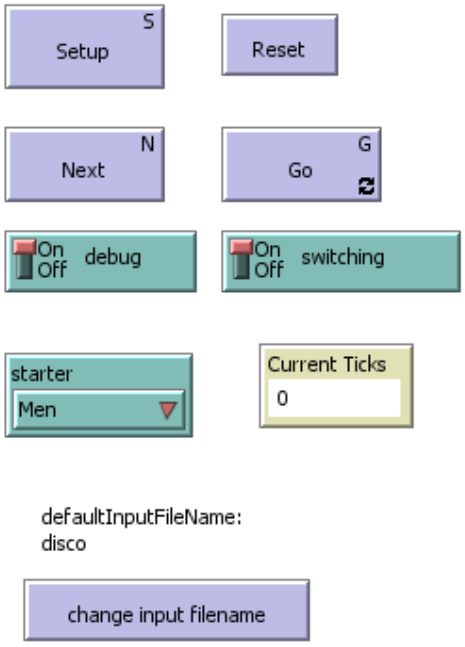

Figure 3.1 shows the control section in the GUI. With the buttons Setup and Reset the simulation model can be initialized or reset after a simulation run. The button Next advises the program to perform one simulation step, if Go is activated the simulation continues a termination criterion is met. In the disco example the main termination criterion is if every every proposing person is already matched to another person. Another criterion for termination is if the proposing person

already proposed to every person in his partnerList but got rejected every time. The debug-switch enables or disables debug messages which are printed during the simulation run. The speed of a simulation run can be increased if the debug-switch is set to Off. The second switch, switching, controls, if the two parties propose to each other alternatively (On) or if only one party proposes to the other one (Off). With the drop-down list starter the user can determine which party starts proposing. The reporting box Current Ticks shows the current number of simulation steps. The button change input filename raises a dialogue box where the user can enter the filename of an input file which shall be used in the simulation. Together with the number of ticks the filename will also be used as the name for the CSV export file.

Figure 3.2: The information section for the disco example

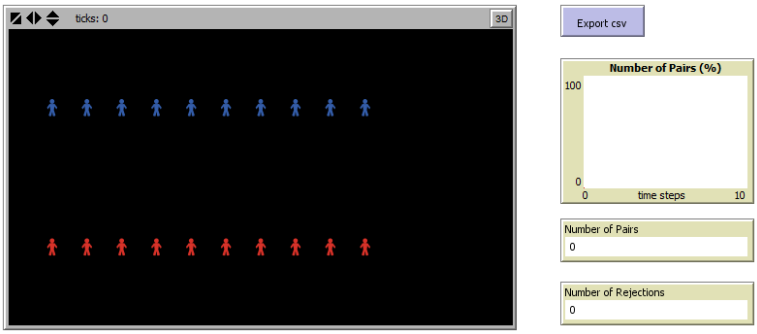

Figure 3.2 shows the information section of the disco example GUI. All created humans are shown in the world area. Men are coloured blue and women are red. On setup of the simulation the proposing site is shown in the top row. As soon as two people are matched their colour changes to green and a link between the two is established. If a couple is divorced the link is removed and the colour changes back to the original one. The plot named Number of Pairs (%) illustrates at each point in time how many of the possible couples exist. If the simulation contains 10 men and women and currently 4 couples exist, the plot shows that 40% are matched. The reporting box Number of Pairs shows the absolute number of couples. The other reporting box, Number of Rejections, indicates how many couples were divorced over time. With the button Export csv the current properties of all humans are exported to a csv file. The filename is a combination of the input filename and the current number of ticks.

Netlogo code

Prototype Evaluation: College Applications in Austria

Subsection

Text...

Prototype Evaluation: Labour Market in the European Union

Subsection

Text...

Summary

Subsection

Text...

References

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

[2] Nicholas Barr. *Economics of the Welfare State*. Oxford University Press, 2012.

[3] Colin Camerer. *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press, 2003.

[4] Robert Gibbons. An introduction to applicable game theory. *The Journal of Economic Perspectives*, 11:127–149, 1997.

[5] Benedikt Fuest. Frequenzauktion droht zum mini-geschaeft zu werden, 2015.

[6] Alvin E Roth and Marilda A Oliveira Sotomayor. *Two-sided matching: A study in game-theoretic modeling and analysis*. Number 18. Cambridge University Press, 1992.

[7] Brian Christian and Tom Griffiths. *Algorithms to Live By: The Computer Science of Human Decisions*. Henry Holt and Co., 2016.