A Smart Market for Study-Room Allocation at Reykjavik University

Research report

Autumn 2011

Anna Sigga Lúðvíksdóttir and Sævar Jónasson

Instructors:
Henning Arnór Úlfarsson
Jón Þór Sturluson
External referee:
Magnús Már Halldórsson

B.Sc. final project
School of Computer Science

Abstract. Reykjavik University has eight study rooms for approximately 3000 undergraduate students. These study rooms are very popular and most often in use. Utilization of the study rooms could be better because there are often only 1 or 2 students in a room, that has space for 6 students, for hours or even a whole day. A better arrangement would be preferable for the study rooms. We propose an auction to allocate the rooms. With an auction, it can be prevented that students who get a room can keep it as long as they want. Those who want to stay alone in the rooms have to pay high prices for the room while groups that are big, pay lower price per member and can thus bid more often. In this project it will be examined how an auction can be useful to allocate the study rooms. A website was made to implement the auction.

Útdráttur. Í Háskólanum í Reykjavík eru átta hópavinnuherbergi fyrir u.þ.b. 3000 nemendur. Herbergin eru mjög vinsæl og eru oftast í notkun. Nýtingin á herbergjunum gæti verið betri þar sem herbergi sem rúma 6 nemendur eru oft frátekin af 1 eða 2 nemendum. Betra fyrirkomulag væri ákjósanlegt. Við mælum með uppboði til að úthluta herbergjunum. Með uppboði er hægt að forðast það að nemendur sem ná herbergjunum geti haldið þeim eins lengi og þau vilja. Þeir sem vilja vera einir í herbergjunum þurfa að borga hátt verð, meðan stórir hópar borga minna á hvern hópmeðlim. Í þessu verkefni verður skoðað hvernig uppboð getur verið gagnlegt til að úthluta hópavinnuherbergjunum. Til að framkvæma prófun á uppboðinu var gerð vefsíða.

Acknowledgements. First we would like to thank our advisors Jón Þór Sturluson for the original idea of the project and Henning Úlfarsson for all his help on the algorithm and time spent on the project with us. We would also like to thank Daníel Brandur Sigurgeirsson for his assistance programming the auction website and Hjalti Magnússon for his help with the typesetting.

Contents

1. Introduction

A combinatorial auction is an auction where people bid on multiple units. If you have many units but don't want to sell them all at once, you rather want to offer one or more units. Then the bidders can bid on the units they want.[3, Ch. 1]

The purpose of this project is to design an auction format for the study rooms in Reykjavik University, were the currency is credits. There is a need for another arrangement than the one in use at the time, to improve utilization and fairness so that all undergraduate students have equal access to the rooms.

In this project there are four study rooms for small groups, some with different attributes and sizes. Some rooms have windows and some have a wall mounted monitor in it. The combinatorial problem here are the time units. The bidder bids on the time units, because we don't want to auction off a whole day, but prefer to auction one time unit. It is possible to bid on the whole day if the bidder can afford it. A single time unit would be one time slot according to the school time table. So each bidder, who can be a student or a group of students, bids on a room and the time units each bidder needs.

2. About smart markets

A combinatorial auction is a type of a smart market which uses mathematical optimization, such as linear, integer programming *etc.* It was used in the early 1980s when S.J. Rassenti, V.L. Smith and R.L. Bulfin designed an auction application to utilize the airport time slots of four major airports.[6, p.402-417] With the emergence of the internet and faster computers, the smart market has great potential. Smart markets are used in many markets *e.g.*, where natural gas and electricity is sold and for allocating access to rail roads.[2]

3. Why auction the study rooms?

An auction is often an advantageous possibility, *e.g.*, if there are difficulties in dividing a product or if the value of the product is unknown[7, Ch.1]. In this project, it is difficult to divide the product. Currently the students that get a room, can keep it as long as they want, leading to unfairness. Often the study rooms are occupied during most of the day, by 1 or 2 students, leading to poor utilization. We hope this can be avoided if groups of students bid on the study rooms and the highest bidding group wins the time period they bid on.

A survey was submitted to all students earlier this semester. Students were asked for their opinion of the current utilization of the study rooms. About 70% of the students who don't use the rooms, say it is because the rooms are always occupied when they need them. There is a need for a different arrangement for the study rooms to increase fairness.

4. Types of auctions

A sealed bid auction is an auction where bidders bid on specific units. They have to decide what they are willing to pay and at the end of the auction the winners are announced. There are many types of these auctions *e.g.*, [7, Ch.10]

- Vickrey auctions, and
- Discriminatory auctions

These auctions are called standard auctions. In both Vicrey auction and discriminatory auction the highest bidder or bidders win. The difference between these two auctions are the pricing rules.[7, Ch.10]

Vickrey auction. A Vickrey auction is a sealed bid auction where the winner pays the highest loosing bid. Bidders submit the amount they are willing to pay for the unit. If there is only a single unit for sale then the Vickrey auction becomes a second-price sealed-bid auction.[7, Ch.10]

Discriminatory auction. A discriminatory auction is a multi-unit auction where the bidder must pay the bid amount if the bidder wins. If there is only a single unit for sale, then a discriminatory auction becomes a first-price auction.[7, Ch.10]

The choice was between Vickrey auction and discriminatory auction. Discriminatory auction would not fit since the bidder pays the bid amount. The auction form chosen was a Vickrey auction. The goal is to get a better utilization of the study rooms and to increase fairness. In Vickrey auction the bidders bid what they are willing to pay, thus the bid amount is the bidders true value of the units [1]. Thus if one bidder bids a large amount and another bidder bids a low amount the winning bidder would not necessarily pay the bid amount. The winning bid was the actual amount that the bidder was willing to pay for the unit. In this project it would not be reasonable for the winner to pay the highest amount.

5. Optimization

To find the winners of the auction or the best solution, we used linear and integer programming. A variable is made for each bid and an objective function is generated, because the objective is to maximize the amount bid on the rooms. Next, constraints need to be generated because there are conditions that need to be fulfilled, like only one group of students can get one room at the same time.

Linear programming. The most common type of models in linear programming (**LP**) is to allocate limited resources in the best possible way. That is used for many different kinds of problems *e.g.*, in production, shipping pattern and agricultural planning *etc.* A mathematical model uses linear programming to find the best solution. Thus *linear programming does not refer to computer programming*, it is more a planning tool to find the optimal solution.[4, Ch.3]

Integer programming. In linear programming the variables can take any positive value, not only integers but also real numbers. There has to be a restriction to integers in the model. It is not possible to divide one room in half on a specific time unit for two or more different groups. Thus it is necessary to use integer programming (**IP**). In this model the variables must be either one or zero *i.e.*, *binary integer progamming* (**BIP**).[4, Ch.11]

Python and PuLP. Python is a simple programming language which is easy to use, with many modules and packages[1]. PuLP is a package for Python with linear programming models. PuLP has built-in solvers to solve the LP problem[2]. The solvers in PuLP solve only LP problems, so it is needed to restrict the LP model to an IP model by restricting the variables to integers, not real numbers[3].

---

[1] http://www.python.org,2011

[2] PuLP 1.4.7 , http://pypi.python.org/pypi/PuLP/1.4.7,2011

[3] Optimisation Concepts, http://packages.python.org/PuLP/main/optimisation_concepts.html,2011

6. The mathematical model

The mathematical model has to ensure that no bidder gets more than one room at the same time. Groups cannot have two or more rooms at the same time either. This can be formulated using integer programming. [3, Ch.3]

Each bidder bids on a room and a set of time units. Let T be the set of all time units and $t \subseteq T$, where t is a subset of time units. Let R be the set of all rooms and $r \in R$, where r is a single room. Let I be the set of all bidders and $i \in I$, where i is a bidder. If $A_{(t,i,r)}$ is the amount, bidder i is willing to pay for room r on time t then $[A_{(t,i,r)}, t, i, r]$ is the bid, bidder i submitted. We assume that $A_{(t,i,r)} \geq 0$. A variable $X(t,i,r)$ is constructed for each bid. Its value is 1 in the solution if the bid is one of the winning bids, otherwise its value is 0.

Equation (1) is the objective function, as we want to maximize the amount bid on the rooms. Next, constraint (2) ensures that no group gets the same room at the same time. Constraint (3) ensures that no group gets two or more rooms at the same time. Finally, constraint (4) ensures that all variables are equal or less than one.[3, Ch.3]

$$\text{(1)} \qquad \text{maximize} \qquad \sum_{i \in I} \sum_{t \subseteq T} \sum_{r \in R} A_{(t,i,r)} \cdot X(t,i,r)$$

$$\text{(2)} \qquad \text{such that} \qquad \sum_{\substack{t \ni j \\ i \in I}} X(t,i,r) \leq 1, \forall r \in R \text{ and } \forall j \in T$$

$$\text{(3)} \qquad \sum_{\substack{t \ni j \\ r \in R}} X(t,i,r) \leq 1, \forall i \in I \text{ and } \forall j \in T$$

$$\text{(4)} \qquad X(t,i,r) \leq 1$$

When the IP-solver has calculated which bids are the winning bids, we need to compute the payment P_i that the bidder i pays. As covered earlier in the report, a Vickrey auction is a second price auction, thus the bidder pays the highest losing bid[7, Ch.10]. The bidder does not necessarily pay the bid amount, since that was the amount the bidder was willing to pay. If A_i is the value i is willing to pay for the unit and V is the objective value, next i's bid is removed and the model executed again to find the objective value $V^{'}$ where i hasn't participated in the auction. So i pays

$$P_i = A_i - (V - V^{'})$$

or the value that would win, if the bidder has not participated in the auction [5].

Let's look at a simple example. There are eight bidders bidding on four rooms. We make a list for each bidder, where the list for a bid is $[A, t, i, r]$.

The bids are

$$\begin{aligned}
\mathrm{B}_1 &= [5, [1,2,3], \mathrm{A}, 1] \\
\mathrm{B}_2 &= [6, [2,3,4,5], \mathrm{B}, 1] \\
\mathrm{B}_3 &= [9, [5,6,7,8], \mathrm{C}, 2] \\
\mathrm{B}_4 &= [8, [7,8,9], \mathrm{D}, 2] \\
\mathrm{B}_5 &= [15, [3,4,5,6], \mathrm{E}, 3] \\
\mathrm{B}_6 &= [17, [3,4,5,6,7], \mathrm{F}, 3] \\
\mathrm{B}_7 &= [10, [8,9,10], \mathrm{G}, 4] \\
\mathrm{B}_8 &= [20, [2,3,4,5,6,7,8], \mathrm{H}, 4]
\end{aligned}$$

Next we formulate this by using the algorithm above. First we set up the object function

$$\begin{aligned}
\text{maximize} \quad & 5 \cdot X_{[1,2,3][\mathrm{A}][1]} + 6 \cdot X_{[2,3,4,5][\mathrm{B}][1]} + 9 \cdot X_{[5,6,7,8][\mathrm{C}][2]} \\
& + 8 \cdot X_{[7,8,9][\mathrm{D}][2]} + 15 \cdot X_{[3,4,5,6][\mathrm{E}][3]} + 17 \cdot X_{[3,4,5,6,7][\mathrm{F}][3]} \\
& + 10 \cdot X_{[8,9,10][\mathrm{G}][4]} + 20 \cdot X_{[2,3,4,5,6,7,8][\mathrm{H}][4]}
\end{aligned}$$

Next we generate the constraints

$$\begin{aligned}
\text{such that} \quad & X_{[1,2,3][\mathrm{A}][1]} + X_{[2,3,4,5][\mathrm{B}][1]} \leq 1 \\
& X_{[5,6,7,8][\mathrm{C}][2]} + X_{[7,8,9][\mathrm{D}][2]} \leq 1 \\
& X_{[3,4,5,6][\mathrm{E}][3]} + X_{[3,4,5,6,7][\mathrm{F}][3]} \leq 1 \\
& X_{[8,9,10][\mathrm{G}][4]} + X_{[2,3,4,5,6,7,8][\mathrm{H}][4]} \leq 1 \\
& X_{[t][i][r]} \leq 1
\end{aligned}$$

By executing the model the winners will be

$$X_{[2,3,4,5][\mathrm{B}][1]} = 1,$$
$$X_{[5,6,7,8][\mathrm{C}][2]} = 1,$$
$$X_{[3,4,5,6,7][\mathrm{F}][3]} = 1,$$
$$X_{[2,3,4,5,6,7,8][\mathrm{H}][4]} = 1.$$

So rooms are assigned to bidders B, C, F, H and the objective value is

$$V = 6 + 9 + 17 + 20 = 52.$$

Next we need to remove each winning bidder from the auction and find out what they have to pay. If bidder B is removed, A, C, F, H would win and the objective value is, $V^{'} = 51$, so B pays

$$P_B = 6 - (52 - 51) = 5.$$

Now C is removed, and bidders B, D, F, H win and the objective value is $V^{'} = 51$, so C pays

$$P_C = 9 - (52 - 51) = 8.$$

Now F is removed, and bidders B, C, E, H win and the objective value is $V^{'} = 50$, so F pays

$$P_F = 17 - (52 - 50) = 15.$$

Finally H is removed, and bidders B, C, F, G win and the objective value is $V^{'} = 50$, so H pays

$$P_H = 20 - (52 - 42) = 10.$$

Now, each bidder pays the bid amount minus the difference on the objective value, or each bidder pays the bid amount minus the effect the bidder had on the auction.[3, Ch.3] If only one bidder bids on the room then the bidder gets it for free. If there is a tie between bidders the winner will be chosen randomly by the algorithm.

Coding. You can see the code in Appendix C. The model is formulated using PuLP, LP-solver for Python which is restricted to an IP model. First the IP-model finds the winners of the auction. Next, a function checks if the winning bids are legal *i.e.*, if the winners have enough balance. If not, that bid is removed and the model executed again. When all the bids are legal, the next step is to find out what each winner has to pay. Thus it's needed to remove each winning bid from the auction and execute the model to find the actual price.

Model execution time. Figure 1 shows how the execution time changes with different number of bids. For each number of bids, random bids were generated and the model executed five times and average execution time calculated.

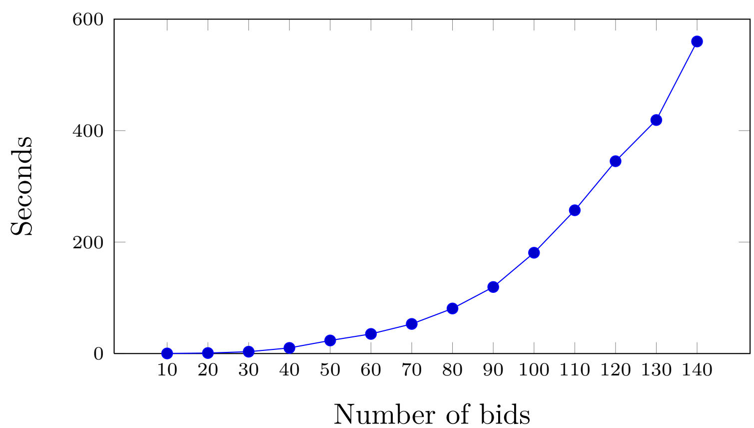


FIGURE 1. Execution time of the model.

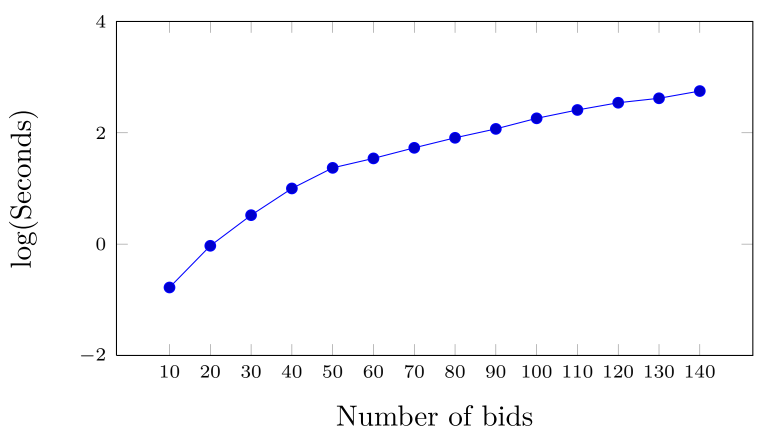


FIGURE 2. Logarithmic function of the execution time.

Like Figure 1 shows, the execution time of the model increases fast. Figure 2 shows that the execution time is not increasing exponentially, although this might change with very large number of bids. In this comparison, a laptop was used. The execution time would probably decrease if a more powerful computer would be used.

7. The auction site

To implement the auction methods a website was made, in order to use it in an experiment. The website had to be as simple as possible, in order to get students more involved in the experiment. The auction site was built with a program called Django which is a web framework that was made by the web department of the Lawrence Journal-World newspaper, Kansas in 2003[4]. Three different programming languages were used to program the website. The main programming language is Python and in the earliest version of Python there is a built-in SQL software library which makes it easy to create databases and store all the information that is submitted from the website. In order to get a simple user interface the HTML programming language was used.

A short guide through the website would be

- Students go to `https://auction.ru.is` and log in with their Reykjavik University user name and password.
- Students create a group and add friends to the group.
- When students use the auction site for the first time, 1000 credits are assigned to their balance.
- Students choose whether to bid on "any room" or a "specific room"
- If the students bid on a "specific room", they can choose between four rooms, like Figure 3 shows.
- Students had to select time, choose correct group and input an amount to successfully submit a bid.

[4] `http://docs.djangoproject.com,2011`

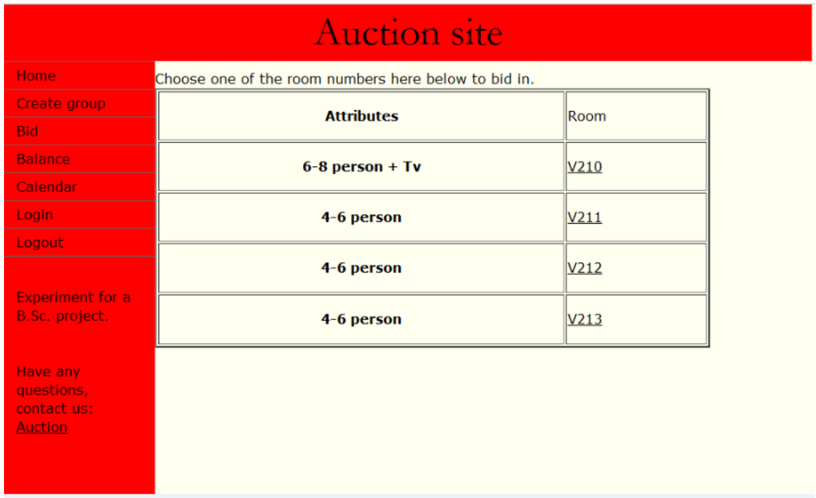


Figure 3

In Appendix B the website process is shown step by step.

8. Experiment

The experimented was run for one week during the semester, with four study rooms with different attributes. Students needed to bid on the rooms to get them, they could either bid on a "specific room" or "any room". Each student began with 1000 credits and it depended on the size of each group, how much they could bid on a room. If the group was large, the group could bid higher on a room and was therefore more likely to get the room.

As mentioned before a Vickrey auction was used in the experiment. Each student or a group of students bid on a room or all the rooms if the attributes of the room didn't matter. Bids were stored in a database, and the information was converted to a correct form and manually sent to the auction algorithm. The process was carried out after the auction closed at 8 PM every weekday. The results from the auction were displayed on the website and in front of each room.

Results from the experiment. In the experiment there were four study rooms, but the undergraduate students have access to eight study rooms. If a bidder bid on "any room" a bid was created for all rooms. A group will only get one room if the group wins the bid, like the constraint mentioned earlier in the report shows.

After one week 176 students had signed in to the website, and a total of 67 groups were created. The average group size was 3.25 students. There were 63 groups that made a bid on a certain time interval during the experiment. Figure 4 shows which room choice was most popular.

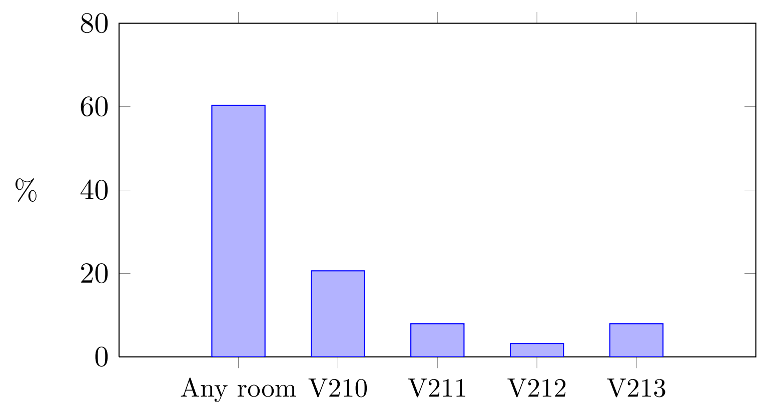


FIGURE 4. Room choice popularity

In Figure 5 the average price paid is shown, per time unit for each day.

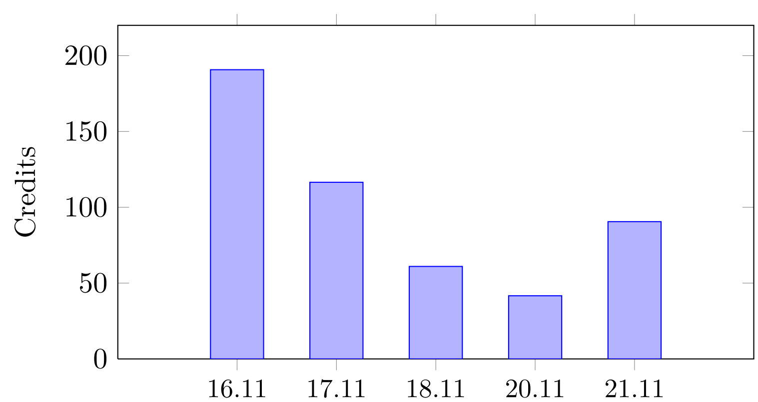


FIGURE 5. Average time unit price

The bids per time unit lowered after each day and rose again the last day. Which could mean that students were trying to spend their remaining balance on the last day. The average winning bid during the entire experiment was approximately 100 credits for each time unit.

9. Results

Before the experiment started we expected the groups that use the rooms to be bigger than with the previous arrangement. One of the questions asked in the survey, was "*When you use the study rooms, how many are you on average?*" In Figure 6 we can see the results from the survey.

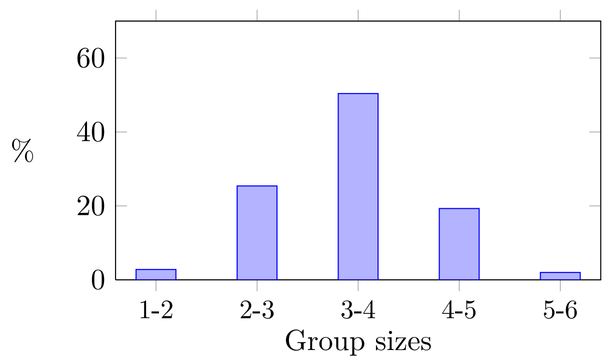


Figure 6

In Figure 7 the sizes of winning groups from the experiment are shown.

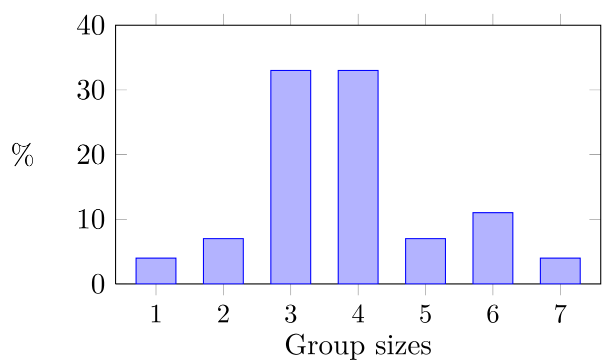


Figure 7

As Figure 7 shows, larger group sizes are more likely. In the beginning the goal was to get larger groups than usually use the study rooms like Figure 6 shows. The actual results shows that the average group size was approximately 3.9 students. As Figure 6 shows around 2% of students are in group of size 5 or 6. The results from the experiment show that there are approximately 18% of students that are in group of size 5 or 6. These results are very positive since the group sizes are bigger in the experiment than the group sizes in the survey.

10. Future work

There are a lot of possibilities using an auction form to get better utilization for the study rooms. The auction form tested was only one of many choices that could be used. It would be interesting to try a live Vickrey auction and display rooms that have been bid on, on the website. For that to happen the system must be completely automatic. Today the bids are downloaded manually from the database and sent manually to the algorithm and executed. It would be better to have more automation in this process in order to save time and have possibilities of a live auction.

The website and algorithm could be improved so students could bid on

- Any room
- Big room with TV
- Any room with TV

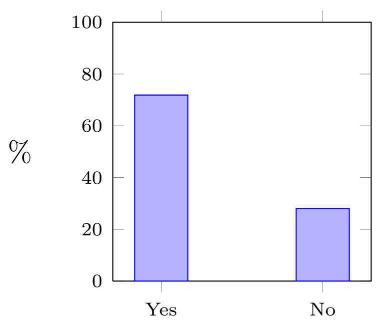
not just "any room" or "specific room" like it was in the experiment. It's also necessary, when a student creates a group that an automatic confirmation email is sent to all group members and students have to accept or decline membership to the group. It would also be a good addition if students could see the groups that they are registered in and their balance on the website.

11. Conclusion

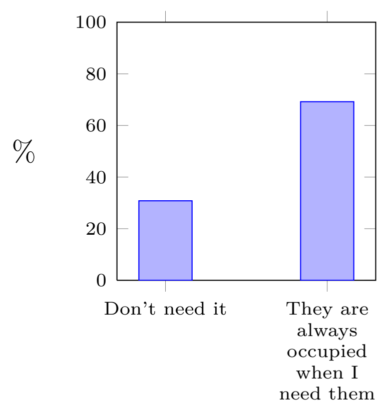
An auction form would be a good choice for the university to implement, in order to get better utilization for the study rooms. If an auction is implemented there can be some obstacles. One of them is that students could exchange credits if some students are not using it. Also the auction form could be a live auction, like we mentioned earlier, instead of an auction that is closed the previous night. If the auction is always live, students cannot guarantee a room for two or more time slots, since other students can bid on the study room while they are in it. Thus we would recommend a sealed bid auction that closed the previous night. As the results from the survey show, the majority of the students would like a different arrangement for the study rooms. The results from the experiment show that the utilization of the study rooms improved, thus an auction is a preferable choice to implement. Some students don't have the possibility of catching the rooms early in the morning, for example parents and students that work during the day. An auction would increase fairness and also create a competitive and positive atmosphere.

12. Appendix A

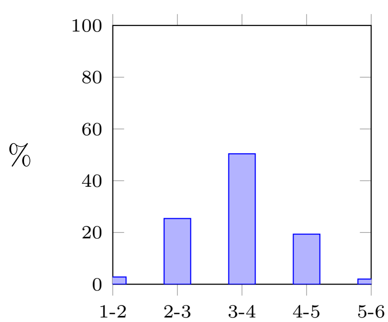
Early in the semester we sent out a survey to 3004 students in Reykjavik University and asked how they liked the current utilization for the study rooms. Five questions were asked and 310 students responded. The results are shown below.



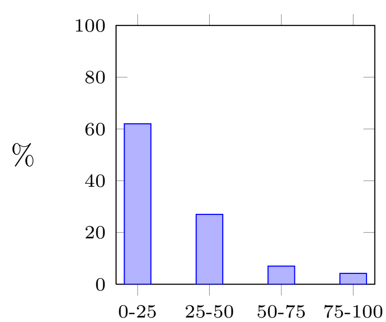
(a) Question 1: Do you use the study rooms?



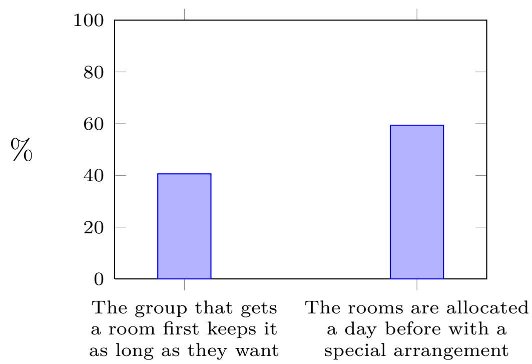
(b) Question 2: If no, why?



(c) Question 3: When you use the study rooms how many are in your group on average?(Do not apply if don't use the study rooms)



(d) Question 4: When you wanted to use the study rooms, how often, on average, has a room been available?



(e) Question 5: Which of following would you consider a better arrangement

Below you can see the actual survey in Icelandic.

Hvernig er nýting hópherbergjanna?

Notar þú hópherbergin?

		Response Percent	Response Total
1	Já	71.94%	223
2	Nei	28.06%	87
		answered	310
		skipped	-310

Ef ekki, afhverju?

		Response Percent	Response Total
1	Hef ekki þörf á þeim	30.77%	32
2	Þau eru alltaf upptekinn þegar ég þarf á þeim að halda	69.23%	72
		answered	104
		skipped	-104

Þegar þú notar hópherbergin, hversu mörg eruð þið að meðaltali? (Á ekki við ef þú notar ekki hópherbergin)

		Response Percent	Response Total
1	1-2	2.82%	7
2	2-3	25.40%	63
3	3-4	50.40%	125
4	4-5	19.35%	48
5	5-6	2.02%	5
		answered	248
		skipped	-248

Í þeim tilvikum sem þú hefur ætlað að nota hópherbergi, hversu oft, að meðaltali, hafa þau verið laus.

		Response Percent	Response Total
1	0 til 25%	61.89%	177
2	25 til 50%	26.92%	77
3	50 til 75%	6.99%	20
4	75 til 100%	4.20%	12
		answered	286
		skipped	-286

Hvert af eftirtöldu telur þú heppilegra fyrirkomulag:

		Response Percent	Response Total
1	Sá hópur sem fyrstur nær hópherbergi fær að halda því eins lengi og hann vill.	40.67%	122
2	Hópherberjum ráðstafað með dags fyrirvara með sérstöku fyrirkomulagi	59.33%	178
		answered	300
		skipped	-300

13. Appendix B

Figure 8 shows the home-page of the website, where the instructions are.

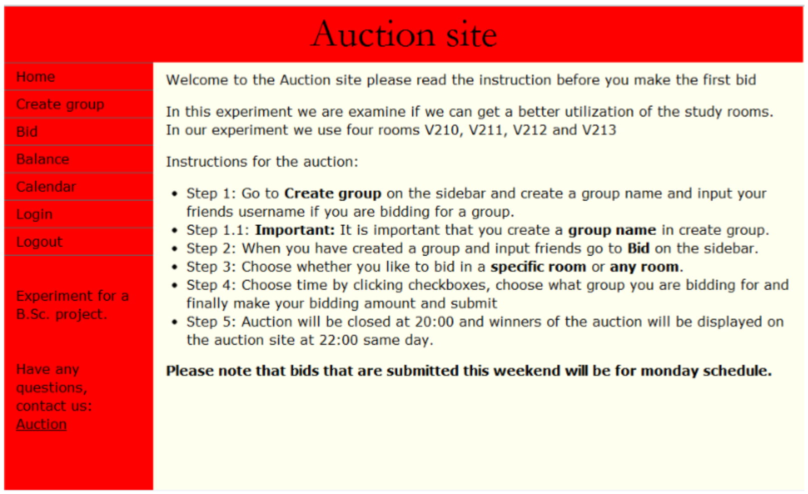


Figure 8. The home-page

Here the students were asked to create a group and add friends to the group. In order to make a legal bid students have to input friends' RU user names, *i.e.*, the user name students use for the intranet of Reykjavik University.

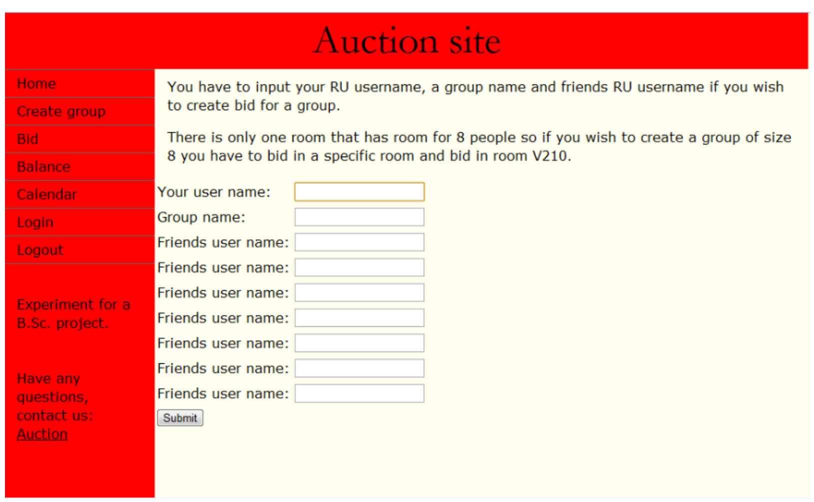


Figure 9. Create a group

When a group has been submitted, the user chooses “any room” or “specific room” to bid on.

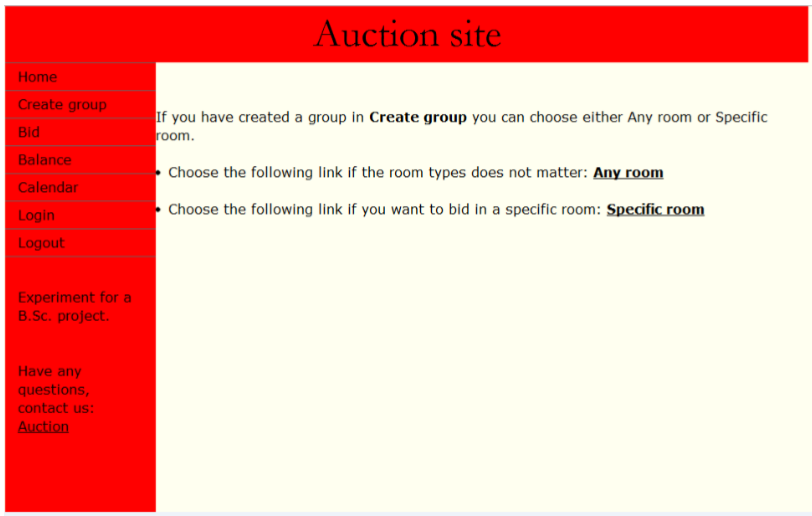


FIGURE 10. Choose “any room” or a “specific room”

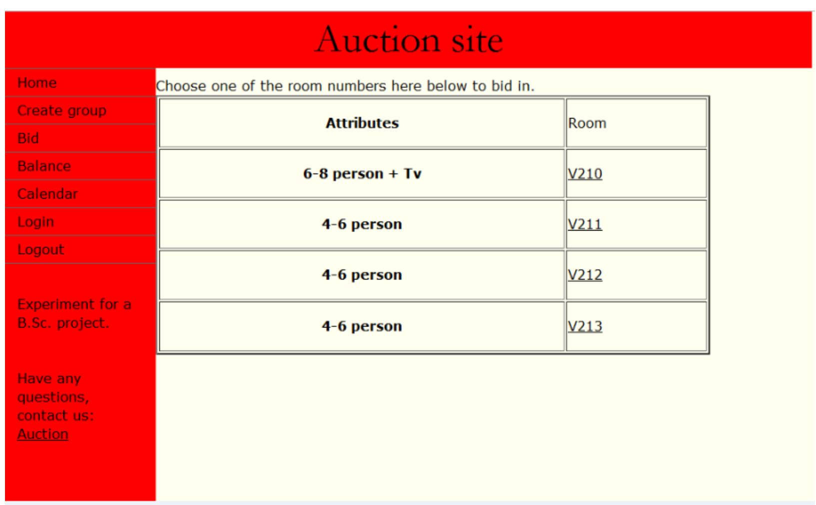


FIGURE 11. If “specific room” was chosen, the user needs to choose attributes

FIGURE 12. User selects a time, bid, group and presses submit

The user has successfully made a bid. User sees the results of the auction at 10 PM the same day.

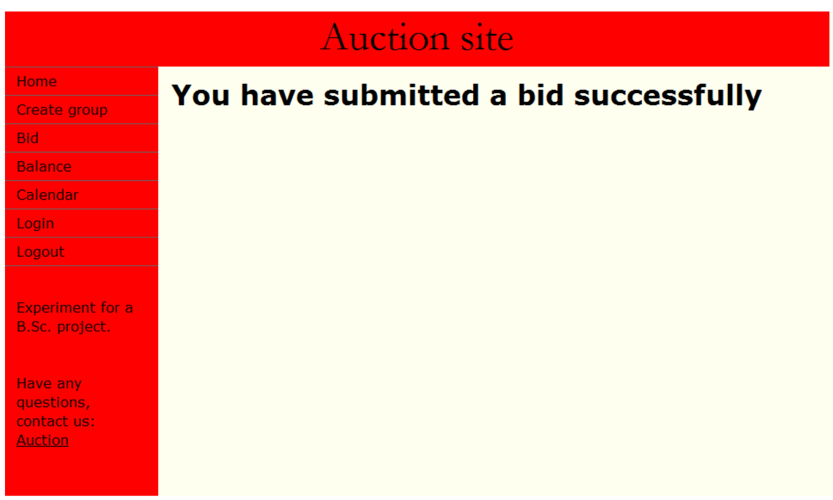
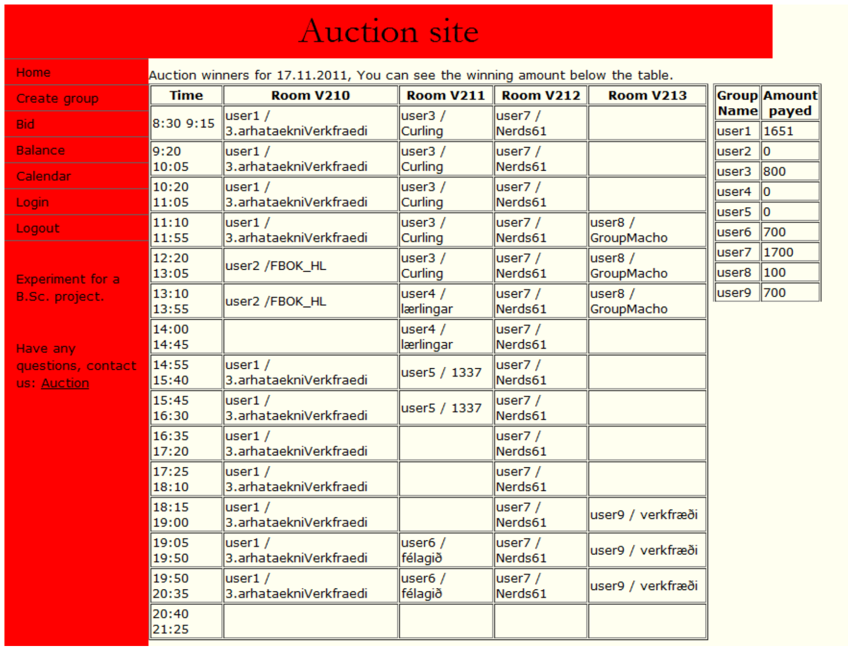


FIGURE 13. You have made a bid



Auction site

Home
Create group
Bid
Balance
Calendar
Login
Logout

Experiment for a B.Sc. project.

Have any questions, contact us: Auction

Auction winners for 17.11.2011, You can see the winning amount below the table.

Time	Room V210	Room V211	Room V212	Room V213
8:30 9:15	user1 / 3.arhataekniVerkfraedi	user3 / Curling	user7 / Nerds61	
9:20 10:05	user1 / 3.arhataekniVerkfraedi	user3 / Curling	user7 / Nerds61	
10:20 11:05	user1 / 3.arhataekniVerkfraedi	user3 / Curling	user7 / Nerds61	
11:10 11:55	user1 / 3.arhataekniVerkfraedi	user3 / Curling	user7 / Nerds61	user8 / GroupMacho
12:20 13:05	user2 /FBOK_HL	user3 / Curling	user7 / Nerds61	user8 / GroupMacho
13:10 13:55	user2 /FBOK_HL	user4 / lærlingar	user7 / Nerds61	user8 / GroupMacho
14:00 14:45		user4 / lærlingar	user7 / Nerds61	
14:55 15:40	user1 / 3.arhataekniVerkfraedi	user5 / 1337	user7 / Nerds61	
15:45 16:30	user1 / 3.arhataekniVerkfraedi	user5 / 1337	user7 / Nerds61	
16:35 17:20	user1 / 3.arhataekniVerkfraedi		user7 / Nerds61	
17:25 18:10	user1 / 3.arhataekniVerkfraedi		user7 / Nerds61	
18:15 19:00	user1 / 3.arhataekniVerkfraedi		user7 / Nerds61	user9 / verkfræði
19:05 19:50	user1 / 3.arhataekniVerkfraedi	user6 / félagið	user7 / Nerds61	user9 / verkfræði
19:50 20:35	user1 / 3.arhataekniVerkfraedi	user6 / félagið	user7 / Nerds61	user9 / verkfræði
20:40 21:25				

Group Name	Amount payed
user1	1651
user2	0
user3	800
user4	0
user5	0
user6	700
user7	1700
user8	100
user9	700

FIGURE 14. Calender for 17.11.2011

In Figure 14 we can see the calendar for 17.11.2011 and the amount paid by each group.

14. Appendix C

The auction algorithm:

```python
from pulp import*
import time
Clock = time.clock()
# A function that takes in the BIDS and variable name
def auction(inBIDS,var_names = "Bid_s"):

    BIDS = list(inBIDS)

    #Empty lists for time units, Groups, Room, amount
    T = []
    TO = []
    G = []
    R = []
    OBbid = []
    OBkey = []

    for b in BIDS:
        #Convert the list of time units to string
        newTO = str()

        for i in range(0,len(b[1])):

            if i < len(b[1])-1:
                newTO = newTO + str(b[1][i]) + ','
            else:
                newTO = newTO + str(b[1][i])
        # Add timeunits in the list
        T.append(b[1])
        # list of strings
        TO.append(newTO)
        # Add groups in a list
        G.append(b[2])
        # Add auctioned room in the list
        R.append(b[3])
        # Add bidding amount in a list
```

```
        OBbid.append(b[0])
        # A list of time , room, and group
        OBkey.append([newT0,b[2],b[3]])


    n = len(BIDS)                    # Set the lenght

    # Create the 'prob' variable to contain the problem
        data
    prob = LpProblem("The auction", LpMaximize)

    # A dictionary called 'Bid_sets_vars' is created to
        contain the referenced Variables
    Bid_sets_vars = LpVariable.dicts(var_names,(T0,G,R) ,
        lowBound = 0, upBound = 1,cat = LpInteger)

    #Each variable is 0 or 1 is added to 'prob'
    for i in range(0,n):

        prob += Bid_sets_vars[OBkey[i][0]][OBkey[i][1]][
            OBkey[i][2]] <= 1

    # The objective function is added to 'prob'

    prob += sum(OBbid[i]*Bid_sets_vars[OBkey[i][0]][OBkey
        [i][1]][OBkey[i][2]] for i in range(0,n))

    # The constraints are added to 'prob'

    for i in range(n):
        for j in range(i+1,n):
            # Never assign the same room on the same time
            if R[i]==R[j] and set(T[i]).isdisjoint(set(T[
                j]))== False:
                if G[i] != G[j]:
```

```
                prob += sum(Bid_sets_vars[TO[i]][G[i]][R[i]] + Bid_sets_vars[TO[j]][G[j]][R[j]])<=1
            else:
                prob += sum(Bid_sets_vars[TO[i]][G[i]][R[i]] + Bid_sets_vars[TO[j]][G[j]][R[j]])<=1
        # Never assign more then one set the same time units to the same group
        elif G[i] == G[j] and set(T[i]).isdisjoint(set(T[j]))== False:
            prob += sum(Bid_sets_vars[TO[i]][G[i]][R[i]] + Bid_sets_vars[TO[j]][G[j]][R[j]])<=1

prob.writeLP("The auction model.lp")

# Solve the prob
prob.solve()
print "Status:", LpStatus[prob.status]

# Empty list for the variables that wins
s_h = []

# Solution

for v in prob.variables():

    if v.varValue == 1:

        print v.name, "=" , v.varValue
        s_h.append(v)

obj = value(prob.objective)
print "objective = ", obj

```

```
    # Returns the list of time units, room, group,
        bidding amount and variables that won and the
        objective solution
    return [OBkey, OBbid, Bid_sets_vars,s_h, obj]

# Convert a string to list
def str_to_list(stre):
    return eval('['+stre+']')

# A function to check if the bid is legal
def is_legal(inBIDS, U, kred):

    BIDS = list(inBIDS)

    # Empty dictonary with payment
    payment = dict()

    # Each seat will be zero first
    for g in kred.keys():
        payment[g] = 0

    # Get the lists from auction
    OBkeyU = U[0]
    OBbidU = U[1]
    Bid_sets_varsU = U[2]

    # Check all the bids
    for k in range(len(OBkeyU)):
        # List of time units, groups and room
        K = OBkeyU[k]
        # Group
        g = K[1]

        # Make a variable
        badVar = Bid_sets_varsU[K[0]][g][K[2]]

        # Check if the variable is a winner
```

```
        if badVar.varValue == 1:

            # What the bidder has to pay
            amount = payment[g] + OBbidU[k]

            # If the bidder does not have enough credits
            if amount > kred[g]:

                # Convert the str to list in order to find the bid
                lst2K0 = str_to_list(K[0])

                # Find the bid
                bid = [OBbidU[k], lst2K0, g, K[2]]

                # Remove the illegal bid
                BIDS.remove(bid)

                return (False,BIDS)

            else:
                payment[g] = amount

    return (True,BIDS)

# A function that checks if all the bids are legal and return the actual price winner has to pay
def bidders_payment(inBIDS,kred):

    check = False

    BIDS = list(inBIDS)

    # Execute the first function
    U = auction(BIDS)
    U1 = list(U)

```

```
    # Check if there is any illegal bids
    while check == False:

        (check,BIDS) = is_legal(BIDS, U1, kred)

        # Execute the auction while there exists an
            illegal bid
        U1 = auction(BIDS)

    print 'Legal BIDS list is ' + str(BIDS)

    # An empty dict to find the actual price that winner
        has to pay
    actual_payment = dict()
    for g in kred.keys():
        actual_payment[g] = 0

    for j in range(len(U1[0])):
        # Check all list in OBkey
        J = U1[0][j]
        g = J[1]
        r = J[2]
        lst = str_to_list(J[0])

        # An empty list for the bids we need to remove to
            find the price
        b = []

        # If the variable is a winner
        if U1[2][J[0]][g][r].varValue == 1:
            for k in range(len(BIDS)):
                # Find all the bids with the same bidder
                    ID to remove from the BIDS list
                if g == BIDS[k][2]:

                    b.append(BIDS[k])

```

```
            # Function to find the actual price
            find_price = second_highest(U1,BIDS,b,j)

            actual_payment[g] += find_price

    print 'return of bidders_payment '
    print U1[3],actual_payment

    return (U1[3],actual_payment)

# Remove the elements from the list and execute the
    auction function again to find the price

def second_highest(U,inBIDS,b,i):
    # If b is a list of lists, remove all the lists from
        BIDS
    if type(b[0]) is list:
        BIDS = list(inBIDS)
        for b1 in b:

            BIDS.remove(b1)

        # Execute auction again and find what the removed
            winner has to pay
        return (b[0][0]-(U[4]-auction(BIDS,'temp_'+str(i)
            )[4]))

    else:
        BIDS = list(inBIDS)

        # Remove the list from BIDS
        BIDS.remove(b)

        # Execute auction again and find what the removed
            winner has to pay
        return (b[0][0]-(U[4]-auction(BIDS,'temp_'+str(i)
            )[4]))
```

```

# List of the bids
BIDS1 = []
# Balance each group has
kred = dict()

result = bidders_payment(BIDS1,kred)

clock_end = time.clock()

print clock_end - Clock
```

References

[1] Laurens M. Ausubel and Paul Milgrom. The Lovely but Lonely Vickrey Auction. Stanford University, Stanford, 2004. SIEPR Discussion Paper No. 03-36.

[2] Paul J. Brewer. Decentralized computation procurement and computational robustness in a smart market. *Economic Theory*, 13:41–92, 1999.

[3] Sven de Vries and Rakesh Vohra. Combinatorial auctions: A survey. *INFORMS journal on computing[1091-9856]*, 15:284, May 2003.

[4] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to operations research*. McGraw-Hil international editionl, 8 edition, 2005.

[5] Jonathan Levin. Notes on the Vicrey auction. `http://www.stanford.edu/~jdlevin/Econ%20136/Vickrey%20Auction%20Notes.pdf`, 2010.

[6] S.J. Rassenti, V.L. Smith, and R.L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *The Bell Journal of Economics*, 13:402–417, 1982.

[7] Guaqiang Tian. Lecture notes, auction theory. `http://econweb.tamu.edu/tian/auction%20theory-10-01.pdf`, 2010.