Course Allocation by Proxy Auction[⋆]

Scott Duke Kominers[1⋆⋆], Mike Ruberry[2⋆⋆⋆], and Jonathan Ullman[2†]

[1] Department of Economics, Harvard University, and Harvard Business School
skominers@hbs.edu
[2] Harvard University School of Engineering and Applied Sciences
{mruberry,jullman}@seas.harvard.edu

Abstract. We propose a new *proxy bidding mechanism* to allocate courses to students given students' reported preferences. Our mechanism is motivated by a specific strategic *downgrading* manipulation observed in the course allocation mechanism currently used at Harvard Business School (HBS). The proxy bidding mechanism simplifes students' decisions by directly incorporating downgrading into the mechanism.
Our proxy bidding mechanism is Pareto efficient with respect to lexicographic preferences and can be extended to allow for a more expressive preference language which allows complementarity, substitutability, and indifference. Simulations suggest that the proxy bidding mechanism is robust to the manipulations observed in the HBS mechanism and may yield welfare improvements.

1 Introduction

Course allocation is a *combinatorial assignment problem* that assigns students to courses, given students' preferences over course schedules. Unfortunately, any strategyproof and efficient mechanism for this problem must be dictatorial ([6]), with poor outcomes ([3]).[3]

One course allocation system is the *Bidding Points mechanism*, in which students "bid" for courses using an artificial currency ([4] [7]). Although these mechanisms are commonly used in practice, they require strategic play by students and have meager welfare guarantees.

An alternative course allocation mechanism is the *draft*, in which students take turns selecting individual courses from those with available seats, following in a "draft order." Such a mechanism is used by Harvard Business School

[⋆] We thank Susan Athey, Eric Budish, Yiling Chen, John William Hatfield, David Parkes, and Alvin Roth for helpful conversations. We thank Utku Ünver for graciously providing us with data.

[⋆⋆] Supported by an NSF Graduate Research Fellowship and a Yahoo! Key Scientific Challenges Program Fellowship.

[⋆⋆⋆] Corresponding author.

[†] Supported by NSF grant CNS-0831289.

[3] It is known that on restricted preference domains, better solutions may be achieved. For example, under unit-demand (the *school choice problem*), non-dictatorial strategyproof mechanisms are known (see [1] [5]).

(HBS). The draft mechanism offers strong welfare guarantees under truthful play, but is easily manipulable: Budish and Cantillon [3] demonstrated that students successfully and substantially manipulate the HBS draft, to the benefit of sophisticated students at the expense of social welfare.[4]

We reinterpret the HBS draft as a game amongst naïve proxy agents who act on behalf of students. We develop a *proxy bidding mechanism* for course assignment in which students' proxies pick strategically, unlike in the draft, obviating the need for certain types of manipulations. Our proxies' behaviors are inspired by the desire to have a mechanism play optimal draft strategies on students' behalves.

We prove that the proxy bidding mechanism is Pareto efficient when students have lexicographic preferences over schedules. We present simulations that show that the proxy bidding mechanism performs favorably relative to the HBS draft in the presence of the manipulations identified by Budish and Cantillon [3]. Finally, we extend our mechanism to allow a bidding language in which complementarity, substitutability and indifference can be expressed. For brevity, most proofs are deferred to the Appendix, which is available on the authors' websites.

2 The Proxy Bidding Mechanism

Our proxy bidding mechanism provides a course allocation for a problem consisting of a set of courses $\mathcal{C}$, a vector of course capacities $(q_c)_{c\in\mathcal{C}}$, a set of players $\mathcal{N}$, as well as a set of strict ordinal preferences over courses, $(\prec_i)_{i\in\mathcal{N}}$, specified by the players.

The mechanism takes as input a set of bidding priorities B_i for each player i; these will typically be allocated randomly for reasons of fairness. The bid sets $\{B_i\}_{i\in\mathcal{N}}$ form a partition of a finite global bid set $\mathcal{B}\subset\mathbb{R}$. The maximum number of courses a player can be allocated is $|B_i|$. In principle the choice of bid sets may be arbitrary, but for our purposes we only consider bids in correspondence with turns of the HBS draft, choosing

$$B_i=\{-i,-(2\mathcal{N}-i),-(2\mathcal{N}+i+1),\ldots\}.$$

The mechanism maintains a set of multi-unit auctions $(A_c)_{c\in\mathcal{C}}$ (interpreted as sets of winning bids, for convenience) for course seats.

Note that agents' bids are indivisible. We think of the set $\mathcal{B}$ as representing a sequence of course selection opportunities; the set B_i is the set of opportunities in $\mathcal{B}$ at which i may select a course. In this notation the HBS draft is the mechanism in which students choose courses in the sequence $\mathcal{B}$, with each student i selecting a course at each opportunity $b\in B_i$.

Our proxy bidding mechanism proceeds in rounds, with players' bids being reallocated in response to the current profile of course "prices" $(p(A_c))_{c\in\mathcal{C}}$,

[4] In a separate work, Budish [2] introduces the effective but complex *Approximate Competitive Equilibrium from Equal Incomes* mechanism which ameliorates many of the issues discussed above.

where[5]

$$p(A_c) := \begin{cases} 0 & |A_c| < q_c, \\ \min A_c & |A_c| = q_c. \end{cases}$$

The formal proxy bidding mechanism specification is given in Algorithm 1. We take this opportunity to explain where our mechanism diverges from the draft mechanism. The inner loop of our mechanism (lines 9- 29) allows players' proxies to place the *lowest sufficient bid* into an auction to avoid overpaying for a course. In the language of the draft mechanism, overpaying for a course means selecting that course too early. If we were to replace this inner loop with a simpler procedure that places the *maximum available bid* into each successful course, then we would recover something closer to the draft mechanism.

We now give an example of the proxy bidding mechanism on simple input. Consider the following input: $\mathcal{N} = \{1, 2, 3\}$, $\mathcal{C} = \{c_1, c_2, c_3\}$, $q_{c_1} = q_{c_2} = q_{c_3} = 2$,

$$\begin{aligned} c_3 \prec_1 c_2 \prec_1 c_1, &\quad B_1 = \{6, 1\}, \\ c_3 \prec_2 c_2 \prec_2 c_1, &\quad B_2 = \{5, 2\}, \\ c_1 \prec_3 c_3 \prec_3 c_2, &\quad B_3 = \{4, 3\}. \end{aligned}$$

The mechanism will run as illustrated below.[6]

Round	i	State
1	1	$A_{c_1} = \{1\}, A_{c_2} = \{1\}, A_{c_3} = \emptyset$
2	2	$A_{c_1} = \{2, 1\}, A_{c_2} = \{2, 1\}, A_{c_3} = \emptyset$
3	3	$A_{c_1} = \{2, 1\}, A_{c_2} = \{3, 2\}, A_{c_3} = \{3\}$
4	1	$A_{c_1} = \{2, 1\}, A_{c_2} = \{6, 3\}, A_{c_3} = \{3\}$
5	2	$A_{c_1} = \{2, 1\}, A_{c_2} = \{6, 5\}, A_{c_3} = \{3\}$
6	3	$A_{c_1} = \{3, 2\}, A_{c_2} = \{6, 5\}, A_{c_3} = \{3\}$
7	1	$A_{c_1} = \{6, 3\}, A_{c_2} = \{5, 1\}, A_{c_3} = \{3\}$
8	2	$A_{c_1} = \{6, 5\}, A_{c_2} = \{2, 1\}, A_{c_3} = \{3\}$
9	3	$A_{c_1} = \{6, 5\}, A_{c_2} = \{3, 2\}, A_{c_3} = \{3\}$
10	1	$A_{c_1} = \{6, 5\}, A_{c_2} = \{3, 2\}, A_{c_3} = \{3, 1\}$

In the first round, the price of every course is 0, so player 1 bids for his two most-preferred courses. Notice that he uses his *lowest sufficient bid* in both auctions but that his biding is consistent with his actual bid set. That is, he only needs to bid 1 to win each course, but *could* win both courses as he has bids of 6 and 1 available. In the third round, player 3 sees that his most-preferred course has a price of 1 and his second most-preferred course has a price of 0; he therefore bids 3 on each. But then, player 1 no longer has a winning bid in A_{c_2}. In the fourth round, player 1 observes that the price of c_2 is now 2, so he

[5] Our auction procedure has a cleanup step to guarantee that $0 \leq |A_c| \leq q_c$ for every $c \in \mathcal{C}$, hence it is only necessary to define the price function on this range.

[6] Each row of the right column represents the states of the auctions *after* player i has had a chance to update his bids.

bids 6 in A_{c_2}. The mechanism continues in this fashion until it terminates six rounds later.[7] The final allocation C is given by $C_1 = \{c_1, c_3\}$, $C_2 = \{c_1, c_2\}$, $C_3 = \{c_2, c_3\}$.

Our mechanism preserves some of the draft's positive properties. In particular our simultaneous auction is guaranteed to converge and results in a Pareto efficient allocation.

Proposition 1 (Convergence). *Algorithm 1 terminates.*

If bids fail to converge there must be a "cycle," that is, a series of auction states which is repeated over the course of the mechanism. Since bids are discrete and unique, there is for any such cycle a highest bid b^* cast in any stage of the bid cycle. Moreover, at some point in the cycle the student $i \in \mathcal{N}$ holding bid b^* must cast b^* for some course c which i ranks most highly among all courses i bids for during the course of the cycle. But since b^* is maximal among all cast during the cycle, it cannot be displaced once made in the course-c auction A_c; this would contradicts the involvement of b^* in the cycle. Note that this argument directly uses the fact that the bidding mechanism disallows the withdrawl of undisplaced bids.

Proposition 2 (Pareto Efficiency). *The allocation produced by Algorithm 1 is Pareto efficient with respect to the lexicographic preferences induced by input preferences* $(\prec_i)_{i \in \mathcal{N}}$.

For any Pareto inefficient allocation, there is (by definition) a sequence of course trades which consitute a Pareto improvement. Of the students participating in these trades, one student i^* buys the course c_{i^*} he wishes to trade away with the largest bid $b_{c_{i^*}}$ used to buy a course in the Pareto-improving trade. But under the bidding mechanism, i^* should instead have bid $b_{c_{i^*}}$ in the course for which i seeks to trade. Thus, the outcome of the proxy bidding mechanism must be Pareto efficient.

3 Welfare Properties

In this section we use simulations to analyze the welfare properties of the proxy bidding mechanism. Our simulation environment produces correlated preferences for 1000 students over a set of 110 courses. Each student demands 10 courses and each course offers 100 seats. This problem is roughly the size of that of Harvard Business School. Full details of our simulation environment are deferred to the Appendix.

We use two welfare measures averaged over the population to evaluate assignments: *average rank*, the average preference rank of the courses allocated to each student; *lexicographic rank*, the highest-ranked course received. Note that since we interpret preferences as rank-order lists from 0-th to $(|B_i| - 1)$-st in the

[7] Technically, the auction will not terminate until all players have declined the chance to change their bids.

Algorithm 1

Input: $\mathcal{C}$, $(q_c)_{c\in\mathcal{C}}$, $\mathcal{N}$, $(\prec_i)_{i\in\mathcal{N}}$, $\{B_i\}_{i\in\mathcal{N}}$

Output: $(C_i)_{i\in\mathcal{N}}$

```
for $c \in \mathcal{C}$ do
    $A_c := \emptyset$
end for

repeat
    active := false
    for $i \in \mathcal{N}$ do
        $B'_i = \emptyset$
        for $c$ in order of $\prec_i$ do
            if $B_i \cap A_c = \emptyset$ then
                $b^* := \min_{b\in B_i \setminus B'_i}\{b > p(A_c)\}$
                if $b^*$ exists then
                    add $b^*$ to $B'_i$
                    add $\min_{b\in B_i}\{b > p(A_c)\}$ to $A_c$
                    if $|A_c| > q_c$ then
                        remove $\min_{b\in A_c}$ from $A_c$
                    end if
                    active := true
                end if
            else
                $b^* := B_i \cap A_c$
                $b^{**} := \min_{b\in B_i \setminus B'_i}\{b \geq b^*\}$
                if $b^{**}$ exists then
                    add $b^{**}$ to $B'_i$
                else
                    remove $b^*$ from $A_c$
                end if
            end if
        end for
    end for
until active = false

for $i \in \mathcal{N}$ do
    $C_i := \{c \mid B_i \cap A_c \neq \emptyset\}$
end for
```

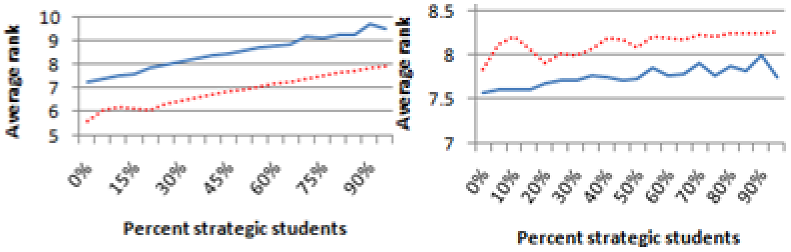

Fig. 1. A simulation of downgrading in the HBS draft (left) and proxy bidding (right) mechanisms. The solid and dotted lines respectively represent the average welfare levels of truthful and downgrading students.

example presented in the previous section, the average rank of the allocation is 2/3 and the lexicographic rank is 0.[8]

3.1 Comparison with the HBS Draft

First we demonstrate that the HBS draft mechanism is susceptible to strategic play in our environment by simulating the *downgrading* manipulation highlighted by Budish and Cantillon [3], in which (some) students partially reorder their preferences in correspondence with course popularity levels. In Figure 1, average rank outcomes are plotted for both the HBS draft and the proxy bidding mechanism. The dotted and solid line respectively plot the outcome to strategic and non-strategic students. In Figure 1 (and in Figure 1, below), the horizontal axis indicates the fraction of students playing the manipulative strategy and the vertical axis indicates the outcome.

As expected, students who play the downgrading manipulation in the HBS draft receive substantially lower-ranked courses on average than students who report their preferences straightforwardly. In the proxy bidding mechanism, however, the opposite result obtains, demonstrating resilience of the proxy bidding mechanism to this manipulation. These results hold regardless of the fraction of strategic students.

Figure 2 plots cross-population welfare statistics in the presence of downgrading by part of the population. The solid line charts the performance of the proxy bidding mechanism; the dashed line charts the HBS draft. While both populations' welfare decreases as more students manipulate and at any point the welfare of proxy bidding is lower, the downgrading manipulation only benefits students in the draft, as shown in Figure 1. Once over 30% of students play downgrading strategies, average welfare in the HBS draft is worse than would be achieved in the proxy bidding mechanism under truthful play.

[8] In the example, one student received his first- and third-most-preferred courses and two received their first- and second-most-preferred courses. Considering the most-preferred course to have rank 0, the lexicographic rank is 0 (all students received their most-preferred courses). The average rank is $((0+2)/2+(0+1)/2+(0+1)/2)/3 = 2/3$.

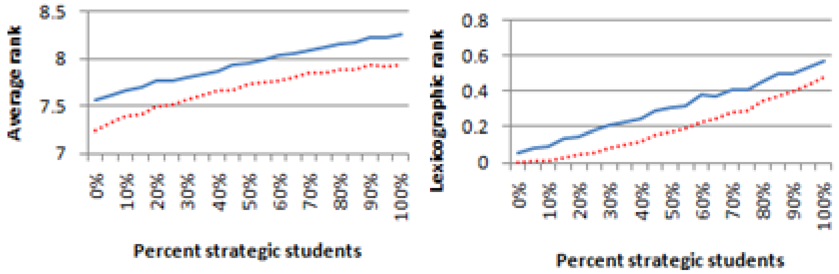

Fig. 2. Simulated average welfare levels of the entire population when part of the population plays the downgrading strategy. The dotted line is the HBS draft and the solid line proxy bidding. While at any point on the curve proxy bidding lags the draft, its resilience to this manipulation, as seen in Figure 1, suggests it will be played truthfully.

3.2 Strategic Play in the Proxy Bidding Mechanism

Although our proxy bidding mechanism is apparently robust to the downgrading manipulation, it is not strategyproof.

Consider the following input: $\mathcal{N} = \{1, 2\}$, $\mathcal{C} = \{c_1, c_2, c_3, c_4\}$, $q_{c_1} = q_{c_2} = q_{c_3} = q_{c_4} = 1$,

$$c_4 \prec_1 c_3 \prec_1 c_2 \prec_1 c_1, \quad B_1 = \{4, 1\},$$
$$c_1 \prec_2 c_4 \prec_2 c_3 \prec_2 c_2, \quad B_2 = \{3, 2\}.$$

The final allocation will be C given by $C_1 = \{c_1, c_2\}$, $C_2 = \{c_3, c_4\}$. If player 2 reports the preferences $c_4 \prec_2' c_3 \prec_2' c_2 \prec_2' c_1$, then the final allocation C' given by $C_1' = \{c_1, c_4\}$, $C_2' = \{c_2, c_3\}$ is obtained. Thus a player with preferences $\prec_2$ receives a more-preferred allocation by reporting $\prec_2'$ than by reporting honestly.

4 Extended preference support

Lexicographic preferences over single courses provide an elegant model and allow quick computation but are unlikely to represent students' true preferences. To partially address this problem, we extend the input space of the proxy bidding mechanism to allow players to express conditional demand for courses. Specifically, we introduce `ANY`, `IF`, and `NOTIF` statements, which we illustrate in Figure 3 and define formally in the Appendix.

The `ANY` statement is an exclusive-or over a set of courses. This feature allows players to indicate indifference over a set of courses, such as identical sections of a course, or equally-preferred courses that meet at the same time. Such a feature seems especially important, as in current bidding systems it is sometimes possible ex-post for students to obtain a section of every course offered.[9] The `IF`

[9] This ironic situation occurs as an outcome of the University of Michigan bidding points mechanism, which has been studied by Krishna and Unver [4].

Order	Modifier	Course name
0		Modern Art
1	IF Modern Art	Modern Art Criticism
2	NOTIF Modern Art Criticism	Renaissance Art
3	ANY	{Ancient History, Modern History Classical History}

Fig. 3. A student's extended preferences. The student's most-preferred courses are Modern Art and Modern Art Criticism, but she can only take the latter if she takes the former. If she does not receive Modern Art Criticism, she would like Renaissance Art instead. She also must take one art history course but is indifferent between three choices.

(respectively, NOTIF) statement is a conditional which allows a player to demand a course c if he holds (respectively, does not hold) a more-preferred course c'.

Our proxy bidding mechanism suitably extends to accommodate this more-expressive preference language, however the outcome produced may not be Pareto efficient if ANY statements are used (see the Appendix).

5 Conclusion and Future Work

We have introduced a new proxy bidding mechanism for course allocation which offers attractive welfare possibilities. It is relatively simple and extends naturally to allow for a more expressive preference language than is typically used in course allocation. Although our mechanism is resistant to the strategic manipulations that have been observed in the HBS draft, a full analysis of strategic play under the proxy bidding mechanism requires further study.

It seems likely that our approach of replacing agents by proxies who strategize on the agents' behalves would find applications in other domains. A general theory of proxy mechanisms seems appropriate for future work.

References

1. Abdulkadiroglu, A., Pathak, P.A., Roth, A.E., Sonmez, T.: Changing the boston school choice mechanism: Strategy-proofness as equal access (2006)
2. Budish, E.: The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes (2010)
3. Budish, E., Cantillon, E.: Strategic behavior in multi-unit assignment problems: Lessons for market design (2009)
4. Krishna, A., Unver, M.U.: Improving the efficiency of course bidding at business schools: Field and laboratory studies. Marketing Science 27(262), 262–82 (2008)
5. Pápai, S.: Strategyproof assignment by hierarchical exchange. Econometrica 68(6), 1403–33 (2000)
6. Pápai, S.: Strategyproof and nonbossy multiple assignments. Journal of Public Economic Theory 3(3), 257–71 (2001)
7. Sonmez, T., Unver, U.: Course bidding at business schools. International Economic Review 51(1), 99–123 (2010)

A Properties of the Proxy Bidding Mechanism

For notational efficiency we will say that one *timestep* has elapsed at each pass through line 29. We will use the *state at timestep t* to refer to $\sigma_t = (A_c^{(t)})_{c\in\mathcal{C}}$ at the t-th pass through line 29.

A.1 Convergence

The first, and essential result about our mechanism is that it does indeed reach a steady state and halt whenever the input is valid. For an input to be *valid*, we require that

1. for every $i, i' \in \mathcal{N}$, $B_i \cap B_{i'} = \emptyset$,
2. for every $i \in \mathcal{N}$, $\prec_i$ is a strict preference relation over a subset $\mathcal{C}' \subseteq \mathcal{C}$.

Condition 1 requires that no two players have bids of equal value. Condition 2 requires that every player's preference relation over courses be strict.

Proposition 1 (Convergence). *Algorithm 1 terminates on all valid inputs.*

Proof. Since Algorithm 1 is memory free, it suffices to show that there do not exist timesteps $t \neq t'$ such that

1. $\sigma_t = \sigma_{t'}$, and
2. $\sigma_{t+1} \neq \sigma_{t'}$.

So we assume the contrary, there there exist $t \neq t'$ satisfying these conditions.

Without loss of generality, suppose $t' > t$, and consider the cycle of states

$$\sigma_t, \sigma_{t+1}, \sigma_{t+2}, \ldots, \sigma_{t'}. \tag{1}$$

We define the set of bids that are *fixed* in this cycle as

$$F = \left\{ b \in \mathcal{B} \mid \exists c \; \forall t \leq t_1 \leq t' \text{ s.t. } b \in A_c^{(m)} \right\}. \tag{2}$$

These bids are purchasing the same course in every state of the cycle. Consider the largest bid that is not fixed in this cycle, $b^* = \max_{b\in\mathcal{B}\backslash F}$; such a bid exists since $\sigma_t \neq \sigma_{t+1}$. Since b^* is not fixed, and maximal in $\mathcal{B}\backslash F$, there exists $t \leq t^* < t'$ and courses $c, c' \in \mathcal{C}$ such that

$$b^* \in A_c^{(t^*)}, \quad b^* \in A_{c'}^{(t^*+1)}. \tag{3}$$

Assume without loss of generality $b^* \in B_i$.

Consider the case where $c' \prec_i c$. Then (3) is inconsistent with the specification of Algorithm 1. Since c would come at an earlier iteration of the loop beginning at line 9, Algorithm 1 would add b^* to $A_c^{(t^*+1)}$ at line 14, a contradiction.

Now consider the case where $c \prec_i c'$. Since we have assumed there is a cycle, there must exist t^{**} and course $c'' \in \mathcal{C}$ such that $c'' \prec_i c'$ and

$$b^* \in A_{c'}^{(t^{**})}, \; b^* \in A_{c''}^{(t^{**}+1)}. \tag{4}$$

However this is also inconsistent with the specification of Algorithm 1.

A.2 Pareto Efficiency

In order to reason about Pareto efficiency, we impute full preferences over bundles from the ordinal preferences $(\prec_i)_{i\in\mathcal{N}}$ specified by players. Specifically, we take the bundle preferences of a player i to be lexicographic with respect to $\prec_i$.

To demonstrate that allocations produced by Algorithm 1 are Pareto efficient, we introduce the notion of a *trade cycle*.

Definition 1. *An allocation $(C_i)_{i\in\mathcal{N}}$ contains a* trade cycle *$(\mathcal{C}',\mathcal{N}')$, $\mathcal{C}' \subseteq \mathcal{C}$, $\mathcal{N}' \subseteq \mathcal{N}$, with respect to preferences $(\prec_i)_{i\in\mathcal{N}}$ if*

1. *for every $c_i \in \mathcal{C}'$, $c_i \in C_i$ for $i \in \mathcal{N}'$, and*
2. *the set of players $i \in \mathcal{N}'$ can be ordered as $i_1,\ldots,i_{|\mathcal{C}'|}$ such that for every $c_{i_\ell} \in \mathcal{C}'$, $c_{i_\ell} \prec_{i_\ell} c_{i_{\ell+1}}$.*

It will suffice to restrict our attention to trade cycles because of the following lemma, the proof of which is immediate (and hence is omitted).

Lemma 1. *An allocation $(C_i)_{i\in\mathcal{N}}$ is Pareto efficient with with respect to the lexicographic preferences induced by $(\prec_i)_{i\in\mathcal{N}}$ if and only if it does not contain a trade cycle.*

We now proceed with the proof of Pareto efficiency.

Proposition 2 (Pareto Efficiency). *The allocation produced by Algorithm 1 is Pareto efficient with respect to the lexicographic preferences induced by input preferences $(\prec_i)_{i\in\mathcal{N}}$.*

Proof. Suppose the allocation is not Pareto efficient. By Lemma 1, there exists a trade cycle $(\mathcal{C}',\mathcal{N}')$. Consider the state of the mechanism, σ_t, at termination. For every $c_i \in \mathcal{C}'$, let

$$b_{c_i} := B_i \cap A^{(t)}_{c_i} \tag{5}$$

be the bid used by player i to purchase course c_i. We now define the player with the maximal bid in the cycle as

$$i^* = \operatorname{argmax}_{i\in\mathcal{N}'} b_{c_i} \tag{6}$$

By assumption $c_{i^*} \prec_{i^*} c_{i^*+1}$, hence since $c_{i^*+1} \notin C_{i^*}$, it must be the case that $p(A^{(t)}_{c_{i^*+1}}) > b_{c_{i^*}}$. But this is impossible, since $b_{c_{i^*}} > b_{c_{i^*+1}} \geq p(A^{(t)}_{c_{i^*+1}})$, a contradiction.

B Extended Preference Support

B.1 Preliminaries

Let $\{\mathscr{C}_j\}$ be a partition of $\mathcal{C}$; we refer to the elements of $\{\mathscr{C}_j\}$ as *level sets*. A bundle of courses $C \subseteq \mathcal{C}$ is *feasible with respect to* $\{\mathscr{C}_j\}$ if $|C \cap \mathscr{C}_j| \leq 1$ for every j, i.e. if the bundle contains no more than one course from each set in the partition.

For a partition $\{\mathscr{C}_j\}$, we write

$$G(\{\mathscr{C}_j\}, E_{\texttt{IF}}, E_{\texttt{NOTIF}})$$

for the directed graph on $\{\mathscr{C}_j\}$ with two disjoint edge sets $E_{\texttt{IF}}$ and $E_{\texttt{NOTIF}}$. We call such a graph a *conditional demand graph*[10], interpreting directed edges

1. $(\mathscr{C}', \mathscr{C}) \in E_{\texttt{IF}}$ as meaning that a course in $\mathscr{C}$ is desired only if a course in $\mathscr{C}'$ is held and
2. $(\mathscr{C}', \mathscr{C}) \in E_{\texttt{NOTIF}}$ as meaning that a course in $\mathscr{C}$ is desired only if a course in $\mathscr{C}'$ is not held.

For notational convenience we define the *`IF` neighborhood* and *`NOTIF` neighborhood* of a level set $\mathscr{C}$, respectively denoted $E^{\mathscr{C}}_{\texttt{IF}}$ and $E^{\mathscr{C}}_{\texttt{NOTIF}}$:

$$E^{\mathscr{C}}_{\texttt{IF}} := \{\mathscr{C}' \in \{\mathscr{C}_j\} \mid (\mathscr{C}', \mathscr{C}) \in E_{\texttt{IF}}\},$$
$$E^{\mathscr{C}}_{\texttt{NOTIF}} := \{\mathscr{C}' \in \{\mathscr{C}_j\} \mid (\mathscr{C}', \mathscr{C}) \in E_{\texttt{NOTIF}}\}.$$

We say that a bundle of courses $C \subseteq \mathcal{C}$ is *feasible with respect to* $G(\{\mathscr{C}_j\}, E_{\mathit{IF}}, E_{\mathit{NOTIF}})$ if it is feasible with respect to $\{\mathscr{C}_j\}$ and furthermore

1. if $\mathscr{C}' \in E^{\mathscr{C}}_{\texttt{IF}}$ and $C \cap \mathscr{C} \neq \emptyset$ then $C \cap \mathscr{C}' \neq \emptyset$,
2. if $\mathscr{C}' \in E^{\mathscr{C}}_{\texttt{NOTIF}}$ and $C \cap \mathscr{C} \neq \emptyset$ then $C \cap \mathscr{C}' = \emptyset$.

B.2 Extended Preferences

Using the concepts defined in the previous section, we now extend players' preferences to conditional statements over partitions.

Definition 2. *An* extended preference relation π *is a ranking* $\prec$ *over* $\{\mathscr{C}_j\}$ *along with a conditional demand graph* $G(\{\mathscr{C}_j\}, E_{\mathit{IF}}, E_{\mathit{NOTIF}})$ *such that if* $(\mathscr{C}_j, \mathscr{C}_{j'}) \in E_{\mathit{IF}}$ *or* $(\mathscr{C}_j, \mathscr{C}_{j'}) \in E_{\mathit{NOTIF}}$ *then* $\mathscr{C}_{j'} \prec \mathscr{C}_j$*.*

An allocation $C \subseteq \mathcal{C}$ is *feasible with respect to* π if it is feasible with respect to the conditional demand graph underlying π.

From an extended preference relation π, we impute the following weak preferences, Π, over feasible allocations: a feasible allocation C is preferred to another feasible allocation C' if the most preferred level set not achieved under C' is higher than the most preferred level set not achieved under C. More precisely, let

$$\mathfrak{D} := \{\mathscr{C}_j \mid C \cap \mathscr{C}_j \neq \emptyset\} \cap \{\mathscr{C}_j \mid C' \cap \mathscr{C}_j \neq \emptyset\}\,.$$

Allocation C is preferred to C' under Π, denoted $C \Pi C'$, if and only if

$$\max_{\pi} \{\mathscr{C}_j \in \{\mathscr{C}_j\} \setminus \mathfrak{D} \mid C \cap \mathscr{C}_j \neq \emptyset\}$$

is either empty or preferred to

$$\max_{\pi} \{\mathscr{C}_j \in \{\mathscr{C}_j\} \setminus \mathfrak{D} \mid C' \cap \mathscr{C}_j \neq \emptyset\}$$

under π.

[10] For simplicity, we have required demand for a level set to be conditional on whole sets. However, this definition and our algorithm easily extend to demand that is conditional on individual courses.

B.3 Extended Mechanism

In this section we modify Algorithm 1 to compute feasible, efficient allocations with respect to a profile of extended preference relations $(\pi_i)_{i\in\mathcal{N}}$. We define the *extended proxy bidding mechanism* to be Algorithm 1 with the inner loop (lines 9–29) replaced with the extended bidding procedure detailed in Algorithm 2 below. The extended algorithm takes as input extended preferences $(\pi_i)_{i\in\mathcal{N}}$ instead of strict preferences $(\prec_i)_{i\in\mathcal{N}}$.

B.4 Convergence

As for our original mechanism, we must prove that the extended proxy bidding mechanism actually converges.

Proposition 3 (Convergence). *The extended proxy bidding mechanism terminates on all valid inputs.*[11]

Proof. As in the proof of Proposition 1, suppose there is a cycle of states

$$\sigma_t, \sigma_{t+1}, \sigma_{t+2}, \ldots, \sigma_{t'}. \tag{7}$$

with $t+1 \neq t < t'$. We define the set of bids that are *fixed* in this cycle as in (2). Consider the largest bid that is not fixed in this cycle, $b^* = \max_{b\in\mathcal{B}\setminus F}$; such a bid exists since $\sigma_t \neq \sigma_{t'}$. Assume without loss of generality $b^* \in B_i$.

Let $G(\{\mathscr{C}_j\}, E_{\mathtt{IF}}, E_{\mathtt{NOTIF}})$ be the conditional demand graph associated with π_i. We define the following procedure that identifies the most preferred level set for which b^* is bid. For each $\mathscr{C} \in \{\mathscr{C}_j\}$ in descending order of $\prec_i$, there are three cases:

Case 1: For every $t \leq t_1 < t_2 \leq t'$ and every $c \in \mathscr{C}$

$$B_i \cap A_c^{(t_1)} = B_i \cap A_c^{(t_2)}. \tag{8}$$

Case 2: For every $\mathscr{C}' \in E_{\mathtt{IF}}^{\mathscr{C}}$ and every $t \leq t_1 \leq t'$ there exists $c \in \mathscr{C}'$ such that

$$B_i \cap A_c^{(t_1)} \neq \emptyset, \tag{9}$$

and for every $\mathscr{C}' \in E_{\mathtt{NOTIF}}^{\mathscr{C}}$ and every $c \in \mathscr{C}'$ and $t \leq t_1 \leq t'$

$$B_i \cap A_c^{(t_1)} = \emptyset, \tag{10}$$

and for every $c \in \mathscr{C}$ and $t \leq t_1 \leq t'$

$$b^* \notin A_c^{(t_1)}. \tag{11}$$

[11] The notion of valid input extends straightforwardly to this context.

Algorithm 2 Extended Bidding Procedure

```
for $\mathscr{C}$ in order of $\prec_i$ do
    if
        $$\left|\bigcup_{\mathscr{C}' \in E^{\mathscr{C}}_{\mathtt{IF}}} (B_i \cap (\cup_{c \in \mathscr{C}'} A_c))\right| \neq \left|E^{\mathscr{C}}_{\mathtt{IF}}\right|$$
    or
        $$\left(\bigcup_{\mathscr{C}' \in E^{\mathscr{C}}_{\mathtt{NOTIF}}} (B_i \cap (\cup_{c \in \mathscr{C}'} A_c))\right) \neq \emptyset$$
     then
        $b^* := B_i \cap (\cup_{c \in \mathscr{C}} A_c)$
        $\{A_c\}_{c \in \mathscr{C}} := \{A_c \setminus b^*\}_{c \in \mathscr{C}}$
        $B_i' := B_i' \setminus b^*$
        Move to the next iteration
    end if
    if $B_i \cap (\cup_{c \in \mathscr{C}} A_c) = \emptyset$ then
        choose $c^* \in \operatorname{argmin}_{c \in \mathscr{C}} p(A_c)$
        $b^* := \min_{b \in B_i \setminus B_i'} \{b > p(A_{c^*})\}$
        if $b^*$ exists then
            add $b^*$ to $B_i'$
            add $\min_{b \in B_i} \{b > p(A_{c^*})\}$ to $A_{c^*}$
            if $|A_{c^*}| > q_{c^*}$ then
                remove $\min_{b \in A_{c^*}}$ from $A_{c^*}$
            end if
            active := true
        end if
    else if $B_i \cap (\cup_{c \in \mathscr{C}} A_c) \neq \emptyset$ then
        $b^* := B_i \cap (\cup_{c \in \mathscr{C}} A_c)$
        $b^{**} := \min_{b \in B_i \setminus B_i'} \{b \geq b^*\}$
        if $b^{**}$ exists then
            add $b^{**}$ to $B_i'$
        else
            remove $b^*$ from all $\{A_c\}_{c \in \mathscr{C}}$
        end if
    end if
end for
```

Case 3: The first two conditions of case 2 hold, and there exists $c \in \mathscr{C}$ and $t \le t_1 \le t'$ such that

$$b^* \in A_c^{(t_1)}. \tag{12}$$

Since $b^* \in \mathcal{B} \backslash F$, there must exist a level set $\mathscr{C} \in \{\mathscr{C}_j\}$ for which Case 3 holds. Let $\mathscr{C}^*$ be the most-preferred such level set. Since (12) holds and $b^* \in \mathcal{B} \setminus F$, there must exist $t \le t^* \le t'$ and $c^* \in \mathscr{C}^*$ such that

$$b^* \notin A_{c^*}^{(t^*)}, \quad b^* \in A_{c^*}^{(t^*+1)}. \tag{13}$$

Since every course more-preferred to $\mathscr{C}^*$ by player i falls into Case 1, or Case 2, $\mathscr{C}^*$ does not satisfy the condition of the **if**-statement in line 2 of Algorithm 2 in any state σ_{t_1} for $t^* + 1 \le t_1 \le t'$. Since b^* is maximal in $\mathcal{B} \setminus F$, it can never be removed from A_{c^*} in line 15. Thus the condition on line 19 will always be satisfied and the sub-condition on line **??** will never be satisfied. It follows that b^* can never be removed from an auction after timestep $t^* + 1$, i.e. for every $t_1 > t^* + 1$,

$$b^* \in A_{c^*}^{(t_1)};$$

this contradicts the fact that (7) is a cycle.

B.5 Pareto Efficiency

In this section we show that if every player only has preferences over level sets containing single courses (called *trivial level sets* in the sequel), then the extended bidding procedure produces Pareto efficient allocations.

We proceed by demonstrating, in analogy to Lemma 1, that the inexistence of trade cycles in the sense of Definition 1 guarantees Pareto efficiency.

Lemma 2. *Assume $(\pi_i)_{i \in \mathcal{N}}$ contains no non-trivial level sets. Then if an allocation $(C_i)_{i \in \mathcal{N}}$ is not Pareto efficient with with respect to the preference relation Π induced by $(\pi_i)_{i \in \mathcal{N}}$, it contains a trade cycle.*

Proof. Since we have assumed that $\{\mathscr{C}_j\}$ contains no non-trivial level sets, we abuse notation and write c to represent the unique $c \in \mathscr{C}$.

Suppose the allocation profile $(C_i)_{i \in \mathcal{N}}$ is not Pareto efficient. Then there exists a subset of players $\mathcal{N}' \subseteq \mathcal{N}$ and a feasible allocation profile $(C_i')_{i \in \mathcal{N}}$ such that

$$\cup_{i \in \mathcal{N}'} C_i' = \cup_{i \in \mathcal{N}'} C_i, \tag{14}$$

and for every $i \in \mathcal{N}$, $C_i' \Pi_i C_i$.

Consider the sequence $\{c_i\}_{i \in \mathcal{N}'}$ where

$$c_i = \max_{\prec_i} \{c \in C_i'\}. \tag{15}$$

Since $C_i' \Pi_i C_i$, it must be that $c_i \notin C_i$. Since c_i is maximal in C_i' and C_i' is feasible, it must be that $E_{\mathtt{IF}}^{c_i} = \emptyset$ and $E_{\mathtt{NOTIF}}^{c_i} \cap C_i = \emptyset$. Consider player i' such that $c_{i'} \in C_i$. Then, $c_{i'} \prec_i c_i$. It follows that $(\mathcal{N}', \{c_i\}_{i \in \mathcal{N}'})$ is a trade cycle.

Proposition 4 (Pareto Efficiency). *Assume that $(\pi_i)_{i \in \mathcal{N}}$ contains no non-trivial level sets. Then the extended proxy bidding mechanism produces an allocation which is Pareto efficient with respect to the preferences $(\Pi_i)_{i \in \mathcal{N}}$ induced by $(\pi_i)_{i \in \mathcal{N}}$.*

Proof. The argument used in the proof of Proposition 2 demonstrates that no trade cycle can occur in the final allocation profile returned by the extended proxy bidding mechanism. Thus by Lemma 2, this mechanism is Pareto efficient.

C Simulation Environment

Our simulated problems consist of 1000 students and 110 courses. Each student demands 10 courses and each course has 100 available seats.

We generate preferences in the following way:

1. For each course $c_j \in \mathcal{C}$, sample $g_j \sim \mathcal{G}(0,1)$ independently where $\mathcal{G}(0,1)$ is the standard Gaussian distribution.
2. For each player $i \in \mathcal{N}$ and course $c_j \in \mathcal{C}$, sample $v_{i,j} \sim \mathcal{G}(0,1)$ independently.
3. For each player i his *value* for each course c_j, his valuation $u_{i,j} = wv_{i,j} + (1-w)g_j$ for a constant $w \in [0,1]$.
4. Compute a rank-order-list by sorting the values $u_{i,j}$ for each player.

By choosing $w < 1$ we introduce correlation into the preferences by averaging each student's individual valuation with the global valuation. For our simulations we use $w = 1/2$ when computing the true valuation and true ranking. To compute "strategic valuations" we recompute the values $u_{i,j}$ with $w = 1/3$ (making the valuations more correlated). To produce the strategic ranking we resort the *ten most-preferred courses* according to their strategic valuations. Thus students will reorder their top ten courses to account for their overall popularity.

In our simulations, we begin by producing true rankings for each student. We then select a set of students from the population at random and have them report their strategic ranking instead of their true ranking. As we increase the fraction of students playing strategically, we strictly enlarge the set of students who are reporting their strategic ranking. This has the effect of reducing the variance of the simulation results.

The simulation environment is implemented in C#. Running on a low end desktop, each run of the proxy bidding mechanism completes in about 30 seconds. We deviate slightly from the mechanism as stated in Algorithm 1 by randomly shuffling the order we iterate through the students in the main loop (lines 7-30). We emphasize that we are only shuffling the turns in which proxies get to make bids, and are not making any change to the set of bids assigned to each player. The modification does not affect the analysis of our mechanism, however it does in practice rapidly decrease the number of rounds required for convergence.