AN ITERATIVE ONLINE AUCTION FOR AIRLINE SEATS

MARTA ESO
IBM RESEARCH DIVISION
THOMAS J. WATSON RESEARCH CENTER
PO BOX 704
YORKTOWN HEIGHTS, NY 10598
MARTAESO@US.IBM.COM

Abstract. Due to their low cost and ease of access, online auctions are a very popular way of selling perishable excess inventory in the travel industry. We analyze online auctions for airline seats where leftover seat capacity on flights between two given cities is traded. In addition to the number of tickets requested and the bid amount, bids may specify a *set* of alternative flights, each equally acceptable for the bidder. A winning bid will have all requested seats allocated on the same flights. We study an iterative mechanism where the decision to accept and reject bids and to provide minimum bid suggestion for rejected and displaced bids has to be made in real time. Each iteration of the mechanism can be thought of as a general combinatorial auction where customers bid on bundles of flights. We discuss heuristics as well as exact solution methods for solving the underlying Integer Program. We show that the model can be easily extended to incorporate more general settings. Preliminary computational results for synthetic data are also presented.

Key words. Mechanism design, Combinatorial auctions, Integer Programming

1. Introduction.

Due to their low cost and ease of access, online auctions are a very popular way of selling perishable excess inventory in the travel industry. Auctions for specific tickets, trips or even vacation packages are common at online travel agencies (e.g., Bid4Trips.com, SkyAuction.com) while other sites offer reverse auctions for the very flexible traveller (e.g., Priceline.com). Most major airlines provide online booking at their web sites but up to our best knowledge none of them offer automated online auctions (even though more than half of the tickets booked online are purchased through these sites). We believe that customer traffic at the sites of major airlines may generate enough demand so that excess seats on different flights can be pooled together and traded within the same auction (rather than trading pre-specified tickets one by one) allowing customers to express their preferences better and the airlines to achieve a more efficient allocation of resources and thus more revenue.

This paper addresses the question of designing a simple iterative auction for trading excess seat capacity for an airline. Our study originated in the following real-life application where the airline was interested in auctioning off leftover seat capacity on flights *between two given cities.* The proposed format is an iterative sealed bid auction where bidders get instant feedback, including minimum bid suggestion for declined bids. Customers bid on round-trip tickets by specifying the number of tickets they wish to purchase and a per ticket bid amount. Bidders can express their preferences by submitting several flights, assumed to be equally acceptable, for both the inbound and the outbound leg. Winners are assigned all the seats they requested on one flight in each direction. Each bid is evaluated instantly: it is either rejected or temporarily accepted. Temporarily accepted bids may be displaced during the course of the auction by other bids, in which case the bidder is immediately notified. Note that bids can be displaced indirectly; that is, displaced bids do not even have to contain any of the flights that the new bid displacing them does. A minimum bid suggestion is provided for rejected and displaced bids; these bids may be resubmitted with a higher bid amount, otherwise they are not considered again. The auction lasts for a given amount of time or until inactivity, whichever is later. Temporarily accepted bids become winners at closing, the owners of these bids are notified of their allocation and pay what they have bid.

Every iteration of the problem presented above can be modelled as a multi-unit combinatorial auction where the *bundles* customers bid on consist of the required number of tickets on acceptable flight pairs. With one bid a customer specifies an entire set of bundles which can be obtained by enumerating all the acceptable flight pairs. Combinatorial auctions received much attention lately, see [2] for a survey and [4] for background and computational complexity. Determining the winners (computing the best allocation of resources to the bidders) in a multi-unit combinatorial auction is NP-hard since the problem can be formulated as a multi-knapsack problem [3].

The above problem could be stated in a more general way. The airline could offer its leftover seat capacity on any number of flights, not just between two given cities. (Other products, like hotel rooms and rental cars could also be included in the bundles.) Customers could bid on arbitrary bundles of flights (not just pairs) and they could express a complex preference ordering on these flight combinations (rather than valuing all acceptable flight pairs the same). These generalizations do not make the winner determination problem computationally any more difficult (having a complex preference ordering might even help to avoid degeneracy of the multi-knapsack problem since these preferences translate to non-uniform objective coefficients). The remainder of the paper will mainly focus on the original problem setting but we will refer back to the extensions discussed in this paragraph from time to time.

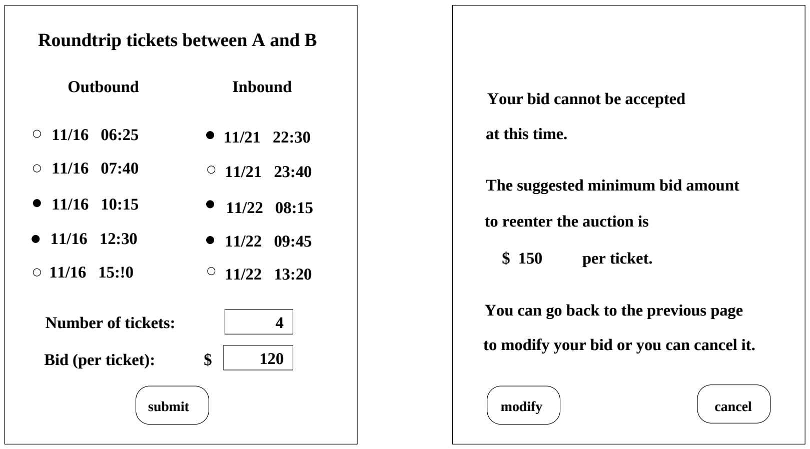
The proposed mechanism is iterative: after a bid is submitted the winner determination problem is solved and feedback is provided in the form of a minimum bid suggestion for rejected bids. In general, bid submission, solving for the best allocation and providing some feedback for bid reformulation are repeated in a loop (until termination) in an iterative auction. The auction mechanism introduced above could be improved upon in all three components of this loop. First, bids are captured in our problem in the form of flights that are equally acceptable for the bidder. More sophisticated bid descriptions that capture customer preferences better could be included as long as all the possible flight combinations along with their valuations can be efficiently enumerated. (For instance, limits on the time spent away from the departure city could be easily incorporated.) Second, since winner determination is the computationally intensive part of the loop it would make sense to execute it less often. That is, the allocation problem could be solved for batches of new bids that are collected over a given (short) period of time. This can result in more profit for the airline (since bids that are rejected when considered one by one cannot be used in a later iteration) and in more consistent bid reformulation signals (the minimum bid suggestion will fluctuate less, see Section 2.3). On the other hand it will take longer for the bidders to receive a feedback. Third, feedback for bid reformulation could be provided in a more general form. It is well known that no equilibrium item prices (i.e., prices for seats on individual flights) exist in a combinatorial auction if bidders have complementarities in their preferences (see e.g. [5]). However, suggestions could include extended flight sets that subsume those in the bid (which flights to include in the extended set requires business intelligence tools that are beyond the scope of this paper).

In a practical application of mechanism design it is highly desirable for the auction to be simple (the rules are transparent and easy to understand), efficient (the final allocation maximizes the participants' overall valuations), difficult to manipulate (by a single participant or by collusion) and incentive compatible (bidders reveal their true valuations). Of these properties simplicity and difficulty of manipulation is true for our iterative design. We believe that efficiency holds in the limit (in each iteration bids are processed after everyone has submitted a bid and the auction terminates only at inactivity) if bids reflect true valuations. However, incentive compatibility cannot be shown for this setup. We plan to investigate the economic properties further.

In what follows we will first describe the roles of the participants in the auction, then discuss the winner determination problem in detail. We provide an Integer Programming formulation and outline heuristics and exact methods for solving it. Finally we present some computational results.

2. Participants of the auction. There are three roles in the auction, the auction organizer, the seller and the bidders. The airline plays the first two roles and the customers the third. The two roles of the airline should be treated separately since it might be advantageous to have an independent auctioneer. Having an independent auctioneer could increase the bidders' trust in the auction process since the identity, number and payment information of the bidders need not be disclosed to the airline. Also, an independent auctioneer will be able to handle offers from more than one airline.

2.1. The seller. Before the auction starts the seller needs to specify for the auctioneer the items for sale and any requirements that might restrict the feasibility of allocations. The airline needs to identify seat ca-


Roundtrip tickets between A and B

Outbound	Inbound
○ 11/16 06:25	● 11/21 22:30
○ 11/16 07:40	○ 11/21 23:40
● 11/16 10:15	● 11/22 08:15
● 11/16 12:30	● 11/22 09:45
○ 11/16 15:!0	○ 11/22 13:20

Number of tickets: 4

Bid (per ticket): $ 120

Your bid cannot be accepted at this time.

The suggested minimum bid amount to reenter the auction is $ 150 per ticket.

You can go back to the previous page to modify your bid or you can cancel it.

FIG. 1. *a. Page for entering a bid. b.Rejection page with minimum bid suggestion.*

pacity and reservation prices for the flights to be offered at the auction. In our setting reservation prices are the same for all flights but they could be differentiated based on historical demand information. Requirements on the feasibility of allocations could include limits on the number of winners, or, on the proportion of winners requiring at least a certain number of tickets. Also, a limit could be placed on the number of displaced bids when a new bid is accepted.

2.2. The bidders. Interested customers visit the auction site and browse through the schedule of flights available for auction. If they would like to participate then payment information (e.g., credit card) has to be submitted since winning bids are binding. After specifying some initial information (departure and destination airports, approximate dates for travel) bidders could be presented with a page like the one sketched out on Figure 1.a where they could fine tune their flight selection. The number of tickets requested and the bid amount (optional since minimum bid suggestion is provided otherwise) are also entered through this page. The system will immediately return with a page informing the bidder of acceptance or rejection (see Figure 1.b for a possible rejection page).

If a bid is rejected the customer may choose to increase the bid amount or cancel the bid. Cancelled bids are never used again by the system. If the bid is temporarily accepted the bidder will learn only this fact and not the temporary allocation. If a bid is later displaced the auction notifies the bidder (for instance, via e-mail) and also supplies a minimum bid amount needed for re-acceptance (assuming other bidders don't change their bids). Similarly as for the rejected bids, the bidder can either increase the bid amount or cancel the bid. Note that the bidder might want to change other factors of his bid than the bid amount (after a rejection or displacement).

In the current setup this can be accomplished by cancelling the bid and submitting a new one, but any part of the bid could be modified in general (the minimum bid suggestion does not provide enough information for this, however).

We assume that bidders have private valuations for the goods and that they behave rationally. Also, if their bid is rejected or displaced bidders resubmit the bid as long as the suggested minimum bid amount is below their valuation.

2.3. The auctioneer. The main tasks of the auction organizer are to collect information from the seller, open the auction, collect the bids and respond to them, close the auction and announce the winners. The closely related problems of determining whether a bid is accepted or rejected and computing the minimum bid amount are the computationally challenging operations.

While the auction is open a list of temporarily accepted bids along with the corresponding seat assignment is maintained by the auctioneer. When a new bid arrives it is evaluated and is either accepted or rejected. Evaluation (also called winner determination) means computing the "best" allocation (which maximizes the seller's utility) given the temporarily accepted bids and the new bid. (Comparing allocations and solving the winner determination problem will be discussed in Section 3.) If the new best allocation is better than the best allocation from the previous iteration then the new bid is accepted and the best allocation is updated to be the new allocation. Otherwise the bid is rejected and the best allocation remains the same.

To compute the minimum bid suggestion we have to determine an amount for the new bid at which the new best allocation becomes preferable over the previous one. This can be accomplished by pretending that the new bid is accepted at reservation prices and computing the new best allocation. If none of the temporarily accepted bids is displaced (the allocation can be different however) then the minimum bid amount needed to enter the auction is the sum of reservation prices on the assigned flights. Otherwise there are displaced bids and the bidder must compensate the seller for the lost revenue or pay the reservation price, whichever is higher. We propose two different methods for compensating the seller: pay the full amount needed for the new best allocation to overtake the previous allocation, or pay a little more (per ticket) than the highest payment from the displaced bids.

The first method of fully compensating the seller has the advantage that successive allocations become better and better for the seller at each iteration. However, the minimum bid amounts do not necessarily increase monotonically which might be counterintuitive for bidders familiar with single-item ascending price auctions. The example of Figure 2 illustrates that fluctuations may happen with both of the minimum bid computation rules. For the example we assume that there are no reservation prices and

flights	seats	Bid 1	Bid 2	Bid 3		Bid 4	Bid 2	Bid 3
out 1	2	2		1		1		1
out 2	1		1		Bid 4		1	
in 1	2	2			replaces	1		
in 2	1		1	1	Bid 1		1	1
		\$105	\$100	??		\$220	\$100	??
compensate				\$310				\$100
overbid				\$105				\$100

FIG. 2. *Example of fluctuating minimum bid amounts.*

that the seller wants to maximize the revenue collected from the bidders. There are two outbound and two inbound flights with corresponding seat availabilities. When Bid 3 arrives the first time it has to displace both Bid 1 and Bid 2 to be accepted. Thus the bidder would need to pay 2*\$105 + \$100 = \$310 (plus a bid increment of say \$1) to get accepted. Assume that this is above the bidder's valuation and he stays away. Now Bid 4 arrives and replaces Bid 1, leaving one seat on both out 1 and in 1 unallocated. Should the new bidder resubmit his bid now, only Bid 2 would need to be displaced and the bidder pays \$100 (+\$1).

With the second method of (per ticket) overbidding the highest displaced bid the minimum bid suggestion may still fluctuate; in the above example Bidder 3 would need to pay \$105 (+\$1) in the first case and \$100 (+\$1) in the second case. In addition to this the seller's revenue might not be monotone (he collects only \$105 (+\$1) compared to \$310 if the new bid offers at least the minimum bid amount). This method may also leave more unallocated seats at the end of the auction.

As we have mentioned in the Introduction, solving the winner determination problem for batches of new bids will result in less fluctuation in the minimum bid amounts (for the full compensation case). Indeed, optimizing with more than one new bid at once rather than incrementally provides better allocations with more of the resources assigned. Several new bids together may displace a set of accepted bids, sharing the cost of compensation. Note that even if bids are evaluated in batches, the minimum bid suggestion is computed for each bid one by one, assuming that the other bids remain the same. Thus some bid amount needs to be known for all new bids. While reservation prices may play this role the resulting minimum bids would be over-estimated in this case.

We have experimented with both minimum bid computation rules (see Section 3.4) and arrived to the conclusion that the full compensation method is superior if bidders can accept fluctuating minimum bids.

3. The winner determination problem.

In this section we will focus on how to find the best allocation given the set of temporarily accepted bids and the new bid(s). We discuss how to compare different allocations, formulate the winner determination problem as an Integer Program and outline heuristics and exact methods for solving it. Computational results are also presented at the end of this section.

3.1. Comparing allocations. The first question to ask is that given a set of bids and seat (resource) availability, what objective should the seller use to compare different allocations. These objectives could be the seller's profit (the total amount collected from the bidders minus the reservation prices of the allocated seats), the number of accepted bids or the number of seats allocated. The profit of the seller is the best base for comparison since otherwise the auction could be easily manipulated and the outcome would not be efficient. (If the allocation with the higher number of accepted bids is preferred then bidders wishing to purchase multiple tickets would have a better chance of winning by submitting multiple bids for single tickets instead of one bid for multiple tickets. If the allocation with the more seats assigned is preferred then in case of high enough demand no new bids are accepted after all the seats are temporarily allocated.) An allocation is optimal if it is best with respect to the chosen objective for the given the set of bids. Other objectives than profit maximization could be incorporated as part of a composite objective or as side constraints. We used profit maximization as the seller's objective in our computational experiments.

3.2. IP formulation of winner determination. We can formulate the Winner Determination problem as an Integer Program. Let us introduce some notation first. The flights available for auction are given as $S = \{1, \ldots, m\}$ with corresponding seat availability (capacity) $c_1, \ldots, c_m$ and reservation prices $r_1, \ldots, r_m$. The flights are indexed by j. There are n bidders, bid i consists of the set of acceptable flights in both directions $E_i \subset S$ and $F_i \subset S$, the number of tickets requested t_i and the per ticket bid amount b_i. Let us enumerate the acceptable flight pairs and index them with $k = 1, \ldots, k_i$ where $k_i = |E_i||F_i|$. If the bid is temporarily accepted then the bidder will be allocated t_i seats on one flight in E_i and one flight in F_i and will pay $t_i b_i$ if the bid is a winner.

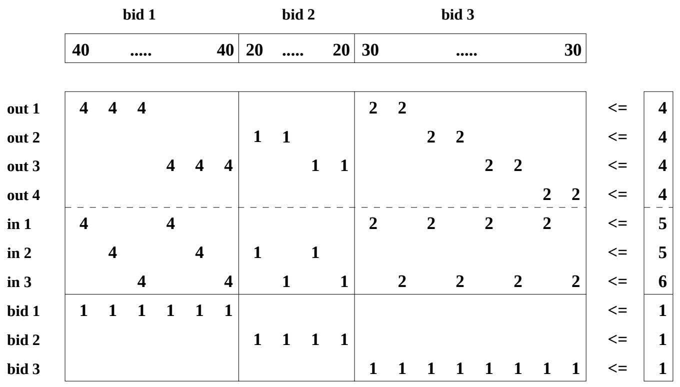
To formulate the winner determination problem we introduce binary variables for each acceptable flight pair of each bidder indicating whether the flight pair is chosen in the solution or not:

$x_{i,k} = 1$ if flight pair k is chosen for bidder $i; k = 1, \ldots, k_i$

There are two sets of constraints: the seat availability cannot be exceeded for any of the flights and at most one of the flight pairs can be chosen for any bidder. Introduce $a^j_{i,k}$ to denote the number of tickets on flight j for flight pair k of bidder i (that is, $a^j_{i,k}$ is t_i if j is one of the two flights in the pair and 0 otherwise). With this notation the resource limit constraints can be written as

$$\sum_{i=1}^{n} \sum_{k=1}^{k_i} a^j_{i,k} x_{i,k} \leq c_j, \quad j = 1, \ldots, m;$$

and the second set of constraints (choose at most one flight combination for each bidder) as



	bid 1						bid 2				bid 3									
	40		.....			40	20	.....		20	30			.....				30		
out 1	4	4	4								2	2							<=	4
out 2							1	1					2	2					<=	4
out 3				4	4	4			1	1					2	2			<=	4
out 4																	2	2	<=	4
in 1	4			4							2		2		2		2		<=	5
in 2		4			4		1		1										<=	5
in 3			4			4		1		1		2		2		2		2	<=	6
bid 1	1	1	1	1	1	1													<=	1
bid 2							1	1	1	1									<=	1
bid 3											1	1	1	1	1	1	1	1	<=	1

FIG. 3. *Integer Programming formulation of the winner determination problem.*

$\sum_{k=1}^{k_i} x_{i,k} \leq 1, \quad i = 1, \dots, n.$

We also include the integrality restrictions on the variables:

$x_{i,k} \in \{0, 1\}, \quad i = 1, \dots, n; \;\; k = 1, \dots, k_i.$

The objective is to maximize the seller's profit (payment collected from winners minus reservation prices):

$\max \sum_{i=1}^{n} \sum_{k=1}^{k_i} \left(t_i b_i - \sum_{j=1}^{m} a_{i,k}^{j} r_j \right) x_{i,k}$

We provide a sketch of the formulation in Figure 3 for a simple case where the reservation prices are assumed to be constant (so they can be omitted from the objective function). Columns of the matrix correspond to the acceptable flight pairs. Note that the flights can be divided into two sets, the outbound and inbound flights. Each flight pair will have one entry in each set, the number of tickets required by the bidder. The sketch makes it clear that this is a multi-unit combinatorial auction where the bids are for bundles of flights (we can introduce a fictitious flight for each bidder to account for the second set of constraints.) Observe that the problem matrix has $m + n$ rows, $\sum_{i=1}^{n} k_i$ columns and three nonzeros per column (thus the problem matrix is very sparse).

As we have mentioned in the Introduction, any set of flights could be offered for auction. In this case the set of rows could be subdivided so that flights in the same subset have the same origin and destination. Also, customers could bid on general bundles, probably containing at most one flight from each subset. Finally, complex preference orderings on the acceptable flight combinations could be incorporated into the objective function.

To enforce that a certain bid is accepted (which is needed for minimum bid computation, see Section 2.3) exactly one of the flight pairs must be chosen for the bid. This can be achieved by setting the constraint in the second set corresponding to this bidder to equality:

$\sum_{k=1}^{k_i} x_{i,k} = 1$ to accept bid i.

Restrictions imposed by the seller (mentioned in Section 2.1) can be added in the form of additional (side) constraints. For instance, limits on the total number of accepted bids can be expressed as

$L \leq \sum_{i=1}^{n} \sum_{k=1}^{k_i} x_{i,k} \leq U$

Note that flight pairs with non-positive objective coefficients (i.e., reservation price exceeds bid amount) will never participate in any optimal allocation. Therefore, columns (and variables) corresponding to such flight pairs can be simply left out of the formulation before solving the problem.

3.3. Solving the winner determination problem. Here we will outline heuristic and exact solution methods for the winner determination problem. We will discuss how to solve the problem for one new bid or a batch of new bids, and how to use the same algorithm to compute the minimum bid suggestion for one rejected/displaced bid. The drawback of the first two heuristics described below is that they cannot efficiently handle the winner determination problem for a batch of new bids.

Another difficulty that my arise when using heuristics is that small changes in the formulation might require the development of different specialized algorithms. This is however not the case with the heuristics discussed here.

We will present the heuristics and the exact methods in increasing order of their computational complexity.

3.3.1. Obvious inclusion. This is a very straightforward algorithm. Given an already existing allocation the new bid is accepted only if there is enough capacity to accommodate it without reallocating the already (temporarily) accepted bids. This happens when there is enough unallocated seating capacity for both segments of an acceptable flight pair. We call this the *obvious inclusion* of the new bid. If there is enough capacity on more than one acceptable flight pair then one has to be chosen based on some rule (e.g., choose the flight pair with the smallest reservation price, or choose at random). Side constraints can be incorporated by rejecting the new bid if its acceptance would violate the constraint.

If the winner determination problem is solved for a batch of new bids then order the bids (e.g., randomly, or in the order of their arrival) and accept/reject them one by one using obvious inclusion. This procedure could be repeated for different orderings of the new bids (if there are few new bids, all orderings could be considered, otherwise repeat only a constant number of times or until some time limit is reached).

Since the existing allocation cannot be modified by definition of this method, minimum bid suggestions are made only if there is enough unallocated capacity for the bid but the bid amount does not reach the reservation price.

We check for obvious inclusion *before* any other method since it is computationally very cheap. However, our experiments confirm that both the seller's profit and the allocated capacity is weak if this heuristics is used alone.

3.3.2. Simple greedy heuristics. This heuristics uses obvious inclusion repeatedly. We consider the already accepted bids and the new bid(s) together, order them randomly and accept/reject them one by one using obvious inclusion. This procedure is repeated a constant number of times or until a time limit is reached. If the overall best allocation is better than the allocation with the previously accepted bids then the new allocation replaces the old one.

To compute the minimum bid suggestion for a rejected or displaced bid consider only those orderings of this bid and the bids in the new allocation where this bid is the first. Again, a sequence of obvious inclusions is applied to these orderings and the overall best solution is used for minimum bid computation.

This heuristics is computationally inexpensive and results in a better quality solution than the previous one.

3.3.3. IP based heuristics. The compute the new best allocation the winner determination IP of Section 3.2 needs to be solved for the already accepted bids and the new bid(s). The new allocation replaces the old one if it is better. To compute a minimum bid suggestion the IP is considered with the bids in the new allocation and the rejected/displaced bid (which is forced into the solution by setting its constraint to equality).

Solving the winner determination IP to optimality might not be a practical approach if the size of the input is too large. During a typical IP solution process (like Branch-and-Bound) linear relaxations of the IP have to be solved repeatedly. Since the computational effort to solve the linear relaxations is determined by the row dimension of the problem matrix, our IP based heuristics uses only a submatrix of the original formulation. In particular, the heuristics considers only those flights (rows) that are acceptable for bidders who are currently assigned seats on flights acceptable for the new bid. We achieve this by "fixing" the allocation to other bids that currently do not have any seats assigned to them on flights the new bidder would accept. The fixing is accomplished by setting the variables corresponding to the flight assignments to one.

Any side constraint that was included in the original IP formulation can also be added to the restricted IP. This heuristics is still computationally intractable (NP-hard) in theory since it is still a multi-knapsack problem. However, the size of the problem is expected to be much smaller than the size of the original formulation and thus we expect to obtain a solution faster. Any of the methods sketched out in the next section could be applied.

3.3.4. Exact solution of the IP. If the problem size is reasonable, commercial IP packages (which employ a Branch-and-Bound backbone) can be used to solve the Integer Program to optimality. This is what we did in our experiments. However, it is likely that more sophisticated methods have to be devised especially if the set of flights offered for auction is extended or if the the number of flights or the flight capacities are large.

A generic IP solver could be significantly enhanced. Using problem specific information a custom code could be built that uses, for instance, column generation, better branching rules and warmstarting. In a column generation scheme (also called pricing) we start with a small, promising subset of the variables and add other variables to the formulation iteratively. Our implementation is under way using BCP, an LP formulation based parallel Branch-and-Cut-and-Price framework for Mixed Integer Programs. BCP is a module in COIN-OR, an Open Source initiative for the OR community (see [1]). Special branching rules can be devised that take the problem structure into consideration. For instance, instead of branching on a variable (which assigns a flight pair to a bidder) we could branch on whether a particular flight leg is assigned to a bidder or not. Warmstarting is a method to use information from a previously solved problem to speed up computations in the current one. Since the subsequent IPs differ only in a few bids the search tree of the previous optimization is likely to be a good starting point for the current problem.

3.4. Some computational results. We have used uniformly generated data in our tests. First we constructed the flight availability information (m, c_j, r_j) by hand, with the seat capacity uniformly distributed among the flights. The number of flight segments that could be chosen in both directions $(|E_i|, |F_i|)$ and the number of tickets (t_i) were both uniformly distributed between one and four. Per ticket bid amounts (b_i) were chosen uniformly from the interval $[\$200, \$300]$ The number of bids (n) to generate was chosen so that the total demand is twice or four times the supply (the first case representing low demand while the second high demand). The bids were taken in the order they were generated and either temporarily accepted or rejected. Rejected and displaced bids were never resubmitted in our tests. Reservation prices were set to \$200 for all the flights and we assumed a minimum bid increment of \$1.

We have implemented the three heuristics and the exact method for solving the winner determination problem, as discussed above. We solved the IPs using IBM OSL, a commercial solver package. Both minimum bid computation rules were implemented (fully compensate the seller and overbid the highest displaced bid). In half of the experiments we included a side constraint that limits the number of displaced bids to two in each iteration. Our tests were carried out on an IBM RS6000 43P Model 40 computer.

Figures 4 and 5 summarize our results for a small example. There are

	compensate no limit on displ			overbid no limit on displ		
	alloc cap	profit	time	alloc cap	profit	time
obvious	44.6	11.20	0	42.8	10.73	0
greedy	46.2	11.80	0.2	42.2	11.19	0.1
IP heur	48.8	12.66	6.8	46.6	12.27	8.6

	compensate displ ≤ 2			overbid displ ≤ 2		
	alloc cap	profit	time	alloc cap	profit	time
obvious	43.2	11.21	0	42.2	10.45	0
greedy	46.4	12.06	0.1	41.8	10.97	0.1
IP heur	49.0	13.07	6.1	48.6	12.69	5.7

FIG. 4. *Experiments: 50 seats, 40 bids (low demand)*

	compensate no limit on displ			overbid no limit on displ		
	alloc cap	profit	time	alloc cap	profit	time
obvious	48.2	12.02	0	48.2	12.22	0
greedy	48.8	12.65	0.4	45	12.55	0.3
IP heur	50	13.66	32.9	49.4	13.82	17.1

	compensate displ ≤ 2			overbid displ ≤ 2		
	alloc cap	profit	time	alloc cap	profit	time
obvious	49	12.60	0	48.8	12.23	0
greedy	48.8	12.76	0.3	46.2	12.59	0.3
IP heur	50	13.62	38.6	49.6	13.75	19.3

FIG. 5. *Experiments: 50 seats, 80 bids (high demand)*

50 seats available, 40 bids were generated for the low demand and 80 bids for the high demand case. The tables contain the allocated capacity, the seller's profit and the running time for the three heuristics, for both minimum bid computation rules and with or without the limit on the number of displaced bids. That is, there are altogether 12 experiments for each problem instance. The table contains averages for five instances.

Using our synthetic data we can conclude that most of the seats can be allocated at the end of the auction in the high demand case no matter which heuristics we use. However, the seller's profit improves dramatically with the more sophisticated heuristics. Obviously, running times are the exact opposite. Observe also that the allocated capacity is higher with the "compensate the seller" minimum bid computation rule than with the "overbid the highest displaced bid" rule. Profits are also significantly higher with the first rule in the low demand case. Also note that limiting the number of displaced bids does not hurt the allocated capacity or profits in the high demand case.

4. Conclusions. We have presented an iterative design for airline seat auctions which we believe could be profitable to introduce in the travel industry, it is easy to understand and is trustable for consumers and can be implemented using readily available optimization software. In the future we plan to further investigate the economic properties of the design and the computational feasibility of more complex settings outlined in the paper.

Acknowledgements Thanks to Manoj Kumar for posing the original problem and to all who read the draft of the paper and provided valuable suggestions.

REFERENCES

[1] *COIN-OR: Common Optimization Interface for Operations Research*, http://www.coin-or.org.

[2] S. de Vries, R. Vohra, *Combinatorial Auctions: A Survey*. Technical report, MEDS, Kellogg Graduate School of Management, Nothwestern University. 2000.

[3] M.R. Garey, D.S. Johnson, *Computers and Intractability*. W.H. Freeman, 1979.

[4] M.H. Rothkopf, A. Pekeč, R.M. Harstad, *Computationally Manageable Combinatorial Auctions*. Management Science, 44(8) pp 1131–1147, 1998.

[5] M.P. Wellman, W.E. Walsh, P.R. Wurman, J.K. MacKie-Mason, *Auction Protocols for Decentralized Scheduling*. Manuscript, to appear in Games and Economic Behavior.