Consiglio Nazionale delle Ricerche

Generating Realistic Data Sets for Combinatorial Auctions

A. Bonaccorsi, B. Codenotti, N. Dimitri, M. Leoncini, G. Resta, P. Santi

IIT TR-26/2002

Technical report

Dicembre 2002

Istituto di Informatica e Telematica

Generating Realistic Data Sets for Combinatorial Auctions

A. Bonaccorsi* B. Codenotti† N. Dimitri‡ M. Leoncini§ G. Resta¶
P. Santi¶

November 2002

Abstract

We consider the generation of realistic data sets for combinatorial auctions. This problem has been recognized as central to enhance the contribution of the computer science community to the field. We put forward the notions of structure and budget as main guidelines towards the generation of succinct and realistic input data. We describe a computational framework for the analysis of existing algorithms against realistic benchmarks, and use it in the context of two real world scenarios, i.e. real estate and railroad track auctions.

1 Introduction

Combinatorial auctions (CA) have recently emerged as a possible mechanism to improve the auction efficiency when many items are on sale. In a CA, a bidder can present bids on bundle of items, and thus may easily express valuations where some goods are substitutes (i.e., the bidder wants A or B but not both) or complement each other (i.e., the bidder valuates A and B together more than the sum of what she valuates A alone and B alone). The problem faced by the auctioneer is to determine the allocation of goods in order to maximize the seller's revenue. This problem, also known as winner determination, is NP-hard. The optimal solution can be obtained by solving an Integer Linear Program (see below); a very interesting approximation algorithm by Nisan and Zurel has been shown to be fast, and accurate on the average. However, this algorithm has been tested against random inputs, which need not reflect the structure of real bids.

In this paper, we have used economics as a guide for restrictions on the parameters of realistic combinatorial auctions. We generated realistic bids using these simple rules, and performed statistical experiments (simulations) that, we believe, shed light on the performance of the Nisan and Zurel heuristic, and on the economic efficiency of combinatorial auctions. Our results suggest that for some realistic situations it is feasible to obtain the optimal solution, while the Nisan and Zurel heuristic may be as much as 15% off, a penalty that may be unacceptable for many applications. We also show that in realistic situations there may be a problem with the economic efficiency of (single round) combinatorial auctions, suggesting the use of different mechanisms not because of computational difficulties, but because it may be an inadequate economic mechanism for the problem at hand.

We view our results as evidence for the importance of further studies along these lines.

1.1 Background

Let S be the set of items to be auctioned (with $m = |S|$) and let $N = \{1, 2, \dots, n\}$ denote the set of bidding agents. Agent j can place a bid for any combination $I \subseteq S$, denoted by $b_j(I)$ (we may assume $b_j(I) = 0$ if j does not submit a bid on the combination I). Also, let $\bar{b}(I)$ denote the highest bid received

*Scuola Superiore S. Anna, Pisa (Italy).
†Department of Computer Science, The University of Chicago (On leave from IIT-CNR, Pisa, Italy)
‡Dipartimento di Economia Politica, Università di Siena, Siena (Italy)
§Dipartimento di Scienze Sociali, Cognitive e Quantitative, Università di Modena e Reggio Emilia, Reggio Emilia (Italy)
¶IIT-CNR, Pisa (Italy)

for combination I. Then the optimal allocation is given by the solution vector $x = [x_I]_{I \subseteq S}$ to the following problem (see [21]).

$$\begin{aligned} &\max_x \sum_{I \subseteq S} \bar{b}(I) x_I \\ s.t. \quad & x_I \in \{0,1\}, \forall I \subseteq S \qquad (1) \\ & \sum_{I | i \in I} x_I \leq 1, \forall i \in S \end{aligned}$$

Formulation (1) of the winner determination problem is correct only when $b_j(I_1)+b_j(I_2) \leq b_j(I_1 \cup I_2)$, but can be easily modified to take subadditivity into account, too. To obtain a more compact representation, we may consider only the sets $I \subseteq S$ such that a bid is received on I (instead of all the subsets of S). It is easy to see that (1) is the same problem as *weighted set packing*, which is *NP*-hard [5].

Researchers in Computer Science and Economics have studied CAs from a number of viewpoints. One important body of work has focused on the economic efficiency of the auction mechanism[1]. The well known Vickrey auction, and the associated VCG mechanism (named after Vicrey-Clarke-Groves [2, 4, 23]) assigns the good to the highest bidder at the second highest price. This mechanism has the desirable property that an optimal strategy for a bidder is to report her true valuations, i.e., it is *truthful*. VCG can be generalized to the combinatorial setting: each bidder receives the subset of goods assigned to her in the partition which maximizes the total revenue and pays the amount she bid for the subset received less the increment to the maximum total revenue due to her participation in the auction [12]. The generalized pricing rule requires the solution to a different instance of the winner determination problem for each bidder who wins some goods. Unfortunately, if one applies this rule to a suboptimal allocation, the resulting mechanism is not truthful. Hence, in the context of VCG-based mechanisms, truthfulness requires finding the exact solution to up to $n + 1$ instances of a computationally intractable problem, where n is the number of bidders (see [9, 15, 18]).

No other mechanism is known that is truthful and as economically efficient as VCG. In view of this, and the above mentioned result on the computational complexity of the winner determination problem, we may classify the known contributions on CAs according to one of the following approaches:

1. restrict the input space, in order to reach the optimal solution in polynomial time (so that the VCG pricing rule can still be adopted);

2. keep the input space arbitrary and search for heuristics that improve the running time of finding optimal solutions (again, in order to retain the VCG pricing rule and truthfulness);

3. abandon optimality and devise efficient approximation algorithms, keeping the input space arbitrary but sacrificing truthfulness and/or efficiency (using either VCG or a different mechanism, such as the ones proposed in [9, 13]).

Following the first approach, it has been shown that optimal allocations can be computed in polynomial time if any of the following constraints is met: the bidders present bids on at most two items; bundles consist of consecutive items in a given one-dimensional ordering of items; for any two bids, they are either disjoint or one is a subset of the other [19]. [6] proves that if bidder's valuations satisfy the *gross substitutes* property (i.e., the demand for an item does not decrease when the price of other items increases), then the optimal solution can be found in polynomial time. Note that the gross substitutes property prevents bidders from expressing complementarities in their bids, so that one of the main potential advantages of combinatorial auctions may be impaired.

Algorithms for determining the optimal solution on arbitrary inputs (approach 2) have been investigated in [3, 21, 22]. The running time of such algorithms (which usually adopt a branch and bound approach and differ mainly in the techniques used to prune the search space) can be exponential on bad inputs. However, all of them enjoy the so called *anytime property*: if stopped at any time, they return a valid allocation, whose quality (i.e., seller's revenue) improves over time. Hence, optimal allocation algorithms can also be regarded as (possibly highly inefficient) approximation algorithms.

[1]The mechanism defines the bidding, good allocation, and pricing rules.

Approach 3 is adopted in a number of papers. In [8], a performance guaranteed polynomial time approximation algorithm is presented for restricted CAs (the authors impose constraints on bidder's valuations, ruling out complementarities). In general, an approximation guaranteed to be within a factor of $k^{1-\epsilon}$ from the optimum (where k is the number of bids on distinct subsets of items and arbitrary positive constant ϵ) can not be computed in polynomial time unless $NP = ZPP$ [21]. Hence, heuristics that provide good approximations (on the average) in a reasonable amount of time have been investigated (see, e.g., [3, 20, 24] and the references therein).

In most of the papers mentioned above, the problem of relevance of the proposed solutions with respect to real world auctions is not explicitly addressed. Typically, in order to guarantee optimality or computational efficiency, restrictions on one or more auction design variables are placed (e.g., highly restrictive assumptions on the number or sequence of bids or on the number of bundles). However, restrictions do not necessarily generate realistic scenarios. To the best of our knowledge, no systematic effort has been made to check the quality of the allocations generated by various algorithms against realistic sets of data.

1.2 Our contribution

In this paper we develop an experimental framework for the generation of realistic bids in relevant economic scenarios, by carefully defining criteria for the design of the auction parameters. Our framework builds upon some of the ideas and techniques presented by Leyton-Brown et al. in [10]. In particular, the case studies we will develop are based on the paths and proximity in space examples studied in [10]. A major development which distinguishes our work lies in an extensive use of economic parameters to model the auctions. In particular, while we concur with Leyton-Brown et al. that problem scaling is important when evaluating the performance of an algorithm, we determine the actual (maximum) sizes according to the type of auction. As an example, auctions where the goods on sale are expensive industrial "items" (e.g., UMTS frequencies or drilling rights) can be hardly expected to involve more than a few dozens bidders. We coherently determine the space of problem instances as a function of certain parameters that we regard as strictly dependent on the economic setup (i.e., the type of auction). The parameters we consider are: the number and value of items, the number of bidders (both absolute and relative to the number of items), and the size of bidders (buyer's wealth or, for corporate agents, total net assets).

A second difference, with respect to [10], is in the actual bid generation scheme. We first consider the problem of choosing the items to be inserted in each bid. We observe that, in a number of important cases, a plausible criterion is to first select a subset consisting of the "most wanted" items, which can then be complemented with some other items. For instance, when configuring and buying a PC on the web, one would definitely select CPU, RAM, and disk drive. But then, depending on the budget, one may have to decide whether to add, say, a printer or more RAM and a CD recording drive. This behavior, cast to the CAs scenario, lead us to define the notion of *sunflower bids* (see Appendix A for the proper definitions). In one such bid we identify a subset of items (the core) and two or more additional disjoint sets (the petals). In our case studies, we will also allow bids formed by more than one sunflower, with the constraint that at most one will be declared winner. In this way a bidder can think of alternative ways to use her full budget. A second issue, that determines the number of items requested in a bid, is the budget. In a real estate auction scenario, for instance, there could be high budget operators, such as hotel companies, whose aim is at buying many real estate pieces to build in different parts of a city. On the other hand, there could be private bidders, with limited budget, who are interested in getting a reasonable lot where to build a house. Altogether, our general scheme for generating bids will consist of: (1) defining a reasonable size (hence budget) distribution for the bidders, and (2) compute one or more (mutually disjoint) bids with the sunflower structure and whose values do not exceed the available budget. Clearly, as in [10], any buyer will have her own preferences (determined by private values that may or may not coincide with a common, theoretical value). We shall also consider possible complementarities in bundles.

As a third contribution, we perform two different sets of experiments. In the first set, we aim at comparing the exact vs approximate approach to the winner determination problem using our data sets. More precisely, we compare the results obtained with the LP-SOLVE package (freely available under Linux) with those obtained using the Nisan and Zurel's software (available for download) that implements their well-known approximation algorithm [24]. Our results confirm that the approximation algorithm is indeed very fast and also accurate, on the average. Hoverer, under the hypothesis (that we base on economic considerations) that the asymptotic analysis is not always adequate, the experiments show that optimal algorithms are often feasible. On the other hand, though very accurate in most cases, approximation algorithms may occasionally fall 15% off the optimum, a fact that can be a serious obstacle for practical goals.

The second set of experiments has to do with the economic efficiency of CAs. It is often remarked that CAs make it possible for a bidder to assign her true valuations to bundles of goods, which contributes to make a CA economically efficient. However, our results confirm that, in a realistic setting, there may be the problem of a high percentage of unassigned items (we will refer to this as to the *coverage problem*). This seems to be one of the reasons for adopting more complex mechanisms in practice, with respect to the simple one-round approach. Altogether, the experimental results seem to suggest that, in practical situations, the nature of the obstacles to the use of (one round) CAs is more economic than computational.

1.3 Paper structure

This paper is organized as follows. In Section 2 we analyze a number of important auction parameters from an economic viewpoint. We take advantage of these economic considerations especially when addressing the problem of budget distribution. In Section 3 we first present a high level procedure for the generation of sunflower bids in structured domains, i.e., when the items on sale can be regarded as the elements of a space with some structural properties (the sunflower bid structure is defined formally in Appendix A). Then, we describe in details two real life scenarios to which we apply our general bid generation technique together with the economic considerations developed in Section 2. The first such scenario is real estate (Section 3.1), while the second scenario deals with the rights to use railroad tracks (Section B). In Section 4 we show and discuss the results of the experiments performed on the input data generated for the two aforementioned scenarios. We conclude in Section 5 outlining directions for further work.

2 Economic setup

In the following, we define a number of economic parameters that we will use to generate synthetic yet realistic bids. Specifically, we understand as important the following quantities: the *number of auctioned items* and their indicative *market value* (also known as the appraised or expected resale value); the *number of bidders* and, in case of budget limited bidders, the *budget distribution*; the maximum *number of bids* a bidder may generate. Throughout this section, we assume that there is a single auctioneer, who owns all the items on sale. However, most of the considerations below extend to the case of multiple sellers.

Number of items. A reasonable figure for the number of items depends on the economic scenario. We can distinguish between two typical situations:

- auctions involving either consumer products or complex and/or expensive industrial items (e.g., spectrum frequencies, intermediate goods); in this case, it is reasonable to assume that the total number of items is of the order of few dozens (a typical value may be around 20);
- auctions involving a long list of heterogeneous products (e.g., a list of spare parts for electric power generating systems); in this case, the number of items does not exceed a few hundreds (here typical figures are 100 to 200).

In principle one cannot rule out the feasibility of auctions involving a larger number of items. However, due to cognitive limitations, producing reasonable valuations for bundles of items in this setting would be extremely difficult for the potential buyers. Hence, it is likely that in this case items are quasi-homogeneous, and the auction can be seen as a multi-unit auction of a relatively small number of item classes.

Value of items. Although (bundle) items valuations are subjective to the bidder, they can be considered to be drawn from a common distributions across bidders. In fact, it is reasonable to assume that items have a common value (also called *market value*), and that subjective bidder's valuation is a variation around this value. We therefore adopt this canonical convention, but we specify the distribution for each type of good. Since we specify a valuation function that is itself probabilistic (see below the paragraph on Bidder evaluation) we can also consider situations where benchmark values per items are uniquely determined a priori. In this case we will define the benchmark value by following a realistic procedure, and then apply a probabilistic perturbation on these values.
There are no general rules about the item market value: auctions have been applied to objects of very small value (e.g., stickers or collection cards), of medium value for consumers (e.g., wine bottles), of high value for consumers (e.g., real estate pieces), up to high value even for corporate buyers (e.g., spectrum frequencies). Useful guidelines are the following.

- If the average value is small, then it is likely that there are few items with comparatively large value (e.g., a wine bottle of a prestigious vintage), and hence the ratio r between the most and the least expensive items satisfies $r \gg 1$. In this case, the actual values can be generated using a left skewed distribution, such as the Log-Normal or Pareto distributions.

- On the other hand, when the values of the items are very high (e.g., spectrum frequencies), their market value is likely to be relatively homogeneous, and hence the ratio r is relatively small, say $r = 2$ or $r = 3$ (even though larger values are in principle possible); in this case the actual values can be generated using a symmetric distribution, such as the uniform or normal distribution (with reasonably thin tails).

Number of bidders. Large on-line auction houses such as e-Bay (market leader) or Amazon and Yahoo have millions of regular subscribers. E-Bay closes hundreds of thousands auctions per day. The number of bidders participating to each auction, however, is not known in general.

Denoting with n/m the ratio of the number of bidders to the number of items, we obtain, however, two clear bounds. In consumer online auctions one can found situations in which a few items are of interest to many potential buyers. In reported on-line auctions of collectible trading cards, "hundreds of different sellers ran auctions and each seller typically auctioned dozens of independent card lots simultaneously" ([11]). A similar situation may be found in business-to-business auctions of industrial components that may be of interest to many industries at a time (e.g. electric components). In this case the upper bound of the n/m ratio is in the order of dozens, with maximum level at, say, 100.

On the other hand, auctions of railroad segments, real estate, or complex industrial components have typically many items and few buyers. Here the ratio n/m lies below 1–2, with lower bound close to zero.

Another consideration regards the absolute number of bidders, which is in general larger for consumer than for corporate buyers. Absolute numbers not exceeding 10000 in the former case, and not exceeding 100 in the latter are realistic.

Bidder budget. This is also called the *size* of the potential buyer. In case of individuals, the size is the buyer's wealth, while for a corporate agent is total net assets. In the case of consumers, we must assume that the value of items never exceeds their size, while with corporate buyers debt in principle would be possible. However, it is better to exclude this possibility due to limitations of the financial market (excess risk). In the following, we will then always assume that the total value of the bids presented by any bidder does not exceed their budget.
It should be observed that the bidder budget plays a significant role only when the ratio of the (average) bidder size to the (average) item market value, denoted S/V, is relatively small. Suppose $S/V = 10$; then, the bidders invest a significant amount of their money in the bids (in general, above 10% of her size), and the price offered for each bundle of items must be carefully evaluated in order not to exceed the budget. Conversely, if S/V is very large (say, 10000), as it could be the case when the auctioned items have very small values (e.g., stickers), we might think of bidders with unlimited budget, and budget considerations do not apply.
A last remark regards the distribution of the bidder sizes. In general, we can assume that this distribution is left skewed, both for individual income and firm size. In fact, it is a widely accepted fact that wealth

is asymmetrically distributed. The support of the distribution can be $[0, \infty]$, taking care of setting the distribution parameters in such a way that the right tail is thin enough. As an alternative, we can use a truncated left skewed distribution with the following support:

- for income, 0-1 billion dollars;
- for assets, 0-100 billion dollars.

Bidder valuation. We model the "subjective" factors that affect the valuation made by each bidder in term of probabilistic perturbations; that is, a bidder valuation for a given bundle is given by a probabilistic perturbation of the sum of the market values of the items included in that bundle. The probabilistic distribution chosen is normal, with mean the sum of the market value and very small deviation. Super-additivity (or even sub-additivity) can be also modelled depending on the particular types of goods (see Section 3). We observe, however, that complementarities are likely to occur on bundle composed by few items (due to bidders' cognitive limitations), and that the "added value" of a complementarity is usually a fraction (hardly exceeding 10%) of the sum of the single item market values.

We conclude this section by summarizing the steps we will follow to generate realistic set of bids.

Guidelines for `Realistic bid generation`

1. Fix the number m of items, according to the economic setting.
2. Fix the value for n/m, thus defining the number of bidders.
3. Depending on the average item market value in the particular setup, choose the appropriate item values distribution (left skewed for small values, symmetric for high values).
4. If the size of bidders is relevant, choose a left skewed distribution for the bidder sizes; the parameter of the item values and bidder sizes distributions should be chosen is such a way the the ratio S/V is coherent with the economic setup.
5. For every bidder, generate a reasonable set of bids, taking care not to exceed the budget (in case that budget limitation applies to the economic setup). The price offered for every (bundle) bid is generated using the methodology described above.

3 Case studies

The abstract procedure described at the end of the previous section must be made precise not only by assigning specific values to the economic parameters, but also by clearly defining the procedure used to generate the actual items on sale and the bids. In turn, this will depend on the specific setting and the nature of the goods. In this section we outline the general ideas underlining the item and bid generation process in case the goods can be regarded as geometric elements in a Euclidean space. For the two case studies of Section 3.1 and Appendix B, the items will be rectangles and lines segments, respectively.

We first assign a (market) value to the goods on sale. For each item, we compute it by combining two figures: (1) a measure of the geometric object itself and (2) a quantity related to the (average) distance of the object from a set of "strategic points", that we will refer to as the *centers of interest*.

We then define, for each potential buyer, the set of items she is interested in. What we actually do is to compute a different ranking of the items for each bidder. To do this, we use essentially the same ideas adopted for assigning the market values to the items, but using only some (randomly picked) centers of interest, which we regard as strategic to her. We stress that the computed distances from the subset of centers of interest selected are used only to rank the items, and not to assign values to them.

Depending on the budget, we now proceed to form the actual (sunflower) bids. A fraction of the budget will go to the core and the rest to the petals. The goal is to select set of items that are close together in some sense. For instance, in the real estate scenario, bidders may try to form bids in which the whole surface (core plus any petal) is as regular as possible. Once the bundles have been selected, the offered price is determined according to the rules given in Section 2.

In the next subsections we specialize the high level bid generation schemes described above to the generation of realistic test sets in two specific settings: real estate auctions and the allocation of the rights to use railroad tracks. In doing so, we will introduce a number of technical parameters that can be used to fine tune the experiments. When we describe any experimental result, we clearly indicate the parameter values for which it holds.

3.1 Real estate auctions

Real estate sales represent a realistic case study for CAs (see [10, 16]), and one where the items auctioned can be regarded as geometric objects. We can thus apply both the economic guidelines of Section 2 and the structural considerations at the beginning of this section.

Real Estate map generation. We use a different technique with respect to that used in the proximity in space distribution of [10]. We start with an empty rectangular region R, which represents the overall auctioned land. Then, we subdivide it iteratively into smaller lots. The process is driven by two size parameters, M and μ, and a probability of further splitting p. The detailed generation algorithm is the following.

Algorithm `Lots generation`

1. Let m be the number of desired lots (the items on sale).
2. Let $R = R_0$ denote the whole auctioned area (a square region of unit side).
3. Mark R_0 as unvisited.
4. Repeat
   (a) Let R_i be an unvisited lot.
   (b) If both sides of R_i are shorter than μ, mark R_i as visited;
   (c) if both sides of R_i are larger than M, split R_i into R_j and R_k and mark them as unvisited;
   (d) otherwise, split R_i (as in 4c) with probability p.
   (e) If the number of lots is m then report SUCCESS and STOP.

   until there are unvisited lots.
5. Report FAILURE

In case R_i must be subdivided (steps 4c and 4d), we first select one of its sides, along which R_i will be cut. If one of the sides is smaller than μ, then we select the other side. Otherwise, we randomly choose one of the two sides. The selected side is cut at a random point, but with the constraint that the two resulting rectangles have sides of length at least $\mu/2$. By appropriately choosing parameters M, μ, and p we can control the average number of lots generated and some requested output parameters (e.g., how much they differ in shape and area). An example of map generated according to this procedure is shown in Figure 1, which is characterized by the following normalized parameters: $M = 0.34$, $\mu = 0.18$, and $p = 0.5$. A sunflower bid is also shown in Figure 1, which is formed by a core (regions labelled C) and three disjoint petals (lots labelled 1 to 3).

We now discuss in details the parameters defined in Section 2.

Number of items and number of bidders. We consider real estate lots as high-value items for consumer (see Section 2). Therefore, we assume that $m \leq 100$ and fix $n/m \leq 2$.

Value of items. Our item (market) value generation technique is driven by the following simple observations:

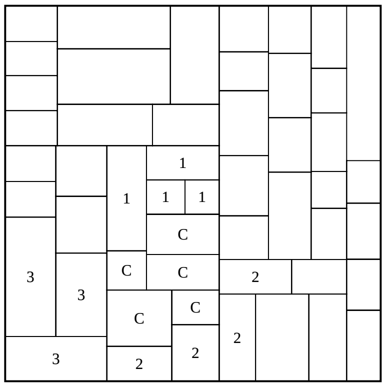

Figure 1: Real estate sample map.

- the value of a lot is proportional to its area;
- lots that are close to centers of interest (e.g., tourist spots, business centers, and so on) are more valuable.

In order to generate the item values, we first randomly and uniformly distribute c centers of interest (points in the plane) in R. For each center of interest c_i, we randomly generate two numbers k_i and s_i, which represent the relative strength of c_i and how fast the interest generated by c_i decreases with the distance. More specifically, given a lot R_j on sale, the "added value" generated by c_i on R_j is computed as:

$$V_i(R_j) = k_i \iint_{R_j} e^{\frac{(x-x_i)^2+(y-y_i)^2}{s_i}} dx\, dy \ ,$$

where (x_i, y_i) are the coordinates of the point c_i. The value of the real estate piece R_j is defined as the sum of the values generated by all the centers of interest. Formally:

$$V(R_j) = \sum_{c_i} V_i(R_j) \ . \tag{2}$$

Size of bidders. We generate the size of each bidder at random starting from Log-Normal deviates. We assume that the wealthiest bidder has enough budget to buy a fraction ϕ of all the goods on sale. To generate a budget b_i for bidder B_i we first draw a random number r_i from the Log-Normal distribution with parameters $\mu = 1$ and $\sigma = 0.4$ and scale the result so that the probability that it exceeds ϕ is negligible. Then, we assign to bidder i the budget

$$b_i = \frac{r_i \times t_c}{\lambda_\phi},$$

where λ_ϕ is the scaling factor, and t_c is the sum of the values of all the lots on sale. In the experiments reported in the next section, we set $\lambda_\phi = 20$.

Bid generation. We assume that the goal of each bidder is the acquisition of a certain amount of area as closest as possible to one of her centers of interest. Clearly, given her budget b_i, bidder B_i can buy a fraction b_i/t_c of all the area on sale, on the average. Since different lots will have different values, B_i may not be able to satisfy her wishes (in this case, the bidder will not send any offer). To limit this effect, we reduce the requirement of area by a factor $1 + u_i$, where u_i is a value drawn uniformly at random from $[0, \epsilon]$, for some small parameter ϵ. In the experiments reported in the next section, we set $\epsilon = 0.2$.

Having set the goals, the last step is the actual generation of the bids, which will have the sunflower structure. There is still the problem of "modeling" the different interests (i.e., different buyers are interested in different lots) and valuations (i.e., the subjective perception of the market values) of the bidders. We are aware that any mathematical solution to this problem is likely not to be adequate. We propose the following strategy. We have already identified c centers of interest, and used them to compute the (market) value of items. Now we use the centers of interest also to form the bidder's lot preferences. More precisely, for each sunflower she will submit, a bidder select (at random) a subset of the c centers of interest and compute, as in (2), a value for each lot on sale. She then uses this value to rank the pieces of land; the sorted (from the most to the less valuable) list will define her preferences. Once the ranking has been generated, the private item values are discarded. In fact, in accordance with the guidelines of Section 2, the bidder valuation function is based on the market value of the items, and is generated as follows. First, the valuation of each single item is determined as a uniform random perturbation of the market value. This models the fact that, due to cognitive limitations, the subjective perception of a market value can be different from its "real" value. For example, a big company can hire a team of experts to precisely estimate the market value of an item; conversely, a private bidder who wants to buy a piece of land to build her house has in general only a rough idea of its real market value. Once the valuations for single items have been defined, the valuation of a bundle bid is generated as the sum of the valuations of its components, plus a sub-additive or super-additive term that depends on the regularity of the bid area (see details below).

To summarize, each sunflower bid is generated according to the procedure outlined below.

Algorithm `Bid generation for real estate auctions`

1. Selects the best ranked item not yet considered; if its rank is below a certain threshold, this bid generation fails.
2. Choose a number of lots adjacent to the one selected in 1, until a fraction f_c of the desired area A is reached. These will form the core of the sunflower.
3. Build 2 up to 4 disjoint petals around the core, until the desired area (core plus a petal) is reached.
4. For each (core,petal) pair, compute the price to be offered as the sum of the perturbed marked values[2].
5. If the price to be offered exceeds the budget, repeat step 3. If this repeatedly fails, backtrack to step 1.

As a final remark, we observe that the lots that will form the sunflower core are selected in a such way that the shape of the core is as regular as possible.

3.2 Railroad track auctions

Due to space limitations, this section is reported in Appendix B.

4 Experimental results

We have performed a significant number of experiments, where the inputs to the allocation algorithms have been computed according to the scheme described in the previous sections. The goal of these experiments was two-fold:

- to investigate how existing optimal and approximate winner determination algorithms perform on realistic data sets. Performance is measured in terms of execution time and, in case of approximate algorithms, in terms of accuracy of the solution provided;
- to investigate the auction coverage problem (as defined in Section 1.2) in two realistic settings.

[2]Sub or super-additivity can be considered here, which we regard as dependent on the regularity of the resulting region in the bundle; clearly, regular shapes give rise to super-additive valuations, while highly irregular ones induce sub-additive valuations.

Regarding the first issue, we have taken into account the ALPH approximate algorithm proposed by Zurel and Nisan in [24], whose source code is available on the web (`http://www.cs.huji.ac.il/~zurel`). This algorithm is known to perform extremely well, both in term of speed and accuracy, on large size random instances and on the instances of [10]; thus, it is of great interest to test it against more realistic benchmarks. The performance of ALPH (expressed in terms of both accuracy of the solution and running time) has been compared to that of an optimal winner determination algorithm, namely the freely available Linux package LPSOLVE. Our current focus is more on algorithm accuracy than speed; for this reason, we have not yet compared the performance of ALPH with that of other existing packages, like CPLEX or the algorithm of [21]. For the experiments we have used a Linux machine with a 1.4Ghz Athlon processor and 1Gb of RAM.

The results of our experiments have shown that:

- ALPH is very fast (running time below 0.13 seconds) for every combination of n (number of bidders) and m (number of items) in both the real estate and the railroad case study.
- ALPH provides very good accuracy on the average (above 99% in the real estate scenario and above 99.5% in the railroad case). However, its worst-case accuracy could be an issue: in some cases (railroad setting with $n = 20$ and $m = 200$), ALPH generates less than 85% of the maximum possible revenue is some cases. We remark that worst-case accuracy is an important parameter from the auctioneer's point of view. Indeed, the seller needs to get a minimum guarantee on the auction efficiency.
- although less efficient, LPSOLVE running time resulted acceptable (below 228 seconds) for every simulated scenario.

Overall, the results of the first set of experiments have shown that computational efficiency should not be and issue in both the real estate and the railroad setting, since the optimal solution can be calculated in a reasonable time.

In the second set of experiments, we have evaluated the percentage of items sold (*coverage*) in the real estate and railroad case studies for different combinations of n and m. In order to keep the simulation time reasonable, we have used the fast ALPH allocation algorithm, which generates (in most cases) a nearly optimal solution. Overall, the results of our experiments have shown that coverage is an issue, especially in the railroad setting, where it can be as low as 31%, and it is only 81% in the most favorable scenario. We remark that, if items were to be auctioned sequentially, the coverage would be very close to 100%. This means that a CA that allocates only, say, 50% of the items is very likely to generate less revenue than the corresponding sequential auction (which is also much easier to manage). Thus, the results of our experiments suggest that in some settings (e.g., railroad track auctions), multi-round CAs are the only viable tool to improve the auction efficiency.

For lack of space, we show the details of our experiments in Appendix C.

5 Further work

This paper is a preliminary step towards the development of a test suite for the analysis of allocation algorithm against economically sound benchmarks. Much work has still to be done. On the one side, we are currently working on other case studies, which could add different insights into the problems. On the other side, we are analyzing different properties of the input data sets which could be responsible for significant variations in the running time of the allocation algorithms. In particular, it seems natural to assume a dependence, in the behavior of the integer programming solver, on the intersections between petals of sunflowers submitted by different bidders.

Acknowledgments

The authors wish to thank Janos Simon for carefully reading an earlier version of this paper, and for providing very useful suggestions.

References

[1] P. J. Brewer and C. R. Plott, A binary conflict ascending price (BICAP) mechanism for the decentralized allocation of the rights to use railroad tracks. *International Journal of Industrial Organization* 14 (1996), 857–886.

[2] E. H. Clarke, Multipart pricing of public goods, *Public Choice* 11 (1971), 17–33.

[3] Y. Fujishima, K. Leyton-Brown, Y. Shoham, Taming the Computational Complexity of Combinatorial Auctions: Optimal and Approximate Approaches, *Proc. Int. Joint Conference on Artificial Intelligence* (1999), 548–553.

[4] T. Groves, Incentives in teams, *Econometrica* 41 (1973), 617–631.

[5] R. M. Karp, Reducibility among combinatorial problems. In: R. E. Miller, J. W. Thatcher (Eds.), *Complexity of Computer Computations*, Plenum, New York (1972), 85-103.

[6] A.S. Kelso, V.P. Crawford, Job Matching, Coalition Formation and Gross Substitutes, *Econometrica* 50 (1982), 1483–1504.

[7] J. Kleinberg, Authoritative sources in a hyperlinked environment, *J. of the ACM* 46 (1999), 604–632.

[8] B. Lehmann, D. Lehmann, N. Nisan, Combinatorial Auctions with Decreasing Marginal Utilities, *Proc. 3rd ACM Conference on Electronic Commerce (EC)* (2001), 18–28.

[9] D. Lehmann, L. Ita O'Callaghan, Y. Shoham, Truth Revelation in Approximately Efficient Combinatorial Auctions, *Proc. 1st ACM Conference on Electronic Commerce (EC)* (1999), 96–102.

[10] K. Leyton-Brown, M. Pearson, Y. Shoham, Towards a Universal Test Suite for Combinatorial Auction Algorithms, *Proc. 2nd ACM Conference on Electronic Commerce (EC)* (2000), 66–76.

[11] D. Lucking-Reiley, Auctions on the Internet: What's being auctioned, and how?, *Vanderbilt University, Department of Economics*, Working Paper (1999).

[12] J.K. MacKie-Mason, H.R. Varian, Generalized Vickrey Auctions, *Tech. Rep.*, University of Michigan, July 1994.

[13] A. Mu'alem, N. Nisan, Truthful Approximation Mechanisms for Restricted Combinatorial Auctions, *to appear in Proc. AAAI* (2002).

[14] N. Nisan, Bidding and Allocation in Combinatorial Auctions, *Proc. 2nd ACM Conference on Electronic Commerce (EC)* (2000), 1–12.

[15] N. Nisan, A. Ronen, Algorithmic Mechanism Design, *Proc. STOC* (1999), 129–140.

[16] D. Quan, Real estate auctions: a survey of theory and practice, *J. of Real Estate Finance and Economics* 9 (1994), 23–49.

[17] S.J. Rassenti, V.L. Smith, R.L. Bulfin, A combinatorial auction mechanism for airport time slot allocation, *Bell Journal of Economics*, 13, no. 2 (1982), 402–417.

[18] N. Nisan, A. Ronen, Computationally feasible VCG mechanisms, *Proc. 2nd ACM Conference on Electronic Commerce (EC)* (2000), 242-252.

[19] M.H. Rothkopf, A. Pekec, R.H. Harstad, Computationally Managable Combinatorial Auctions, *DIMACS Tech. Rep. 95-09*, April 1995.

[20] Y. Sakurai, M. Yokoo, K. Kamei, An Efficient Approximate Algorithm for Winner Determination in Combinatorial Auctions, *Proc. 2nd ACM Conference on Electronic Commerce (EC)* (2000), 30–37.

[21] T. Sandholm, Algorithm for Optimal Winner Determination in Combinatorial Auctions, *Artifical Intelligence* 135 (2002), 1–54.

[22] T. Sandholm, S. Suri, A. Gilpin, D. Levine, CABOB: A Fast Optimal Algorithm for Combinatorial Auctions, *Proc. Int. Joint Conference on Artificial Intelligence (IJCAI)* (2001), 1102–1108.

[23] W. Vickrey, Counterspeculation, auctions, and competitive sealed tenders, *J. of Finance* 16 (1961), 8–37.

[24] E. Zurel, N. Nisan, An Efficient Approximate Allocation Algorithm for Combinatorial Auctions, *Proc. 3rd ACM Conference on Electronic Commerce (EC)* (2001), 125–136.

A The sunflower bid structure

Imposing a strong structure on the allowable bids is one possible way to render the implementation of a CA computationally manageable. Unfortunately, the required constraints are often quite unnatural, since they are not designed to reflect a structure that emerges from a realistic economic scenario. Actually, we suspect that the computational complexity of the winner determination problem remains high in many settings. On the other hand, we do concur with [10] that the availability of realistic data sets is crucial in order to evaluate or compare the proposed (exact or approximate) allocation algorithms. In this respect, we think that a structured approach is indeed useful to simplify the process of data synthesis. According to this viewpoint, we will impose some constraints on the data generation process in order to produce artificial bids that, in our opinion, reflect the structure of actual bids in some auction scenarios.

Sometimes, when dealing with CAs, the potential buyers tend to express a strong interest on a few items, which they regard as strategic. The values of these items could be further increased if some non-strategic items (whose value as singletons is relatively low) are used to form a bundle. For example, in the case study of Section 3.1, we assume that a prospective real estate buyer is interested primarily in one specific lot of land (or one among a few possible centers of interests); then, depending on a number of other elements, which include budget, she may be also interested in some other lots adjacent to the main target. Hence, from a generic bidder's viewpoint, the goods on sale can be divided into two categories: (1) strategic items, having a considerable singleton value, and (2) non-strategic items, which complement strategic items and are used to form high value bundles. The non strategic items can be regarded as substitutable goods.

This leads in a natural way to a sunflower bid structure, which we now formally define.

Definition 1 (Sunflower bids). *Let $\mathcal{S} = \{i_1, \ldots, i_m\}$ be a set of items, and let $S_1, \ldots, S_k$ be subsets of $\mathcal{S}$. We say that $S_1, \ldots, S_k$ have the sunflower property (or, equivalently, that $(S_1, \ldots, S_k)$ is a sunflower bid) if and only if the following holds:*

- $\bigcap_{i=1\ldots k} S_i = \bar{S} \neq \emptyset$;
- $\bar{S}_i \cap \bar{S}_j = \emptyset$, *for every* $i \neq j \in \{1, \ldots, k\}$, *where* $\bar{S}_i = S_i - \bar{S}$.

Set $\bar{S}$ is the sunflower core, while sets $\bar{S}_i$ are called petals.

In words, a family of subsets forms a sunflower bid if all the subsets share a common core, but besides the core they are pairwise disjoint.

It is easy to see that, even in the simple case in which sunflowers are made of the sole core and each bidder is allowed to bid on a single sunflower, the winner determination problem is still *NP*-hard.

B Railroad track auctions

Railroad track auctions (see [1, 10]) form another possible scenario where the items on sale can be regarded as geometric objects. Hence, as in Section 3.1, we will use the underlying geometric properties to both assign values to the auctioned goods and to generate the bids.

As in the case of real estate, the first step is the generation of the railroad map. We propose an algorithm to generate families of graphs that, in our opinion, reflects the way actual railroad networks have developed, i.e., by the merging of local and/or regional (or even national) networks, which were originally independent or very weakly interconnected.

Railroad network generation We model a railroad network as a graph, where the nodes represent cities and edges represent connections between cities. Given an Euclidean graph G, let the distance between two different connected components G_1 and G_2 of G be defined as the minimum, over all pairs (a, b) such that $a \in G_1$ and $b \in G_2$, of the distances between a and b. The generation algorithm is the following.

Algorithm `Network generation`

1. Randomly place h_1 points on a (unit) square region and connect each point with its k closest neighbors.
2. Repeat the following:
   (a) check whether the graph is connected;
   (b) if connected then goto 3;
   (c) make each connected component biconnected, and
   (d) connect each (bi)connected component with the component at minimum distance;
   (e) goto 2a.
3. Make the graph biconnected.
4. Randomly place h_2 points.
5. Connect each new point with its closest neighbor in the initial set of random points.

In the random points placement (step 1 and 4), the actual implementation obeys the constraint that no closest neighbor is closer than a given threshold ($1/\sqrt{h_1 + h_2}$ in our experiments). An example network generate according the algorithm above is shown in Figure 1 (right). The figure reports also a sunflower bid, with the core formed by the bold edges and the petals by dashed edges. The reason why we make each connected component biconnected is that a railroad network hardly contains articulation points. Note, however, that a realistic railroad network has a number of nodes of degree one; for this reason, we further add h_2 points and we connect each of them with its closest node in the biconnected part of the network being defined.

We now discuss in details the parameters defined in Section 2.

Number of items. The goods on sale are the edges of the railroad network, that we regard as expensive items. Thus, we regard $m = 100$ as a reasonable upper bound. A technical note is in order, though. Observe that the network generation algorithm presented above is parametrized by the number $h = h_1 + h_2$ of nodes (i.e., cities), rather than edges. To exactly obtain the requested number m of edges, we adopt a heuristic approach. For each value of k we (pre)compute the average number of edges generated as a function of h_1 (in steps 2 and 3). Then, when m edges are required, we choose h_1 so that the average number of edges generated is between $(1 - l_1)m$ and $(1 - l_2)m$, with $0 < l_2 < l_1 < 1$. If this actually happens, we then add the appropriate number h_2 of nodes of degree one. Otherwise (with low probability) we discard this generation.

Number of bidders. We set n in the range $5 - 40$.

Value of items. We begin by observing that the value of a train path depends not only on its length (which also implies greater operation and maintenance costs), but also on the expected number of passengers that travel on it in the unit of time. In turn, this depends on the importance of the cities touched by that path. The value of a city does depend on "external" (with respect to the railway system) factors but also, in a recursive way, on the value of the nodes which is connected to. Hence we assign market values to the edges of the graph by first assigning values to the nodes. After that, we assign an edge $e = (c_1, c_2)$ the value $V(e) = (V(c_1) + V(c_2))d(c_1, c_2)$, where V denotes the market value function (for both nodes and edges) and $d(\cdot, \cdot)$ denotes the Euclidean distance.

The problem then reduces to assigning values to the cities. Our approach consists essentially in applying the (truncated) power method to the adjacency matrix of the graph, starting from a normalized (1-norm) vector. Node i will then get the value of the corresponding entry of the normalized eigenvector (more precisely, the approximation obtained after four iterations). Note that this argument is commonly used in different contexts (e.g., to compute the "authority weight" of web sites [7]).

Size of bidders. The size of bidders is generated as in Section 3.1.

Bid generation The actual bid generation follows the same general rules defined in Section 2 with the technicalities introduced in Section 3.1. However, differently from the real estate case, where the bidder's goals can be stated in terms of area, here the goal is simply the acquisition of some paths as close as possible to one of the bidder's centers of interests. Thus, for each sunflower bid, a bidder B_i randomly selects a few centers of interests and ranks all the edges by calculating their average distance[3]. Now, given a budget and an ordering of the edge set, a bidder formulates a bid according to the algorithm given below.

Algorithm `Bid generation for railroad track auctions`

1. Selects the best ranked item not yet considered; if its rank is below a certain threshold, this bid generation fails.
2. Select other edges that form (together with the edge devised at step 1) a path in the graph; stops when at least a fraction f_c of the budget is used. This path is the core of the sunflower.
3. Select a few edges that form a path incident to the core. Each connection is a petal of the sunflower.
4. Assign a value to a sunflower by computing the sum of the bidder's valuations of the edges included in the sunflower. As in Section 3.1, such valuations are determined by slightly (and randomly) perturbing the market prices[4].
5. If the budget constraint is violated, repeat step 3. If this happens repeatedly, go back to step 1.

C Experiments

Winner determination algorithm performance. In the first set of experiments we have tested the performance of the optimal winner determination algorithm LpSolve and of the approximate ALPH algorithm [24] with the data sets generated according to our real estate and railroad settings.

First we have considered the real estate scenario. We have generated several data sets, with m, the number of items, which in this case are real estate pieces, equal to 50 and 100, and n, the number of bidders, ranging from 10 to 100. Going far beyond these figures seems unrealistic for this case study. For each combination of n and m, we have generated 100 different data sets, where every bidder presents exactly one sunflower bid, and executed both LpSolve and ALPH on each data set.

The results of our experiments are reported in Tables 1 and 2. ALPH turned out to be very fast for every combination of n and m: even in the most critical scenario ($n = m = 100$), the worst-case ALPH running time was 0.04 seconds. The average ALPH accuracy was always above 98.9%. However, the worst-case accuracy could be an issue: when the number of bidders is above 10, ALPH generates less than 92% of the maximum possible revenue is some cases. We remark that worst-case accuracy is an important parameter from the auctioneer's point of view. In fact, the seller is interested in having a minimum guarantee on the auction efficiency.

LpSolve running time, although relatively high for the largest problem instances, never exceeded a reasonable bound: when $n = m = 100$, the worst-case running time was 117.9 seconds, with an average of only 7.42 seconds. Hence, we can conclude that in our real estate scenario computational efficiency should not be an issue, since the optimal solution can be calculated in a reasonable time.

We have then considered the railroad scenario. We have generated several data sets, with m, the number of items, which in this case are railroad tracks, ranging from 50 to 200, and n, the number of bidders, ranging from 5 to 20. In this scenario, going beyond the number of 20 bidders seems unrealistic. For each combination of n and m, we have generated 100 different data sets, where every bidder presents exactly one sunflower bid, and executed both LpSolve and ALPH on each data set.

The results of our experiments for 50 and 200 items are reported in Tables 3 and 4. Also in this case ALPH turned out to be very fast for every combination of n and m: in the most critical instances ($n = 20$

[3]Here we mean the distance in the graph, not the Euclidean distance. More precisely, the distance between two edges e_1 and e_2 is defined as the minimum distance in the graph between endpoints of e_1 and e_2.

[4]The value of the bid generated can be increased by a small random amount (not more than 10% of the value computed so far) as a function of the rank of the edges included. This can actually produce super-additive bids.

	n=10		n=20		n=50		n=100	
	Avg	Worst	Avg	Worst	Avg	Worst	Avg	Worst
LPS run time	< 0.01	0.01	< 0.01	0.03	0.04	0.3	0.15	1.56
ALPH run time	< 0.01	0.01	< 0.01	0.02	0.01	0.02	< 0.01	0.01
ALPH accuracy (%)	99.91	96.88	99.75	95.86	99.28	93.3	98.97	94.46

Table 1: LpSolve and ALPH running times, and ALPH accuracy, in the real estate scenario with 50 items on sale and different number of bidders. Running times are expressed in seconds.

	n=10		n=20		n=50		n=100	
	Avg	Worst	Avg	Worst	Avg	Worst	Avg	Worst
LPS run time	< 0.01	0.02	0.01	0.11	0.37	3.02	7.42	117.9
ALPH run time	< 0.01	0.01	0.01	0.02	0.01	0.02	0.02	0.04
ALPH accuracy (%)	99.90	96.55	99.56	91.39	99.17	93.28	99.03	91.56

Table 2: LpSolve and ALPH running times, and ALPH accuracy, in the real estate scenario with 100 items on sale and different number of bidders. Running times are expressed in seconds.

and $m = 200$), the worst-case ALPH running time was 0.23 seconds. Compared to the real estate case study, the average ALPH accuracy was slightly higher (always above 99.5%). However, the worst-case accuracy was more critical for certain combinations of n and m. For example, when $n = 20$ and $m = 200$, ALPH generates less than 85% of the maximum possible revenue is some cases.

For what concerns the running time of LpSolve, we have observed that, also in this setting, it never exceeds a reasonable bound: when $n = 20$ and $m = 200$, the worst-case running time was 227 seconds, with an average of only 6.3 seconds. Hence, also in the railroad setting computational efficiency should not be an issue.

Auction coverage. In the second set of experiments, we evaluated the auction coverage, which, we recall, is defined as the percentage of items sold.

As in the previous set of experiments, we have considered both the real estate and the railroad settings. In the real estate scenario, we have generated data sets with 50 and 100 items, and number of bidders ranging from 5 to 100 in the first case, and from 10 to 200 in the second case. In the railroad track setting, we have generated data sets with 50 and 100 items, and number of bidders ranging from 5 to 20 when $m = 50$, and from 10 to 40 when $m = 100$. In both scenarios, we have considered a further parameter in the data generation phase, i.e. the number of sunflower per bidder presented, which varies from 1 to 8. We remark that sunflowers presented by a single bidder are XORed, i.e., any bidder can win at most one sunflower. In order to keep the simulation time reasonable, we have used the fast ALPH allocation algorithm, which generates a nearly optimal (in most cases) solution. The results of our experiments are reported in Tables 5 and 6 for the real estate setting, and in Tables 7 and 8 for the railroad setting. As in the previous section, results are averaged over 100 experiments.

In the case of real estate, the coverage is proportional to the n/m ratio. Moreover, given any value of n/m, coverage increases with the number of sunflowers per bidder presented. However, unless the number of bidders is very low (5–10), allowing more than 5 sunflowers per bidder provides little benefit to the auction coverage. When n/m is far below 1 (e.g., 0.1), a considerable percentage of items remains unsold (up to 54% when $m = 50$, $n = 5$ and every bidder bids on a single sunflower), even when the

	n=5		n=10		n=20	
	Avg	Worst	Avg	Worst	Avg	Worst
LPS run time	0.01	0.02	0.01	0.09	0.06	0.26
ALPH run time	0.02	0.09	0.03	0.14	0.07	0.13
ALPH accuracy (%)	99.99	99.90	99.91	94.40	99.76	95.71

Table 3: LpSolve and ALPH running times, and ALPH accuracy, in the railroad scenario with 50 items on sale and different number of bidders. Running times are expressed in seconds.

	n=10		n=20	
	Avg	Worst	Avg	Worst
LPS run time	0.17	0.77	6.30	227.77
ALPH run time	0.08	0.22	0.09	0.23
ALPH accuracy (%)	99.99	99.85	99.66	84.94

Table 4: LpSolve and ALPH running times, and ALPH accuracy, in the railroad scenario with 200 items on sale and different number of bidders. Running times are expressed in seconds.

	number of sunflowers per bidder							
n	1	2	3	4	5	6	7	8
5	45.76	58.00	65.00	69.80	72.74	74.46	77.48	79.08
10	62.80	75.78	81.16	84.34	87.48	88.80	90.10	91.62
25	80.56	88.74	92.54	94.26	95.34	96.30	96.58	96.94
50	89.42	94.90	96.32	96.98	97.70	98.20	98.06	98.38
75	93.00	96.72	97.60	98.08	98.36	98.54	98.74	98.34
100	94.90	97.20	98.14	98.44	98.54	98.34	98.76	98.68

Table 5: Auction coverage (expressed as percentage of item sold) in the real estate scenario with 50 lots on sale and different number of bidders.

number of sunflowers per bidder is 8. Hence, in these cases multi-round CAs should be used to increase efficiency. When the ratio n/m is 1 and above, the auction coverage is in general good (above 90%). However, obtaining 100% of coverage in a single CA round seems to be very difficult: our results clearly show a coverage convergence phenomenon, which is around 98% when $m = 50$ and around 95% when $m = 100$.

Coverage resulted much more critical in the railroad tracks case study: even in the more favorable scenario ($n = 20$, $m = 50$, and 8 sunflowers per bidder), almost 20% of the items remain unsold. Hence, in this setting multi-round mechanisms seem to be the only way to improve the auction efficiency.

	number of sunflowers per bidder							
n	1	2	3	4	5	6	7	8
10	60.07	70.17	74.98	77.15	79.24	81.17	82.36	82.88
20	70.90	77.92	81.81	84.24	85.78	86.63	87.32	88.08
50	81.08	86.29	88.15	90.02	90.07	90.69	91.54	91.83
100	86.39	89.75	91.03	92.15	92.97	93.14	93.05	93.19
150	88.92	91.63	92.36	94.02	93.53	93.66	94.08	94.1
200	90.24	92.53	94.05	93.97	94.58	94.86	95.05	95.17

Table 6: Auction coverage (expressed as percentage of item sold) in the real estate scenario with 100 lots on sale and different number of bidders.

	number of sunflowers per bidder							
n	1	2	3	4	5	6	7	8
5	31.10	39.44	44.80	49.46	52.50	54.00	56.48	58.82
10	41.48	53.02	58.48	62.32	65.62	68.12	69.72	71.48
20	54.10	64.06	69.64	72.92	76.02	77.94	79.70	81.18

Table 7: Auction coverage (expressed as percentage of item sold) in the railroad scenario with 50 tracks on sale and different number of bidders.

	number of sunflowers per bidder							
n	1	2	3	4	5	6	7	8
10	31.32	39.16	43.02	46.42	47.77	49.58	51.55	52.57
20	39.74	47.02	50.74	53.71	55.44	56.71	57.92	58.57
40	47.24	53.95	57.17	59.48	60.79	62.14	63.45	64.71

Table 8: Auction coverage (expressed as percentage of item sold) in the railroad scenario with 100 tracks on sale and different number of bidders.