

# Preferences of Agents in Decentralized Task Allocation

Mark Hoogendoorn<sup>a</sup> Maria Gini<sup>b</sup>

<sup>a</sup> *Vrije Universiteit Amsterdam, Department of Artificial Intelligence, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands, email: mhoogen@cs.vu.nl*

<sup>b</sup> *University of Minnesota, Department of Computer Science and Engineering, 200 Union Street SE, Minneapolis, MN 55455, United States of America, email: gini@cs.umn.edu*

The ability to express preferences for specific tasks in multi-agent auctions is an important element for potential users who are considering to use such auctioning systems. This paper presents an approach to make such preferences explicit and to use these preferences in bids for reverse combinatorial auctions. Three different types of preference are considered: (1) preferences for particular durations of tasks, (2) preferences for certain time points, and (3) preferences for specific types of tasks. We study empirically the tradeoffs between the quality of the solutions obtained and the use of preferences in the bidding process, focusing on effects such as increased execution time. We use both synthetic data as well as real data from a logistics company.

Keywords: auctions, agent preferences, scheduling tasks

## 1. Introduction

Auctions are used in multi-agent systems, among other things, to perform allocation of tasks (see e.g. [14] and [15]). Such reverse auctions, where the buyer is the auctioneer, can be of a combinatorial type, allowing for bidding on bundles of tasks. Sandholm [13] notes that reverse auctions are not economically efficient because optimal bundling depends on suppliers preferences, which traditionally cannot be expressed. Enabling the agents to express the preferences of their users is an important requirement for actual companies and people to use agents for bidding.

In this paper we propose a concrete preference function to be used by an agent to express preferences over tasks. This function expresses preferences for specific properties of tasks and it is used in a decentralized task allocation setting. We introduce a bidding algorithm, where an agent bids on its most preferred tasks that are feasible given its current commitments. This algorithm

uses a pricing mechanism which depends on the actual cost to perform the tasks and on the preference for the task. The influence of preferences on the price can be varied by setting a parameter (look at the role of the parameter  $p$  in the algorithm in Section 3.5).

Using this algorithm, we investigate the impact of preferences upon other aspects of task execution, such as execution time. We use both synthetic as well as real data from a logistics company and present results for four types of market situations (resource shortage vs. resource overflow, wide vs. narrow time windows) with different settings of the parameter  $p$  that affects the influence preferences have on price.

This paper is organized as follows. First, the auctioning system used throughout the paper is introduced in Section 2. Section 3 introduces a function to express preferences and a bidding algorithm based upon such preferences. Experiments to evaluate the bidding algorithm and to study the trade-off between preferences and efficiency of task execution are presented in Section 4. Section 6 discusses related work, and finally, Section 7 concludes the paper.

## 2. The MAGNET System

The approach we present exploits some unique features of the MAGNET [4] system that allows autonomous agents to negotiate over coordinated tasks with precedence and time constraints. The MAGNET system consists of: (1) a *customer* agent, which puts tasks up for auction. The tasks have time constraints and other restrictions; (2) *suppliers* agents, which bid on the tasks and execute them if awarded; and (3) the MAGNET market server, which keeps track of the activities of the agents and of the auctions. The main in-

interactions between agents in the MAGNET system are as follows:

- A customer agent issues a *Request for Quotes* (RFQ) which specifies the tasks, their precedence relations, and a time line for the bidding process. For each task, a time window specifies the earliest time the task can start and the latest time the task can end.
- Supplier agents submit bids. A bid includes one or more tasks, a price, the portion of the price to be paid as a non-refundable deposit, and the estimated duration and time window for task execution. Bids reflect supplier resource availability and constrain the customer’s scheduling process.
- The customer agent decides which bids to accept. Each task needs to be mapped to one bid and the constraints of all awarded bids must be satisfied in the final schedule. In MAGNET the customer can choose from a collection of winner-determination algorithms (A\*, IDA\* [2], simulated annealing, and integer programming [3]).
- The customer agent awards bids and specifies the work schedule.

### 3. Preference Algorithm

In the bidding algorithm we propose, price is used as a mechanism to express preferences for tasks. Preferences in our case can be a combination of the following: (1) a preference for tasks of a particular duration (e.g. I hate performing very short tasks), (2) a preference for tasks at particular times during the day (e.g. I love getting up early in the morning, so give me tasks that ought to start early in the morning), and (3) a preference for particular types of tasks (e.g. I really hate to perform a task like that).

We show how to express these preferences and how to combine them. The preference for a task is referred to as  $\phi_{task}$ , which we express using a real number in the interval [0,1]. Hereby,  $\frac{1}{2}$  indicates a neutral preference, 0 is not preferred, and 1 is fully preferred. Since humans typically do not think in terms of a number when specifying preferences, we provide for each of the preference types covered a more intuitive formulation using piecewise linear functions, as explained next. The specifics of how preferences are computed could be adapted for different domains, while keeping the approach.

#### 3.1. Preferences for Duration

Let the preference to perform tasks of a certain duration be an integer. Such an integer can indicate either a minimum or a maximum duration. Let  $d_{min}$  be the minimum duration you want a task to last, i.e. you want the task to last longer than  $d_{min}$ . Durations below  $d_{min}$  are not preferred. If the duration is precisely  $d_{min}$  the preference is  $\frac{1}{2}$ , i.e. neutral. Let  $d_{close}$  be an integer that indicates how longer than  $d_{min}$  the task should last to be fully preferred. Tasks with duration in the range  $[d_{min}, d_{min} + d_{close}]$  are more preferred than neutral, but not fully preferred. Any duration longer than  $d_{min} + d_{close}$  is fully preferred. Then the preference  $\phi_{duration}$  of a task with duration  $d_{task}$  can be calculated using a piecewise linear function as follows:

- preference for minimum duration  $d_{min}$ :
  - if  $d_{task} \geq d_{min}$  then
 
$$\phi_{duration,task} = \frac{1}{2} + \min\left(\frac{1}{2} \times \frac{d_{task} - d_{min}}{d_{close}}, \frac{1}{2}\right)$$
  - if  $d_{task} < d_{min}$  then
 
$$\phi_{duration,task} = \max\left(\frac{d_{task}}{d_{min}} - \frac{1}{2}, 0\right)$$
- preference for maximum duration  $d_{max}$ :
  - if  $d_{task} \leq d_{max}$  then
 
$$\phi_{duration,task} = \frac{1}{2} + \min\left(\frac{1}{2} \times \left(\frac{d_{max} - d_{task}}{d_{close}}\right), \frac{1}{2}\right)$$
  - if  $d_{task} > d_{max}$  then
 
$$\phi_{duration,task} = \max\left(\frac{d_{max}}{d_{task}} - \frac{1}{2}, 0\right)$$

#### 3.2. Preferences for Time Points

Let the preference for particular time points be indicated by a time of the day (e.g. 6.30 a.m.). Such a preference can indicate that the time needs to be before a particular time point  $t_{before}$ , or after a time point  $t_{after}$ . Let  $t_{close}$  indicate a time which is considered close to a particular time point. Again, the preference for a task which is precisely at the specified time point  $t_{before}$  or  $t_{after}$  is  $\frac{1}{2}$ , i.e. preference neutral. The preference for a given start time  $t_{task}$  can now be calculated as follows (note that for calculations using time points these are represented in seconds of the day):

- preference for a task time before  $t_{before}$ :
  - if  $t_{task} \leq t_{before}$  then
 
$$\phi_{time,task} = \frac{1}{2} + \min\left(\frac{1}{2} \times \left(\frac{t_{before} - t_{task}}{t_{close}}\right), \frac{1}{2}\right)$$
  - if  $t_{task} > t_{before}$  then
 
$$\phi_{time,task} = \max\left(\frac{t_{before}}{t_{task}} - \frac{1}{2}, 0\right)$$
- preference for a task time after  $t_{after}$ :

if  $t_{task} \geq t_{after}$  then  
 $\phi_{time,task} = \frac{1}{2} + \min(\frac{1}{2} \times \frac{t_{task} - t_{after}}{t_{close}}, \frac{1}{2})$   
 if  $t_{task} < t_{after}$  then  
 $\phi_{time,task} = \max(\frac{t_{task}}{t_{after}} - \frac{1}{2}, 0)$

### 3.3. Preferences for Tasks

The last way to express preferences is for specific types of tasks. Let  $type_{task}$  be the type of a task. The type of a task is specified by means of a range of integers, whereby integers are ordered based upon similarity of the tasks. For example, if tasks are represented on the interval  $[0, 100]$ , then the task identified with 1 is completely different from the task identified with 100, but has great similarity with the task identified with 2. Let the preferred tasks include a certain range of tasks  $[type_{lower}, type_{upper}]$ . Furthermore, let  $type_{close}$  be an integer that expresses when a task is close to another task. The preference is calculated as follows:

- if  $(type_{lower} \leq type_{task} \leq type_{upper})$  then  
 $\phi_{type,task} = 1;$
- if  $type_{lower} > type_{task}$  then  
 $\phi_{type,task} = \min(\frac{type_{close}}{type_{lower} - type_{task}}, 1)$
- if  $type_{upper} < type_{task}$  then  
 $\phi_{type,task} = \min(\frac{type_{close}}{type_{task} - type_{upper}}, 1)$

### 3.4. Combining Preferences

The preferences specified above are usually combined. We assume that preferences are independent of each other and we combine them using a weighted sum of the preferences, setting the weight to 0 if a preference is not expressed. The weights are set by the user according to their relative importance.

$$\phi_{task} = w_{duration} \times \phi_{duration,task} + w_{time} \times \phi_{time,task} + w_{type} \times \phi_{type,task},$$

where  $w_{duration} + w_{time} + w_{type} = 1$

### 3.5. Bidding Algorithm with Preference for Tasks

We assume that the supplier agent owns a single resource with a specific capability (with which, of course, a number of different task types can be performed, as explained earlier). Furthermore, the resource has an availability slot (i.e. a begin and end time) as well as a specific  $type_{begin}$  when the resource

is initially setup and an end  $type_{end}$  at which the use of the resource needs to end. The supplier agent maintains a schedule of the tasks planned for its resource.

We now present a bidding algorithm that takes preference values  $\phi_{task}$  into account. The algorithm is a greedy algorithm, supplier agents bid upon as many tasks as feasible to maximize the usage of their resource. The algorithm uses a parameter,  $p$ , to vary the influence of the preference upon the eventual price bid.

The tasks within an RFQ are first ordered based upon their preference. If some tasks have identical preferences, they are ordered according to the earliest start time specified in the RFQ. We assume that there exists a function  $SwitchTime: TASKTYPE \times TASKTYPE \rightarrow DURATION$  that calculates the switching time from one task type to another (when it can be performed on the resource). Furthermore,  $ExecTime: TASKTYPE \rightarrow DURATION$  expresses the time needed to perform the task.

#### Bidding Algorithm

The bidding algorithm allocates tasks to empty slots in the schedule for the resource in a greedy way, taking one task at a time in order of preference and trying to fit it in the earliest empty slot available in the schedule. The algorithm does not guarantee an optimal allocation but is simple and fast.

Let  $LF_{prior}$  be the latest finish time in the current schedule of the task which is scheduled just before the current empty slot (or the schedule start time if no such task exists). Let  $type_{prior}$  be the type of the previous task (or  $type_{begin}$  in case of no prior task),  $LS_{next}$  be the latest start time of the next task (or the schedule end time if no such task exists), and  $type_{next}$  be the type of the next task (or  $type_{end}$  in case of no next task).

For each task  $task_{current}$  in the ordered list of tasks  
 if  $task_{current}$  can be done using the resource  
 for each empty slot  $s_{current}$  in the schedule  
 if  $task_{current}$  fits in  $s_{current}$  (see below)  
 then insert  $task_{current}$  in the bid,  
 add its time parameters to the schedule,  
 and compute the bid price (see below)  
 else if  $LF_{current} \leq LF_{next}$   
 then break  $\{task_{current}$  is abandoned}

To see if  $task_{current}$  fits in slot  $s_{current}$  in the schedule, check if the following holds, where ES indicates the earliest start time, LS the latest start time, and LF the latest finish time of a task:

$$\begin{aligned}
& [(LF_{prior} + \text{SwitchTime}(\text{type}_{prior}, \text{type}_{current})) \\
& \leq LS_{current}] \\
& \wedge [(ES_{current} + \text{ExecTime}(\text{type}_{current}) + \\
& \text{SwitchTime}(\text{type}_{current}, \text{type}_{next})) \leq LS_{next}] \\
& \wedge [LF_{prior} + \text{SwitchTime}(\text{type}_{prior}, \text{type}_{current}) + \\
& \text{ExecTime}(\text{type}_{current}) + \\
& \text{SwitchTime}(\text{type}_{current}, \text{type}_{next}) \leq (LS_{next})]
\end{aligned}$$

Figure 1 shows graphically the constraints.

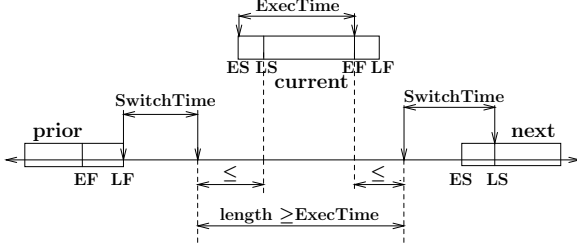


Fig. 1. Constraints for a task to fit into an empty slot.

The price of a task in the bid is computed as follows (note the use of the parameter  $p$ ):

$$\begin{aligned}
\text{price}_{task} = & (1 + (p \times (1 - \phi_{task}))) \times \\
& [\text{SwitchTime}(\text{type}_{prior}, \text{type}_{current}) + \\
& \text{ExecTime}(\text{type}_{current}) + \\
& \text{SwitchTime}(\text{type}_{current}, \text{type}_{next},)]
\end{aligned}$$

We have shown earlier how to calculate the value of  $\phi_{task}$  for the different types of preferences. This price equation assumes a certain standard price for each minute of time spent. In case these costs vary, the cost per minute can be included as an additional parameter.

## 4. Experimental Setup

We now describe the effect of adjusting the parameter  $p$  in the bidding algorithm defined above. Furthermore, we study the effect of the preferences on the duration of task execution, which is an indicator of how efficiently the tasks are being performed. Together these form the utility function of the suppliers. Of course it is expected that having more preferences awarded will result in a less efficient execution. We are interested in assessing the severeness of these effects. We performed experiments using synthetic data, and experiments using a real dataset obtained from a trucking company.

### 4.1. Experimental Setup with Synthetic Data

We start by describing the parameters in the setup with synthetic data, and specify the actual settings used. There are many parameters that can influence the results. Many of them influence the difficulty of the task allocation problem in general. These include:

1. The number of tasks to be allocated.
2. The number of resources available.
3. The ratio between the resources required to perform the tasks and the availability of those resources (e.g. one resource might be more scarce than another). This also includes the specification of duration of tasks, switching time, and initial resource settings.
4. The tightness of the time windows specified in the tasks. Wider time windows allow more flexible scheduling of tasks, therefore finding a solution is easier.

The preference value itself is influenced by other parameters, including the following:

1. The parameter setting for the preference functions (e.g. what is considered to be a close-by task, the stricter this norm is, the more easily preferences can be met).
2. The variation of tasks that exist (i.e. more variation means that it will be easier to get your preferences met).

Finally, other parameter settings can be varied, such as the number of iterations, and the value of the parameter  $p$ , which is used in the bidding algorithm to determine the influence of preferences on price.

#### 4.1.1. Parameter Settings Used

We set the parameters of the preference functions and the variety of tasks to fixed values. This means that the preference function itself remains constant over time, so that the influence of the parameter  $p$  is the only variation regarding the preference function.

We used several variations of the difficulty of the task allocation throughout the experiments. In particular we considered a market where more than sufficient resources are available (*overflow*) versus a market where resources are

insufficient (*shortage*). Furthermore, the tightness of the time windows was varied by either setting them very tight or setting them wide. More precisely, the following parameters have been used to affect tightness of tasks:

1. the number of tasks was fixed to 10.
2. the number of resources available varied between 12 (tight market) and 50 (plenty of resources available).
3. the ratio between the required resources to perform tasks and the availability of those resources was fixed. We had three types of resources, each generated with equal probability. The number of different tasks per resource was set to 9999. The maximum time to change from one task to another was set to 100 minutes. Task types were generated in a random fashion with an equal probability as well.
4. The tightness of the time windows specified in the tasks was varied between just sufficient time to perform the task to twice the time needed plus two full hours.

The parameter setting for the preferences are set so that initially the preference for tasks is around 60%, equally divided over the different preferences. Each of the agents is assigned one type of preference at random. The parameter  $p$  varied between 0 and 5.

#### 4.2. Trucking data

Besides synthetic data, we tested our approach using a real company dataset from the trucking domain. The dataset consists of a number of container transports that need to take place. Tasks require a certain transportation from one zip code (the pickup location) to an intermediate location (the delivery location), ending at a third location (the return location). Therefore, a task description does not consist of one integer specifying the task, as before, but of three integers.

Furthermore, each task is associated with a certain earliest start time and a particular deadline at which the container needs to be returned at the return location. In addition to the containers that require transportation, the dataset also specifies which trucks are available. These can carry one container at a time (so only one type of resource is available), and have a certain availability slot when the truck becomes available, and when the truck needs to be returned. A location is also specified where the truck starts, and where it has to end. This nicely maps to the algorithm specified.

The performance time is now defined as the time to go from the pickup to the delivery location, plus the time to go from the delivery location to the return location. The switching time is no longer an artificial time, but it is the actual driving time from one zip code to another.

The characteristics of the dataset are shown in Table 1 and Table 2. The only artificial data which we have generated are the preferences of the various trucks. This is done according to the method mentioned for the synthetic data. Finally, the preference for type of tasks is the average of the three different integers included in the task description (i.e. pickup, delivery, and return location).

Traveling distance per task (in km)	% tasks within distance
0-10	6.4
10-30	26.3
30-60	36.1
60-120	4.1
120+	27.1

Table 1

Traveling distance required to perform tasks in the dataset.

Traveling distance between tasks (in km)	% of tasks
0-5	48.3
5-10	41.8
10+	9.9

Table 2

Distance between tasks in the dataset.

## 5. Results

This Section presents the results on the two datasets. First, we discuss the synthetic data, followed by the trucking data.

### 5.1. Results Synthetic Data

The results for the preference algorithm with various settings for  $p$ , type of market, and tightness of time windows are shown in Table 3. Note that for this task allocation process on the shortage market in 8 out of 50 runs no fully covered solution exists for the wide time window case. For the narrow time window case this increases to 19 out of 50 runs. Hence, the allocation is difficult in the shortage market. The first column in the table shows the tightness of the time windows, whereas the second column specifies the type of market. The setting for  $p$  is shown in the third column. The fourth column specifies the average prefer-

ence value for awarded tasks (i.e. the average value of  $\phi_{task}$  for awarded tasks), given the setting of  $p$ . We can see that the average preference value per task increases as the value of  $p$  increases. The standard deviation of this average preference is also shown. The column marked solution shows the average time needed to perform all the tasks (again, we assume a fixed cost of 1 per minute of using a resource). This column in fact represents the effectiveness of the solution found, and shows that as the value of  $p$  goes up, so does the time needed to perform all the tasks, which is what was expected. Furthermore, the standard deviation of the solution is shown, and the total evaluation time needed to allocate the tasks to suppliers. Graphical results are presented below.

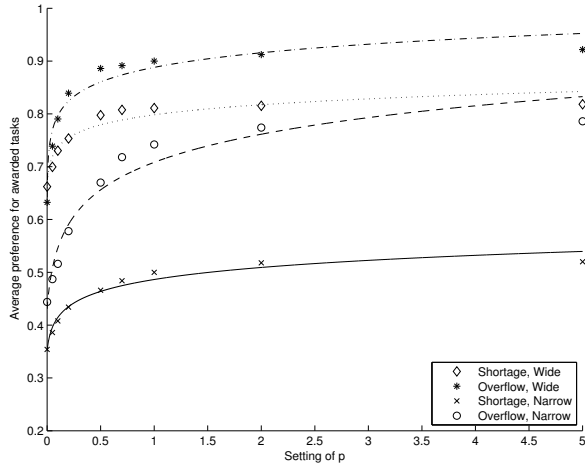


Fig. 2. Preferences met for varying values of  $p$

Figure 2 shows the average preferences for tasks with varying  $p$  for the different market types and time window settings. As can be seen, the easiest way to get the preferences met is the overflow market with wide time windows. The most difficult is the shortage market with tight time windows. The curves of the shortage market are less steep compared to the overflow market. The influence of the time windows on the average preference value is that the curve is basically lower by a certain constant value. The shape of the curve does not change for varying time window settings (i.e. in both the shortage market and the overflow market, the shape of the curve is the same for narrow and wide time windows).

In Figure 3 the influence of the setting of  $p$  upon the efficiency of the overall solution found is shown. Hereby the  $y$ -axis shows the percentage of increase in the total time required to perform the tasks. As can be

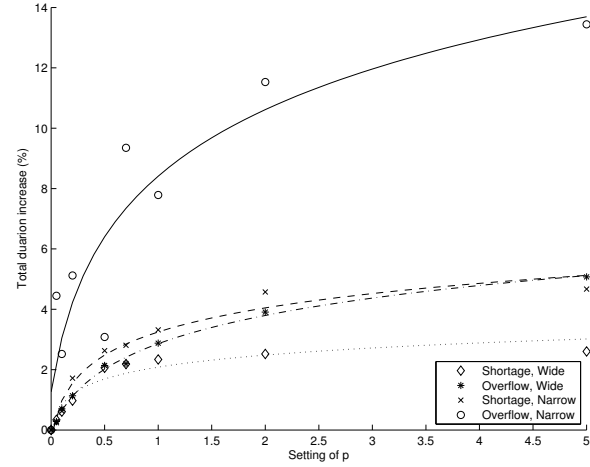


Fig. 3. Duration increase to perform tasks for varying values of  $p$ .

seen, the overall efficiency decreases as the value of  $p$  increases. The setting whereby this increase is highest is the case for the overflow market with narrow time windows. This is due to the fact that there are typically only a few suppliers on the market for which a particular task fits well into the current schedule (especially since narrow time windows are specified), thereby reducing the switching time, and thus the total duration. If these start to increase their price due to preference, less suited suppliers (with more switching time) will get their bid awarded, resulting in a lower quality solution. In case the time windows are wider, typically more suppliers are present that can fit the task well into their schedule (there are simply more time options at which the task can be performed), therefore, there will almost always be one that prefers the task, resulting in fewer increase of the duration. When looking at the shortage market, the reverse is true. In the case of wider time windows the duration increases more than it increases with narrower time windows. This has to do with the fact that in the case of a wider time window there is a higher probability of finding an alternative solution (thus increasing the duration), whereas for a more narrow time window this probability is very small due to the limited number of suppliers.

Figure 4 shows the average preference for tasks on the  $x$ -axis and the increase in the average duration to perform the tasks on the  $y$ -axis. This clearly shows the trade-off between preferences awarded and the efficiency of task execution. All curves look similar (an  $x^n$  type shape) except for the point where the huge increase starts, which varies for the different types of markets. The only exception is the curve of the over-

Time windows	Market type	p	Avg. Preferences	$\sigma$	Solution	$\sigma$	Eval Time (msec)
Wide	Shortage	5.0	0.826	0.336	1482.3	120.4	11.7
Wide	Shortage	2.0	0.815	0.337	1481.1	123.3	10.6
Wide	Shortage	1.0	0.811	0.337	1478.4	125.1	11.1
Wide	Shortage	0.7	0.808	0.339	1476.3	125.0	11.2
Wide	Shortage	0.5	0.798	0.346	1474.5	124.9	12.4
Wide	Shortage	0.2	0.753	0.378	1458.7	121.3	9.36
Wide	Shortage	0.1	0.730	0.393	1453.5	122.5	10.4
Wide	Shortage	0.05	0.700	0.407	1449.5	124.3	9.80
Wide	Shortage	0.0	0.662	0.422	1444.7	121.4	11.2
Wide	Overflow	5.0	0.922	0.230	1381.6	121.9	38.7
Wide	Overflow	2.0	0.912	0.240	1366.3	122.1	31.9
Wide	Overflow	1.0	0.900	0.247	1352.7	126.4	33.5
Wide	Overflow	0.7	0.891	0.251	1343.8	125.5	31.1
Wide	Overflow	0.5	0.886	0.257	1343.0	126.2	25.8
Wide	Overflow	0.2	0.839	0.303	1329.8	122.9	20.9
Wide	Overflow	0.1	0.790	0.351	1324.1	126.8	24.9
Wide	Overflow	0.05	0.739	0.389	1318.6	123.1	15.2
Wide	Overflow	0.0	0.632	0.432	1314.9	122.9	20.3
Narrow	Shortage	5.0	0.520	0.405	1565.0	103.4	12.9
Narrow	Shortage	2.0	0.518	0.406	1563.6	101.4	17.0
Narrow	Shortage	1.0	0.500	0.407	1544.8	103.5	13.8
Narrow	Shortage	0.7	0.484	0.403	1537.2	98.4	11.5
Narrow	Shortage	0.5	0.466	0.407	1534.5	98.7	14.4
Narrow	Shortage	0.2	0.434	0.409	1520.9	94.8	12.3
Narrow	Shortage	0.1	0.408	0.409	1504.6	97.9	13.1
Narrow	Shortage	0.05	0.386	0.409	1498.8	97.1	17.1
Narrow	Shortage	0.0	0.354	0.401	1495.2	98.8	12.0
Narrow	Overflow	5.0	0.786	0.312	1471.7	329.5	14.4
Narrow	Overflow	2.0	0.774	0.314	1446.9	323.5	14.4
Narrow	Overflow	1.0	0.742	0.329	1398.3	378.9	21.5
Narrow	Overflow	0.7	0.718	0.345	1418.6	320.3	16.9
Narrow	Overflow	0.5	0.670	0.366	1337.3	410.2	16.2
Narrow	Overflow	0.2	0.578	0.396	1363.7	306.6	19.3
Narrow	Overflow	0.1	0.516	0.410	1330.0	363.5	16.8
Narrow	Overflow	0.05	0.487	0.414	1355.0	306.3	19.4
Narrow	Overflow	0.0	0.444	0.418	1297.3	407.1	23.0

Table 3

Results using synthetic data. The table shows for different values of tightness of time windows, types of market, and  $p$  what is the average preference value and its standard deviation in the solution. In addition it shows the average time needed in the solution to perform all the tasks (measured assuming a fixed cost of 1 per minute) and its standard deviation, and the time taken to compute the solution.

flow market with narrow time windows. In this case the results are less stable compared to the other results. The curve with the lowest preference value, after which a steep increase is observed, is the one in the shortage

market with narrow time windows. This makes sense because there is hardly any room for allocating tasks to other agents. The curve with the highest point is the overflow market with wide time windows, in which

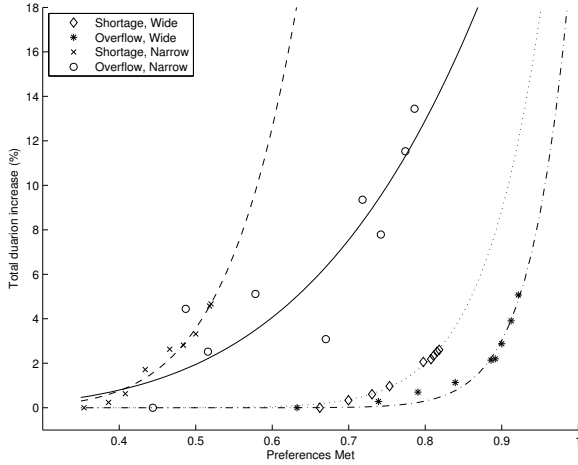


Fig. 4. Preferences met versus increase in duration

there is plenty of space to express preferences and get them awarded.

## 5.2. Results Trucking Data

The results using the trucking dataset are shown in Table 4. The columns shown are the same as the columns specified in the table regarding the synthetic dataset.

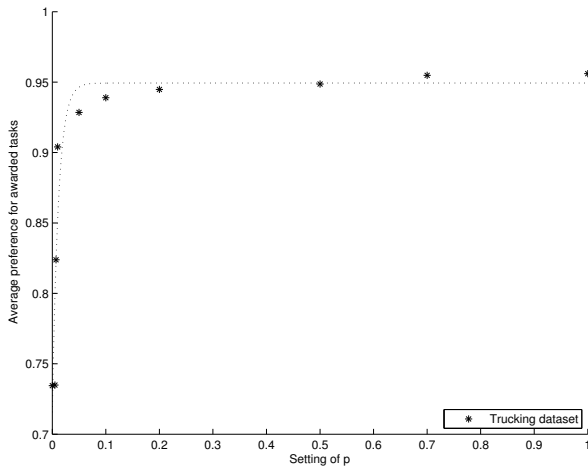


Fig. 5. Preferences met for the trucking dataset, for varying values of  $p$ .

Figure 5 shows how the value of  $p$  affects the average preference for tasks. It can be seen that the value of  $p$  required to increase the average preference significantly is much lower than for the random dataset. Furthermore, the limit seems to be comparable with the overflow market with wide time windows.

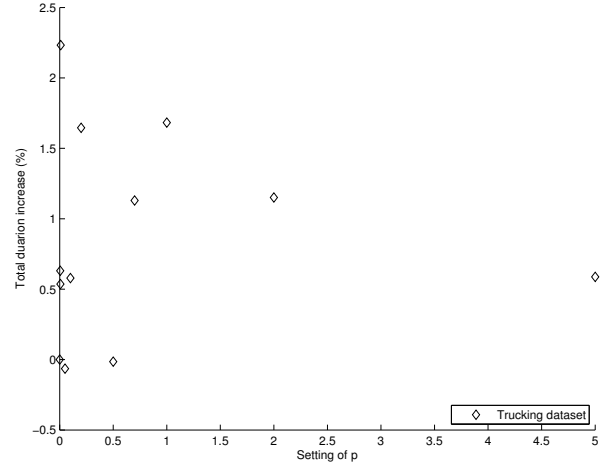


Fig. 6. Duration to perform tasks for the trucking dataset, for varying values of  $p$ .

The relation between the value of  $p$  and the relative increase in duration is shown in Figure 6. As can be seen, it is hard to find a correlation between the value of  $p$  and the increase in duration of the solution, unlike in the random dataset. This is considered to be a specific characteristic of this dataset, due to the fact that the return and pickup locations of the tasks in the dataset are typically close to each other.

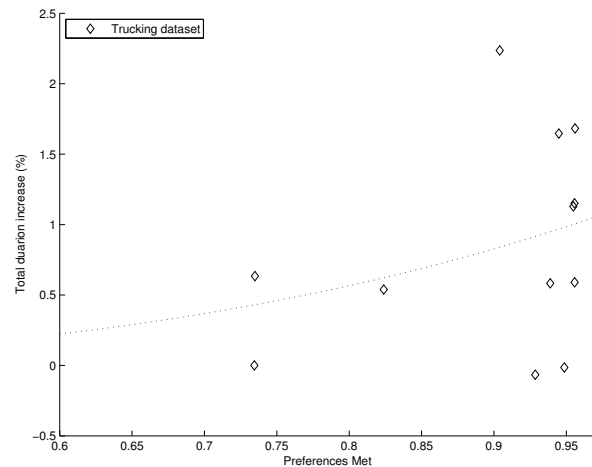


Fig. 7. Average task preference versus duration of performing the tasks.

Figure 7 shows the preference value versus the average duration increase, i.e. the trade-off between preferences met, and the efficiency of execution. It can be seen that there is hardly any correlation between the average preference value of the trucks and the average



Preference setting	Avg. Preferences	$\sigma$	Solution	$\sigma$	Eval Time
5	0.956	0.141	1380.9	531.8	27.0
2	0.956	0.141	1388.6	559.6	24.7
1	0.956	0.141	1395.9	526.7	26.8
0.7	0.955	0.142	1388.3	542.5	20.6
0.5	0.949	0.149	1372.6	518.0	18.4
0.2	0.945	0.154	1395.4	559.1	22.1
0.1	0.939	0.161	1380.8	518.6	25.5
0.05	0.929	0.173	1371.9	517.9	23.5
0.01	0.904	0.214	1403.5	541.3	20.0
0.007	0.824	0.314	1380.2	533.4	19.1
0.005	0.735	0.391	1381.5	518.6	21.1
0	0.735	0.390	1372.8	519.3	21.4

Table 4

Results using the trucking dataset for varying values of  $p$ .

increase in duration. This is of course very good news for the trucking company because this means they can award drivers their preferences without increasing the total driving time. This is assuming that preferences are equally divided amongst the truckers, as in the experimental setup.

## 6. Related Work

In the field of combinatorial auctions, a lot of attention has been devoted to finding out the exact preference for particular bundles of tasks (see e.g. [5] and [11]). In general a certain preference for each of the bundles is assumed, but no detail is given on how the bidder comes up with such a preference value. In this paper we introduce a preference function that allows for a more intuitive specification of preferences, thereby taking multiple aspects of the tasks into account. Preferences for different aspects of the tasks are combined using a weighted average to produce a single preference value.

In research on preference elicitation, typically the impact on selling is addressed, but the precise influence of preferences upon the quality of the solution is not. In this paper, we show how the allocation of tasks in a decentralized fashion directly influences the quality of the solution, and we explore the relationship between the average preference of tasks and the solution quality. In [7] an approach for scheduling a meeting between agents is proposed, which takes into account the preferences of the agents. The relationship between such preferences and the quality of the solution is addressed,

but the problem is not studied from the perspective of combinatorial auctions.

Task allocation can also be performed from a centralized perspective, using preferences as soft constraints. See, for example, [9] for an approach to consider preferences in decision making. There are decentralized variants of constraint optimization (e.g. [12]) but the agents in our case are not necessarily cooperative. In the field of planning and scheduling preferences have been considered as well. Languages have been developed that allow for the specification of preferences and soft constraints (see e.g. [8]).

The logistic domain we use for our experiments has been researched for quite some time (see e.g. [10]), mainly focusing on calculating optimal solutions from a centralized perspective. For instance, in [6] the problem addressed is to find optimal routes for transportation orders of a large set of users. Orders have to be picked up and delivered at specific locations, within a given time window, and using a limited number of trucks. The solution proposed is centralized, and it is used to support a human dispatcher.

The current trend in logistics requires an even more distributed setting because of the use of fourth party logistics (4PL) [1]. 4PL companies sign contracts with large companies to arrange their entire transportation demand. These companies, however, do not have sufficient resources on their own to arrange all these transports and therefore distribute many of those tasks to other (partner) companies. Centralized calculation might no longer be feasible due to lack of complete information (availability of resources is too sensitive for a company to communicate) as well as the complexity of calculating an optimal solution within a short period (time is crucial in the business).

## 7. Conclusions

We have presented an approach to specify preferences for tasks in a combinatorial auction setting. Allowing users to specify such preferences is essential for them to use auctions and to increase the economic efficiency of reverse auctions, as reported, for instance, in [13]. We propose a preference function and use it in a bidding algorithm where bids on non-preferred tasks have a higher price.

We evaluated our approach in two ways, first by rigorously testing it with synthetic data. Several parameters have been varied, namely the tightness of the time windows (i.e. how easy it is to fit a task within a certain schedule), and the relative availability of resources. (i.e. how many suppliers can perform a task). It was shown that it was easiest to get preferences awarded in markets with wide time windows. The trade-off between meeting preferences and overall execution time has been studied in depth. We have shown that the overall efficiency of the solution in terms of execution time is influenced most in the case of the overflow market, due to the fact that in the shortage market there are hardly any alternatives at hand and therefore, although the agent might not prefer a task, it will still get its bid awarded. The curves observed tend to have the same shape when the time window setting changes but the market type remains the same. For different market types, the curves vary in steepness.

Besides testing with synthetic data, we have also used a real company dataset from the trucking domain. We have shown that the bidding algorithm is effective in awarding suppliers more preferred tasks. The influence of this preference on the overall solution quality was not observed using the real dataset. Hence, in this setting the preferences being met have much less influence on the efficiency of the solution found. For future work, it would be interesting to find out whether other real datasets would show the same results as the dataset used in this paper. Furthermore, exploring how well the companies can express their preferences using these functions would be interesting as well.

**Acknowledgments:** Partial support is gratefully acknowledged from the National Science Foundation under grant IIS-0414466.

## References

- [1] P. Briggs. The hand-off: the future of outsourced logistics may be found in the latest buzzword [fourth party logistics]. *Canadian Transportation Logistics*, 102(5):18, 1999.
- [2] J. Collins, G. Demir, and M. Gini. Bidtree ordering in IDA\* combinatorial auction winner-determination with side constraints. In J. Padget, O. Shehory, D. Parkes, N. Sadeh, and W. Walsh, editors, *Agent Mediated Electronic Commerce IV*, volume LNAI2531, pages 17–33. Springer-Verlag, 2002.
- [3] J. Collins and M. Gini. An integer programming formulation of the bid evaluation problem for coordinated tasks. In B. Dietrich and R. V. Vohra, editors, *Mathematics of the Internet: E-Auction and Markets*, volume 127 of *IMA Volumes in Mathematics and its Applications*, pages 59–74. Springer-Verlag, New York, 2001.
- [4] J. Collins, W. Ketter, and M. Gini. A multi-agent negotiation testbed for contracting tasks with temporal and precedence constraints. *Int'l Journal of Electronic Commerce*, 7(1):35–57, 2002.
- [5] W. Conen and T. Sandholm. Preference elicitation in combinatorial auctions. In *Proc. First Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, volume 1, pages 168–169, Bologna, Italy, July 2002.
- [6] K. Dorer and M. Calisti. An adaptive solution to dynamic transport optimization. In *Proc. Fourth Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, pages 45–51, 2005.
- [7] M. Franzin, E. Freuder, F. Rossi, and R. Wallace. Multi-agent meeting scheduling with preferences: efficiency, privacy loss, and solution quality. In *Proc. of AAAI Workshop on Preference in AI and CP*, 2002.
- [8] A. Gerevini and D. Long. Preferences and soft constraints in PDDL3. In *Proc. ICAPS Workshop on Planning with Preferences and Soft Constraints*, 2006.
- [9] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, 1976.
- [10] T. Magnanti. Combinatorial optimization and vehicle fleet planning: Perspectives and prospects. *Networks*, 11:179–214, 1981.
- [11] D. C. Parkes. Auction design with costly preference elicitation. *Annals of Mathematics and Artificial Intelligence*, 44:269–302, 2005.
- [12] A. Petcu and B. Faltings. DPOP: A scalable method for multi-agent constraint optimization. In *Proc. of the 19th Joint Conf. on Artificial Intelligence*, pages 266–271, Edinburgh, Scotland, Aug 2005.
- [13] T. Sandholm. Expressive commerce and its application to sourcing: How we conducted \$35 billion of generalized combinatorial auctions. *AI Magazine*, 28(3):45–58, Fall 2007.
- [14] R. G. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Trans. Computers*, 29(12):1104–1113, December 1980.
- [15] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason. Auction protocols for decentralized scheduling. *GEB: Games and Economic Behavior*, 35(1–2):271–303, 2001.