Decentralized task allocation using magnet: An empirical evaluation in the logistics domain

Conference Paper · August 2007

DOI: 10.1145/1282100.1282161 · Source: DBLP

3 authors:

Mark Hoogendoorn
Vrije Universiteit Amsterdam

Maria Gini
University of Minnesota Twin Cities

Catholijn M. Jonker
Delft University of Technology

Decentralized Task Allocation using MAGNET: An Empirical Evaluation in the Logistics Domain

Mark Hoogendoorn
Vrije Universiteit Amsterdam
Department of Art. Intelligence
De Boelelaan 1081a
Amsterdam, The Netherlands
mhoogen@cs.vu.nl

Maria L. Gini
University of Minnesota
Department of C.S. and Eng.
200 Union Street SE
Minneapolis, USA
gini@cs.umn.edu

Catholijn M. Jonker
Delft University of Technology
Department of Man-Machine Interaction
Mekelweg 4
Delft, The Netherlands
catholijn@mmi.tudelft.nl

ABSTRACT

This paper presents a decentralized task allocation method that can handle allocation of tasks with time and precedence constraints in a multi-agent setting where not all information needed for a centralized approach is shared.

In our MAGNET-based approach agents distribute tasks via first-price reverse combinatorial auctions, where the auctioneer is whatever agent has tasks to be allocated. The choice of MAGNET is based on its uniqueness to handle auctions for allocation of tasks which include time windows and precedence constraints.

Empirical evaluations based on real data obtained from a logistics company show that the system performs well. The costs of the allocations obtained by our approach are on average within 5% from the optimal allocation. The computation time is linear in the number of tasks, while computing the optimal allocation is an NP-hard problem.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence; K.4.4 [**Computers and Society**]: E-commerce

General Terms

Algorithms, Economics, Theory

Keywords

Automated auctions, multi-agent contracting, logistics

1. INTRODUCTION

There are many real-world problems in which agents need to plan in advance and schedule multiple tasks. Think of logistics, hospital schedules that have to be changed with new patients coming in, manufacturing on demand, and design of complex systems. We are interested in situations where an agent recruits other agents to carry out tasks which commonly have precedence and time constraints, such as in logistics, hospitals, and manufacturing on demand. Such recruitment is indeed a form of electronic commerce, in which one agent seeks bids to have certain tasks executed and other agents offer to perform these tasks for a price.

The field of planning has contributed several centralized heuristic algorithms for optimal task allocation. For example, algorithms have been created for the Vehicle Routing Problem and its instances (see e.g. [10]), and the Dial-a-Ride problem [17]. The main disadvantage of such algorithms is their centralized nature, since a centralized allocation of tasks to multiple agents is not always possible. It may be computationally infeasible to find an optimal allocation or agents may be unwilling to share complete information about their resources and commitments may invalidate the algorithm.

Decentralized task allocation has been a topic of research for quite some time, see e.g. [28], [29], and [24]. However, so far, the decentralized task allocation literature has not addressed the problem of task precedence relations and time constraints between the tasks.

This paper presents a decentralized way of allocating tasks that does deal with precedence and time constraints. The method exploits the unique feature of the MAGNET [7] system that allows autonomous agents to negotiate over complex coordinated tasks, with precedence and time constraints, in an auction-based market environment [5].

In our method MAGNET agents participate in market mediated first-price reverse combinatorial auctions, where the agent which allocates the tasks to other agents is the auctioneer. Any agent can be an auctioneer, so any agent can, at any moment in time, attempt to allocate its tasks to other agents via auctions.

The method has been thoroughly evaluated by means of empirical analysis using data obtained from a logistical company. This choice of domain allowed us to test specifically the effectiveness of the method to deal with precedence relations and time constraints, while delivering solutions that are nearly optimal.

In logistics, the tasks that require allocation have different types of time constraints. The transportation devices (ships, trucks, plains, trains) are not cost effective while they are being used as storage room. Furthermore, devices are often not allowed to stay at the same place for long. For example, in the harbor of Rotterdam, ships are assigned specific slots

ICEC'07, August 19–22, 2007, Minneapolis, Minnesota, USA.

for off loading their goods. Due to the nature of ships and harbors, last minute rescheduling of slots and ships is impossible. Ships and harbor have to know the schedule much longer in advance, see e.g. [27]. The industry and/or companies that need the goods are increasingly producing on demand instead of keeping a large stock. This implies that the logistics process starts no sooner than when an order comes in, while the customer still expects a speedy delivery.

Furthermore, the goods themselves can impose additional time constraints on the logistics. For example, perishable goods like flowers have to cross the world in hours to be still of value at the point of delivery.

Finally, the logistics process itself can cause precedence constraints; a good cannot be transported from a particular location before it has arrived at that particular location. Similar precedence constraints can be caused by production on demand processes requiring different raw materials or half-fabricates.

As a result, transportation tasks typically have time windows specified, stating when particular goods can be picked up, and when they need to be delivered. Other aspects relevant for logistics are the locations (from, to), and some indication of the type of load to be transported (e.g., the type of container), so that an appropriate transportation device can be selected and scheduled.

This paper is organized as follows. Section 2 gives a brief overview of the MAGNET system. The application of the MAGNET system in the field of logistics is presented in Section 3. Results of the empirical evaluation using a dataset obtained from a logistics company are presented in Section 4. Section 5 discusses related work. Finally, Section 6 presents our conclusions and gives directions for future work.

2. THE MAGNET SYSTEM

The MAGNET architecture provides a framework for secure and reliable commerce among self-interested agents. MAGNET shifts the burden of market exploration, auction handling, and preliminary decision analysis from human decision-makers to a network of heterogeneous agents.

The MAGNET system architecture, shown in Figure 1, consists of: (1) a *customer* agent, which allocates tasks to other agents. The tasks have time constraints and other restrictions; (2) *suppliers* agents, which bid on the tasks and execute them when awarded; and (3) the MAGNET market server, which keeps track of the activities of the agents and of the auctions.

The main 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 is specified giving 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. Supplier data 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 work schedule. In MAGNET the customer can chose from a collection of winner-determination algorithms (A*, IDA*[1] [6], simulated annealing, and integer programming [5]).

- The customer agent awards bids and specifies the work schedule.

3. MAGNET AND LOGISTICS

In this Section, the domain of logistics is introduced and thereafter the application of the MAGNET system within this domain is presented.

3.1 The Logistic Domain

The field of logistics is a domain in which task allocation is part of the core operations [18]. Orders that arrive demand a set of specific transportation tasks to take place. These transportation tasks need to be assigned to a particular resource (e.g. a truck or a ship).

The logistic domain has been a topic of research in classical planning for quite some time (see e.g. [21]), mainly focusing on calculating optimal solutions or approximating them from a centralized perspective. For instance, in [12] 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.

Distributed planning has been popular in distributed AI applications (see, for instance, [13]), where agents are assumed to be cooperative, but coordinating the plans of individual agents is still a challenging task [9]. When the agents are not cooperative, auction based approaches to allocation of tasks are more commonly used (for instance, [29, 1]).

A trend has now emerged in the field of logistics which requires a more distributed setting: Fourth party logistics (4PL) [2]. Fourth party logistics 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. A rapid assignment of tasks to particular resources is essential for these 4PL companies. Orders typically arrive at the company by phone, and being able to immediately inform the customer on when the task will be performed gives a competitive advantage.

Given this setting, centralized calculation of the optimal solution might no longer be feasible due to the lack of complete information (availability of resources which is too sensitive for a company to communicate) as well as the complexity of calculating this optimal solution within a short period (time is crucial in the business). The latter especially holds due to the fact that constraints, such as time windows and precedence constraints, are also specified for these tasks, making calculation of the optimum even harder.

[1]Iterative Deepening A* (IDA*) [19] is a variant of A* which uses the same heuristic function in a depth-first search, and which keeps in memory only the current path from the root to a particular node. In each iteration of IDA*, search depth is limited by a threshold on the value of the heuristic function.

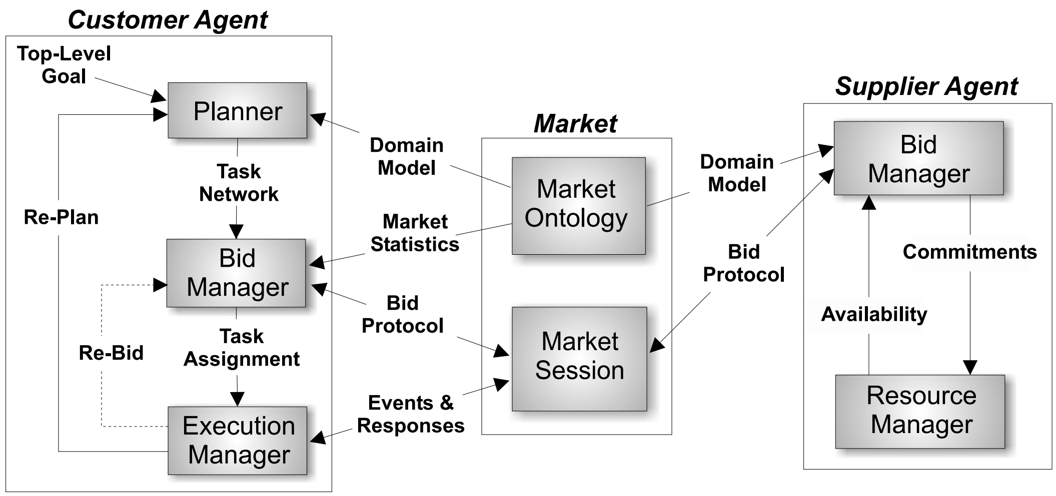

Figure 1: MAGNET architecture.

3.2 Using MAGNET in Logistics

Given the problems observed in the previous Section (i.e. centralized calculation of solutions is not feasible), the MAGNET system can help overcome these problems since it allows the companies to do task allocation in a distributed way while maintaining their own schedule. Furthermore, the strength of the MAGNET system is that it is also able to handle time windows as well as precedence constraints which is essential in this domain.

Other task allocation methods based on auctions assign only the tasks needed for the immediate time period and run auctions every time a new task becomes available [11]. Because of this, they do not produce optimal allocations. MAGNET avoids this problem by soliciting bids for tasks spanning over time, and accepting the optimal combination of bids that fits the overall schedule.

Following the description of the logistical domain and the move towards 4PL presented above, task allocation can be performed as follows: The 4PL company (i.e. the *customer*) issues an RFQ, sends it to partner companies (i.e. the *suppliers*) who can bid on one or more tasks included in the RFQ. Since the price per kilometer of driving for each partner firm is fixed, the price they bid equals the amount of driving required to perform the task(s).

Based upon this viewpoint, implementations of both *supplier* and *customer* agents have been created.

3.2.1 Supplier Agent

The *supplier* agent maintains a schedule for its resources and generates bids based upon that schedule. The schedule specifies when resources are available as well as the start location when the resource becomes available and the end location when the availability slot ends. During that availability time, the schedule consists of entries that specify when tasks are scheduled to be performed (i.e. start and end time), and furthermore what the start and end location of that particular task is.

Once an RFQ arrives, the tasks in the RFQ are sorted based upon the early start time, and the following algorithm is performed by the *supplier* agent:

For all tasks in the RFQ do the following:

- if the schedule for the current day is empty
  - Check whether the task can be performed in an empty schedule.
    This means determining whether it can be performed between the start time (starting at the specified location) and the end time of the total schedule (ending at the specified end location). Therefore, calculate the total time required from the start location of the task to perform the task and to return from the task to the end location. Thereafter, calculate whether this task fits into the schedule and whether the specified times for the task (earliest/late start time and the deadline) can be achieved. If the task indeed fits, mark the task as *potential include*.
- else
  - Check whether the current task can be performed within the current schedule (which is not empty). This is done by going through the schedule from the start time and determining whether the task can fit in somewhere. A task fits in somewhere in case it is possible to go from the previous task (or the start location in case no previous task exists) to the task location, perform the task, and return to the next task in the schedule (or to the end location if such a task does not exist) after the deadline of the previous task (or the start time) and before the scheduled start time of the next task (or before the overall end time). If the task indeed fits, mark the task as *potential include*.
- if the task is marked as *potential include*
  - if the strategy is set to RANDOM bidding (which includes a task in a bid with a certain probability)
    - ∗ Generate a random number. If the number is above the threshold, mark the task as *include*.
  - if the strategy is set to CLOSEBY bidding (which includes only those tasks that are close, given a

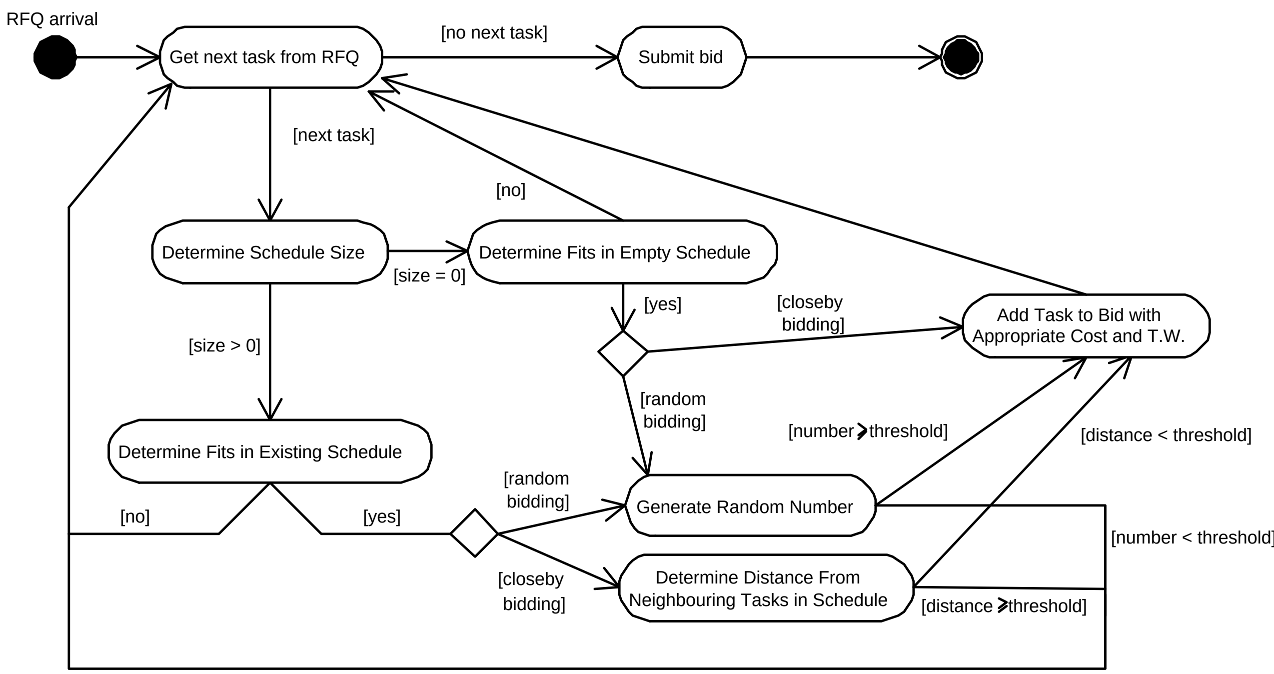

Figure 2: UML Activity Diagram of Supplier Algorithm (Note that T.W. is an abbreviation for Time Window.)

certain distance measure, to the start or end location of the tasks that are already part of the schedule)

∗ if the schedule for the current day is empty, mark the task as *include*. Otherwise, mark the task as *include* if the task is close enough.

- if the task is marked as *include*
  - Insert the time windows according to the new schedule, taking into account the windows specified for the task within the RFQ. As cost for the task, we use the sum of the distance of traveling to the start location of the task, performing the task, and returning from its end location.

Figure 2 shows the algorithm in the form of a UML activity diagram. Note that preferences of *suppliers* can also be taken into consideration. This, however, is not the focus of this paper. In [15] for example, bidding strategies are specified that do take such preference into account.

3.2.2 Customer Agent

The *customer* agent simply creates RFQ's for tasks matching the orders that have been received, and evaluates the bids that have been received based upon the evaluation algorithms part of the MAGNET system. Since it could happen that certain tasks are not bid upon, dummy bids for each task are added to the bid set for this evaluation process with an extremely high price. In case such a dummy bid gets awarded the task needs to be sent again, possibly attracting some *suppliers* that did not get their bid awarded. Each RFQ which is sent the same day is later referred to as a *cycle*.

4. EMPIRICAL EVALUATION

To see how the setup within the field of logistics described in the previous Section would work in a real life setting, data has been obtained from a 4PL company. The characteristics of the data are described first. Thereafter, results of using the data as input for the system are presented as well as comparisons between the solutions found and the optimal solution. In this case, the optimal solution could be calculated as all information is centrally available in one dataset. This is not necessary for the MAGNET algorithm, but it enables us to compare the distributed with the centralized approach, giving us insight in the quality of the distributed solution. Furthermore, the time required for the computations can be compared as well.

4.1 Dataset Description

The dataset has been obtained from a company within the field of logistics. The company is a mid-size company that focuses on transport of various types of goods, including perishable goods, and containers. Transportation of containers has been a growing global market over the last decades [25], and 4PL companies need to transport many containers as part of the contracts they have with their customers. The company owns over 200 trucks for the various operations it performs, and has contracts with numerous partners which can be contacted in case more trucks are needed on a particular day.

The dataset we have obtained from the company concerns container transports. Each morning the company receives a set of tasks concerning transportation of containers on that specific day from a specific pickup location (for instance a container terminal), to a specific destination location where the container is either unloaded or loaded. Thereafter, the container needs to be transported to a third location where it is left behind. Besides these locations, time points are also specified, indicating after which time point the container becomes available at the pickup location and when the container needs to be returned to the third location. Note that the dataset only concerns the transport of entire containers, it does not concern the load inside the container. The challenge is to combine these container transport jobs in such a way that the minimum amount of driving time is required. The amount of orders received upon a day is on average just above 20 of such transportation tasks. The size of the dataset concerns 100 such days, totaling to approximately 2000 tasks that need allocation.

To characterize the nature of the tasks in the dataset, Table 1 shows the distance that needs to be traveled by trucks in order to visit all three specified locations (i.e. pickup, destination, and return location). As can be seen, the tasks have a large variety in performance time, both shorter trips (10 - 60 kilometers) as well as longer hauls (120 kilometers and up) are common in the dataset. In order to get more insight in how easily these tasks can be combined, Table 2 characterizes the distances between the return and pickup locations of tasks. The table shows that most return and pickup locations are quite close to each other. This is due to the fact that this company mainly picks up and returns containers within the Rotterdam port area, where container terminals are quite close to each other. Typically, a containers is picked up at one of these terminals, the content of the container is delivered somewhere in The Netherlands, and the container is returned to one of the container terminals again. The traveling distance between the tasks is however still a significant part of the traveling time, especially when considering that the majority of the tasks require less than 60 kilometers of driving.

Traveling distance per task (in km)	**Percentage of 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.

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

Table 2: Distance between tasks.

Note that the complexity of the scheduling for this company is clearly not in the amount of tasks to be scheduled. The main problem here is speed and incompleteness of information. As described above 4PL companies do not have the trucks themselves, they have to negotiate with the companies that do have trucks. In fact what happens is that different 4PLs compete with each other for work. They can only be effective if their interaction with truck owning companies is time effective. Similarly, the truck owning companies have to compete with each other for work, and, again, time is of the essence.

Besides tasks that need to be performed, the dataset also includes resources that can be allocated to such tasks. In this particular dataset, trucks are specified that can be used as a resource on a particular day to perform tasks. Note that these trucks are the trucks owned by the company itself as well as trucks that can potentially be hired. For each of these trucks, an availability slot is given, including a start time when the resource is available, and an end time after which the truck is no longer available. The capacity of such a truck is that it can carry one container at the same time. Each truck starts at the headquarters of the company at the beginning of the availability slot, and needs to reach the headquarters at the end of the slot.

Table 3 shows how close the tasks (pickup and return locations) within the dataset are to the headquarters of the company. As can be seen, most tasks are between 15 and 30 kilometers from the headquarters. On average, approximately half the amount of trucks are available compared to the number of tasks that need to be performed. This is more than sufficient to perform all transports while still meeting the requirements that have been set for these orders.

Traveling distance to headquarters (range in km)	**Percentage of tasks**
0-15	0.4
15-20	40.1
20-25	15.7
25-30	41.3
30+	2.5

Table 3: Distance between tasks and company headquarters.

Given this dataset, both types of companies are represented, namely the truck owning companies (the trucks in the dataset) and the 4PL company (the orders in the dataset). Each truck is represented by exactly one *supplier* agent within the MAGNET system. Each *supplier* calculates the distance between different locations, using the same distance function used by the other *suppliers*. Finally, each *supplier* uses the same definition of locations considered to be close to each other (in case of the CLOSEBY algorithm), which is based upon a definition given by planners within the company.

4.2 Results

The results reported in this Section concern usage of the full dataset (i.e. 100 days of operations with on average 20 orders, meaning approximately 2000 orders). Since we are interested in how well our algorithms scale up, we want to vary the amount of tasks that require allocation upon a day. This means that we perform runs over the full 100 days. For each run we keep the number of tasks that require allocation upon one day constant (e.g. 5 tasks per day). Therefore, each day selections are made of the total number of tasks that are available from the dataset, where the size of the selection equals the number of tasks we want to investigate. We've performed runs using 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 15, and 20 tasks. Also, selections of resources have been made to make the runs as realistic as possible. As already mentioned, on average half the amount of resources is available compared to the number of tasks that require allocation. This means that the number of resources available upon a day during such a run is set to 1, 1, 2, 2, 3, 3, 4, 4, 5, 6, 7, and 10 respectively.

The experiments have been conducted on a Sun UltraSPARC-IIIi 1062 MHz CPU with 8 GB of memory. Calculation of the optimal result is performed by means of a brute force algorithm, which does not scale up well with the number of tasks that require allocation upon a day. Theoretical results show that the type of problem, called the capacitated dial-a-ride problem is NP-hard to solve [3]. As a result of this, such calculations could only be performed up to 10 tasks per day.

Regarding the MAGNET system, the IDA* algorithm has been used for evaluation of the bids that have been submitted by the trucks. IDA* is an admissible (i.e. it finds an otimal solution given a heuristics which understimates the remaining cost), memory-bounded, heuristic search algorithm. Its time complexity is hard to characterize, since it depends on how good the heuristic used is [19]. Its space complexity is linear in the depth of the solution. This makes IDA* a good choice when an optimal solution is needed in a large state-space where A* would run out of memory.

4.2.1 *Comparison to optimal solution*

Figure 3 shows the average deviation from the optimal solution of the solution produced by the distributed MAGNET algorithm when using the Closeby bidding and running the algorithm on the full 100 days.

A result of 1 means that the average solution found is equal to the optimal (which is the lowest cost for performing the tasks), whereas 1.05, for instance, means the average result found is 5% above optimal.

The results are presented for varying number of tasks that require allocation upon a day. As can be seen, the deviation of the solution found compared to the optimal one initially increases with the number of tasks. However, the steepness of this increase in deviation from the optimal result decreases as the number of tasks that need allocation increases. This decrease is due to the fact that more tasks increase the probability of the trucks finding a task which nicely fits within their schedule, avoiding large driving distances from one task to another.

Besides the average deviation from the optimal solution, the standard deviation has been calculated as well, and is shown in Table 4. Furthermore, the Table shows the exact deviation from the optimal solution for the Closeby strategy.

Simulations using the Random bidding algorithm have also been performed. The results are significantly worse compared to the Closeby bidding algorithm. For 5 tasks for example, the deviation from the optimal solution is 1.17 and increases with the number of tasks. Hence, solving the task within such a small margin from the optimal solution is not a trivial task that can be performed by simple Random bidding, but advanced strategies such as Closeby bidding are needed.

Besides comparing the quality of the solution found, the difference in search time is also a crucial element within the field of logistics. As already mentioned before, being able to immediately inform customers over the phone gives a competitive advantage.

Number of tasks	Average deviation from optimum	Standard deviation
2	1.008	0.029
3	1.014	0.030
4	1.019	0.031
5	1.025	0.033
6	1.032	0.037
7	1.043	0.066
8	1.034	0.033
9	1.037	0.032
10	1.038	0.035

Table 4: Detailed results using Closeby bidding and comparison of the solution found by the distributed MAGNET algorithm to the optimal solution. Results are over 100 days.

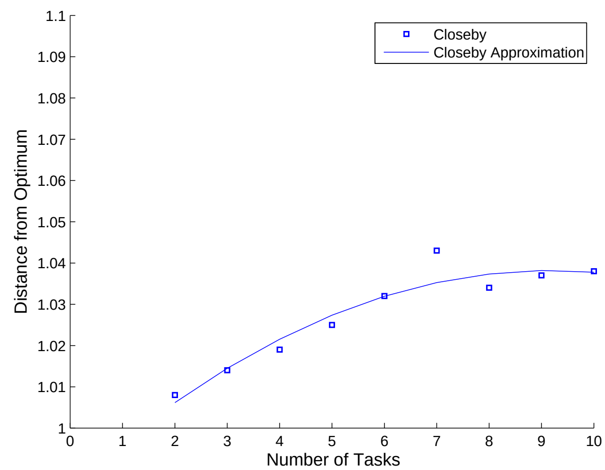

Figure 3: Distance of the solution computed by the MAGNET algorithm to the optimal solution over different numbers of tasks per day.

In Figure 4 the average total evaluation time (i.e. the sum of the evaluation time for all *cycles* upon a day in the case of the MAGNET algorithm) over 100 days for varying number of tasks is shown. Again, only the Closeby bidding algorithm is shown as the Random algorithm scales in a similar fashion. As can be seen, the algorithm for optimal performance does not scale well, whereas the MAGNET algorithm scales very well, it can even be approximated by a linear function. Note again that IDA* has been used here.

When considering a maximum waiting time of approximately 1 minute on the phone, no more than 8 orders can be placed in case of the centralized algorithm. For the decentralized MAGNET algorithm however, 20 orders can certainly be handled which is currently the maximum number of orders received by the company.

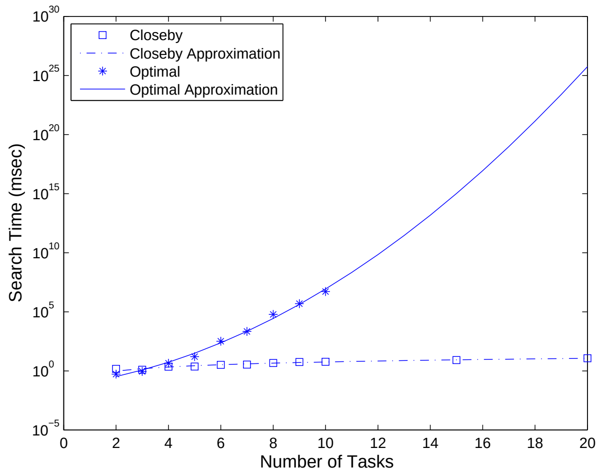

Figure 4: Total evaluation time needed for the optimal algorithm and for the MAGNET algorithm (i.e. sum of evaluation time of all cycles needed). Note the logarithmic scale.

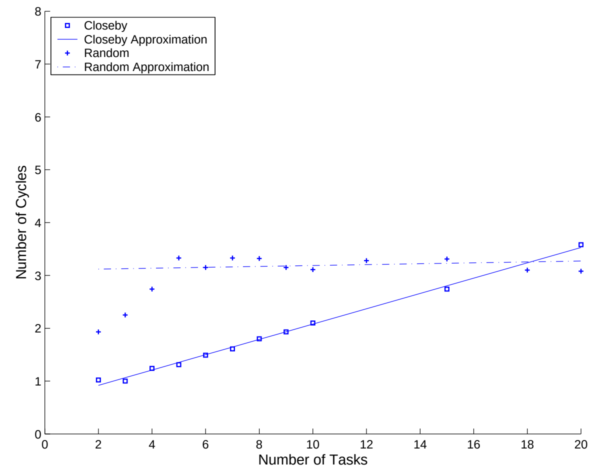

Figure 5: Cycles needed by the Closeby **and** Random **bidding strategies. The number of tasks is the number of tasks to be allocated in a single day. The number of cycles is averaged over the 100 days within a run.**

4.2.2 *MAGNET Bidding Strategy Characteristics*

Besides comparing the quality and search time of the solution found by the MAGNET based system with the centralized approach, the characteristics of the two different bidding algorithms (i.e. Closeby and Random) have been compared as well. As already mentioned, the Closeby algorithm finds solutions of a much higher quality than the Random algorithm. Furthermore, the search times scale approximately the same for both bidding strategies. A third measure for comparison is the number of *cycles* needed (i.e. how many times an RFQ with tasks needs to be sent to have a fully covered task allocation for a day).

The number of *cycles* needed, averaged over the 100 days within a run, is shown in Figure 5 for a varying number of tasks that require allocation upon a day. As can be seen, the number of cycles needed for the Random algorithm remains approximately constant. This can be explained by the fact that the ratio between trucks and tasks is constant. For the Closeby algorithm however, the number of cycles increases with the number of tasks. This is the result from the initial location being identical for all trucks, therefore the trucks without tasks awarded bid for the same tasks and hence, more cycles are needed before all tasks are covered.

4.2.3 *MAGNET Evaluation Algorithm Characteristics*

Finally, results are shown on the average performance of the MAGNET evaluation algorithm (IDA* in this case) within one *cycle*. Figure 6 shows the performance for varying number of tasks in the RFQ. The algorithm scales very well and can be approximated by a linear function.

The characteristics of the bids that are evaluated are shown in Table 5, including detailed average evaluation times as well as the standard deviation and maximum search times. The table shows that as the number of tasks increases, so does the average number of bids that have been received. This is logical because more tasks are presented, and therefore the probability of trucks being able to perform at least one of the tasks increases. Note that the average number of bids for a certain number of tasks can exceed half the amount of tasks (i.e. the number of trucks available when starting with that task size) as this concerns averages over all *cycles* and all amount of tasks that need to be scheduled upon a day (i.e. 2 to 20 tasks). It might for instance be the case that for a run with 20 tasks, multiple *cycles* are needed in which the last *cycle* only concerns 2 tasks whereas 10 trucks can still bid. Furthermore, the average number of tasks per bid increases with the number of tasks as well, which is due to the fact that tasks can more easily be combined. The standard deviation increases with the number of tasks being evaluated. Furthermore, the maximum search time is significantly higher than the average search time. These values however still allow for real time computation using the presented system.

5. RELATED WORK

Work done in centralized task allocation or planning involves finding efficient algorithms for solving (or approximating a solution for) specific problems. One specific family of problems is that of vehicle routing problems (VRP). A variant of the VRP that is close to the task allocation problem used as an empirical evaluation in this paper include the capacitated VRP with pick-up and deliveries and time windows (CVRPPDTW). Furthermore, the dial-a-ride problem (DARP) generalizes a number of such vehicle routing problems [8] and when including capacities maps to the problem addressed in this paper. This problem is known to be an NP-hard problem to solve. See for example [3], [17], and [12] for algorithms that solve such problems from a centralized perspective. Solving the vehicle routing problem from a centralized perspective might however not always be feasible, resulting in research focusing on decentralized task allocation as well.

Distributed constraint optimization algorithms have been

Number of Tasks	Avg. number of Bids	Avg. Tasks per Bid	Avg. Search Time (msec)	Standard Deviation	Max Search Time (msec)
2	4.15	1.30	1.43	2.22	16.0
3	5.20	1.43	1.37	1.46	12.0
4	7.11	1.79	2.15	2.92	17.0
5	7.81	1.97	2.05	2.48	21.0
6	9.68	2.26	2.40	3.05	16.0
7	10.98	2.45	2.56	2.84	20.0
8	12.88	2.71	3.48	3.72	21.0
9	13.53	2.69	3.79	3.98	23.0
10	14.65	2.65	4.06	4.00	29.0
15	22.11	3.33	4.71	5.60	36.0
20	30.00	3.99	5.85	5.36	35.0

Table 5: MAGNET evaluation characteristics.

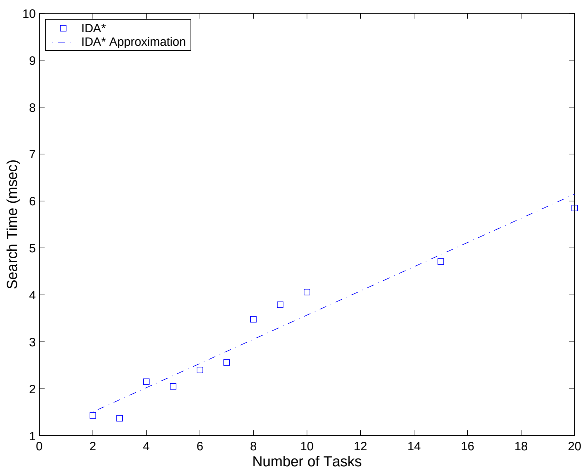

Figure 6: Evaluation time needed by MAGNET for different numbers of tasks.

proposed for task allocation (see, for instance, ADOPT [23] and OptAPO [22]). These algorithms are appropriate in domains where optimality is essential, but have high communication costs. [26] proposes an approximate algorithm for distributed task allocation which trades off optimality for reduced communication costs and which is specially suited for large teams in simulated search and rescue.

Auctions [20] have been suggested for allocation of computational resources since the 60's. The Contract Net [28] is perhaps the most well known and widely used bidding protocol for distributed problem solving. Many multi-agent and distributed systems use some form of auction to allocate resources. Auction-based methods for allocation of tasks are becoming popular as an alternative to other allocation methods, such as centralized scheduling [4], distributed planning [13, 9], or application-specific methods, which do not easily generalize. An advantage of auctions is they are a distributed mechanism and draw on a large body of analytical results from economics. In addition, one-shot auctions are efficient in the case of low bandwidth and unreliable communications.

Scheduling plays an important role in task allocation, since before accepting a task an agent has to find how to fit it into its existing schedule. In [16] combinatorial auctions are used for the initial commitment decision problem, which is the problem an agent has to solve to decide whether to accept or refuse a new task. In [14] scheduling decisions are made not by the agents, but instead by a central authority, which has insight into the states and schedules of the agents. In MAGNET, there is no central authority; the market is used only as a repository of statistical information.

Despite the abundance of work in auctions, limited attention has been devoted to auctions over tasks with complex time constraints and interdependencies, as in MAGNET. Auctions for decentralized scheduling have been studied extensively by Wellman's group. The emphasis of their work is in the supply-chain construction, more than dealing with time, and in analyzing strategies using game-theoretic techniques. A protocol for combinatorial auctions for supply chain formation is proposed in [29]. Complex task networks are allowed, but they do not include time constraints. A protocol for decentralized scheduling is proposed in [31]. The study is limited to scheduling a single resource, while we are interested in multiple resources. In [30] agents bid for individual time slots on separate, simultaneous markets.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an approach to perform decentralized task allocation using the MAGNET system. There is already a vast amount of literature on performing such task allocation using negotiation, see e.g. [28] and [29], however, the unique feature of the system presented here concerns the negotiation about complex tasks including time window and precedence constraints. In a variety of domains such constraints are vital for task allocation, such as for the field of logistics. Implementations are created for both the *supplier* and *customer* agent where for the former two different bidding strategies are implemented, namely one which takes the distance to tasks into account (i.e. only bidding on tasks that are close to a task you already perform) called CLOSEBY, and a RANDOM bidding algorithm.

To evaluate the proposed approach, a comparison is made to a central task allocation scheme which is able to calculate the optimal solution. Such an evaluation could be performed on a randomly generated dataset, in this paper however, the choice is made to use empirical data. This choice results in a dataset with characteristics that indeed occur in the real world, giving more insight in the usability of the approach in real life.

The evaluations show that the approach using Closeby bidding comes very close to the optimal result. The maximum average deviation found is just over 4% of the optimal result, whereas the trend is that this deviation from the optimum is not (or hardly) increasing for greater amount of tasks. The Random bidding does not perform that well, showing that taking distances into account when bidding is very effective for the quality of the solution found. When looking at the computation time needed to come to the solution found, the MAGNET algorithm scales very well (linear), whereas calculation of the optimal solution does not (NP-hard). For 20 tasks, the maximum observed in the dataset, the MAGNET algorithm took a total of just under 12 msec.

For future work, we want to investigate how giving the *supplier* agents a preference for tasks would affect the distance from the optimal solution. In the logistical domain for example, drivers of trucks tend to have particular preferences for tasks which is often taken into consideration by human planners. Another interesting part of future work is to see how the ordering of tasks before feeding them into the *supplier* algorithm influences the overall results. More advanced ordering methods might result in better overall results. Furthermore, we want to investigate the scaling of the algorithms for very large datasets, consisting of for instance thousands of tasks that need to be allocated. Finally, a comparison of the approach presented in the paper with current (non-optimal) approaches being used in the logistics domain (e.g. simulated annealing or genetic programming) would be interesting to investigate as well.

Acknowledgments

The authors would like to thank the logistics company for providing the data set. Furthermore, the authors thank the anonymous reviewers for their valuable comments. The research of Mark Hoogendoorn and Catholijn Jonker has been conducted as part of the DEAL (Distributed Engine for Advanced Logistics) project, funded by the Dutch Ministry of Economic Affairs. The research of Maria Gini has been partially funded by the National Science Foundation under grant IIS-0414466.

7. REFERENCES

[1] A. Babanov, J. Collins, and M. Gini. Asking the right question: Risk and expectation in multi-agent contracting. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 17(4):173–186, September 2003.

[2] 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.

[3] M. Charikar and B. Raghavachari. The finite capacity dial-a-ride problem. In *39th Annual Symposium on Foundations of Computer Science*, page 458, Los Alamitos, CA, USA, 1998. IEEE Computer Society.

[4] S. Chien, A. Barrett, T. Estlin, and G. Rabideau. A comparison of coordinated planning methods for cooperating rovers. In *Proc. of the Fourth Int'l Conf. on Autonomous Agents*, pages 100–101. ACM Press, 2000.

[5] J. Collins. *Solving Combinatorial Auctions with Temporal Constraints in Economic Agents*. PhD thesis, University of Minnesota, June 2002.

[6] 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.

[7] 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.

[8] J. Cordeau and G. Laporte. The dial-a-ride problem (darp): Variants, modeling issues and algorithms. *4OR*, 1, 2003.

[9] J. S. Cox, E. H. Durfee, and T. Bartold. A distributed framework for solving the multiagent plan coordination problem. In *Autonomous Agents and Multi-Agent Systems*, pages 821–827, 2005.

[10] M. Desrochers, J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2):342–354, 1992.

[11] M. B. Dias, R. M. Zlot, N. Kalra, and A. T. Stentz. Market-based multirobot coordination: A survey and analysis. Technical Report CMU-RI-TR-05-13, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2005.

[12] K. Dorer and M. Calisti. An adaptive solution to dynamic transport optimization. In *Proc. of AAMAS05*, pages 45–51, 2005.

[13] E. H. Durfee. Scaling up agent coordination strategies. *IEEE Computer*, 34(7):39–46, July 2001.

[14] A. Glass and B. J. Grosz. Socially conscious decision-making. In *Proc. of the Fourth Int'l Conf. on Autonomous Agents*, pages 217–224, June 2000.

[15] M. Hoogendoorn and C. M. Jonker. Formation of virtual organizations through negotiation. In *Proceedings of the Fourth German Conference on Multiagent Technologies (MATES 2006)*, pages 135–146. Springer, 2006.

[16] L. Hunsberger and B. J. Grosz. A combinatorial auction for collaborative planning. In *Proc. of 4th Int'l Conf on Multi-Agent Systems*, pages 151–158, Boston, MA, 2000. IEEE Computer Society Press.

[17] J. J. Jaw, A. Odoni, H. Psaraftis, and N. Wilson. Heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B*, 20B:243–257, 1986.

[18] R. Kasilingam. *Logistics and Transportation: Design and Planning*. Springer, 1999.

[19] R. E. Korf. Depth-first iterative deepening: An optimal admissible tree search. *Artificial Intelligence*, 27:97–109, 1985.

[20] V. Krishna. *Auction Theory*. Academic Press, London, UK, 2002.

[21] T. Magnanti. Combinatorial optimization and vehicle fleet planning: Perspectives and prospects. *Networks*, 11:179–214, 1981.

[22] R. Mailler and V. Lesser. Solving distributed constraint optimization problems using cooperative mediation. In *Proc. of AAMAS04*, 2004.

[23] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. An asynchronous complete method for distributed constraint optimization. In *Proc. of AAMAS03*, 2003.

[24] T. A. Moehlman, V. R. Lesser, and B. L. Buteau. Decentralized negotiation: An approach to the distributed planning problem. *Group Decision and Negotiation*, 1:161–191, 1992.

[25] T. Notteboom. Container shipping and ports: An overview. *Review of Network Economics*, 3(2):86–106, 2004.

[26] P. Scerri, A. Farinelli, S. Okamoto, and M. Tambe. Allocating tasks in extreme teams. In *Proc. of AAMAS05*, pages 727–734, 2005.

[27] M. C. Schut, M. Kentrop, M. Leenaarts, M. Melis, and I. Miller. Approach: Decentralised rotation planning for container barges. In *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 755–759, 2004.

[28] 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.

[29] W. E. Walsh, M. Wellman, and F. Ygge. Combinatorial auctions for supply chain formation. In *Proc. of ACM Conf on Electronic Commerce (EC'00)*, pages 260–269, October 2000.

[30] M. Wellman, J. MacKie-Mason, D. Reeves, and S. Swaminathan. Exploring bidding strategies for market-based scheduling. In *Proc. of Fourth ACM Conf on Electronic Commerce*, 2003.

[31] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Economic Behavior*, 35:271–303, 2001.