Agents Preferences in Decentralized Task Allocation (extended abstract) [1]

Mark Hoogendoorn [a, b]   Maria L. Gini [b]

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

[b] *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*

1 Introduction

Auctions are used in multi-agent systems, among other things, to perform allocation of tasks (see e .g. [2] and [3]). Such reverse auctions, where the buyer is the auctioneer, can be of a combinatorial type, allowing for bidding on bundles of tasks. Sandholm [1] 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 namely (1) duration, (2) task type, and (3) start and end time. The function 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.

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.

Below, a brief overview of the approach is given, including the results using the synthetic and the real life dataset.

2 Bidding using Preferences

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). Hereby, each of these preferences can be specified in a natural way, which is mapped (using particular functions) to a real number on the interval [0,1]. Given these three ingredients of the preference function, a weighed sum is taken, resulting in a preference value for the specific task (ϕ_{task}). In the bidding algorithm we propose, price is used as a mechanism to express these preferences for tasks.

The bidding algorithm proposed starts when a request for quotes (RFQ) arrives. The tasks within this RFQ are then ordered based upon their preference. If some tasks have identical preferences, they are ordered

[1] The full version of the paper appeared in: M. Ghallab, C.D. Spyropoulos, N. Fakotakis and N. Avouris (eds.), Proceedings of the 18th European Conference on Artifi cial Intelligence, ECAI'08. IOS Press, 2008, pp. 398-402.

according to the start time specified in the RFQ for the tasks included. We assume that there exists a function *switch_time:* TASK_TYPE × TASK_TYPE → DURATION that calculates the switching time from one task type to another (when it can be performed on the resource). Furthermore, *performance_time:* TASK_TYPE → DURATION expresses the time needed to perform the task.

Bidding Algorithm

For each preference ordered task:
  Check if task (*current*) can be done using the resource.
  If yes, see if it fits in the current schedule (see below).
  From the beginning of the schedule and for each empty slot
  in the schedule do:
    If the task fits in the current empty slot in the schedule
    then insert the task in the bid,
      add its time parameters to the schedule, and
      compute the price of the bid (see below)
    else if $latest\_end\_time_{current} > latest\_end\_time_{next}$
      then continue with the next slot
      else continue with the next task.
The price of the bid is computed as follows (note the parameter *p*):

$$price_{task} = (1 + (p \times (1 - \phi_{task}))) \times [switch\_time(type_{previous}, type_{current}) + switch\_time(type_{current}, type_{next},) + performance\_time(type_{current})]$$

3 Results

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 in which the tasks need to be performed and the relative availability of resources. It was shown that it was easiest to get preferences awarded for 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 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.

Acknowledgments: Partial support is gratefully acknowledged from NSF under grant IIS-0414466.

References

[1] Tuomas 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.

[2] 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.

[3] William Walsh and Michael Wellman. A market protocol for decentralized task allocation and scheduling with hierarchical dependencies. In *Proc. of 3th Int'l Conf on Multi-Agent Systems*, 1998.