

Global Optimization for Multiple Agents

(Extended Abstract)

Brammert Ottens, Boi faltings
Artificial Intelligence Laboratory, EPFL
Lausanne, Switzerland
firstname.lastname@epfl.ch

ABSTRACT

In many situations, agents optimize their own operations locally but their local problems are interdependent. We consider the problem of coordinating these local problems to find a globally optimal solution. We model the coordination problem as a Distributed Constraint Optimization Problem that can take advantage of the locality of interactions, and then show how incremental elicitation and solving techniques can minimize the effort required. We illustrate the approach on an example of coordinating logistics service providers, for example couriers in a delivery company.

1. INTRODUCTION

In any cooperative multi agent setting, agents need to coordinate their actions. Take as an example a logistics setting, where individual couriers pick up and deliver packets. The decision on who is *assigned* which packet must be coordinated, i.e. no two couriers should try to pick up the same packet, because this leads to a waste of resources. Furthermore, no two *coordinated* assignments are the same, i.e. some assignments use the available resources more efficiently than others. Also, individual couriers could have individual preferences over different routes. They might prefer one restaurant over another for lunch, know which roads are more likely to be congested and so on and so forth. This latter information is difficult to formalize, and hence it would be best if each courier would be able to plan its own route. We thus have a *global* coordination problem with *local* preferences.

More generally, one can say that coordination problems consist of two highly interrelated parts: the local problems of each particular agent, that determine the preferences of individual packet assignments, and the coordination problem that determines which assignments are compatible. In this paper we argue that the DCOP paradigm [2] is a natural model for coordination problems. DCOP algorithms, however, have been designed without regard of the possible difficulty of solving the local problems of the agents. This paper evaluates a range of DCOP algorithms on problems with non-trivial local problems.

2. MULTI AGENT COORDINATION PROBLEMS

Many coordination problems are resource or task allocation problems: given a finite set of resources or tasks, distribute them over

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Copyright © 2012, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

the agents. Often, however, multiple outcomes are feasible, and the agents have preferences over these outcomes. A solution to a coordination problem is thus a feasible outcome that maximizes the local preferences of the different agents.

An example of a coordination problem can be found in logistics. Being able to efficiently distribute goods using couriers has large practical value. The types of problems we are looking at in this paper are inspired by a project with a courier company in a large European city, where couriers make independent decisions about their routes. The particular model used in this paper is an adaptation of the Truck Task Coordination (TTC) problem as given in [4].

We take the perspective of a single company, consisting of a group of couriers, dispersed over a geographical area. Each courier has its own *garage*, from which it operates. Customers offer packets for pickup and delivery to the company, but there are restrictions on which couriers are allowed to service them: packets will only be offered to couriers whose garages are within a certain range of the pickup and delivery locations. Furthermore, the range of the couriers will also be limited. These two restrictions together make that not all possible packet assignments are feasible. The main difference with standard VRP problems [6] is that not all couriers are able to service all packets. Furthermore, the goal is to maximize utility and not to minimize driving distance, where the utility is defined as the payment obtained from delivering a set of packets, minus the cost incurred by driving.

The DCOP paradigm is well suited to model the TTC problem. Let $P_i \subseteq \mathcal{P}$ be the set of packets that has been offered to courier t_i , and let $T_j \subseteq \mathcal{T}$ be the set of couriers that have been requested to deliver packet p_j . Then for each courier t_i and every packet p_j such that $p_j \in P_i$ and $|T_j| > 1$, t_i owns a binary variable x_j^i . Packets p_j for which $|T_j| = 1$ are assumed to be delivered by this courier, if within *courierRange*. If $x_j^i = 1$, then t_i will service packet p_j , and if $x_j^i = 0$ it will not service it. If a packet is not serviced, a penalty γ is incurred. For each packet, this is modeled through a $|T_j|$ -ary constraint, running over all variables that represent the particular packet. The coordination constraint must enforce that no two couriers will deliver the same packet. This is captured by the following hard constraint.

$$x_k^i + x_k^j \leq 1 \quad (1)$$

The local utility of an agent depends on the set of packets it is assigned, and on the route that it will take. Only packets that are within a certain distance are to be serviced. Let $distance(t_i, p_j) = true$ when both the pickup and delivery city of packet p_j are within *courierRange*, and *false* otherwise. Then $ownPackets_i = \{p_j | distance(t_i, p_j) = true \wedge |T_j| = 1\}$ is the set of packets that only courier t_i can service, and $coordinationPackets_i = P_i \setminus ownPackets_i$ is the set of packets courier t_i needs to co-

ordinate over. For every delivered packet p_j , a courier gets paid αw_{p_j} , where w_{p_j} is the weight of packet p_j and α is the payment per unit weight. Hence it is guaranteed a payment of $pr = \sum_{p \in \text{ownPackets}_i} \alpha w_p$. The cost of the route is provided by the local solver. If the assignment contains a packet whose pickup or delivery city is outside of the courierRange, or when the capacity constraint cannot be met, it is infeasible and this is indicating by setting the cost to ∞ .

$$vrp_i(\text{ownPackets}_i, x_{j_1}^i, \dots, x_{j_m}^i) = \quad (2)$$

$$pr + \sum_{k=1}^m x_{j_k}^i \alpha w_{p_{j_k}} - \text{cost of the route}$$

3. EXPERIMENTAL RESULTS

The main goal of the experiments presented in this section is to investigate the influence of the presence of non trivial local problems on the complexity of the coordination problem. Performance is measured using both simulated time [5] and Non Concurrent Constraint Checks (NCCC) [1].

3.1 Experimental Setup

We evaluated several DCOP algorithms on the TTC problem. We created a map of size 1000x1000, with 50 cities, 6 couriers and 16 packets randomly dispersed over the cities. The customer range and courier range are taken from {200, 250, 300, 350, 400} and for each combination of parameters we generated a 101 instances. We evaluated DPOP, O-DPOP, DSA, MGM and MGM2 on all these problems, setting the penalty for querying the local problem at 60 seconds per query. The local solver used for O-DPOP is not guaranteed to generate packet assignments in a best first order, and any violations are recognized and the utilities capped. The local search algorithms are run until they converge.

All algorithms have been written in Java, and have been implemented in the Frodo [3] framework. All experiments have been run on a four core Intel xeon 3 Ghz machine running linux, and each run was allocated 2 Gb of internal memory.

3.2 Results

Due to space constraints we show only a fraction of the experimental results. Figure 1 shows the solution quality obtained by the algorithms, and Figure 2 shows the simulated runtime. First, note that the local search algorithms are only able to find good solutions for the simplest problems. Furthermore, the lack of results for local search algorithms on problems with a customer range of more than 350 is caused by the fact that they are not able to find feasible solutions. The more complex the problems become, the more suboptimal the found solutions are. As for runtime, it is clear that O-DPOP outperforms DPOP. For the smaller problems it is even faster than the local search algorithms, but never worse.

4. CONCLUSIONS

There is a wide range of methods for solving coordination problems. In this paper we show that the DCOP paradigm is a natural choice for modeling such problems. Experimental results show that DCOP methods allow one to take advantage of the problem structure to obtain optimal solutions with reasonable complexity.

Furthermore, the results show that if the problems do not ask for a great amount of coordination, i.e locally good solutions are part of globally good solutions, the incremental elicitation algorithm O-DPOP needs to perform little to no work to find the optimal allocation of goods. It even outperforms local search algorithms.

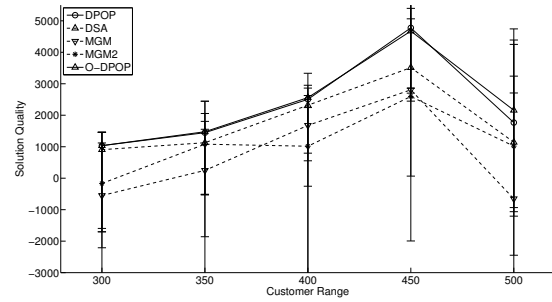


Figure 1: Utility for courier range of 400

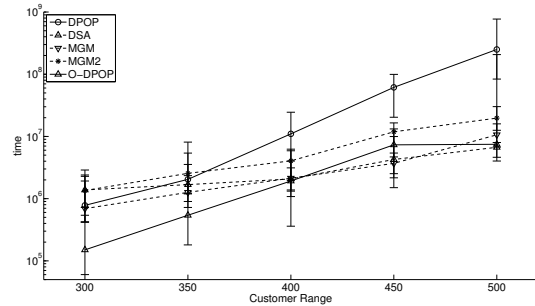


Figure 2: Simulated Runtime for courier range of 400

In future work, we plan to improve the preference elicitation scheme used here to make it more efficient and also allow for anytime performance. We are also considering more efficient data structures for task allocation that could improve the communication efficiency of the process.

5. REFERENCES

- [1] Amir Gershman, Roie Zivan, Tal Grinshpoun, Alon Grubshtein, and Amnon Meisels. Measuring distributed constraint optimization algorithms. In *Proceedings of DCR'08*, pages 17–24, May 13 2008.
- [2] Katsutoshi Hirayama and Makoto Yokoo. Distributed partial constraint satisfaction problem. In *CP'97*, pages 222–236, 1997.
- [3] Thomas Léauté, Brammert Ottens, and Radoslaw Szymanek. FRODO 2.0: An open-source framework for distributed constraint optimization. In Katsutoshi Hirayama, William Yeoh, and Roie Zivan, editors, *Proceedings of the IJCAI'09 Distributed Constraint Reasoning Workshop (DCR'09)*, pages 160–164, July 13 2009.
- [4] Brammert Ottens and Boi Faltings. Coordinating agent plans through distributed constraint optimization. In *Proceedings of the ICAPS'08 Multiagent Planning Workshop (MASPLAN'08)*, September 14 2008.
- [5] Evan A. Sultanik, Robert N. Lass, and William C. Regli. DCOPolis: A framework for simulating and deploying distributed constraint optimization algorithms. In Jonathan P. Pearce, editor, *Proceedings of the 9th Intl Workshop on Distributed Constraint Reasoning (CP-DCR'07)*, September 23 2007.
- [6] Paolo Toth and Daniele Vigo, editors. *The Vehicle Routing Problem*. SIAM, 2001.