The Multi Variable Multi Constrained Distributed Constraint Optimization Framework

(Extended Abstract)

Christopher Portway
University of Michigan, Ann Arbor, MI 48109
portway@umich.edu

Edmund H. Durfee
University of Michigan, Ann Arbor, MI 48109
durfee@umich.edu

ABSTRACT

We extend the MC-DCOP model to problems where each agent controls multiple variables, map a service-oriented computing domain to this MV-MC-DCOP model, and use the solutions as a preprocessing step to an existing inexact MDP solver.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *Multiagent Systems*

General Terms

Algorithms, Experimentation

Keywords

Distributed Constraint Optimization, Multi-agent Coordination, Task Scheduling

1. INTRODUCTION

In service-oriented computing environments, choreographing services with stochastic behaviors (durations and results) can amount to formulating joint policies, a generally intractable problem [1]. In response, suboptimal local search techniques have been developed, such as a greedy approach [5]. Unfortunately, in overconstrained situations (where service requests must be denied), the greedy approach's myopic view can aggravate its suboptimality, as some requests are greedily scheduled that ultimately add no value because other complementary requests are denied.

Our work investigates how distributed constraint optimization (DCOP) techniques can help avoid such situations by first identifying (approximately) optimal combinations of service requests that can be feasibly scheduled together. The idea is to abstract away details about services' interactions and stochastic behaviors in order to formulate and solve the problem as a DCOP, and then use the DCOP solution(s) to guide the joint policy search, allowing faster convergence to better (but still potentially suboptimal) solutions.

We consider the MC-DCOP algorithm [2] as a logical starting point for our preprocessing model, due to its simplicity and separation of constraints into a maximizing function and a limiting function. However, MC-DCOP assumes that each agent controls only a single variable, whereas in our problem service agents can be responsible for granting or denying multiple requests. We thus introduce a more general variant of MC-DCOP, which we call Multi-Variable MC-DCOP (MV-MC-DCOP), that removes the single variable assumption and requires agents to assign values to their local variables that satisfy constraints and maximize reward.

Cite as: The Multi Variable Multi Constrained Distributed Constraint Optimization Framework (Extended Abstract), Christopher Portway and Edmund H. Durfee, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1385-1386

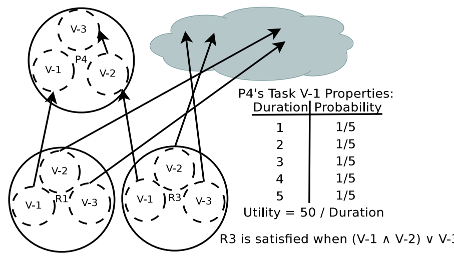

Figure 1: Partial View of a Service Coordination Problem

2. SERVICE COORDINATION DOMAIN

Our motivating domain of service coordination has two types of agents: service providers and requesters. Each agent has internal tasks, which may depend on the prior completion of other tasks, internal or external to the agent. An agent is considered satisfied when any of a group of subsets of its internal tasks are completed, and unsatisfied agents receive no rewards. The task durations can be uncertain, and several task reward structures may exist.

Figure 1 shows a portion of a graph representing a larger problem, with two requesters and one provider visible. Details are shown for the durations and rewards of variable (request) 1 in agent P4, and satisficing requirements for requester R3. There are 3 requests visible, creating 2 task chains: P4's task V-1 provides for R1's task V-1, and P4's task V-3 provides for P4's task V-2, which in turn provides for R3's V-1.

Casting this type of service coordination problem into a DCOP requires abstracting away detailed information about things like uncertainties and task relationships. The DCOP model of each task has a deterministic duration and outcome, chosen to trade off risk of missing a high-utility combination with risk of proposing an infeasible combination. For Figure 1, a pessimistic choice assigns P4's V-1 a duration of 5 and utility of 10.

Details about task relationships, such as in Figure 1 where the condition for R(equester) 3 to be satisfied involves a Boolean expression over which requests are satisfied, are also reduced to where an agent simply has a minimum number of variables that must be satisfied for the agent to receive a reward. In Figure 1, setting R3's *minSatisfied* to 2 (more than) ensures that R3 is satisfied when-

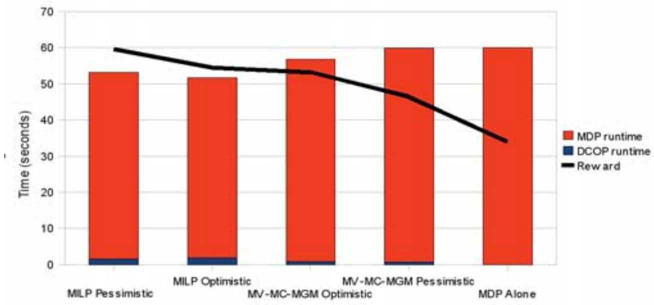

Figure 2: MV-MC-DCOP seeded & unseeded Greedy MDP with $2.\overline{3}$ requests per agent

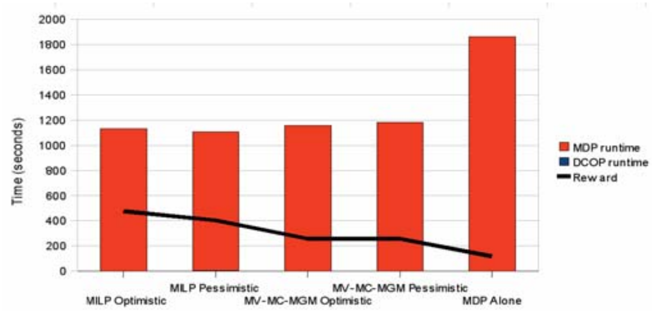

Figure 3: MV-MC-DCOP seeded & unseeded Greedy MDP with 3.2 requests per agent

ever this constraint is met. Further, for our initial implementation described here, we also ignore ordering between tasks within an agent, and prohibit an agent from acting as both a provider and requester. Hence, in Figure 1, tasks V-2 and V-3 are merged, with the new super-task summing the individual utilities and concatenating the durations.

With these transformations, the resulting DCOP essentially requires that agents assign subsets of their local variables to TRUE, satisfying the various constraints (e.g., that the total durations of the tasks an agent agrees to do cannot exceed the available time) and maximizing the total reward across the agents.

3. MV-MC-DCOP ALGORITHMS

3.1 Mixed Integer Linear Program

We have formulated a Mixed Integer Linear Program(MILP) which performs a simplex search of a graph in the MV-MC-DCOP framework [4]. This returns a globally optimal solution, if one exists, but requires all data and computation to be centralized.

3.2 MV-MC-MGM Local Search

We have extended the existing MC-MGM [2] algorithm to find local optimal solutions to MV-MC-DCOPs more quickly than the complete MILP. We call this new algorithm MV-MC-MGM [4], which is a hill-climbing-based local search algorithm that uses a decomposition approach for dealing with multiple variables per agent (see [3] for trade-offs in approaches). In MV-MC-MGM, each variable calculates its best move (variable reassignment) given the values of neighboring variables and its budget allocated by the agent it is subordinate to, and broadcasts the gain of this move. If no directly connected neighbor has a higher gain, then the agent moves.

4. EVALUATION AND DISCUSSION

We implemented these MV-MC-DCOP algorithms, and tested whether they indeed can be a fruitful preprocessing step to improve the performance of the greedy MDP solver [5]. We generated random problems with 16 requests (and therefore 32 externally connected variables), half of which had high reward and low duration, and the other half the opposite. These were distributed among 6 agents. First each agent was assigned to be either a requester or provider, and then the pairs of tasks in a binary request were randomly assigned. Figure 2 compares the performance of the MV-MC-MILP and MV-MC-MGM approaches, for each trying both optimistic and pessimistic strategies for determinizing task durations (hence utilities). We can observe that MV-MC-MILP helps the greedy MDP solver find higher-reward policies faster, but itself has higher runtime, while MV-MC-MGM has a lower runtime, but is somewhat less helpful to the greedy MDP solver. The short dark (blue) bands at the bottom of the bars in Figure 2 represent the MV-MC-DCOP's runtime. Both types of preprocessing, however, cause improvements compared to the greedy MDP solver working alone. Figure 3 presents similar data for the case where the number of agents is reduced to 5 while the number of requests and other parameters are held constant (although reward is scaled by 10 for visibility). This results in an increase from $2.\overline{3}$ requests per agent to 3.2. Now, because even more requests must be selectively declined, the advantages of DCOP preprocessing to decreasing runtime are accentuated, and the dark (blue) band is not even visible.

The MV-MC-MGM algorithm thus shows promise in reducing runtime and increasing reward while keeping computation distributed, as is the case with greedy MDP, and appears to scale linearly as internal complexity and number of agents increase. However, the centralized MILP approach results in lower average MDP-inclusive runtime, and higher reward as well. Our future work includes evaluating these techniques on a wider space of problems, and extending the MV-MC-DCOP formulation (and our algorithms) to model richer task interactions involving precedence ordering and task chains.

5. ACKNOWLEDGEMENTS

This work was supported in part by AFOSR under Contract No. FA9550-07-1-0262 and NSF under Grant IIS-0534280.

6. REFERENCES

[1] D. S. Bernstein, R. Givany, N. Immerman and S. Zilberstein. The Complexity of Decentralized Control of Markov Decision Processes. In *Mathematics of Operations Research*, 27(4):819–840, 2002.

[2] E. Bowring, J. P. Pearce, C. Portway, M. Jain, M. Tambe. On K-optimal Distributed Constraint Optimization Algorithms: New Bounds and Algorithms. In *AAMAS*, pages 607–614, 2008.

[3] D. A. Burke and K. N. Brown. A Comparison of Approaches to Handling Complex Local Problems in DCOP. In *Distributed Constraint Satisfaction Workshop, Riva del Garda, Italy*, pages 27–33, 2006.

[4] C. Portway and E. H. Durfee. The Multi Variable Multi Constrained Distributed Constraint Optimization Framework. In *IJCAI Distributed Constraint Reasoning Workshop*, 2009.

[5] S. J. Witwicki, E. H. Durfee. Commitment-Based Service Coordination. In *Int. J. Agent Oriented Software Engineering*, 3(1):59–87, 2009.