Distributed Constraint Optimization Problems related with Soft Arc Consistency

Patricia Gutierrez (student)
IIIA-CSIC
Universitat Autonoma de Barcelona
08193 Bellaterra, Spain
patricia@iiia.csic.es

Pedro Meseguer (advisor)
IIIA-CSIC
Universitat Autonoma de Barcelona
08193 Bellaterra, Spain
pedro@iiia.csic.es

ABSTRACT

Distributed Constraint Optimization Problems (DCOPs) are commonly used for modeling multi-agent coordination problems. DCOPs can be optimally solved by distributed search algorithms, based on messages exchange. In centralized solving, maintaining soft arc consistency techniques during search has proved to be beneficial for performance. In this thesis we aim to explore the maintenance of different levels of soft arc consistency in distributed search when solving DCOPs.

1. PRELIMINARIES

1.1 DCOPs

Distributed Constraint Optimization Problems (DCOPs) [12] can be used for modeling many multi-agent coordination problems, such as distributed meeting scheduling [10], sensor networks [7], traffic control [8], and others. DCOPs involve a finite number of agents, variables and cost functions. The cost of an assignment of a subset of variables is the evaluation of all cost functions on that assignment. The goal is to find a complete assignment with minimum cost.

Researchers have proposed several distributed search algorithms to optimally solve DCOPs. The first proposed complete algorithm was ADOPT [12], which performs distributed search using a best-first strategy. Later on, the closely related BnB-ADOPT [13] was presented. This algorithm changes the nature of the search from ADOPT best-first search to a depth-first branch-and-bound search strategy, obtaining a better performance. Both algorithms are complete, compute the optimum cost and terminate.

DCOPs are NP-hard, so an exponential time is needed in the worst case to find the optimum. This severely limits the application of existing solving approaches.

1.2 Soft Arc Consistency

In the centralized case, several techniques have been developed to speed up the solving of constraint optimization problems. In particular, search can be improved by enforcing soft arc consistency, which identifies inconsistent values that can be removed from the problem.

Several soft arc consistency levels have been proposed [9]. By enforcing them it is possible to infer that some values are suboptimal and can be removed from the problem. In practical terms, the effect is that the search tree is reduced and there are fewer nodes to explore, but on the other hand more computational work must be done per node. Globally, the overall effect is very beneficial.

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

2. THESIS GOAL

The thesis goal is to include in distributed search algorithms for DCOPs solving some techniques to enforce soft arc consistency during search. Such as it happens in the centralized case, we expect that this combination would cause performance improvements.

We consider soft arc consistencies conceptually equal in the centralized and distributed case. However, maintaining soft arc consistencies during distributed search requires different techniques. While in the centralize case all problem elements are available to the single agent performing the search, in the distributed case agents only know some part of the problem and must exchange information in order to achieve the desired consistency level. In this process, operations that modify the problem structures should be done in such a way that the partial representation of the whole problem remains coherent on every agent.

We measure the efficiency of the proposed algorithms with respect to existing ones in terms of (synchronous) cycles [12], non-concurrent constraint checks (NCCCs) [11], and network load (total number of messages exchanged).

3. RESEARCH DONE

3.1 Contributions to Distributed Search

We have experimented with existing complete DCOP algorithms, namely ADOPT and BnB-ADOPT. As result of this work, we have improved them to a large extent, as explained in the following.

On their execution, ADOPT and BnB-ADOPT exchange a large number of messages, which is a major drawback for their practical use. Aiming at increasing their efficiency, we show that some of these messages are redundant and can be removed without compromising optimality and termination. Removing most of those redundant messages we obtain ADOPT^{+} [3] and BnB-ADOPT^{+} [4]. When tested on commonly used benchmarks, these algorithms obtain large reductions in the number of messages, a slight reduction in NCCCs, and the number of cycles remains constant. BnB-ADOPT^{+} was able to process only half of messages in the worst case and reach the optimal solution in almost the same number of cycles [4], while ADOPT^{+} divided the number of messages by a factor from 1.1. to almost 3 on the benchmarks tested [3].

In addition, we have proposed the new algorithm ADOPT(k) [6], which generalizes ADOPT and BnB-ADOPT. These two algorithms share similar data and message structures, but differ on their search strategies: the former uses best-first search and the latter uses depth-first branch-and-bound search. ADOPT(k) generalizes ADOPT and BnB-ADOPT in the following way. Its behavior depends on the k parameter. It behaves like ADOPT when $k = 1$, like BnB-ADOPT when $k = \infty$ and like a hybrid of ADOPT and BnB-ADOPT when $1 < k < \infty$. We prove that ADOPT(k) is a correct and complete algorithm and experimentally show that ADOPT(k) outperforms ADOPT and BnB-ADOPT in terms of runtime and network load on several benchmarks across several metrics. Additionally, ADOPT(k) provides a good mechanism for balancing the tradeoff between runtime and network load [6].

3.2 Connection with Soft Arc Consistency

We have experimented with BnB-ADOPT$^+$, on top of which we maintain the soft arc consistency levels AC and FDAC. During BnB-ADOPT$^+$ execution, we can assure in some cases that the value of a variable is not in the optimal solution. Then, this value is deleted unconditionally in the agent handling the variable. Unconditional deletions are propagated in such a way that they can be known by other neighboring agents. When deletions are propagated, AC/FDAC is reinforced on neighboring agents, which may generate new deletions that will also be propagated. The global effect is that we search in a smaller space, causing performance improvements.

We presented the new algorithms BnB-ADOPT$^+$- AC and BnB-ADOPT$^+$-FDAC [2], which combine distributed search with the levels of consistencies AC and FDAC. Maintaining AC level (BnB-ADOPT$^+$-AC) we observe a clear decrement in the number of messages and also in the number of cycles. Maintaining FDAC level (BnB-ADOPT$^+$-FDAC) enhances this reduction. In the worst case, when maintaining FDAC our approach divides the number of required messages by a factor of 3, substantially decreasing the number of cycles as well. Although agents need to perform more local computation to maintain consistency and some new messages are introduced to propagate deletions, this is largely compensated by a decrement in the number of messages used to solve the problem and as result, the number of NCCCs shows important reductions as well [2].

More recently, we presented several improvements for BnB-ADOPT$^+$-AC [5]. First, we propose some modifications in the implementation of the algorithm. Secondly, we address the issue of simultaneous deletions. When neighboring agents perform deletions at the same time, the order of projections in both agents is opposite and as a result some costs might be lost from the problem. This costs that were lost could have contributed to identify sub-optimal values. To avoid this, we provide a synchronization mechanism to assure that projections are always done in the same order on every agent. Finally, we propose to search on cost functions that are made AC in a preprocessing step. With this, lower bounds calculated for every value can provide a heuristic for value selection during search.

The aggregation of these three modifications produces a complete algorithm with communication and computation efforts substantially smaller than previous versions of BnB-ADOPT$^+$ (including either AC [2], DP2 heuristics [1] or a combination of both). In most cases, messages and NCCCs are reduced by at least a factor of 2, reaching up to one order of magnitude for some cases [5].

4. FUTURE WORK

We aim to extend our work to higher levels of soft arc consistency. To maintain these levels agents need to have a wider knowledge about the global problem. This may compromise privacy, which is an issue to resolve.

Furthermore, we want to explore soft arc consistency maintenance in problems with global constrains. In this case, performance can be improved, since often local consistency can be achieved more efficiently when global constraints are involved instead of an equivalent binary formulation.

5. REFERENCES

[1] S. Ali, S. Koenig, and M. Tambee. Preprocessing techniques for accelerating the DCOP algorithm ADOPT. *Proc. of AAMAS*, 2005.

[2] P. Gutierrez and P. Meseguer. BnB-ADOPT$^+$ with several soft arc consistency levels. In *ECAI*, pages 67–72, 2010.

[3] P. Gutierrez and P. Meseguer. Saving redundant messages in ADOPT-based algorithms. In *DCR*, pages 53–64, 2010.

[4] P. Gutierrez and P. Meseguer. Saving redundant messages in BnB-ADOPT. In *AAAI*, pages 1259–1260, 2010.

[5] P. Gutierrez and P. Meseguer. Improving BnB-ADOPT$^+$-AC. In *AAMAS*, page To appear, 2012.

[6] P. Gutierrez, P. Meseguer, and W. Yeoh. Generalizing ADOPT and BnB-ADOPT$^+$. In *IJCAI*, pages 554–559, 2011.

[7] M. Jain, M. Taylor, M. Tambe, and M. Yokoo. DCOPs meet the realworld: Exploring unknown reward matrices with applications to mobile sensor networks. In *IJCAI*, 2009.

[8] R. Junges and A. L. C. Bazzan. Evaluating the performance of DCOP algorithms in a real world dynamic problem. In *AAMAS*, pages 599–606, 2008.

[9] J. Larrosa and T. Schiex. In the quest of the best form of local consistency for weighted CSP. *IJCAI*, 2003.

[10] R. Maheswaran, M. Tambe, E. Bowring, J. Pearce, and P. Varakantham. Taking DCOP to the real world: Efficient complete solutions for distributed event scheduling. In *AAMAS*, pages 310–317, 2004.

[11] A. Meisels, E. Kaplansky, I. Razgon, and R. Zivan. Comparing performance of distributed constraints processing algorithms. In *DCR*, pages 86–93, 2002.

[12] P. Modi, W. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161:149–180, 2005.

[13] W. Yeoh, A. Felner, and S. Koenig. BnB-ADOPT: An asynchronous branch-and-bound DCOP algorithm. *Journal of Artificial Intelligence Research*, 38:85–133, 2010.