Pseudo-tree-based Algorithm for Approximate Distributed Constraint Optimization with Quality Bounds

(Extended Abstract)

Tenda Okimoto, Yongjoon Joe, Atsushi Iwasaki, and Makoto Yokoo
Department of Informatics, Kyushu University
Fukuoka, Japan
{tenda@agent., yongjoon@agent.,iwasaki@, yokoo@}is.kyushu-u.ac.jp

ABSTRACT

Most incomplete DCOP algorithms generally do not provide any guarantees on the quality of the solutions. In this paper, we introduce a new incomplete DCOP algorithm that can provide the upper bounds of the absolute/relative errors of the solution, which can be obtained a priori/a posteriori, respectively. The evaluation results illustrate that this algorithm can obtain better quality solutions and bounds compared to existing bounded incomplete DCOP algorithms, while the run time of this algorithm is much shorter.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multi-agent systems*

General Terms

Algorithms, Experimentation

Keywords

Distributed Constraint Optimization Problem, Pseudo-tree, Induced Width

1. INTRODUCTION

A Distributed Constraint Optimization Problem (DCOP) is a fundamental problem that can formalize various application problems in multi-agent systems, e.g., distributed sensor networks [4] and meeting scheduling [6]. Since DCOP is NP-hard, considering faster incomplete algorithms is necessary for large-scale applications. Most existing incomplete algorithms generally do not provide any guarantees on the quality of the solutions. Some notable exceptions are DALO [3], the bounded max-sum algorithm [2], and ADPOP [5]. Among these algorithms, DALO is unique since it can obtain a priori bound. Also, the obtained bound is independent of problem instances. On the other hand, the bounded max-sum algorithm and ADPOP can only obtain a posteriori bound. Having a priori bound is desirable, but a posteriori bound is usually more accurate.

Cite as: Pseudo-tree-based Algorithm for Approximate Distributed Constraint Optimization with Quality Bounds (Extended Abstract), Tenda Okimoto, Yongjoon Joe, Atsushi Iwasaki, and Makoto Yokoo, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1269-1270.

In this paper, we introduce an incomplete algorithm based on a new solution criterion called *p-optimality*. This algorithm can provide the upper bounds of the absolute/relative errors of the solution, which can be obtained a priori/a posteriori, respectively. These bounds are based on the *induced width* p of a constraint graph [1] and the maximal value of each reward function, but they are independent of problem instances. This algorithm utilizes a graph structure called a pseudo-tree, which is widely used in complete DCOP algorithms such as ADOPT [4] and DPOP [6]. This algorithm is a one-shot type algorithm, which runs in polynomial-time in the number of agents n. Furthermore, this algorithm has adjustable parameter p, so that agents can trade-off better solution quality against computational overhead.

DALO is an algorithm based on the criteria of local optimality called *k-size/t-distance optimality* [3]. Compared to this algorithm, our algorithm is a one-shot type algorithm, while DALO is an anytime algorithm. Also, our algorithm can provide tighter bounds a priori. The bounded max-sum algorithm is a one-shot type algorithm. Compared to this algorithm, our algorithm has adjustable parameter p, while this algorithm has no adjustable parameter. Also, our algorithm can obtain a priori bound. Our algorithm is quite similar to ADPOP, which also eliminates edges among variables to bound the size of messages. ADPOP uses a heuristic method to determine which edges to eliminate. As a result, it cannot obtain a priori bound. We can consider p-optimality gives a simple but theoretically well-founded method to determine which edges to eliminate in ADPOP.

2. PRELIMINARIES

A distributed constraint optimization problem is defined by a set of agents S, a set of variables X, a set of binary constraint relations C, and a set of binary reward functions F. An agent i has its own variable x_i. A variable x_i takes its value from a finite, discrete domain D_i. A binary constraint relation (i,j) means there exists a constraint relation between x_i and x_j. For x_i and x_j, which have a constraint relation, the reward for an assignment $\{(x_i,d_i),(x_j,d_j)\}$ is defined by a binary reward function $r_{i,j}(d_i,d_j) : D_i \times D_j \to \mathbb{R}$. For a value assignment to all variables A, let us denote

$$R(A) = \sum_{(i,j)\in C, \{(x_i,d_i),(x_j,d_j)\}\subseteq A} r_{i,j}(d_i,d_j).$$

Then, an optimal assignment A^* is given as $\arg\max_A R(A)$, i.e., A^* is an assignment that maximizes the sum of the value of all reward functions.

A DCOP problem can be represented using a constraint graph, in which a node represents an agent/variable and an edge represents a constraint.

For a graph $G = (V, E)$, a total ordering o, and a node $i \in V$, we call $A(E, o, i) = \{j \mid (i, j) \in E \ \wedge \ j \prec i\}$ as i's ancestors, where we denoted $j \prec i$, if j occurs before i in o. We also denote ord(i) for the i-th node in o.

Definition 1 (Chordal graph based on total ordering). *For a graph $G = (V, E)$ and a total ordering o, we say G is a chordal graph based on total ordering o when the following condition holds:*

- $\forall i, \forall j, \forall k \in V$, *if $j, k \in A(E, o, i)$, then $(j, k) \in E$.*

Definition 2 (Induced chordal graph based on total ordering). *For a graph $G = (V, E)$ and a total ordering o, we say a chordal graph $G' = (V, E')$ based on total ordering o, which is obtained by the following procedure, as an induced chordal graph of G based on total ordering o.*

1. *Set E' to E.*
2. *Choose each node $i \in V$ from the last to the first based on o and apply the following procedure.*
   - *if $\exists j, \exists k \in A(E', o, i)$ s.t. $(j, k) \notin E'$, then set E' to $E' \cup \{(j, k)\}$.*
3. *Return $G' = (V, E')$.*

A parameter called *induced width* can be used as a measure for checking how close a given graph is to a tree. We call $w(G, o)$ as the width of graph G based on total ordering o and it is defined as $\max_{i \in V} |A(E, o, i)|$. Furthermore, we call $w(G', o)$ as the induced width of G based on total ordering o, where $G' = (V, E')$ is the induced chordal graph of G based on total ordering o.

A chordal graph $G = (V, E)$ based on total ordering o can be assumed as a pseudo-tree. We say an edge (i, j) is a back-edge of i, if $j \in A(E, o, i)$ and j is not i's parent. Also, when $(i, j_1), (i, j_2), \ldots, (i, j_k)$ are all back-edges of i, and $j_1 \prec j_2 \prec \ldots \prec j_k$ holds, we call them as first back-edge, second back-edge, ..., k-th back-edge, respectively.

3. BOUNDED INCOMPLETE ALGORITHM BASED ON INDUCED WIDTH

Our proposed incomplete algorithm has two phases:

Phase 1: Generate a subgraph from an induced chordal graph by removing several edges, so that the induced width of the induced chordal graph obtained from the subgraph is bounded by parameter p.

Phase 2: Find an optimal solution to the graph obtained in Phase 1 using any complete DCOP algorithms.

Let us describe Phase 1. To obtain such a subgraph is not easy. One might imagine that we can easily obtain such a subgraph by just removing the back-edges so that all nodes have at most $p - 1$ back-edges. However, by this simple method, we cannot guarantee that the remaining graph is a chordal graph and we might need to add some edges to make it a chordal graph. As a result, the induced width of the induced chordal graph can be more than p.

We develop a method for Phase 1 as follows.

Definition 3 (p-reduced graph). *For a chordal graph $G = (V, E)$ based on total ordering o, we say a graph $G' = (V, E')$ obtained by the following procedure as p-reduced graph of G (where $1 \le p \le w(G, o)$):*

1. *Set E' to E.*
2. *Repeat the following procedure $w(G, o) - p$ times.*
   - *For each $i \in V$ where $p + 1 \le ord(i) \le w(G, o)$ remove the first back-edge in $G' = (V, E')$ from E' if there is one.*
3. *Return $G' = (V, E')$.*

In Phase 1, a p-reduced graph is generated. Then, we can guarantee that the obtained graph is chordal and its induced width is p. Based on the idea of p-reduced graph, we introduce a new solution criterion as follows.

Definition 4 (p-optimality). *We say an assignment A is p-optimal for a distributed constraint optimization problem $\langle X, C, R\rangle$ and a total ordering o, when A maximizes the total rewards in $G'' = (X, C'')$, where $G' = (X, C')$ is an induced chordal graph of $G = (X, C)$ based on total ordering o, and $G'' = (X, C'')$ is the p-reduced graph of G'. More specifically, $\forall A', R_{C''}(A) \ge R_{C''}(A')$ holds.*

To find a p-optimal solution in Phase 2, we can use any complete DCOP algorithms. We use the obtained p-optimal solution as an approximate solution of the original graph.

Furthermore, we estimate two types of errors, i.e., absolute and relative errors of the solution. Absolute error can be obtained a priori. Intuitively, the absolute error is given by the product of the maximal value of each reward function and the maximal number of removed back-edges. Relative error can be obtained a posteriori. We can compute it using a method similar to ADPOP.

In our evaluations, we showed that our algorithm for p=1-optimality can obtain better quality solutions and estimate more accurate error bounds compared with DALO-t for t=1-distance-optimality and the bounded max-sum algorithm. Furthermore, the run time for our algorithm for p=1-optimality is much shorter compared to these existing algorithms.

4. REFERENCES

[1] R. Dechter. *Constraint Processing*. Morgan Kaufmann Publishers Inc., 2003.

[2] A. Farinelli, A. Rogers, and N. R. Jennings. Bounded approximate decentralised coordination using the max-sum algorithm. In *DCR*, pages 46–59, 2009.

[3] C. Kiekintveld, Z. Yin, A. Kumar, and M. Tambe. Asynchronous algorithms for approximate distributed constraint optimization with quality bounds. In *AAMAS*, pages 133–140, 2010.

[4] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artif. Intell.*, 161(1-2):149–180, 2005.

[5] A. Petcu and B. Faltings. Approximations in distributed optimization. In *CP*, pages 802–806, 2005.

[6] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *IJCAI*, pages 266–271, 2005.