Lp-Norm based Algorithm for Multi-Objective Distributed Constraint Optimization

(Extended Abstract)

Tenda Okimoto, Nicolas Schwind, Maxime Clement, Katsumi Inoue
National Institute of Informatics

ABSTRACT

In this paper, we develop a novel algorithm which finds a subset of Pareto front of a Multi-Objective Distributed Constraint Optimization Problem. This algorithm utilizes the L_p-norm method, pseudo-tree, and Dynamic Programming technique. Furthermore, we show that this L_p-norm based algorithm can only guarantee to find a Pareto optimal solution, when we employ L_1-norm (Manhattan norm).

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Theory

Keywords

Multi-Objective DCOP, L_p-Norm

1. INTRODUCTION

A *Multi-Objective Distributed Constraint Optimization Problem* (MO-DCOP) [3] is the extension of a Distributed Constraint Optimization Problem [6] which is a fundamental problem that can formalize various applications related to multi-agent cooperation. In MO-DCOPs, since trade-offs exist among objectives, there does not generally exist an ideal assignment, which maximizes all objectives simultaneously. Thus, the solutions of an MO-DCOP is characterized by using the concept of *Pareto optimality.* An assignment is a *Pareto optimal solution* if there does not exist another assignment that weakly improves all of the objectives. Solving an MO-DCOP is to find the *Pareto front* which is a set of reward vectors obtained by all Pareto optimal solutions.

Finding all Pareto optimal solutions of an MO-DCOP becomes easily intractable for large-scale problem instances. In MO-DCOPs, even if a constraint graph has the simplest tree structure, the number of all Pareto optimal solutions is often exponential in the number of agents. Since finding all Pareto optimal solutions is not realistic, it is important to consider the following two approaches. The first approach is to find a subset of Pareto front instead of finding all Pareto optimal solutions. The second is to develop a fast but incomplete algorithm. This paper focuses on the first approach.

An Aggregate Objective Function (AOF) [2] is a representative classical scalarization method which scalarizes the set of objective functions into a mono-objective function by multiplying each objective with a user supplied weight, and finds an optimal solution for it. It is well known that an optimal solution obtained by AOF is a Pareto optimal solution of the original problem. The L_p-norm method is another classical method [2] which finds a Pareto optimal solution using a distance from a reference point. This method is tractable, since we just need to give the reference point.

In this paper, we develop a novel MO-DCOP algorithm called *Multi-Objective L_p-norm based Distributed Pseudo-tree Optimization Procedure* (MO-DPOP$_{L_p}$) that is based on DPOP [6] and finds a subset of the Pareto front of an MO-DCOP using the L_p-norm and AOF methods. Also, we show that MO-DPOP$_{L_1}$ is the only MO-DPOP$_{L_p}$ algorithm that can guarantee to find a Pareto optimal solution.

2. MULTI-OBJECTIVE DCOP

A *Multi-Objective Distributed Constraint Optimization Problem* (MO-DCOP) [3] is the extension of a mono-objective DCOP. An MO-DCOP is defined with a set of agents S, a set of variables X, multi-objective constraints $C = \{C^1, \ldots, C^m\}$, i.e., a set of sets of binary constraint relations, and multi-objective functions $O = \{O^1, \ldots, O^m\}$, i.e., a set of sets of objective functions (binary reward functions). For an objective l $(1 \leq l \leq m)$, a binary reward function $r^l_{i,j} : D_i \times D_j \rightarrow \mathbb{R}^+$, and a value assignment to all variables A, let us denote

$$R^l(A) = \sum_{(i,j)\in C^l, \{(x_i,d_i),(x_j,d_j)\}\subseteq A} r^l_{i,j}(d_i, d_j), \qquad (1)$$

where $d_i \in D_i$ and $d_j \in D_j$. Then, the sum of the values of all reward functions for m objectives is defined by a reward vector, denoted $R(A) = (R^1(A), \ldots, R^m(A))$. Finding an assignment that maximizes all objective functions simultaneously is ideal. However, since trade-offs exist among objectives, there does not generally exist such an ideal assignment. Thus, the optimal solution of an MO-DCOP is characterized by using the concept of Pareto optimality.

Definition 1 (Dominance). *For an MO-DCOP and two reward vectors $R(A)$ and $R(A')$ obtained by assignments A and A', we say that $R(A)$ dominates $R(A')$, denoted by $R(A') \prec R(A)$, iff $R(A')$ is partially less than $R(A)$, i.e., (i) it holds $R^l(A') \leq R^l(A)$ for all objectives l, and (ii) there exists at least one objective l', such that $R^{l'}(A') < R^{l'}(A)$.*

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

Definition 2 (Pareto optimal solution). *For an MO-DCOP, we call an assignment A is the Pareto optimal solution, iff there does not exist another assignment A', such that $R(A) \prec R(A')$. The set of reward vectors obtained by all Pareto optimal solutions is said to be Pareto Front. Solving an MO-DCOP is to find the Pareto front.*

Lp-Norm

Let $v = (v_1, \ldots, v_m)$ and $w = (w_1, \ldots, w_m)$ be two vectors. An L_p-norm has the following form

$$dis_p(v, w) = (\sum_{i=1}^{m} |v_i - w_i|^p)^{\frac{1}{p}}, \tag{2}$$

where $1 \leq p \leq \infty$. Manhattan ($p = 1$), Euclidean ($p = 2$) and Chebyshev norms ($p = \infty$) are special cases of L_p-norm.

3. L_P-NORM BASED ALGORITHM

In this section, we develop a novel algorithm called *Multi-Objective L_p-norm based Distributed Pseudo-tree Optimization Procedure* (MO-DPOP$_{L_p}$) for MO-DCOPs. This algorithm utilizes the L_p-norm to find a Pareto optimal solution, a pseudo-tree, and is based on Dynamic Programming (DP) technique. Furthermore, we show that L_1-norm (Manhattan norm) based algorithm can guarantee to find a Pareto optimal solution of an MO-DCOP. However, L_p-norm based algorithms with $p \geq 2$, e.g., Euclidean and Chebyshev norm, cannot guarantee to find a Pareto optimal solution.

This algorithm has two phases. In Phase 1, we utilize any complete DCOP algorithm and find an optimal solution for each objective function, respectively. Specifically, for m objective functions of an MO-DCOP, we give the m weights, i.e., $(1, 0, \ldots, 0)$, $(0, 1, 0, \ldots, 0)$, $\ldots$, $(0, \ldots, 0, 1)$, and make the m weighted objective functions $o^1, \ldots, o^m$. Then, we find an optimal solution for each o^i ($1 \leq i \leq m$). That is, we utilize AOF technique [2, 5] and solve m DCOP problems independently. In this paper, we denote the m optimal values as $R^1_{max}, R^2_{max}, \ldots, R^m_{max}$ and call the reward vector $R^* = (R^1_{max}, R^2_{max}, \ldots, R^m_{max})$ as the *utopia point* [4]. In general, since trade-offs exist among objectives, there does not exist such an ideal point. In this paper, we use the DPOP algorithm [6] which is a representative pseudo-tree based inference algorithm that adapts the bucket elimination principle [1] to a distributed setting.

In Phase 2, we use the L_p-norm and the utopia point R^*, and find an assignment A so that the distance between a reward vector $R(A)$ and the utopia point is minimal. In this algorithm, we assume that each agent knows the utopia point. For a reward vector $R(A)$ and the utopia point R^*, we define the L_p-norm between $R(A)$ and R^* as follows:

$$dis(R^*, R(A)) = \sum_{l=1}^{m} (R^l_{max} - R^l(A)) \tag{3}$$

Theorem 1. *Manhattan norm based MO-DPOP$_{L_1}$ is the only MO-DPOP$_{L_p}$ algorithm that can guarantee to find a Pareto optimal solution.*

Proof Sketch. Every L_p-norm used here can be seen as an aggregation function f that associates an m-vector of values with a single value. The fact that MO-DPOP$_{L_1}$ guarantee to find a Pareto optimal solution comes from the following idea: the L_1-norm corresponds to the aggregation function $f = \Sigma$, which commutes with itself (given two aggregation functions f and g, f commutes with g if $f(g(x_{1,1}, \ldots, x_{1,q}), \ldots, g(x_{p,1}, \ldots, x_{p,q})) = g(f(x_{1,1}, \ldots, x_{p,1}), \ldots, g(x_{1,q}, \ldots, x_{p,q})))$, also used to aggregate the "local" rewards associated with each constraint (cf. Equation 1). Under such a condition, we can prove that MO-DPOP$_{L_1}$ always computes Pareto optimal solutions. When $2 \leq p \leq \infty$, this is not the case anymore that the underlying aggregation function f commutes with Σ. Considering the example given in Figure 1, it can be verified that the reward tables $UTIL^2_3$, $JOIN^1_2$ and $UTIL^1_2$ are same for every $2 \leq p \leq \infty$. Then, after $VALUE$ propagation, MO-DPOP$_{L_p}$ ($2 \leq p \leq \infty$) provides either the assignment $\{(x_1, a), (x_2, b), (x_3, a)\}$ or $\{(x_1, b), (x_2, a), (x_3, b)\}$, which are not Pareto optimal solutions. $\square$

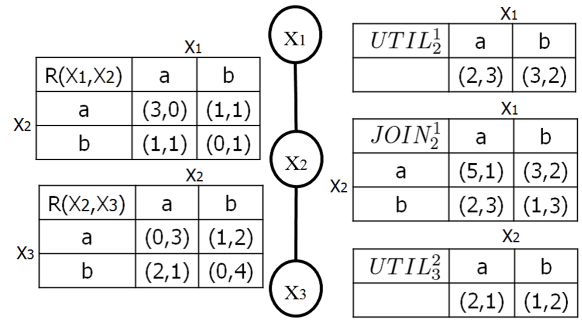

Figure 1: An example where MO-DPOP$_{L_p}$ cannot compute Pareto optimal solutions for $2 \leq p \leq \infty$.

In summary, we show that MO-DPOP $_{L_1}$ is the only MO-DPOP$_{L_p}$ algorithm that can guarantee to find a Pareto optimal solution. Intuitively, using MO-DPOP$_{L_1}$ (Phase 2), we can find a Pareto optimal solution which maximizes the average of the reward values of all objectives, and for Phase 1, we obtain extreme Pareto optimal solutions that optimizes one criterion, while it is really bad for another criterion.

4. CONCLUSION

In this paper, we developed a novel algorithm for MO-DCOPs which utilizes Dynamic Programming technique and the L_p-norm method to find a subset of Pareto front. Furthermore, we showed that MO-DPOP$_{L_1}$ is the only MO-DPOP$_{L_p}$ that can guarantee to find a Pareto optimal solution. Our plans for future work include developing an incomplete algorithm. We will also develop an extended MO-DPOP$_{L_1}$ which finds several Pareto optimal solutions by adding several reference points.

5. REFERENCES

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

[2] M. Ehrgott. *Multicriteria Optimization*. Springer, 2005.

[3] F. M. D. Fave, S. Stranders, A. Rogers, and N. R. Jennings. Bounded decentralised coordination over multiple objectives. In *AAMAS*, pages 371–378, 2011.

[4] T. Marler and J. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.

[5] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.

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