Multi-Objective Distributed Pseudo-Tree Optimization

Extended Abstract

Maxime Clement
National Institute of Informatics
Tokyo, Japan
maxime-clement@nii.ac.jp

Tenda Okimoto
Kobe University
Kobe, Japan
tenda@maritime.kobe-u.ac.jp

Katsumi Inoue
National Institute of Informatics
Tokyo, Japan
inoue@nii.ac.jp

ABSTRACT

In this paper, we develop a novel MO-DCOP algorithm based on dynamic programming techniques which guarantees to find the complete Pareto front. We also propose a bounded version which reduces the size of the messages using an adjustable parameter.

ACM Reference Format:
Maxime Clement, Tenda Okimoto, and Katsumi Inoue. 2018. Multi-Objective Distributed Pseudo-Tree Optimization. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

A *Distributed Constraint Optimization Problem* (DCOP) [2, 6, 8] is a fundamental problem that can formalize various applications related to multi-agent cooperation. Multi-Objective DCOP (MO-DCOP) [1, 5] was proposed as an extension of DCOP with multiple objectives. In this problem, since trade-offs exist among objectives, the goal is to find the *Pareto front* which is the set of cost vectors obtained by Pareto optimal solutions.

Compared to DCOPs, there exist few algorithms for solving MO-DCOPs. MO-ADOPT [5] generalizes the ADOPT algorithm [6] to the multi-objective case and is the state-of-the-art complete algorithm. The Bounded Multi-Objective Max-Sum (B-MOMS) [1] extends the Bounded Max-Sum algorithm [10] and is the state-of-the-art approximation algorithm.

In this paper, we develop a novel complete algorithm for MO-DCOPs called *Multi-Objective Distributed Pseudo-tree Optimization Procedure* (MO-DPOP). This algorithm extends DPOP, the representative dynamic programming algorithm for DCOPs. We also provide an incomplete version of our algorithm that uses a bounding function to reduce the size of the messages exchanged between the agents, reducing the memory complexity while still guaranteeing to find a subset of the Pareto front.

2 PRELIMINARIES

2.1 Multi-objective Distributed Constraint Optimization Problem

Definition 1 (Multi-Objective DCOP). A *Multi-Objective Distributed Constraint Optimization Problem* (MO-DCOP) [1, 5] with m objectives is defined as a tuple $MO\text{-}DCOP = (X, V, \mathcal{D}, \mathcal{F})$ where $X = \{x_1, \ldots, x_n\}$ is a set of agents, $V = \{v_1, \ldots, v_n\}$ is a set of variables, $\mathcal{D} = \{D_1, \ldots, D_n\}$ is a set of domains, and $\mathcal{F} = \{f_1, \ldots, f_c\}$ is a set of multi-objective cost functions. The scope of function f_k, denoted $var(f_k) \subseteq V$ is the set of arguments of f_k, indicating that the variables in $var(f_k)$ share a constraint relation. A multi-objective cost function $f_k \in \mathcal{F}$ is then defined as $f_k : \bigtimes_{\forall v_i \in var(f_k)} D_i \rightarrow \mathbb{R}^m$. These functions produce cost vectors of the form $(u_1, \ldots, u_m)$ where u_l is the cost of objective l.

The assignment of a variable $v_i \in V$ with the value $d_i \in D_i$ is denoted $v_i \leftarrow d_i$. An assignment $A = \{v_i \leftarrow d_i, \ldots, v_j \leftarrow d_j\}$ is a set of assignments to different variables and we denote the set of variables included in an assignment $var(A)$. A is said to be *partial* if $var(A) \subset V$ and *complete* if $var(A) = V$. The cost vector of a complete assignment is calculated by the objective function $\mathcal{F}(A) = \sum_{f_k \in \mathcal{F}} f_k(A)$, where the sum of vectors is the usual component-by-component sum. □

Optimal solutions of a MO-DCOP are characterized using the concept of *Pareto optimality*.

Definition 2 (Pareto Dominance). Given two vectors $\mathbf{u} = (u_1, \ldots, u_m)$ and $\mathbf{w} = (w_1, \ldots, w_m)$, we say that $\mathbf{u}$ dominates $\mathbf{w}$, denoted by $\mathbf{u} \prec \mathbf{w}$, iff it holds $u_l \leq w_l$ for all objectives l, and there exists at least one objective l', such that $u_{l'} < w_{l'}$. □

Definition 3 (Pareto optimal solution). For a MO-DCOP, a complete assignment A is a *Pareto optimal solution* iff there does not exist another complete assignment A', such that $\mathcal{F}(A') \prec \mathcal{F}(A)$. □

Solving a MO-DCOP consists in finding the set of cost vectors obtained by Pareto optimal solutions $\mathcal{PF}$ (called Pareto front), and for each vector $\mathbf{u} \in \mathcal{PF}$ at least one complete assignment A such that $\mathcal{F}(A) = \mathbf{u}$.

An MO-DCOP can be represented using a constraint graph which has a node for each variable and where an edge connects any two nodes whose variables appear in the scope of the same function. Considering this graph,

the corresponding *pseudo-tree* structure[12] is a rooted tree with the same nodes as the constraint graph (corresponding to agents) and with the property that nodes adjacent in the graph must belong to the same branch of the pseudo-tree. Such structure can be obtained using a depth-first traversal of the constraint graph.

After such structure is generated, each agent x_i is aware of its parent P_i, its set of children CH_i, and its set of pseudo-parents PP_i. An agent x_j is a pseudo-parent of x_i if and only if it is an ancestor of P_i in the pseudo-tree and a neighbor of x_i in the constraint graph.

An important concept of pseudo-trees for the algorithm presented in this paper is the *separator* of an agent.

Definition 4 (Separator). In a pseudo-tree, the separator Sep_i of a node x_i is the set of all ancestors of x_i which are pseudo-parents of either x_i or one of its descendants:

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

$$\mathrm{Sep}_i = \mathrm{Ancestors}_i \cap (\{P_i\} \cup \mathrm{PP}_i \cup (\bigcup_{x_j \in \mathrm{Descendants}_i} \mathrm{PP}_j)) \qquad \square$$

3 MO-DPOP

In this section, we present the Multi-Objective Distributed Pseudo-tree Optimization Algorithm (MO-DPOP). This algorithm is based on DPOP [8] and we only present here how we modify the UTIL and VALUE phases of DPOP.

3.1 *UTIL* propagation

Starting from the leaf agents. *UTIL* messages carrying the best cost vectors of each agent's subproblem are sent up the pseudo-tree.

Definition 5 (UTIL *message).* A message UTIL_i^j sent from agent x_i to agent x_j is a multi-dimensional matrix with one dimension for each variable in Sep_i and we denote $var(\mathrm{UTIL}_i^j)$ the set of variables considered by the message.

For an assignment A, $var(A) \subseteq var(\mathrm{UTIL}_i^j)$, $\mathrm{UTIL}_i^j[A]$ is a matrix of dimension $var(\mathrm{UTIL}_i^j) \setminus var(A)$ such that:

$$\mathrm{UTIL}_i^j[A] = \bigcup_{\forall A', var(A')=var(\mathrm{UTIL}_i^j)\setminus var(A)} \mathrm{UTIL}_i^j[A \cup A'] \qquad \square$$

A message $UTIL_i^j$ expresses the best cost vectors that can be obtained by the sub-tree rooted at x_i based on the values taken by the variables in the separator Sep_i. *UTIL* messages are built from cost functions and we assume that a function $f_k : \bigtimes_{\forall v_i \in var(f_k)} D_i \to \mathbb{R}^m$ is represented as a matrix of dimension $var(f_k)$. When the root agent x_r computes the message $UTIL_r^{null}$, it contains the best cost vectors that can be obtained by the whole tree, corresponding to the Pareto front of the problem.

To compute the *UTIL* message it will send to its parent, an agent x_i has to *join* all messages received from its children as well as the cost functions it shares with its parent and pseudo-parents:$\{f_k \in \mathcal{F} | var(f_k) \subseteq \{P_i\} \cup PP_i \cup \{v_i\}, v_i \in var(f_k)\}$.

Definition 6 (Join Operator in MO-DPOP). Joining two matrices M and M', written $M \oplus M'$, produces a new matrix M'' such that $var(M'') = var(M) \cup var(M')$ and:

$\forall A, var(A) = var(M'')$,

$M''[A] = \{\mathbf{u} + \mathbf{u}' | \mathbf{u} \in M[A], \mathbf{u}' \in M'[A]\}$ $\square$

Before sending an *UTIL* message, an agent x_i projects variable v_i out of the matrix, reducing its dimension by one and merging the content of some cells, which are filtered using Pareto dominance. For simplicity, we consider a function which takes a set of vectors $\mathcal{U}$ and returns the corresponding set of non-dominated vectors: $ND(\mathcal{U}) = \{\mathbf{u} \in \mathcal{U} | \nexists \mathbf{u}' \in \mathcal{U}\ s.t.\ \mathbf{u}' \prec \mathbf{u}\}$.

Definition 7 (Projection Operator in MO-DPOP). Projecting variable v_i out of matrix M, written $M\bot_{v_i}$ and requiring $v_i \in var(M)$, is the projection of the matrix M along the v_i dimension such that

$\forall A, var(A) = var(M) \setminus \{v_i\}, M\bot_{v_i}[A] = \mathrm{ND}(M[A])$ $\square$

The root agent x_r receives from each of its children an *UTIL* message of dimension $\{v_r\}$ which, when joined, provide the Pareto front of the problem.

3.2 Limiting the Size of Messages

UTIL messages being of exponential space complexity, we propose a technique to limit their size using a bounding function B_b which takes a set of vectors $\mathcal{U}$ and returns a set $\mathcal{W} \subseteq \mathcal{U}, s.t. |\mathcal{W}| \leq b$. This guarantees an upper bound for the size of the messages and, depending on the function used, can still guarantee to find Pareto optimal solution with our algorithm.

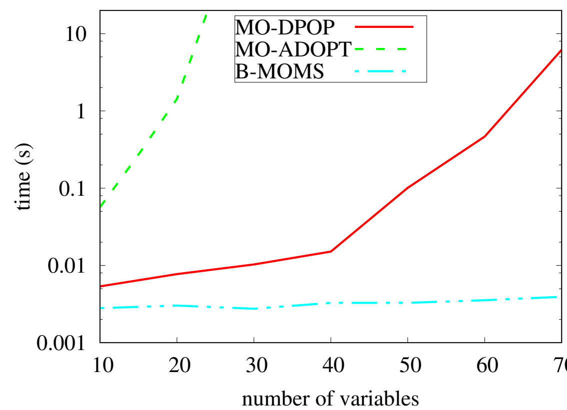

Figure 1: Runtime varying the number of variables

Property 1 (Maximum Bounded Message Size). *If the subset $\mathcal{U}$ yielded by B is bounded in size ($|\mathcal{U}| \leq b$), the maximum message size becomes bounded by the maximum separator size $|Sep_{max}|$ with a space complexity in $O(bm \times |D_{max}|^{|Sep_{max}|})$.*

For example, bounding functions based on weighted-sums [4] or lexicographic orderings [3] guarantee to still find some Pareto optimal solutions when bouding the messages of MO-DPOP.

4 EXPERIMENTS

To evaluate MO-DPOP and its extension with bounded messages, we conducted experiments on the extended graph-coloring problem [1]. We compared MO-DPOP with the existing algorithms MO-ADOPT [5], and B-MOMS [1].

Algorithms were implemented in Java and experiments were carried on a 4.2 GHz 8 cores CPU, measuring the average simulated runtime [13] over 40 random instances.

Figure 1 shows the simulated runtime when varying the number of variables with problems of low density (0.01). We observe that our algorithm provides a significant improvement over the previous complete algorithm, allowing us to solve instances of up to 70 variables within 10s whereas MO-ADOPT cannot solve problems of 25 variables within that time.

5 CONCLUSION

In this paper, we developed a new complete algorithm for MO-DCOP and provided a technique to reduce the size of its messages. In our experiments, we showed that our complete algorithm outperforms the state-of-the-art complete MO-DCOP algorithm.

In future works, we will study additional ways to reduce the complexity of our algorithm by considering techniques such as the Mini-Bucket Elimination [11] or the Memory-Bounded DPOP [9] and p-reduced graph technique [7].

REFERENCES

[1] Francesco M. Delle Fave, Ruben Stranders, Alex Rogers, and Nicholas R. Jennings. 2011. Bounded Decentralised Coordination over Multiple Objectives. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*. 371–378.

[2] F. Fioretto, E. Pontelli, and W. Yeoh. 2016. Distributed Constraint Optimization Problems and Applications: A Survey. *ArXiv e-prints* (Feb. 2016). arXiv:cs.AI/1602.06347

[3] Peter C Fishburn. 1974. Lexicographic orders, utilities and decision rules: A survey. *Management science* 20, 11 (1974), 1442–1471.

[4] R Timothy Marler and Jasbir S Arora. 2010. The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization* 41, 6 (2010), 853–862.

[5] Toshihiro Matsui, Marius Silaghi, Katsutoshi Hirayama, Makoto Yokoo, and Hiroshi Matsuo. 2012. Distributed Search Method with Bounded Cost Vectors on Multiple Objective DCOPs. In *Proceedings of the 15th International Conference on Principles and Practice of Multi-Agent Systems*. 137–152.

[6] Pragnesh Modi, WeiMin Shen, Milind Tambe, and Makoto Yokoo. 2005. ADOPT: asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence* 161, 1-2 (2005), 149–180.

[7] Tenda Okimoto, Yongjoon Joe, Atsushi Iwasaki, and Makoto Yokoo. 2011. Pseudo-tree-based algorithm for approximate distributed constraint optimization with quality bounds. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*. 1269–1270.

[8] Adrian Petcu and Boi Faltings. 2005. A Scalable Method for Multiagent Constraint Optimization. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. 266–271.

[9] Adrian Petcu and Boi Faltings. 2007. MB-DPOP: A New Memory-Bounded Algorithm for Distributed Optimization. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*. 1452–1457.

[10] Alex Rogers, Alessandro Farinelli, Ruben Stranders, and Nicholas Jennings. 2011. Bounded approximate decentralised coordination via the max-sum algorithm. *Artificial Intelligence* 175, 2 (2011), 730–759.

[11] Emma Rollon and Javier Larrosa. 2006. Bucket elimination for multiobjective optimization problems. *Journal of Heuristics* 12, 4-5 (2006), 307–328.

[12] Thomas Schiex, Hélène Fargier, and Gérard Verfaillie. 1995. Valued Constraint Satisfaction Problems: Hard and Easy Problems. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. 631–639.

[13] Evan A. Sultanik, Robert N. Lass, and William C. Regli. 2007. DCOPolis: A Framework for Simulating and Deploying Distributed Constraint Optimization Algorithms. In *Proceedings of the Ninth International Workshop on DCR*.