DeQED: an Efficient Divide-and-Coordinate Algorithm for DCOP

(Extended Abstract)

Daisuke Hatano, Katsutoshi Hirayama
Kobe University
5-1-1 Fukaeminami-machi, Higashi-Nada-ku, Kobe, Japan 658-0022
daisuke-hatano@stu.kobe-u.ac.jp, hirayama@maritime.kobe-u.ac.jp

ABSTRACT

This paper presents a new DCOP algorithm called DeQED (Decomposition with Quadratic Encoding to Decentralize). DeQED is based on the Divide-and-Coordinate (DaC) framework, where the agents repeat solving their updated local sub-problems (the divide stage) and exchanging coordination information that causes to update their local sub-problems (the coordinate stage). Unlike other DaC-based DCOP algorithms, DeQED does not essentially increase the complexity of local sub-problems and allows agents to avoid exchanging variable values in the coordinate stage. Our experimental results show that DeQED significantly outperformed other incomplete DCOP algorithms for both random and structured instances.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Coherence and coordination

General Terms

Algorithms

Keywords

Distributed Constraint Optimization Problem, Lagrangian Decomposition, Divide-and-Coordinate framework

1. INTRODUCTION

In many applications of distributed problem solving, the agents may want to optimize a global objective while preserving their privacy and security. This problem can be formalized as the Distributed Constraint Optimization Problem (DCOP). For solving DCOP, several complete algorithms have been presented, but one recent trend may be incomplete algorithms [1, 2, 3, 4], due to the needs for finding a high-quality solution quickly for large-scale problem instances.

This paper presents a new DCOP algorithm called DeQED (Decomposition with Quadratic Encoding to Decentralize). DeQED is based on the Divide-and-Coordinate (DaC) framework, where the agents repeat solving their updated local sub-problems (the divide stage) and exchanging coordination information that causes to update their local sub-problems (the coordinate stage). Unlike other DaC-based DCOP algorithms [3, 4], DeQED does not essentially increase the complexity of local sub-problems and allows agents to avoid exchanging variable values in the coordinate stage.

Through comparison with MaxSum [1], DALO [2], and EU-DaC [4], we demonstrate that DeQED works very well both in terms of solution quality and efficiency.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013),* Ito, Jonker, Gini, and Shehory (eds.), May 6–10, 2013, Saint Paul, Minnesota, USA.

2. DCOP

COP is defined by a set X of *variables*, where each variable x_i has a finite *domain* D_i from which it takes its value, and a set F of *binary cost functions*, where each function $f_{i,j} : D_i \times D_j \rightarrow \Re^+$ returns a non-negative cost value for each binary relation between variable x_i's domain and variable x_j's domain.

DCOP is the COP where variables are controlled by a set A of agents. Each variable belongs to some agent who controls it. We denote the fact that variable x_i belongs to agent a by $\mathrm{belong}(x_i) = a$. The goal of COP and DCOP is to find a value assignment to X that minimizes a total sum of the values of cost functions.

3. DEQED

As with DaCSA [3], DeQED exploits the Lagrangian decomposition technique, but the difference between DeQED and DaCSA is the way of encoding of the entire problem. In DeQED, we use *quadratic encoding*, in which an cost function is encoded into the quadratic programming problem.

3.1 Quadratic encoding

Let us assume that every variable has the same domain, say D, without loss of generality. For cost function $f_{i,j} \in F$ between variable x_i and variable x_j, we introduce $|D| \times |D|$ cost matrix $\boldsymbol{F}_{i,j}$ whose elements represent the values of the cost function. Furthermore, for variables x_i and x_j, we introduce new variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ whose domains are the whole set of $|D|$-dimensional unit column vectors, respectively. Namely, we have $\boldsymbol{x}_i \in \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{|D|}\}$ and $\boldsymbol{x}_j \in \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{|D|}\}$, where $\boldsymbol{e}_1$ is $(1, 0, 0, \ldots, 0)^{\mathrm{T}}$, $\boldsymbol{e}_2$ is $(0, 1, 0, \ldots, 0)^{\mathrm{T}}$, and so on. Superscript T means the transpose of a vector. The value of cost function $f_{i,j}$ can be computed by $(\boldsymbol{x}_i)^{\mathrm{T}} \cdot \boldsymbol{F}_{i,j} \cdot \boldsymbol{x}_j$. We also introduce two auxiliary variables $\boldsymbol{\alpha}_i^{i,j}$ and $\boldsymbol{\alpha}_j^{i,j}$ for each cost function $f_{i,j}$. These auxiliary variables are supposed to be the copies of variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in terms of $f_{i,j}$, respectively.

Given this representation, DCOP can be formulated as

$$
\begin{aligned}
\mathcal{DCOP} : \min \quad & \sum_{f_{i,j} \in F} (\boldsymbol{\alpha}_i^{i,j})^{\mathrm{T}} \cdot \boldsymbol{F}_{i,j} \cdot \boldsymbol{\alpha}_j^{i,j} \\
\text{s. t.} \quad & \boldsymbol{x}_i = \boldsymbol{\alpha}_i^{i,j},\ \boldsymbol{x}_j = \boldsymbol{\alpha}_j^{i,j}, \quad \forall f_{i,j} \in F, \\
& \boldsymbol{x}_i, \in \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_{|D|}\}, \quad \forall x_i \in X, \\
& \boldsymbol{\alpha}_i^{i,j}, \boldsymbol{\alpha}_j^{i,j} \in \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_{|D|}\},\ \forall f_{i,j} \in F,
\end{aligned} \tag{1}
$$

where $\boldsymbol{\alpha}$ as well as $\boldsymbol{x}$ are decision variables. Due to space limitations, we omit the last two lines of the above formulation because they just describe the domain of these decision variables.

3.2 Lagrangian Decomposition

We decompose this problem into the sub-problems over the agents. First, we relax a set of copy constraints (1) to produce the *Lagrangian relaxation problem*. Then, we decompose its objective function into the terms on the individual agents and the terms on auxiliary variables. As a result, we get the following function:

$$L(\boldsymbol{\mu}) = \sum_{a \in A} L^a(\boldsymbol{\mu}) + \sum_{f_{i,j} \in F} L^{aux}_{i,j}(\boldsymbol{\mu}), \qquad (2)$$

that computes a lower bound on the optimum of DCOP, where $\boldsymbol{\mu}$ is $|D|$-dimensional real-valued column vector,

$$L^a(\boldsymbol{\mu}) \equiv \min \left\{ \sum_{(x_i,x_j) \in P^a} (\boldsymbol{\mu}^{i,j}_i)^{\mathrm{T}} \boldsymbol{x}_i + \sum_{(x_i,x_j) \in N^a} (\boldsymbol{\mu}^{i,j}_j)^{\mathrm{T}} \boldsymbol{x}_j \right\}$$

for each agent a, where $P^a \equiv \{(x_i, x_j) \mid f_{i,j} \in F, \mathrm{belong}(x_i) = a\}$ and $N^a \equiv \{(x_i, x_j) \mid f_{i,j} \in F, \mathrm{belong}(x_j) = a\}$, and

$$L^{aux}_{i,j}(\boldsymbol{\mu}) \equiv \min \left\{ (\boldsymbol{\alpha}^{i,j}_i)^T \cdot \boldsymbol{F}_{i,j} \cdot \boldsymbol{\alpha}^{i,j}_j - (\boldsymbol{\mu}^{i,j}_i)^{\mathrm{T}} \boldsymbol{\alpha}^{i,j}_i - (\boldsymbol{\mu}^{i,j}_j)^{\mathrm{T}} \boldsymbol{\alpha}^{i,j}_j \right\}$$

for each cost function $f_{i,j}$.

DeQED solves the Lagrangian dual problem, whose goal is to maximize $L(\boldsymbol{\mu})$, a lower bound on the optimum of DCOP, by controlling values of $\boldsymbol{\mu}$

3.3 Problem Distribution

We need to clarify which agent should compute which part of (2). Regarding the *primal phase*, where we solve the minimization problem over $\boldsymbol{x}$ and $\boldsymbol{\alpha}$ with specific values on $\boldsymbol{\mu}$, we propose that

- $L^a(\boldsymbol{\mu})$ should be computed by agent a since it includes only agent a's variables;
- $L^{aux}_{i,j}(\boldsymbol{\mu})$ should be computed by either of the agents who control variables $\boldsymbol{x}_i$ or $\boldsymbol{x}_j$ since it represents cost function $f_{i,j}$ between these agents.

On the other hand, regarding the *dual phase*, where we solve the maximization problem over $\boldsymbol{\mu}$ with specific values on $\boldsymbol{x}$ and $\boldsymbol{\alpha}$, we propose that

- Since Lagrange multiplier vectors $\boldsymbol{\mu}^{i,j}_i$ and $\boldsymbol{\mu}^{i,j}_j$ are related to cost function $f_{i,j}$, both vectors should be controlled by the agents having variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively.

3.4 Minimal Procedure

Below is the minimal procedure of DeQED, where the agents try to find values for $\boldsymbol{\mu}$ that maximize $L(\boldsymbol{\mu})$.

Step 1: The agents initialize their $\boldsymbol{\mu}$ as $(0, \ldots, 0)^{\mathrm{T}}$.

Step 2: Every agent a sends, for each cost function $f_{i,j}$ with belong $(i) = a$, the value of $\boldsymbol{\mu}^{i,j}_i$ to the agent which x_j belongs to. Similarly, it sends, for each cost function $f_{i,j}$ with $\mathrm{belong}(j) = a$, the value of $\boldsymbol{\mu}^{i,j}_j$ to the agent which x_i belongs to.

Step 3: After receiving all of the latest values for $\boldsymbol{\mu}$, every agent a solves $L^a(\boldsymbol{\mu})$ by an exact WCSP solver and $L^{aux}_{i,j}(\boldsymbol{\mu})$ by evaluating all possible pairs of the values for x_i and x_j.

Step 4: If *CanTerminate?* then the agents stop; otherwise they update $\boldsymbol{\mu}$ and go back to Step 2.

We refer to the minimal version of DeQED as DeQED$_m$. On the other hand, the agents in DeQED can exploit the best lower and upper bounds that must be collected through global communication. We refer to this extended version of DeQED as DeQED$_a$.

It is noteworthy that the agents in DeQED$_m$ only exchange $\boldsymbol{\mu}$. Namely, they do not have to exchange DCOP's variables. Considering that one major motivation of DCOP is privacy and security, this property of DeQED$_m$ should be important.

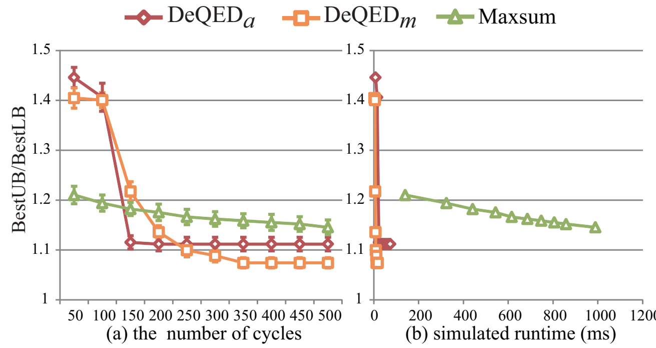

Figure 1: Average quality upper bounds for random networks.

4. EXPERIMENTS

We compared DeQED with DALO [2], EU-DaC [4], and MaxSum [1] on binary constraint networks with *random* topology. We did not adopt DaCSA since it was outperformed by EU-DaC [4].

We created 20 DCOP instances with random network, where the domain size of all variables (nodes) is three and the cost value of binary cost functions (edges) is randomly selected from $\{1, 2, ..., 10^5\}$.

Since all of the algorithms are incomplete, our interest is on how quickly each of these algorithms finds a better solution. Therefore, in our experiments, we observed an average quality upper bound for each algorithm when cutting off a run at a certain cycle bound, which ranges from 50 to 500 cycles in step of 50. Furthermore, since these algorithms clearly have different computational costs in one cycle, we also observed an average quality upper bound against simulated runtime at the above cut-off cycles.

The results are shown in Figure 1, where the left part denoted by (a) shows the average quality upper bound against the number of cycles and the right part denoted by (b) shows the average quality upper bound against simulated runtime. In this figure, we plot the average quality upper bounds with error bars only for MaxSum, DeQED$_m$ and DeQED$_a$ for readability, because the best quality upper bounds of the other algorithms were more than 1.4 even in the best case. Moreover, EU-DaC spent more than simulated runtime of 10,000ms to finish 500 cycles.

Figure 1 shows that DeQED clearly outperformed MaxSum for these instances. We should emphasize that, DeQED converged quite efficiently in simulated runtime. One reason for this efficiency is that the computational cost of each agent in DeQED increases only linearly with the number of its neighbors, while that in MaxSum increases exponentially [1].

5. REFERENCES

[1] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. AAMAS '08, pages 639–646, 2008.

[2] C. Kiekintveld, Z. Yin, A. Kumar, and M. Tambe. Asynchronous algorithms for approximate distributed constraint optimization with quality bounds. AAMAS '10, pages 133–140, 2010.

[3] M. Vinyals, M. Pujol, J. A. Rodriguez-Aguilar, and J. Cerquides. Divide-and-coordinate: DCOPs by agreement. AAMAS '10, pages 149–156, 2010.

[4] M. Vinyals, J. A. Rodriguez-Aguilar, and J. Cerquides. Divide-and-coordinate by egalitarian utilities: Turning DCOPs into egalitarian worlds. OPTMAS '10, 2010.