Embedding Preference Elicitation Within the Search for DCOP Solutions[*]

Extended Abstract

Yuanming Xiao
Washington University in St. Louis
yuanming.xiao@wustl.edu

Atena M. Tabakhi
Washington University in St. Louis
amtabakhi@wustl.edu

William Yeoh
Washington University in St. Louis
wyeoh@wustl.edu

ABSTRACT

A key assumption in *Distributed Constraint Optimization Problem* (DCOP) model is that all constraints are fully specified or known a priori, which may not hold in applications where constraints encode preferences of human users. We extend the model to *Incomplete DCOPs* (I-DCOPs), where some constraints can be partially specified. User preferences for these partially-specified constraints can be elicited during the execution of I-DCOP algorithms, but they incur some elicitation costs. Additionally, we extend the *Synchronous Branch-and-Bound* (SyncBB) algorithm to solve I-DCOPs. Our model extends the state of the art in distributed constraint reasoning to better model and solve distributed agent-based applications with user preferences.

KEYWORDS

Distributed Constraint Optimization; DCOPs; Preference Elicitation

ACM Reference Format:
Yuanming Xiao, Atena M. Tabakhi, and William Yeoh. 2020. Embedding Preference Elicitation Within the Search for DCOP Solutions. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

The field of *Distributed Constraint Optimization Problem* (DCOP) has matured significantly over the past decade since its inception [3, 8, 11]. DCOP researchers have proposed a wide variety of solution approaches, from complete approaches such as distributed search [6, 8, 16] and inference [11] algorithms to incomplete approaches such as distributed local search [17], inference [2, 15], and sampling [9, 10] algorithms. One of the core limitations of all these approaches is that they assume that the constraint costs in a DCOP are specified or known a priori. In some applications, such as meeting scheduling problems [7], constraints encode the preferences of human users. As such, some of the constraint costs may be unspecified and must be elicited from human users.

To address this limitation, researchers have proposed the *preference elicitation problem for DCOPs* [13]. However, this approach suffers from two limitations: First, the authors assume that the cost of eliciting constraints is uniform across all constraints. This is unrealistic as providing the preferences for some constraints may require more cognitive effort than the preferences for other constraints. Second, it decouples the elicitation process from the DCOP solving process since the elicitation process must be completed before one solves the DCOP with elicited constraints. As both the elicitation and solving process are actually coupled, this two-phase decoupled approach prohibits the elicitation process from relying on the solving process.

Therefore, we propose the *Incomplete DCOP* (I-DCOP) model, which *integrates* both the elicitation and solving problems into a single integrated optimization problem. In an I-DCOP, some constraint costs are unknown and can be elicited. Elicitation of unknown costs will incur elicitation costs, and the goal is to find a solution that minimizes the sum of constraint and elicitation costs incurred. To solve this problem, we extend *Synchronous Branch-and-Bounds* (SyncBB) [6], a simple complete DCOP search algorithm.

2 INCOMPLETE DCOPS

An *Incomplete DCOP* (I-DCOP) extends a DCOP by allowing some constraints to be partially specified. It is defined by a tuple $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{F}, \tilde{\mathcal{F}}, \mathcal{E}, \alpha \rangle$:

- $\mathcal{A} = \{a_i\}_{i=1}^p$ is a set of *agents.*
- $\mathcal{X} = \{x_i\}_{i=1}^n$ is a set of decision *variables.*
- $\mathcal{D} = \{D_x\}_{x \in \mathcal{X}}$ is a set of finite *domains* and each variable $x \in \mathcal{X}$ takes values from the set D_x.
- $\mathcal{F} = \{f_i\}_{i=1}^m$ is a set of *constraints*, each defined over a set of decision variables: $f_i : \prod_{x \in \mathbf{x}^{f_i}} D_x \to \mathbb{R} \cup \{\infty\}$, where infeasible configurations have ∞ costs and $\mathbf{x}^{f_i} \subseteq \mathcal{X}$ is the *scope* of f_i. Unlike standard DCOPs, the set of constraints $\mathcal{F}$ are not known to an I-DCOP algorithm. Instead, only the set of partially-specified constraints $\tilde{\mathcal{F}} = \{\tilde{f}_i\}_{i=1}^m$ are known. Each partially-specified constraint is a function $\tilde{f}_i : \prod_{x \in \mathbf{x}^{f_i}} D_x \to \mathbb{R} \cup \{\infty, ?\}$, where ? is a special element denoting that the cost for a given combination of value assignment is not specified. The costs $\mathbb{R} \cup \{\infty\}$ that are specified are exactly the costs of the corresponding constraints $f_i \in \mathcal{F}$.
- $\mathcal{E} = \{e_i\}_{i=1}^m$ is the set of elicitation costs, where each elicitation cost $e_i : \prod_{x \in \mathbf{x}^{f_i}} D_x \to \mathbb{R}$ specifies the cost of eliciting the constraint cost of a particular ? in $\tilde{f}_i$.
- $\alpha : \mathcal{X} \to \mathcal{A}$ is a *mapping function* that associates each decision variable to one agent.

An *explored solution space* $\tilde{\mathbf{x}}$ is the union of all solutions explored so far by a particular algorithm. The *cumulative elicitation cost* $\mathcal{E}(\tilde{\mathbf{x}}) = \sum_{e \in \mathcal{E}} e(\tilde{\mathbf{x}})$ is the sum of the costs of all elicitations conducted while exploring $\tilde{\mathbf{x}}$. The *total cost* $\mathcal{F}(\mathbf{x}, \tilde{\mathbf{x}}) = \alpha_f \cdot \mathcal{F}(\mathbf{x}) + \alpha_e \cdot \mathcal{E}(\tilde{\mathbf{x}})$ is the weighted sum of both the cumulative constraint cost $\mathcal{F}(\mathbf{x})$ of solution $\mathbf{x}$ and the cumulative elicitation

[*]The research was partially supported by NSF grants 1812619 and 1838364.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

$\|\mathcal{A}\|$	# of unknown costs	Without Elicitation Costs				With Elicitation Costs					
		# of unknown costs elicited	runtime	constraint cost	# of nodes expanded	# of unknown costs elicited	runtime	total cost	constraint cost	elicitation cost	#of nodes expanded
10	43	40.62	5.09E-01	51.86	1.65E+03	17.98	3.27E-02	237.90	59.64	178.26	5.92E+01
12	62	59.04	1.99E+00	76.30	6.76E+03	25.60	4.38E-02	349.54	87.96	261.58	1.14E+02
14	86	82.44	8.02E+00	107.14	2.36E+04	35.30	5.95E-02	484.64	122.22	362.42	1.17E+02
16	115	111.74	3.22E+01	145.32	9.35E+04	47.88	8.04E-02	636.10	163.20	472.90	1.58E+02
18	146	140.84	1.18E+02	185.06	3.48E+05	60.52	1.29E-01	803.50	205.50	598.00	2.67E+02
20	182	177.08	1.23E+03	231.64	1.36E+06	70.64	1.63E-01	978.00	258.88	725.12	2.79E+02

Table 1: Preliminary Experimental Results

cost $\mathcal{E}(\tilde{\mathbf{x}})$ of the explored solution space $\tilde{\mathbf{x}}$, where $\alpha_f \in (0, 1]$ and $\alpha_e \in [0, 1]$ such that $\alpha_f + \alpha_e = 1$. The weights represent the tradeoffs between the importance of solution quality and the cumulative elicitation cost. The goal is to find an optimal complete solution $\mathbf{x}^*$ while eliciting only a cost-minimal set of preferences from a solution space $\tilde{\mathbf{x}}^*$. More formally, the goal is to find $(\mathbf{x}^*, \tilde{\mathbf{x}}^*) = \operatorname{argmin}_{(\mathbf{x}, \tilde{\mathbf{x}})} \mathcal{F}(\mathbf{x}, \tilde{\mathbf{x}})$.

3 USING SYNCBB TO SOLVE I-DCOPS

Synchronous Branch-and-Bound (SyncBB) [6] is a complete, synchronous, search-based algorithm that can be considered as a distributed version of a depth-first branch-and-bound algorithm. It uses a complete ordering of the agents to extend a *Current Partial Assignment* (CPA) via a synchronous communication process. The CPA holds the assignments of all the variables controlled by all the visited agents, and, in addition, functions as a mechanism to propagate bound information. The algorithm prunes those parts of the search space whose solution quality is sub-optimal by exploiting the bounds that are updated at each step of the algorithm. In other words, an agent backtracks when the cost of its CPA is no smaller than the cost of the best complete solution found so far. The algorithm terminates when the root backtracks (i.e., the algorithm has explored or pruned the entire search space).

To solve I-DCOPs, we extend SyncBB in the following way: The SyncBB algorithm operates on a search tree, constructed based on the complete ordering of the agents/variables. When SyncBB evaluates a node n after exploring search space $\tilde{\mathbf{x}}$, it considers the cumulative elicitation cost so far $\mathcal{E}(\tilde{\mathbf{x}})$ and the constraint costs of the CPA at node n, which we will refer to as g-values, denoted by $g(n)$. (We use A* notations [5] here.) We refer to the weighted sum of these values as f-values, denoted by $f(n, \tilde{\mathbf{x}}) = \alpha_f \cdot g(n) + \alpha_e \cdot \mathcal{E}(\tilde{\mathbf{x}})$. The algorithm expands the node with the smallest f-value. In other words, it chooses the value from its domain that has the smallest constraint cost. If the constraint cost is unknown, the algorithm replaces the unspecified cost with a lower bound $\mathcal{L}$ on all the constraint costs, and calculates its f-value. If the f-value of the node is smaller than the cost of the best complete solution so far, the algorithm elicits the unknown constraint costs and accumulates the elicitation costs. Otherwise, it prunes the node, and explores the remaining part of the search space.

4 RELATED WORK

Aside from the work proposed by Tabakhi *et al.* [13] discussed in Section 1, the body of work that is most related to ours is the work on *Incomplete Weighted Constraint Satisfaction Problems* (IWCSPs) [12, 14], which can be seen as centralized versions of I-DCOPs. Aside from IWCSPs, similar centralized constraint-based models include *Incomplete Fuzzy CSPs* and *Incomplete Soft CSPs* [4]. Researchers have proposed a family of algorithms based on depth-first branch-and-bound to solve these centralized models.

5 PRELIMINARY EMPIRICAL EVALUATIONS

We evaluate our SyncBB algorithm on random graphs, where we measure various costs of the solutions found – the cumulative constraint costs, cumulative elicitation costs, and their aggregated total costs – the number of unknown costs elicited, the number of nodes expanded after the algorithm terminates, and the runtime (i.e., wall clock time) of the algorithm (in seconds). In all experiments we set $\alpha_f = \alpha_e = 0.5$.

We generate 50 random (binary) graphs [1], where we vary the number of agents/variables $|\mathcal{A}|$ from 10 to 20, set the domain size $|\{D_x\}| = 2$ for all $x \in \mathcal{X}$, the constraint density $p_1 = 0.4$, the tightness $p_2 = 0$, and the fraction of unknown costs in the problem to 0.6. All constraint costs are randomly sampled from $[2, 5]$ and all elicitation costs are randomly sampled from $[0, 20]$. Table 1 tabulates our preliminary empirical results demonstrating the feasibility of this approach. As expected, the runtimes and number of unknown costs elicited by our algorithm increase with increasing number of agents $|\mathcal{A}|$. The reason is that the size of the problem, in terms of the number of constraints in the problem, increases with increasing $|\mathcal{A}|$. And our algorithm needs to elicit more unknown costs and evaluate the costs of more constraints before terminating.

6 CONCLUSIONS AND FUTUREWORK

Distributed Constraint Optimization Problems (DCOPs) have been used to model a variety of cooperative multi-agent problems. However, they assume that all constraints are fully specified, which may not hold in applications where constraints encode preferences of human users. To overcome this limitation, we propose *Incomplete DCOPs* (I-DCOPs), which extends DCOPs by allowing some constraints to be partially specified and the elicitation of unknown costs in such constraints incur elicitation costs. We use the SyncBB search algorithm as the underlying solver for I-DCOPs. In the future, we will investigate the use of various heuristics in conjunction with complete search algorithms to solve I-DCOPs.

REFERENCES

[1] Paul Erdös and Alfréd Rényi. 1959. On random graphs, I. *Publicationes Mathematicae (Debrecen)* 6 (1959), 290–297.

[2] Alessandro Farinelli, Alex Rogers, Adrian Petcu, and Nicholas Jennings. 2008. Decentralised Coordination of Low-Power Embedded Devices Using the Max-Sum Algorithm. In *AAMAS*. 639–646.

[3] Ferdinando Fioretto, Enrico Pontelli, and William Yeoh. 2018. Distributed Constraint Optimization Problems and Applications: A Survey. *Journal of Artificial Intelligence Research* 61 (2018), 623–698.

[4] Mirco Gelain, Maria Silvia Pini, Francesca Rossi, K Brent Venable, and Toby Walsh. 2010. Elicitation strategies for soft constraint problems with missing preferences: Properties, algorithms and experimental studies. *Artificial Intelligence* 174, 3-4 (2010), 270–294.

[5] Paul Hart, Nils Nilsson, and Bertram Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* SSC4, 2 (1968), 100–107.

[6] Katsutoshi Hirayama and Makoto Yokoo. 1997. Distributed partial constraint satisfaction problem. In *CP*. 222–236.

[7] Rajiv Maheswaran, Milind Tambe, Emma Bowring, Jonathan Pearce, and Pradeep Varakantham. 2004. Taking DCOP to the Real World: Efficient Complete Solutions for Distributed Event Scheduling. In *AAMAS*. 310–317.

[8] Pragnesh Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo. 2005. ADOPT: Asynchronous Distributed Constraint Optimization with Quality Guarantees. *Artificial Intelligence* 161, 1–2 (2005), 149–180.

[9] Duc Thien Nguyen, William Yeoh, Hoong Chuin Lau, and Roie Zivan. 2019. Distributed Gibbs: A Linear-Space Sampling-Based DCOP Algorithm. *Journal of Artificial Intelligence Research* 64 (2019), 705–748.

[10] Brammert Ottens, Christos Dimitrakakis, and Boi Faltings. 2017. DUCT: An Upper Confidence Bound Approach to Distributed Constraint Optimization Problems. *ACM Transactions on Intelligent Systems and Technology* 8, 5 (2017), 69:1–69:27.

[11] Adrian Petcu and Boi Faltings. 2005. A Scalable Method for Multiagent Constraint Optimization. In *IJCAI*. 1413–1420.

[12] Atena M Tabakhi. 2019. Parameterized Heuristics for Incomplete Weighted CSPs. In *AAAI*, Vol. 33. 9898–9899.

[13] Atena M. Tabakhi, Tiep Le, Ferdinando Fioretto, and William Yeoh. 2017. Preference Elicitation for DCOPs. In *CP*. 278–296.

[14] Atena M. Tabakhi, William Yeoh, and Makoto Yokoo. 2019. Parameterized Heuristics for Incomplete Weighted CSPs with Elicitation Costs. In *AAMAS*. 476–484.

[15] Meritxell Vinyals, Juan Rodríguez-Aguilar, and Jesús Cerquides. 2011. Constructing a Unifying Theory of Dynamic Programming DCOP Algorithms via the Generalized Distributive Law. *Journal of Autonomous Agents and Multi-Agent Systems* 22, 3 (2011), 439–464.

[16] William Yeoh, Ariel Felner, and Sven Koenig. 2010. BnB-ADOPT: An Asynchronous Branch-and-Bound DCOP Algorithm. *Journal of Artificial Intelligence Research* 38 (2010), 85–133.

[17] Weixiong Zhang, Guandong Wang, Zhao Xing, and Lars Wittenberg. 2005. Distributed Stochastic Search and Distributed Breakout: Properties, Comparison and Applications to Constraint Optimization Problems in Sensor Networks. *Artificial Intelligence* 161, 1–2 (2005), 55–87.