Beyond Uninformed Search: Improving Branch-and-bound Based Acceleration Algorithms for Belief Propagation via Heuristic Strategies

Extended Abstract

Junsong Gao
College of Computer Science, Chongqing University
Chongqing, China
jsgao0126@cqu.edu.cn

Ziyu Chen*
College of Computer Science, Chongqing University
Chongqing, China
chenziyu@cqu.edu.cn

Dingding Chen
College of Computer Science, Chongqing University
Chongqing, China
dingding@cqu.edu.cn

Wenxin Zhang
College of Computer Science, Chongqing University
Chongqing, China
wenxinzhang18@163.com

ABSTRACT

Belief propagation algorithms including Max-sum and its variants are important methods for solving DCOPs. However, they may face a tough challenge when handling n-ary constraints since the computational overheads grow exponentially with the number of variables that a utility function holds.

In this paper, we update the state-of-the-art technique called Function Decomposing and State Pruning (FDSP) which can significantly reduce such an expenditure, by introducing two heuristic techniques. By introducing a round-robin mechanism to control the order of exploration, we propose Concurrent-search-based FDSP (CONC-FDSP). Besides, we propose Best-first-search-based FDSP (BFS-FDSP) by using the A^* search to find the optimal path to the solution. Finally, we demonstrate their efficiency in solving the benchmarks compared with the state-of-the-art.

KEYWORDS

DCOPs; Max-sum; Belief Propagation

ACM Reference Format:

Junsong Gao, Ziyu Chen, Dingding Chen, and Wenxin Zhang. 2022. Beyond Uninformed Search: Improving Branch-and-bound Based Acceleration Algorithms for Belief Propagation via Heuristic Strategies: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

DCOP algorithms can be classified into two categories: incomplete algorithms[11, 13, 14, 20] and complete algorithms[9, 12, 15–17, 19]. Max-sum [6] and its variants are important incomplete algorithms based on the Generalized Distributive Law [1]. They have drawn considerable attention in the DCOP community for their ability to handle $n-$ary constraints and multiple variables per agent [2]. However, they face a tough challenge since the computing for beliefs in Max-sum requires each agent to repetitively maximize the sum of its local constraint function and the incoming utilities to find the locally best configuration of the involved variables. The computation complexity of this process grows exponentially with the number of the variables involved in a constraint function.

To alleviate this issue, two kind of strategies have been proposed to speed up the maximization operation in Max-sum. The first ones include G-FBP [8], GDP [7], GD^{2}P and ART-GD^{2}P [5], all of which require (partially) sorted local utilities, while the second ones, including BnB-MS [18], BnB-FMS [10] and FDSP [3], employ a branch-and-bound technique with estimations. FDSP performs the best among the BnB-based ones, but it could fail to find a high-quality lower bound due to its blind depth-first search and is defeated by ART-GD^{2}P in some scenarios.

Here, we aim to help FDSP to find better lower bounds and enhance its performance by introducing two heuristic strategies, which also means to help BnB-based methods revive again. The first strategy is based on concurrent search[12]. It regards each search space as a process and introduces a round-robin mechanism to help guide the search process. In each round, only the process with the highest estimation can perform exploration. The second strategy makes full use of the estimations constructed by FDSP which is proved optimistic and admissible, and acts like A^* search [4], which always explores the most promising partial assignment.

2 MAX-SUM

[1]Max-sum is an inference-based belief propagation algorithm operating on a factor graph, a bipartite graph representation to a DCOP, which comprises variable nodes representing variables and function nodes representing utility functions, respectively. In Max-sum, the propagation and accumulation of beliefs are implemented by message exchanges between variable nodes and function nodes throughout a factor graph. Formally, the query message sent from a variable node x_i to its neighboring function node $F_m(\mathbf{x_m})$ is defined by $Q_{x_i \to F_m}(x_i) = \alpha_{im} + \sum_{F_j \in N_i \backslash \{F_m\}} R_{F_j \to x_i}(x_i)$ where α_{im} is a

*Corresponding author

[1]For lack of space we do not present a formal definition of DCOP and refer the reader to our recent paper[3]

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

normalization term such that $\sum_{x_i} Q_{x_i \to F_m}(x_i) = 0$, $x_i \in \mathbf{x_m}$ and $N_i \backslash \{F_m\}$ is a set of neighbours of x_i except the target function node F_m. The response message sent from a function-node $F_m(\mathbf{x_m})$ to its neighboring variable node x_i is given by the following:
$R_{F_m \to x_i}(x_i) = \max_{\mathbf{x_m} \backslash \{x_i\}} (F_m(\mathbf{x_m}) + \sum_{x_j \in \mathbf{x_m} \backslash \{x_i\}} Q_{x_j \to F_m}(x_j))$
When a variable node x_i makes a decision, it chooses a value in its domain D_i according to the current beliefs it receives to maximize the total utilities, formalized by $x_i^* = \arg\max_{x_i} \sum_{F_m \in N_i} R_{F_m \to x_i}(x_i)$.

3 FUNCTION DECOMPOSING AND STATE PRUNING

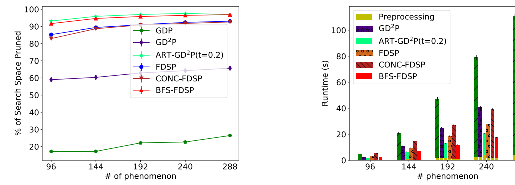
Note that the maximization operation in computing the response message is exponential to the arity of function $F_m(\mathbf{x_m})$, which prohibits Max-sum from handling constraints with large arity. FDSP [3] is the state-of-the-art acceleration algorithm which can greatly reduce such an expenditure. It constructs two kind of function estimations for each variable node $\mathbf{x_{m,i}} \in \mathbf{x_m}$ in a function and conputes estimations for messages dynamically. Then, given an partial assignment, FDSP uses these estimations to build an optimistic and admissible upper bound to prune the search space.

4 CONCURRENT-SEARCH-BASED FDSP

Different from FDSP which explores in a DFS fashion, CONC-FDSP performs a prioritized exploration. It divides the search space into several independent subspaces according to the domain size of the first variable and the exploration of these subspaces is conducted by an individual search process. These processes take turns to execute according to a scheduling strategy which specifies the execution priority of them, and return the control back to the main process according to a round-robin mechanism. Besides, the lower bound found by the pioneer search process is shared to all later search processes and maintained by all of them. In other words, every search process can access the shared lower bound *sharedLB* and update it whenever a better one is found. Besides, we additionally construct $BestEntryView$ to store the full assignment in the utility matrix that corresponds to the best utility for the target assignment to serve as the initial *sharedLB*. Finally, a search process is removed when exhausting all its subspace and CONC-FDSP terminates when no search processes exist.

5 BEST-FIRST-SEARCH-BASED FDSP

We borrow the best first idea from the famous A^* strategies since we notice that the estimations constructed by FDSP is optimistic and admissible. BFS-FDSP, also refered as best-first-based FDSP (BFS-FDSP) in this paper is a more clever way to totally avoid the dependency of lower bounds for FDSP. In each step of exploration, BFS-FDSP will only expand a partial assignment with the highest upper bound (ub) calculated by those estimations constructed by FDSP. To this end, BFS-FDSP maintains a maximum heap to store all the expanded partial assignments, among which it chooses to expand the most promising one (i.e., the partial assignment with the highest ub) and puts the expanding results back to the heap for further expansion. BFS-FDSP terminates when a full assignment is met, which is also the time when the optimal solution is found. In other words, the first full assignment BFS-FDSP found is exactly the optimal solution.



(a) Pruning rate **(b) Runtime**

Figure 1: Performance comparison on 96 Radars

6 EMPIRICAL EVALUATION

We empirically evaluate the performance of our proposed algorithms with GDP, GD^{2}P, ART-GD^{2}P and FDSP for accelerating Max-sum on random DCOPs and netRad systems based on pruned rate and runtime. For the limited space, we only report the experimental results and give a brief description of them. The detailed information about the configuration can be found in papers [3, 5, 7].

For $n-$ary random DCOPs, our proposed methods perform better than FDSP on both sparse and dense problems for all the metrics. On sparse problems, BFS-FDSP helps FDSP match ART-GD^{2}P and outperforms ART-GD^{2}P on large domain and high arity problems for pruning rate, while on dense problem, BFS-FDSP dominates other competitors. Besides, CONC-FDSP can also help narrows the gap between FDSP and ART-GD^{2}P, though not as much as BFD-FDSP does. For runtime, both methods can reduce the it significantly, especially BFS-FDSP which tops on all settings. As a result, all the experimental results indicate the efficiency of our proposed methods.

We then consider the NetRad systems with 96 radars, which is arranged into 8×12 grids [5]. Fig.1 presents the experimental results on 96 radars. It can be seen that BFS-FDSP only performs worse than ART-GD^{2}P and such outperformance narrows with the increase of the phenomena. But, it will cost less time than ART-GD^{2}P, which also indicates that ART-GD^{2}P trades time and memories for efficient pruning. Besides, we can find CONC-FDSP may prune less search space and runs slower than FDSP. It is because that the setting where the maximal arity is just 4 while the maximal domain size is up to 15, leaving the search space large in width but shallow in depth. As a result, even we select the variable with the smallest domain size as the Split Node, there are still too many search processes, which may weaken the advantage of the shared lower bounds held by CONC-FDSP and costs more time on the scheduling for search processes.

7 CONCLUSION

In this paper, we propose two heuristic strategies to speed up the branch-and-bound based acceleration algorithms for belief propagation. To alleviate the paralyzation of FDSP, we propose to generate several search processes for exploration concurrently and maintain a shared lower bound to all of them. Besides, we also propose to use a best-first search strategy to find the optimal path to the soution. Finally, we show the superiority of our proposed methods experimentally.

REFERENCES

[1] Srinivas Aji and Robert McEliece. 2000. The generalized distributive law. *Information Theory, IEEE Transactions on* 46 (04 2000), 325 – 343. https://doi.org/10.1109/18.825794

[2] Jesús Cerquides, Alessando Farinelli, Pedro Meseguer, and Sarvapali Ramchurn. 2013. A Tutorial on Optimization for Multi-Agent Systems. *Comput. J.* 57 (05 2013), 799–824. https://doi.org/10.1093/comjnl/bxt146

[3] Ziyu Chen, Xingqiong Jiang, Yanchen Deng, Dingding Chen, and Zhongshi He. 2019. A Generic Approach to Accelerating Belief Propagation Based Incomplete Algorithms for DCOPs via a Branch-and-Bound Technique. In *The Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, 6038–6045. https://doi.org/10.1609/aaai.v33i01.33016038

[4] Rina Dechter and Judea Pearl. 1985. Generalized best-first search strategies and the optimality of A. *Journal of the ACM (JACM)* 32, 3 (1985), 505–536.

[5] Yanchen Deng and Bo An. 2020. Speeding Up Incomplete GDL-based Algorithms for Multi-agent Optimization with Dense Local Utilities. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, Christian Bessiere (Ed.). ijcai.org, 31–38. https://doi.org/10.24963/ijcai.2020/5

[6] Alessandro Farinelli, Alex Rogers, Adrian Petcu, and Nicholas R Jennings. 2008. Decentralised coordination of low-power embedded devices using the Max-sum algorithm. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and MultiAgent Systems*. 639–646.

[7] Md Khan. 2018. *Speeding up GDL-based distributed constraint optimization algorithms in cooperative multi-agent systems*. Ph.D. Dissertation.

[8] Yoonheui Kim and Victor Lesser. 2013. Improved max-sum algorithm for DCOP with N-ary constraints. *12th International Conference on Autonomous Agents and Multiagent Systems 2013, AAMAS 2013* 1, 191–198.

[9] Omer Litov and Amnon Meisels. 2017. Forward bounding on pseudo-trees for DCOPs and ADCOPs. *Artificial Intelligence* 252 (11 2017), 83–99. https://doi.org/10.1016/j.artint.2017.07.003

[10] Kathryn Macarthur, Ruben Stranders, Sarvapali Ramchurn, and Nicholas Jennings. 2011. A Distributed Anytime Algorithm for Dynamic Task Allocation in Multi-Agent Systems, Vol. 1.

[11] Rajiv T Maheswaran, Jonathan P Pearce, and Milind Tambe. 2004. Distributed Algorithms for DCOP: A Graphical-Game-Based Approach. In *Proceedings of ISCA PDCS'04*. 432–439.

[12] Arnon Netzer, Alon Grubshtein, and Amnon Meisels. 2012. Concurrent forward bounding for distributed constraint optimization problems. *Artificial Intelligence* 193 (12 2012), 186–216. https://doi.org/10.1016/j.artint.2012.09.002

[13] 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.

[14] 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 (TIST)* 8, 5 (2017), 69.

[15] Adrian Petcu and Boi Faltings. 2005. DPOP: A Scalable Method for Multiagent Constraint Optimization. *IJCAI International Joint Conference on Artificial Intelligence*, 266–271.

[16] Adrian Petcu and Boi Faltings. 2006. ODPOP: An algorithm for open/distributed constraint optimization. In *Proceedings of the 21st AAAI conference on artificial intelligence*. 703–708.

[17] Adrian Petcu and Boi Faltings. 2007. MB-DPOP: A New Memory-Bounded Algorithm for Distributed Optimization. (01 2007).

[18] Ruben Stranders, Alessando Farinelli, Alex Rogers, and Nicholas Jennings. 2009. Decentralised Coordination of Mobile Sensors Using the Max-Sum Algorithm. 299–304.

[19] Meritxell Vinyals, Juan Rodríguez-Aguilar, and Jesús Cerquides. 2009. Generalizing DPOP: Action-GDL, a new complete algorithm for DCOPs. 1239–1240. https://doi.org/10.1145/1558109.1558231

[20] Weixiong Zhang, Guandong Wang, Zhao Xing, and Lars Wittenburg. 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.