Improving Solution Quality of Bounded Max-Sum Algorithm to Solve DCOPs involving Hard and Soft Constraints

Md. Musfiqur Rahman\textsuperscript{1} and Mashrur Rashik\textsuperscript{2} and Md. Mamun-or-Rashid\textsuperscript{1} and Md. Mosaddek Khan\textsuperscript{4}

Abstract. Bounded Max-Sum (BMS) is a message-passing algorithm that provides approximation solution to a specific form of decentralized coordination problems, namely Distributed Constrained Optimization Problems (DCOPs). In particular, BMS algorithm is able to solve problems of this type having large search space at the expense of low computational cost. Notably, the traditional DCOP formulation does not consider those constraints that must be satisfied (also known as hard constraints), rather it concentrates only on soft constraints. Hence, although the presence of both types of constraints are observed in a number of real-world applications, the BMS algorithm does not actively capitalize on the hard constraints. To address this issue, we tailor BMS in such a way that can deal with DCOPs having both type constraints. In so doing, our approach improves the solution quality of the algorithm. The empirical results exhibit a marked improvement in the quality of the solutions of large DCOPs.

1 INTRODUCTION

Distributed Constrained Optimization Problems (DCOPs) are a popular framework to coordinate interactions in cooperative multi-agent systems. A number of real-world problems such as distributed event scheduling\textsuperscript{7} and the distributed RLFA problem\textsuperscript{1\textsuperscript{1}} can be modelled with this framework\textsuperscript{7}\textsuperscript{1\textsuperscript{1}}\textsuperscript{5\textsuperscript{2\textsuperscript{1}}}\textsuperscript{1\textsuperscript{3}}. The constraints among the participating agents in these applications, and many others besides, can be both hard and soft. In any case, the algorithms that have been proposed to solve DCOPs can be broadly classified into exact and non-exact algorithms. The former (e.g.\textsuperscript{9}\textsuperscript{10}) always finds a globally optimal solution. In contrast, the latter algorithms (e.g.\textsuperscript{3}\textsuperscript{1\textsuperscript{3}}) trade solution quality at the expense of reduced computation and communication costs.

Among the non-exact approaches, Generalized Distributive Law based algorithms, such as Max-Sum\textsuperscript{3} and Bounded Max-Sum (BMS)\textsuperscript{1\textsuperscript{2}} have received particular attention. Specifically, Bounded Max-Sum is extremely attractive variant of Max-Sum which produces good approximate solution for DCOP problems with cycles. However, BMS does not actively consider such constraints that are hard although there is a number of real-life applications containing hard constraints. We particularly observe that the presence of hard constraints can be utilized to further improve BMS’s solution quality by removing inconsistent values from agents’ domain and thus reduce the upper bound of the global solution. It is worth noting that due to hard constraints the traditional BMS algorithm exhibits a situation where each agent may own a set of allowable assignments in place of a specific assignment. Each combination of the agent’s assignments will experience the same profit for the tree structured graphical representation that is a acyclic graph (e.g. Factor Graph or Junction Tree) of a given DCOP, but produce different profit for that cyclic DCOP. To the best of our knowledge, there exists no method for choosing the best one. In this paper, we propose a novel approach which aims at enforcing consistency and selecting the most preferable combination of agent’s assignments.

2 PROBLEM FORMULATION

A DCOP model can be formally expressed as a tuple (A, X, D, F, α) where A = {a\textsubscript{1}, a\textsubscript{2}, ..., a\textsubscript{n}} is a set of agents, X = {x\textsubscript{1}, x\textsubscript{2}, ..., x\textsubscript{m}} is a set of variables, D = {d\textsubscript{1}, d\textsubscript{2}, ..., d\textsubscript{n}} is a set of domains for the variables in X, F = {f\textsubscript{1}, f\textsubscript{2}, ..., f\textsubscript{m}} is a set of constraint functions. f\textsubscript{1}(x\textsubscript{i}) denotes value for each possible combination of the variables of x\textsubscript{i} ∈ X. The dependencies between the functions and variables can be graphically represented by factor graph FG. Finally, the mapping of variable node to agent is represented by α : X → A where one variable will be assigned to one agent.

Within this model, the main objective of DCOP algorithms such as BMS, is to find the assignment of each variable, \( \hat{x} \) and approximate solution \( \hat{V} \) by maximizing the sum of all functions that is \( \hat{V} = \sum_{\tilde{x}} f_j(\tilde{x}) \). After removing appropriate dependencies from FG according to the phases of BMS, an acyclic graph FG is formed. Additionally, the maximum impact \( B \) is calculated which is used for computing the upper bound on the value of the unknown optimal solution as \( V^m \rightarrow B \). Here \( V^m \) is the solution found by executing BMS on the corresponding acyclic graph. Now, the first objective of our approach is to update the domain of each variables so that the maximum impact is minimized as in (Equation \ref{eq:1}). Here, \( \hat{d}_i \) is inconsistent domain values of variable \( x_i \).

\[
D = \text{argmin}_{d_i \in D} \hat{d}_i \quad \text{s.t.} \forall d_i \in D, d_i = d_i \setminus \hat{d}_i
\]

Due to the presence of hard constraints, the BMS algorithm experiences tie variable assignment(s) after executing Max-Sum on its acyclic graph. Our second objective is to select appropriate variable assignment so that it provides the most preferable solution for the constraint graph (Equation \ref{eq:2}). Here, \( T_j \subseteq D_i \) is set of tie assignments for variable \( x_i \).

\[
\hat{x} = \text{argmin}_{\tilde{x}} \sum_{j=1}^{m} f_j(\hat{x}_j) \quad \text{s.t.} (x_1, x_2, ..., x_n) \in (T_1 \times T_2 \times ..., \times T_m)
\]

3 HARD CONSISTENCY ENFORCED BOUNDED Max-Sum (HBMS)

With the motivation of utilizing hard constraints, our objective is to decrease the upper bound on the optimal solution and also increase the solution quality of BMS. Our first contribution in the upper bound is obtained by changing the maximum impact \( B \). In our first phase, consistency enforcement, we update the variable domains (Equation \ref{eq:1}) by enforcing arc-consistency on the constraint graph. This step also speeds up the execution time of the algorithm. For example,
in Figure 1a, \( F_0 \) calculates its maximum impact \( B_{01} \) using Equation 3. According to this equation, \( x_1 \) selects its value \( x_{1b} \) for maximizing and \( x_{1l} \) for minimizing \( F_{01} \) function. After the consistency enforcement phase, each variable’s domain gets pruned and \((x_{1b}, x_{1l})\) pair changes to \((x'_{1b}, x'_{1l})\) by reducing \( B \). This is true for \( F_{03} \) and \( F_{23} \) in the same way.

\[
B_{01} = \max_{x_1} \left[ \max_{x_{1l}} F_{01}(x_0, x_1) - \min_{x_{1b}} F_{01}(x_0, x_1) \right]
\]  

In the second phase, we generate a spanning tree (i.e. acyclic graph) from the factor graph by removing the most suitable dependencies. We do this following the same way as the BMS algorithm. Then run Max-Sum algorithm on the acyclic factor graph. If the constraint graphs are removed in the previous phase. It consists of \( F_{01}, F_{03}, F_{23} \) and their corresponding variables \((x_0, x_1)\), \((x_0, x_2)\) and \((x_2, x_3)\). At this phase, we execute BMS on this smaller graph and eventually get information about the priority of the tie assignments of each variable that we found in second phase. In Figure 1c, we can see the profit for each assignment received from the third phase (e.g. for \( x_2: 0 : 633, 6 : 715, 8 : 1209, 17 : 943 \)). Finally, In the fourth phase, we use this information to execute Max-Sum on \( G \) again.

This step will finally select the preferable variable assignment which improves solution quality. For instance, the variable assignment is \((x_0: 14), (x_1: 1), (x_2: 8), (x_3: 13)\). The complexity of HBMS is twice of the BMS algorithm since we execute this algorithm two times. The computation cost for the smaller graph in the second phase is negligible. Finally, the complexity of the arc consistency enforcement phase is \( ed^3 \) where \( e \) is the number of edges of the constraint graph and \( d \) is the average domain size.

4 EMPIRICAL EVALUATION

In this section, we empirically evaluate the improvement in solution quality of HBMS in comparison to the Bounded Max-Sum algorithm. To benchmark the result, we run experiment on random constraint graphs. We vary the number of nodes from 5 to 30 in Figure 2 and set the variable domain in \([0, ..., 40]\), 30% hard constraints along with soft constraints and functions’ utility values from 0 to 500. We observe improvement in solution quality around 5-30% on average. However, we experience negative results in some instances. In the future, we would like to explore that area for observing the reasons behind this situation.

5 CONCLUSIONS AND FUTURE WORK

The major finding of this paper is that by taking advantage of the hard constraints, we can significantly improve the solution quality of the Bounded Max-Sum algorithm. Another notable contribution is in the reduction of the upper bound. Our empirical evidence presents that it is possible to improve the solution around 10-30% than BMS. In our future work, we would like to observe the impact of different forms of consistency enforcement, and fix the negative results observed in the evaluation. The final research direction includes extending the potential application domain.

---

**Figure 1:** Worked example of the HBMS algorithm. Here the square nodes and round nodes represent function nodes and variable nodes, respectively. The thick edges of the graph are our main points of interest. The third phase is executed on the sub-graph phase consists of blue color nodes, processes

**Figure 2:** Empirical result for constraint graphs varying number of nodes from 4 to 30. Improvement is calculated in percentage with respect to Bounded Max-Sum.
REFERENCES

[1] Bertrand Cabon, Simon De Givry, Lionel Lobjois, Thomas Schiex, and Joost P. Warners, ‘Radio link frequency assignment’, Constraints, 4(1), 79–89, (1999).

[2] Moumita Choudhury, Saaduddin Mahmud, Md Mosaddek Khan, et al., ‘A particle swarm based algorithm for functional distributed constraint optimization problems’, in AAAI, pp. 7111–7118, (2020).

[3] Alessandro Farinelli, Alex Rogers, Adrian Petcu, and Nicholas R Jennings, ‘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-Volume 2, pp. 659–646. International Foundation for Autonomous Agents and Multiagent Systems, (2008).

[4] Md Khan, Long Tran-Thanh, Nicholas Jennings, et al., ‘A generic domain pruning technique for gdl-based dcop algorithms in cooperative multi-agent systems’, (2018).

[5] Md Khan, Long Tran-Thanh, William Yeoh, Nicholas Jennings, et al., ‘A near-optimal node-to-agent mapping heuristic for gdl-based dcop algorithms in multi-agent systems’, (2018).

[6] Md Mosaddek Khan, Long Tran-Thanh, Sarvapali D Ramchurn, and Nicholas R Jennings, ‘Speeding up gdl-based message passing algorithms for large-scale dcops’, The Computer Journal, 61(11), 1639–1666, (2018).

[7] Rajiv T Maheswaran, Milind Tambe, Emma Bowring, Jonathan P Pearce, and Pradeep Varakantham, ‘Taking dcop to the real world: Efficient complete solutions for distributed multi-event scheduling’, in Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1, pp. 310–317. IEEE Computer Society, (2004).

[8] Saaduddin Mahmud, Moumita Choudhury, Md Khan, Long Tran-Thanh, Nicholas R Jennings, et al., ‘Aed: An anytime evolutionary dcop algorithm’, arXiv preprint arXiv:1909.06254, (2019).

[9] Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo, ‘Adopt: Asynchronous distributed constraint optimization with quality guarantees’, Artificial Intelligence, 161(1-2), 149–180, (2005).

[10] Adrian Petcu and Boi Faltings, ‘A scalable method for multiagent constraint optimization’, Technical report, (2005).

[11] Mashur Rashik, Md Musliqur Rahman, Md Mosaddek Khan, Md Mamun-or Rashid, Long Tran-Thanh, and Nicholas R Jennings, ‘Speeding up distributed pseudo-tree optimization procedures with cross edge consistency to solve dcops’, Applied Intelligence, 1–14, (2020).

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

[13] Weixiong Zhang, Guandong Wang, Zhao Xing, and Lars Wittenburg, ‘Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks’, Artificial Intelligence, 161(1-2), 55–87, (2005).