Max-sum Revisited; The Real Power of Damping

(Extended Abstract)

Liel Cohen and Roie Zivan
Industrial Engineering and Management Department, Ben Gurion University
Beer-Sheva, Israel
lielc@post.bgu.ac.il, zivanr@bgu.ac.il

ABSTRACT

Max-sum is a version of Belief Propagation, used for solving DCOPs. On tree-structured problems, Max-sum converges to the optimal solution in linear time. Unfortunately, on cyclic problems, Max-sum does not converge and explores low quality solutions. Damping is a method, often used for increasing the chances that Belief Propagation will converge. That been said, it was not mentioned in the studies that proposed Max-sum for solving DCOPs.

In this paper we advance the research on incomplete inference DCOP algorithms by investigating the effect of damping on Max-sum. We prove that Max-sum with damping is guaranteed to converge to the optimal solution in weakly polynomial time. Our empirical results demonstrate a drastic improvement in the performance of Max-sum, when using damping. However, in contrast to the common assumption, that it performs best when converging, we demonstrate that non converging versions perform efficient exploration, and produce high quality results, when implemented within an anytime framework.

1. INTRODUCTION

DCOP algorithms generally follow one of two broad approaches: distributed search [7, 2, 6, 19] or inference [9, 8, 1, 14]. The Max-sum algorithm [1] is an incomplete, GDL (Generalized Distributive Law) based inference algorithm that has drawn considerable attention in recent years, including being proposed for multi-agent applications such as sensor systems [16, 14] and task allocation for rescue teams in disaster areas [11]. Max-sum is actually a version of the well known Belief propagation algorithm [18], used for solving DCOPs. Agents in Max-sum propagate cost/utility information to all neighbors. As is typical of inference algorithms, Max-sum is purely exploitive both in the computation of its beliefs and in its selection of values based on those beliefs.

Belief propagation in general (and Max-sum specifically) is known to converge to the optimal solution on acyclic problems. Unfortunately, there is no such guarantee for problems with cycles [18, 1]. Furthermore, when the agents' beliefs fail to converge, the resulting assignments may be of low quality. In fact, many DCOPs that were investigated in previous studies are dense and indeed include multiple cycles (e.g., [7, 2]).

Damping is a method that was combined with Belief propagation in order to decrease the effect of cyclic information propagation. By balancing the weight of the new calculation performed in each iteration and the weight of calculations performed in previous iterations, researchers have reported success in increasing the chances for convergence of Belief propagation when applied in different scenarios [5, 13, 15, 10]. Nevertheless, Damping was not mentioned in the papers that adopted Max-sum for solving DCOPs and proposed extended versions of the algorithm [1, 12, 21].

In this paper we contribute to the development of incomplete inference algorithms for solving DCOPs by investigating the effect of using damping within the Max-sum algorithm.

2. THE MAX-SUM ALGORITHM

[1]Max-sum operates on a *factor-graph*, which is a bipartite graph in which the nodes represent variables and constraints [4]. Each variable-node representing a variable of the original DCOP is connected to all function-nodes that represent constraints, which it is involved in. All nodes are considered "agents" in Max-sum, i.e., they send and receive messages, and perform computation.

A message sent to or from variable-node x (for simplicity, we use the same notation for a variable and the variable-node representing it) is a vector of size $|D_x|$ including a cost for each value in D_x. In the first iteration all messages include vectors of zeros. A message sent from a variable-node x to a function-node f in iteration i is formalized as follows: $Q^i_{x\rightarrow f} = \sum_{f'\in F_x, f'\neq f} R^{i-1}_{f'\rightarrow x} - \alpha$, where F_x is the set of function-node neighbors of variable-node x and $R^{i-1}_{f'\rightarrow x}$ is the message sent to variable-node x by function-node f' in iteration $i-1$. α is a constant that is reduced from all costs included in the message (i.e., for each $d \in D_x$) in order to prevent the costs carried by messages throughout the algorithm run from growing arbitrarily.

A message sent from a function-node f to a variable-node x in iteration i includes for each value $d \in D_x$: $min_{PA_{-x}} cost(\langle x, d\rangle, PA_{-x})$, where PA_{-x} is a possible combination of value assignments to variables involved in f not including x. The term $cost(\langle x, d\rangle, PA_{-x})$ represents the cost of a partial assignment $a = \{\langle x, d\rangle, PA_{-x}\}$, which is: $f(a) + \sum_{x'\in X_f, x'\neq x, \langle x', d'\rangle \in a} Q^{i-1}_{x'\rightarrow f}.d'$, where $f(a)$ is the original cost in the constraint represented by f for the partial assignment a, X_f is the set of variable-node neighbors of f, and $Q^{i-1}_{x'\rightarrow f}.d'$ is the cost that was received in the message sent from variable-node x' in iteration $i-1$, for the value d' that is assigned to x' in a. x selects its value assignment $\hat{d} \in D_x$ following iteration k as follows: $\hat{d} = \arg\min_{d\in D_x} \sum_{f\in F_x} R^k_{f\rightarrow x}.d$.

[1]For lack of space we do not present a formal definition of DCOP and refer the reader to our recent paper [20].

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

3. INTRODUCING DAMPING INTO MAX-SUM

In order to add damping to Max-sum we introduce a parameter $\lambda \in (0, 1]$. Before sending a message in iteration k an agent performs calculations as in standard Max-sum. Denote by $\widehat{m^{k}_{i\to j}}$ the result of the calculation made by agent A_i of the content of a message intended to be sent from A_i to agent A_j in iteration k. Denote by $m^{k-1}_{i\to j}$ the message sent by A_i to A_j at iteration $k-1$. The message sent from A_i to A_j in iteration k is calculated as follows: $m^{k}_{i\to j} = \lambda m^{k-1}_{i\to j} + (1-\lambda)\widehat{m^{k}_{i\to j}}$. Thus, λ expresses the weight given to previously performed calculations with respect to the most recent calculation performed. Moreover, when $\lambda = 0$ the resulting algorithm is standard Max-sum.

In all our implementations damping was performed only by variable-nodes. This allowed us to analyze the level of damping with respect to n (the number of variables/agents in the problem).

4. CONVERGENCE RUNTIME BOUNDS

Standard Max-sum guarantees convergence in linear time to the optimal solution, when the constraint graph (and as a result, the corresponding factor-graph as well) is tree-structured, i.e., contains no cycles. We prove that on such problems, there exists a scenario in which damping slows the convergence to weakly polynomial time. Furthermore, we prove that in the worst case, Damped Max-sum will converge on a tree-structured factor-graphs in weakly polynomial time. The proofs were omitted for lack of space. Similar proofs can be applied to other structures on which Max-sum is guaranteed to converge, e.g., graphs with a single cycle [17] and directed acyclic graphs (on which it converges, but not necessary to the optimal solution) [21].

5. EXPERIMENTAL EVALUATION

We performed a set of experiments comparing different versions of the algorithm, using different λ values with standard Max-sum, two versions that guarantee convergence: Bounded_Max-sum [12] and Max-sum_ADVP [21] and the well known DSA algorithm (we use type C with $p = 0.7$ [19]).

We evaluated the algorithms on random uniform DCOPs and on structured and realistic problems, i.e., graph coloring, meeting scheduling and scale-free, all formulated as minimization problems. At each experiment we randomly generated 50 different problem instances and ran the algorithms for 5,000 iterations on each of them. The results presented are an average of those 50 runs. For each iteration we present the cost of the assignment that would have been selected by each algorithm at that iteration. All algorithms were implemented within the anytime framework proposed in [20], which allowed us to report for each of them the best result it traverses within $5,000$ iterations. Also, in all versions of Max-Sum, we used value preferences selected randomly for the purpose of tie breaking, as was suggested in [1].

Figure 1 presents the solution costs found by all algorithms when solving uniform random problems containing 100 agents with density ($p_1 = 0.1$). Results of higher density problems ($p_1 = 0.7$) showed similar trends. The results per iteration show that Damped Max-sum is inferior to DSA and the guaranteed convergence version Max-sum_ADVP. That been said, the anytime results of Damped Max-sum using high λ values (0.7 and 0.9) significantly outperform DSA and Max-sum_ADVP. This suggests that damping triggers efficient exploration by Max-sum, i.e., that in contrast to the assumptions made in the Belief propagation literature, the best results of Max-sum are not achieved when it converges but rather (like in the case of local search) when there is a balance between exploration and exploitation. On uniform random problems containing 50 agents and on scale free nets, the trends are similar to the results described above. Damped Max-sum improves as more iterations are performed and explores solutions of higher quality. Towards the end of the run, the results per iteration of the version with $\lambda = 0.9$ produces in some iterations better solutions than DSA and similar to Max-sum_ADVP. The anytime results outperform the converging algorithm significantly.

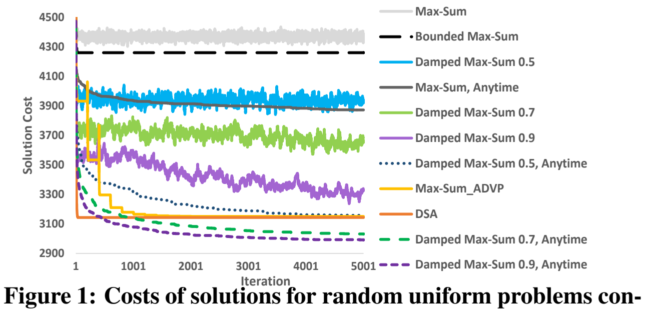

Figure 1: Costs of solutions for random uniform problems containing 100 agents with relatively low density ($p_1 = 0.1$).

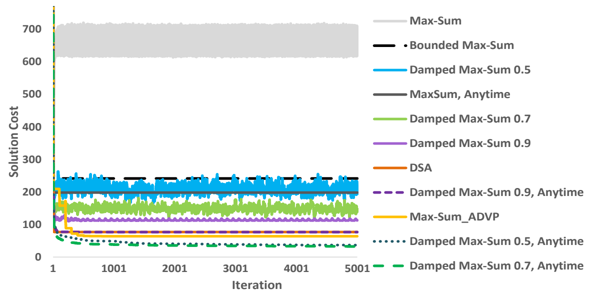

Figure 2: Costs of solutions for Graph coloring problems.

On meeting scheduling and graph coloring problems, the results of the Damped Max-sum versions did not exhibit such an improvement, and explored solutions of similar quality throughout the run. Figure 2 presents results on graph coloring problems. Interestingly, the $\lambda = 0.9$ version on graph coloring seems to perform limited exploration and traverse solutions with similar quality, while the 0.5 and 0.7 versions perform a higher level of exploration.

Our results indicate that, in contrast to the common assumption regarding the role of damping in improving Belief propagation, by increasing its convergence rate, the success of damping is in generating useful exploration of high quality solutions that can be captured by an anytime framework and outperform versions of Max-sum that guarantee convergence, as Max-sum_ADVP.

6. CONCLUSION

Our results indicate that the most successful versions of the algorithm are for relatively high values of λ. Moreover, while damping improved the results of the algorithm drastically, in most cases it did not converge within 5000 iterations. However, when combined with an anytime framework that reports the best solution explored by the algorithm, Max-sum with damping significantly outperforms the best versions of Max-sum, and a standard local search algorithm as well. This is in contrast to the common assumption in the graphical models literature that BP performs best when it converges [3, 10]. Apparently, damping can be used to generate a balance between exploration and exploitation that results in traversal of high quality solutions.

Acknowledgment: We thank the ABC Robotics Initiative at Ben-Gurion University of the Negev, for funding this project.

REFERENCES

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

[2] A. Gershman, A. Meisels, and R. Zivan. Asynchronous forward bounding. *J. of Artificial Intelligence Research*, 34:25–46, 2009.

[3] M. I. J. K. P. Murphy, Y. Weiss. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of Uncertainty in AI*, pages 467–475, 1999.

[4] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:2:181–208, Febuary 2001.

[5] N. Lazic, B. Frey, and P. Aarabi. Solving the uncapacitated facility location problem using message passing algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 429–436, 2010.

[6] R. T. Maheswaran, J. P. Pearce, and M. Tambe. Distributed algorithms for dcop: A graphical-game-based approach. In *PDCS)*, pages 432–439, September 2004.

[7] P. J. Modi, W. Shen, M. Tambe, and M. Yokoo. Adopt: asynchronous distributed constraints optimizationwith quality guarantees. *Artificial Intelligence*, 161:1-2:149–180, January 2005.

[8] A. Petcu and B. Faltings. Approximations in distributed optimization. In P. van Beek, editor, *CP 2005, LNCS 3709*, pages 802–806, 2005.

[9] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *IJCAI*, pages 266–271, 2005.

[10] M. Pretti. A message-passing algorithm with damping. *Journal of Statistical Mechanics: Theory and Experiment*, 11:P11008, 2005.

[11] S. D. Ramchurn, A. Farinelli, K. S. Macarthur, and N. R. Jennings. Decentralized coordination in robocup rescue. *Computer J.*, 53(9):1447–1461, 2010.

[12] A. Rogers, A. Farinelli, R. Stranders, and N. R. Jennings. Bounded approximate decentralized coordination via the max-sum algorithm. *Artificial Intelligence*, 175(2):730–759, 2011.

[13] P. Som and A. Chockalingam. Damped belief propagation based near-optimal equalization of severely delay-spread uwb mimo-isi channels. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE, 2010.

[14] R. Stranders, A. Farinelli, A. Rogers, and N. R. Jennings. Decentralised coordination of mobile sensors using the max-sum algorithm. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 299–304, 2009.

[15] D. Tarlow, I. Givoni, R. Zemel, and B. Frey. Graph cuts is a max-product algorithm. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, 2011.

[16] W. T. L. Teacy, A. Farinelli, N. J. Grabham, P. Padhy, A. Rogers, and N. R. Jennings. Max-sum decentralized coordination for sensor systems. In *AAMAS*, pages 1697–1698, 2008.

[17] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12(1):1–41, 2000.

[18] C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation - an empirical study. *Journal of Machine Learning Research*, 7:1887–1907, 2006.

[19] W. Zhang, Z. Xing, G. Wang, and L. Wittenburg. Distributed stochastic search and distributed breakout: properties, comparishon and applications to constraints optimization problems in sensor networks. *Artificial Intelligence*, 161:1-2:55–88, January 2005.

[20] R. Zivan, S. Okamoto, and H. Peled. Explorative anytime local search for distributed constraint optimization. *Artificial Intelligence*, 211, 2014.

[21] R. Zivan and H. Peled. Max/min-sum distributed constraint optimization through value propagation on an alternating DAG. In *AAMAS*, pages 265–272, 2012.