Towards Improving the Expressivity and Scalability of Distributed Constraint Optimization Problems

William Yeoh
Department of Computer Science and Engineering
Washington University in St. Louis
wyeoh@wustl.edu

Abstract

Constraints have long been studied in centralized systems and have proven to be practical and efficient for modeling and solving resource allocation and scheduling problems. Slightly more than a decade ago, researchers proposed the *distributed constraint optimization problem* (DCOP) formulation, which is well suited for modeling distributed multi-agent coordination problems. In this paper, we highlight some of our recent contributions that are aiming towards improved expressivity of the DCOP model as well as improved scalability of the accompanying algorithms.

1 Introduction

In a *distributed constraint optimization problem* (DCOP), agents need to coordinate the assignments of values to their variables in such a way that maximizes their aggregated utilities [Modi *et al.*, 2005; Petcu and Faltings, 2005a]. They are well suited for modeling multi-agent coordination problems where the primary interactions are between local subsets of agents, such as in meeting scheduling [Maheswaran *et al.*, 2004b], sensor networks [Farinelli *et al.*, 2014], multi-robot coordination [Zivan *et al.*, 2015], coalition formation [Ueda *et al.*, 2010], smart grid [Kumar *et al.*, 2009; Fioretto *et al.*, 2017b], and smart home automation [Rust *et al.*, 2016; Fioretto *et al.*, 2017a] problems.

When DCOPs were introduced slightly more than a decade ago, research efforts were initially focused on the investigation of different algorithmic paradigms to solve the problem, including exact search-based methods [Modi *et al.*, 2005; Gershman *et al.*, 2009; Yeoh *et al.*, 2010; Gutierrez *et al.*, 2011], exact inference-based methods [Petcu and Faltings, 2005a; Vinyals *et al.*, 2011], approximate search-based methods [Maheswaran *et al.*, 2004a; Zhang *et al.*, 2005; Zivan *et al.*, 2014], approximate inference-based methods [Farinelli *et al.*, 2014; Zivan and Peled, 2012], and approximate sampling-based methods [Ottens *et al.*, 2017].

Since then, the field has substantially matured and researchers have begun to propose generalizations to the model and algorithms to better capture and exploit characteristics of more complex applications. The goal of this paper is to briefly highlight our recent contributions in this area. We refer readers to our overview and survey articles [Yeoh and Yokoo, 2012; Fioretto *et al.*, 2018a] for a more comprehensive perspective on the state of the art of DCOPs.

2 DCOP Model

A *distributed constraint optimization problem* (DCOP) [Modi *et al.*, 2005] is a tuple $\langle \mathbf{A}, \mathbf{X}, \mathbf{D}, \mathbf{F}, \alpha \rangle$, where:

- $\mathbf{A} = \{a_i\}_{i=1}^{p}$ is a set of *agents*;
- $\mathbf{X} = \{x_i\}_{i=1}^{n}$ is a set of decision *variables*;
- $\mathbf{D} = \{D_x\}_{x \in \mathbf{X}}$ is a set of finite *domains* and each variable $x \in \mathbf{X}$ takes values from the set $D_x \in \mathbf{D}$;
- $\mathbf{F} = \{f_i\}_{i=1}^{k}$ is a set of *constraints*, each defined over a set of decision variables: $f_i : \prod_{x \in \mathbf{x}^{f_i}} D_x \rightarrow \mathbb{R}_0^+ \cup \{-\infty\}$, where infeasible configurations have $-\infty$ utility, $\mathbf{x}^{f_i} \subseteq \mathbf{X}$ is the *scope* of f_i; and
- $\alpha : \mathbf{X} \rightarrow \mathbf{A}$ is a function that associates each decision variable to one agent.

The goal is to find an optimal solution $\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}} \mathbf{F}(\mathbf{x})$, where $\mathbf{F}(\mathbf{x})$ is the sum of utilities across all constraints.

3 Our Recent Contributions

We now briefly highlight several key contributions that we have made in this area.

3.1 Hierarchical Decomposition for DCOPs

In most complex distributed multi-agent applications, each agent typically needs to solve complex local subproblems [Kim and Lesser, 2013; Giuliani *et al.*, 2014; Amigoni *et al.*, 2015]. While these complex local structures can be captured by the regular DCOP model, through allowing an agent to control multiple variables, unfortunately, many DCOP algorithms commonly assume that each agent controls only one variable. To cope with such restrictions, reformulation techniques are commonly used to transform a regular DCOP into one where each agent controls exclusively one variable. There are two commonly used reformulation techniques [Burke and Brown, 2006; Yokoo, 2001]: *(i) Compilation*, where each agent creates a new *pseudo-variable*, whose domain is the Cartesian product of the domains of all variables of the agent; and *(ii) Decomposition*, where each agent creates a *pseudo-agent* for each of its variables. While both techniques are relatively simple, they can be inefficient. In compilation, the memory requirement for each agent grows exponentially with the number of variables that it controls. In decomposition, the DCOP algorithms will treat two pseudo-agents as independent entities, resulting in unnecessary computation and communication costs.

Therefore, we proposed a novel decomposition method, called *multi-variable agent* (MVA) decomposition, to overcome these limitations [Fioretto *et al.*, 2016b]. Our MVA decomposition enables a separation between the agents' *local subproblems*, which can be solved independently using centralized solvers, and the DCOP *global problem*, which requires coordination between the agents. Thus, it enables the use of different centralized and distributed solvers in a hierarchical and parallel way, where different agents can even use different centralized solvers that exploit their local subproblem structures for efficiency gains. Not surprisingly, our experimental results show that DCOP algorithms using our MVA decomposition have reduced computation and communication costs compared to using the compilation and decomposition methods.

3.2 Linear-space Sampling-based Algorithm

Distributed UCT (DUCT) [Ottens *et al.*, 2017] is the first sampling-based DCOP algorithm that was introduced. Unfortunately, its memory requirement per agent is exponential in the number of agents in the problem, which prohibits it from some multi-agent applications like sensor networks, where agents may have a very limited amount of memory. In response to this limitation, we proposed the *Distributed Gibbs* algorithm [Nguyen *et al.*, 2013], which extends upon the (centralized) Gibbs algorithm [Geman and Geman, 1984], which is used to solve *maximum a posteriori* (MAP) estimation problems in graphical models.

To do this, we first showed that one can map DCOPs to MAP estimation problems [Kumar *et al.*, 2011], where DCOP variables and constraints correspond to random variables and potential functions of MAP estimation problems. Then, any algorithm that solves MAP estimation problems can theoretically be used to solve DCOPs as well. However, as DCOP algorithms must be distributed and agent oriented, MAP estimation algorithms, which are typically centralized, cannot be directly applied to solve DCOPs. Thus, we extended the well-known Gibbs algorithm to a distributed, agent-oriented version to solve DCOPs. A key property of this algorithm is that its memory requirement per agent is *linear* in the number of agents in the problem in contrast to DUCT's *exponential* memory requirement.

3.3 Exploiting Parallelism using GPUs

Sampling-based DCOP algorithms, such as Distributed Gibbs, perform a significant number of sampling operations that are conditionally-independent operations. As such, instead of performing these sampling operations sequentially, they can be done in parallel and sped up through the use of *graphical processing units* (GPUs) [Fioretto *et al.*, 2016a]. We further show that when Distributed Gibbs is used in conjunction with the MVA decomposition, we can achieve further speedups as the local subproblems within each agent can also be sampled in parallel [Fioretto *et al.*, 2016b]. Finally, we also show that other inference-based DCOP algorithms like DPOP can also be sped up through the use of GPUs, as the computation of utility tables that are propagated between agents can be decomposed into conditionally-independent operations [Fioretto *et al.*, 2018b].

The main challenge in this line of work is to parallelize the algorithms in such a way that optimizes the speedup from using GPUs. While it is relatively simple to develop correct programs (e.g., by incrementally modifying a sequential program), it is nevertheless challenging to design an efficient solution. Several factors are critical in gaining performance. Memory levels have significantly different sizes and access times, and various optimization techniques are available (e.g., accesses to consecutive global memory locations by contiguous threads can be coalesced into a single memory transaction). Thus, optimization of these programs requires a thorough understanding of GPU hardware characteristics.

3.4 Constraint Propagation for DCOPs

Constraint propagation [Mohr and Henderson, 1986] is a commonly used technique, especially in the constraint programming community, to speed up the search for solutions in constraint-based models. The key idea is that one can prune portions of the search space if they do not satisfy some subset of hard constraints (i.e., constraints that must be satisfied) in the problem. Motivated by their success in centralized constraint-based models, we proposed a new constraint propagation technique that is tailored for DCOP inference algorithms that use pseudo-trees. Our new branch consistency approach [Fioretto *et al.*, 2014] can be viewed as a generalization of the more traditional arc consistency and a weaker version of path consistency [Mohr and Henderson, 1986], where it is customized for distributed operations of agents that can communicate only with neighboring agents.

Instead of designing specialized constraint propagation techniques, one can also leverage the advances made by other communities to automatically propagate constraints. Towards this end, we proposed the use of *answer set programming* (ASP), developed by the logic programming community, in solving DCOPs [Le *et al.*, 2015; 2017]. When constraint utilities are described in functional form, they can be compactly represented as ASP rules and propagated between agents efficiently, resulting in improved scalability through reduced runtimes and reduced communication overheads. This line of work is especially promising as continuous advancements made by the logic programming community can be automatically adopted to solve DCOPs with little effort.

3.5 Dynamic and Uncertain DCOPs

All our contributions described above are in the context of solving regular DCOPs. However, many multi-agent coordination problems change dynamically over time and with uncertainty. Early efforts in modeling such *dynamic DCOPs* have been to represent them as sequences of regular DCOPs, with changes between subsequent problems, and the goal is to solve each individual DCOP optimally [Petcu and Faltings, 2005b]. As changes between subsequent DCOPs can be small, a large portion of the problem will remain unchanged and, consequently, a large portion of the solution to the previous problem can be reused. With this observation in mind, we proposed incremental search-based approaches that identify and reuse such reusable portions, and focus their search efforts on solving the portion of the problem that changed [Yeoh *et al.*, 2015]. Results show that the speedup obtain is, not surprisingly, correlated with the size of the reusable portion of the previous solution.

In problems where only the constraint utilities change (the agents, variables, domains, scope of constraints, etc. remain unchanged), one can model the changes in utilities as a function of an underlying state that changes over time. For example, in a sensor network application, a dynamically changing phenomena that the network is trying to observe can be represented as an underlying state that changes over time. To solve these problems, called *Markovian dynamic DCOPs*, we proposed distributed reinforcement learning algorithms that learn the underlying state transition function and, consequently, the resulting constraint utilities, thereby allowing them to find good solutions over time [Nguyen *et al.*, 2014]. Finally, when the state transitions functions are known or can be estimated a priori, we proposed proactive DCOP algorithms that take them into account when searching for solutions [Hoang *et al.*, 2016; 2017].

3.6 Preference Elicitation for DCOPs

Existing DCOP algorithms typically assume that all constraint utilities are fully specified a priori as they are provided as part of the problem definition. However, this assumption does not hold in some applications such as smart home scheduling problems [Fioretto *et al.*, 2017a], where the utilities of some constraints represent user preferences and should thus be elicited from users. With this motivation in mind, we proposed the *preference elicitation problem for DCOPs*, which partitions the constraints into constraints with known utilities and constraints with unknown utilities that must either be elicited or approximated [Tabakhi *et al.*, 2017]. We then proposed several heuristics, including one based on minimax regret, to solve this problem.

Following up on this initial work, we also investigated the use of matrix completion algorithms to identify which subset of utilities to elicit in order to best approximate the remaining unelicited utilities [Le *et al.*, 2018]. Further, we model the cognitive bother cost of the user associated with providing the elicited utilities, and optimize the sequence of questions to ask users such that the total bother cost is within some predefined threshold. While both approaches are described for general models, they were evaluated in the context of smart home automation problems, where we show that they perform better than random baseline methods.

3.7 Variable-to-Agent Mappings for DCOPs

Finally, the conventional DCOP model assumes that the mapping of variables to agents is part of the model description and is thus given as an input. This assumption is reasonable in many applications where there are obvious and intuitive mappings. For example, in a smart home scheduling problem, agents correspond to the different smart homes, and variables correspond to the different smart devices within each home. In this case, the agent controls all the variables that map to the devices in its home. However, in other applications, there may be more flexibility in the mapping of variables to agents. For example, imagine an application where a team of *unmanned aerial vehicles* (UAVs) need to coordinate with each other to effectively survey an area. In this application, agents correspond to UAVs and variables correspond to the different zones in the area to be surveyed. The domain for each variable may correspond to the different types of sensors to be used and/or the different times to survey the zone. Since a UAV can survey any zone, there are multiple possible assignments of zones to UAVs (i.e., there are multiple possible mappings of variables to agents).

While choosing a good mapping is important as it can have a significant impact on an algorithm's runtime, choosing an optimal mapping may be prohibitively time consuming as this is an NP-hard problem [Rust *et al.*, 2016]. Considering these issues, coupled with the fact that this step is only a preprocessing step prior to the execution of the actual DCOP algorithm, we developed a generic heuristic-based algorithm that can be executed in a centralized or decentralized manner [Khan *et al.*, 2018]. At a high level, the algorithm uses Fortune's algorithm [Fortune, 1987] to partition the DCOP graph into p partitions, where p is the number of agents in the problem, in such a way where the number of high-degree nodes (i.e., variables that are in the scopes of large numbers of constraints) is balanced across all partitions. Surprisingly, our experimental results show that our heuristic-based approach finds mappings that result in runtimes that are only within 10% of the runtimes found with optimal mappings for GDL-based DCOP algorithms on random and scale-free graphs.

4 Conclusions and Future Directions

While the theoretical foundations and algorithmic improvements for conventional DCOPs have matured significantly over the past decade, their deployment to realistic applications is unfortunately lagging. To make this transition, we hypothesize that researchers will need to make further progress in generalizing the conventional DCOP model, adapting it to specific applications, and, most importantly, developing efficient algorithms for the corresponding extensions. Our efforts described in this paper are motivated by this belief. Other promising directions include investigations in the intersection of DCOPs and game theory, such as the work by Chapman *et al.* [2008], and the use of machine learning techniques for DCOPs, such as the work by Kumar and Zilberstein [2011] and Ghosh *et al.* [2015]. Additionally, orthogonal to anytime [Zivan *et al.*, 2014] and any-space algorithms [Petcu and Faltings, 2007; Yeoh *et al.*, 2009], any-communication algorithms (i.e., algorithms that adapt the number, size, and content of messages sent between agents based on the degree of congestion in the network) should also be of interest, especially in applications with potentially degraded communication channels.

Acknowledgments

The contributions highlighted in this paper are a result of collaborations with many other researchers. Special thanks go to

Ferdinando Fioretto, Khoi Hoang, Md. Mosaddek Khan, Tiep Le, Duc Thien Nguyen, and Atena M. Tabakhi, the student researchers who led their respective research thrusts. The author also gratefully acknowledges support from the National Science Foundation through grants 1345232, 1540168, and 1550662.

References

[Amigoni *et al.*, 2015] Francesco Amigoni, Andrea Castelletti, and Matteo Giuliani. Modeling the management of water resources systems using multi-objective DCOPs. In *Proceedings of AAMAS*, pages 821–829, 2015.

[Burke and Brown, 2006] David Burke and Kenneth Brown. Efficient handling of complex local problems in distributed constraint optimization. In *Proceedings of ECAI*, pages 701–702, 2006.

[Chapman *et al.*, 2008] Archie Chapman, Alex Rogers, and Nicholas Jennings. A parameterisation of algorithms for distributed constraint optimisation via potential games. In *Proceedings of the International Workshop on Distributed Constraint Reasoning*, 2008.

[Farinelli *et al.*, 2014] Alessandro Farinelli, Alex Rogers, and Nicholas Jennings. Agent-based decentralised coordination for sensor networks using the max-sum algorithm. *Journal of Autonomous Agents and Multi-Agent Systems*, 28(3):337–380, 2014.

[Fioretto *et al.*, 2014] Ferdinando Fioretto, Tiep Le, William Yeoh, Enrico Pontelli, and Tran Cao Son. Improving DPOP with branch consistency for solving distributed constraint optimization problems. In *Proceedings of CP*, pages 307–323, 2014.

[Fioretto *et al.*, 2016a] Ferdinando Fioretto, William Yeoh, and Enrico Pontelli. A dynamic programming-based MCMC framework for solving DCOPs with GPUs. In *Proceedings of CP*, pages 813–831, 2016.

[Fioretto *et al.*, 2016b] Ferdinando Fioretto, William Yeoh, and Enrico Pontelli. Multi-variable agent decomposition for DCOPs. In *Proceedings of AAAI*, pages 2480–2486, 2016.

[Fioretto *et al.*, 2017a] Ferdinando Fioretto, William Yeoh, and Enrico Pontelli. A multiagent system approach to scheduling devices in smart homes. In *Proceedings of AAMAS*, pages 981–989, 2017.

[Fioretto *et al.*, 2017b] Ferdinando Fioretto, William Yeoh, Enrico Pontelli, Ye Ma, and Satishkumar Ranade. A distributed constraint optimization (DCOP) approach to the economic dispatch with demand response. In *Proceedings of AAMAS*, pages 999–1007, 2017.

[Fioretto *et al.*, 2018a] Ferdinando Fioretto, Enrico Pontelli, and William Yeoh. Distributed constraint optimization problems and applications: A survey. *Journal of Artificial Intelligence Research*, 61:623–698, 2018.

[Fioretto *et al.*, 2018b] Ferdinando Fioretto, Enrico Pontelli, William Yeoh, and Rina Dechter. Accelerating exact and approximate inference for (distributed) discrete optimization with GPUs. *Constraints*, 23(1):1–43, 2018.

[Fortune, 1987] Steven Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2(1–4):153–174, 1987.

[Geman and Geman, 1984] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[Gershman *et al.*, 2009] Amir Gershman, Amnon Meisels, and Roie Zivan. Asynchronous Forward-Bounding for distributed COPs. *Journal of Artificial Intelligence Research*, 34:61–88, 2009.

[Ghosh *et al.*, 2015] Supriyo Ghosh, Akshat Kumar, and Pradeep Varakantham. Probabilistic inference based message-passing for resource constrained DCOPs. In *Proceedings of IJCAI*, pages 411–417, 2015.

[Giuliani *et al.*, 2014] Matteo Giuliani, Andrea Castelletti, Francesco Amigoni, and Ximing Cai. Multiagent systems and distributed constraint reasoning for regulatory mechanism design in water management. *Journal of Water Resources Planning and Management*, 41(4):04014068, 2014.

[Gutierrez *et al.*, 2011] Patricia Gutierrez, Pedro Meseguer, and William Yeoh. Generalizing ADOPT and BnB-ADOPT. In *Proceedings of IJCAI*, pages 554–559, 2011.

[Hoang *et al.*, 2016] Khoi D. Hoang, Ferdinando Fioretto, Ping Hou, Makoto Yokoo, William Yeoh, and Roie Zivan. Proactive dynamic distributed constraint optimization. In *Proceedings of AAMAS*, pages 597–605, 2016.

[Hoang *et al.*, 2017] Khoi D. Hoang, Ping Hou, Ferdinando Fioretto, William Yeoh, Roie Zivan, and Makoto Yokoo. infinite-horizon proactive dynamic DCOPs. In *Proceedings of AAMAS*, pages 212–220, 2017.

[Khan *et al.*, 2018] Md. Mosaddek Khan, Long Tran-Thanh, William Yeoh, and Nicholas Jennings. A near-optimal node-to-agent mapping heuristic for GDL-based DCOP algorithms in multi-agent systems. In *Proceedings of AAMAS*, 2018.

[Kim and Lesser, 2013] Yoonheui Kim and Victor Lesser. Improved max-sum algorithm for DCOP with n-ary constraints. In *Proceedings of AAMAS*, pages 191–198, 2013.

[Kumar and Zilberstein, 2011] Akshat Kumar and Shlomo Zilberstein. Message-passing algorithms for quadratic programming formulations of MAP estimation. In *Proceedings of UAI*, pages 428–435, 2011.

[Kumar *et al.*, 2009] Akshat Kumar, Boi Faltings, and Adrian Petcu. Distributed constraint optimization with structured resource constraints. In *Proceedings of AAMAS*, pages 923–930, 2009.

[Kumar *et al.*, 2011] Akshat Kumar, William Yeoh, and Shlomo Zilberstein. On message-passing, MAP estimation in graphical models and DCOPs. In *Proceedings of the International Workshop on Distributed Constraint Reasoning*, pages 57–70, 2011.

[Le *et al.*, 2015] Tiep Le, Tran Cao Son, Enrico Pontelli, and William Yeoh. Solving distributed constraint optimization problems with logic programming. In *Proceedings of AAAI*, 2015.

[Le *et al.*, 2017] Tiep Le, Tran Cao Son, Enrico Pontelli, and William Yeoh. Solving distributed constraint optimization problems with logic programming. *Theory and Practice of Logic Programming*, 17(4):634–683, 2017.

[Le *et al.*, 2018] Tiep Le, Atena M. Tabakhi, Long Tran-Thanh, William Yeoh, and Tran Cao Son. Preference elicitation with interdependency and user bother cost. In *Proceedings of AAMAS*, 2018.

[Maheswaran *et al.*, 2004a] Rajiv Maheswaran, Jonathan Pearce, and Milind Tambe. Distributed algorithms for DCOP: A graphical game-based approach. In *Proceedings of PDCS*, pages 432–439, 2004.

[Maheswaran *et al.*, 2004b] Rajiv Maheswaran, Milind Tambe, Emma Bowring, Jonathan Pearce, and Pradeep Varakantham. Taking DCOP to the real world: Efficient complete solutions for distributed event scheduling. In *Proceedings of AAMAS*, pages 310–317, 2004.

[Modi *et al.*, 2005] Pragnesh 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.

[Mohr and Henderson, 1986] Roger Mohr and Thomas Henderson. Arc and path consistency revisited. *Artificial Intelligence*, 28(2):225–233, 1986.

[Nguyen *et al.*, 2013] Duc Thien Nguyen, William Yeoh, and Hoong Chuin Lau. Distributed Gibbs: A memory-bounded sampling-based DCOP algorithm. In *Proceedings of AAMAS*, pages 167–174, 2013.

[Nguyen *et al.*, 2014] Duc Thien Nguyen, William Yeoh, Hoong Chuin Lau, Shlomo Zilberstein, and Chongjie Zhang. Decentralized multi-agent reinforcement learning in average-reward dynamic DCOPs. In *Proceedings of AAAI*, pages 1447–1455, 2014.

[Ottens *et al.*, 2017] Brammert Ottens, Christos Dimitrakakis, and Boi Faltings. DUCT: An upper confidence bound approach to distributed constraint optimization problems. *ACM Transactions on Intelligent Systems and Technology*, 8(5):69:1–69:27, 2017.

[Petcu and Faltings, 2005a] Adrian Petcu and Boi Faltings. A scalable method for multiagent constraint optimization. In *Proceedings of IJCAI*, pages 1413–1420, 2005.

[Petcu and Faltings, 2005b] Adrian Petcu and Boi Faltings. Superstabilizing, fault-containing multiagent combinatorial optimization. In *Proceedings of AAAI*, pages 449–454, 2005.

[Petcu and Faltings, 2007] Adrian Petcu and Boi Faltings. MB-DPOP: A new memory-bounded algorithm for distributed optimization. In *Proceedings of IJCAI*, pages 1452–1457, 2007.

[Rust *et al.*, 2016] Pierre Rust, Gauthier Picard, and Fano Ramparany. Using message-passing DCOP algorithms to solve energy-efficient smart environmentconfiguration problems. In *Proceedings of IJCAI*, pages 468–474, 2016.

[Tabakhi *et al.*, 2017] Atena M. Tabakhi, Tiep Le, Ferdinando Fioretto, and William Yeoh. Preference elicitation for DCOPs. In *Proceedings of CP*, pages 278–296, 2017.

[Ueda *et al.*, 2010] Suguru Ueda, Atsushi Iwasaki, Makoto Yokoo, Marius Silaghi, Katsutoshi Hirayama, and Toshihiro Matsui. Coalition structure generation based on distributed constraint optimization. In *Proceedings of AAAI*, pages 197–203, 2010.

[Vinyals *et al.*, 2011] Meritxell Vinyals, Juan Rodríguez-Aguilar, and Jesús Cerquides. Constructing a unifying theory of dynamic programming DCOP algorithms via the generalized distributive law. *Journal of Autonomous Agents and Multi-Agent Systems*, 22(3):439–464, 2011.

[Yeoh and Yokoo, 2012] William Yeoh and Makoto Yokoo. Distributed problem solving. *AI Magazine*, 33(3):53–65, 2012.

[Yeoh *et al.*, 2009] William Yeoh, Pradeep Varakantham, and Sven Koenig. Caching schemes for DCOP search algorithms. In *Proceedings of AAMAS*, pages 609–616, 2009.

[Yeoh *et al.*, 2010] William Yeoh, Ariel Felner, and Sven Koenig. BnB-ADOPT: An asynchronous branch-and-bound DCOP algorithm. *Journal of Artificial Intelligence Research*, 38:85–133, 2010.

[Yeoh *et al.*, 2015] William Yeoh, Pradeep Varakantham, Xiaoxun Sun, and Sven Koenig. Incremental DCOP search algorithms for solving dynamic DCOPs. In *Proceedings of IAT*, pages 257–264, 2015.

[Yokoo, 2001] Makoto Yokoo, editor. *Distributed Constraint Satisfaction: Foundation of Cooperation in Multi-agent Systems*. Springer, 2001.

[Zhang *et al.*, 2005] Weixiong Zhang, Guandong Wang, Zhao Xing, and Lars Wittenberg. 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.

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

[Zivan *et al.*, 2014] Roie Zivan, Steven Okamoto, and Hilla Peled. Explorative anytime local search for distributed constraint optimization. *Artificial Intelligence*, 212:1–26, 2014.

[Zivan *et al.*, 2015] Roie Zivan, Harel Yedidsion, Steven Okamoto, Robin Glinton, and Katia Sycara. Distributed constraint optimization for teams of mobile sensing agents. *Journal of Autonomous Agents and Multi-Agent Systems*, 29(3):495–536, 2015.