Levels of Cooperation and Equality in Multi-Agent Local Search

Lior Blecher, Yuval Siman-Tov Levy

Lecturer: Nimrod Talmon, Ph.D

07.08.22

Abstract

In a Multi-Agent System (MAS), autonomous agents interact to pursue personal interests and/or to achieve common objectives. Distributed Constraint Optimization Problems (DCOPs) in this regard, have emerged as a prominent agent model to govern the agents' autonomous behavior, where both algorithms and communication models are driven by the structure of the specific problem. A recent paper [19] offers a distributed local search algorithm in order to solve an Asymmetric-DCOP (ADCOP) where agents are partially cooperative (as opposed to the commonly used fully-cooperative and self-interested agents). Based on this approach, we propose a novel implementation of these three heuristics, inspired by natural human interactions, exhibiting an efficient message transferring mechanism between agents. Our local search algorithm outperforms two popular heuristics when measuring the sum of gains, and shows competitive results when measuring the equality in the system. This study thus contributes to the understanding of partial cooperation heuristics against the classic approaches and provides interesting insights with regard to an equality objective.

I Introduction

Many real-world problems are decentralized by nature, making a distributed solution a wiser more convenient choice of action. There are many examples from both natural and artificial systems that show that a composite system consisting of several subsystems can reduce the total complexity of the system while solving a difficult problem satisfactorily [6]. Over the past few decades, researchers in the Artificial Intelligence (AI) community have studied methods for coping with such challenging settings. The collection of which is called Distributed Artificial Intelligence (DAI).

DAI is a subfield AI devoted to researching and developing those distributed solutions through studying interaction of intelligent and distributed agents (au-

tonomous entities performing a task). Because of the decentralized and distributed nature of this field, it is making AI more reachable and scalable, gaining much attention in recent times. DAI is the predecessor of the field of Multi-Agent Systems (MAS).

Distributed Constraint Optimization Problems (DCOP) are a framework that recently emerged as one of the most successful approaches to coordination in MAS [13]. Informally, in DCOP frameworks, the problem is expressed as a set of individual sub-problems, each owned by a different agent. Each agent's sub-problem is connected with some of the neighboring agents' sub-problems via constraints over shared variables and the goal is to find a globally optimal solution. Agents can release information only through message exchange among agents that share relevant information, according to a specified algorithm.

For solving DCOPs, plenty of algorithms have been proposed [9, 16, 12, 8, 1] and they can generally be grouped into two categories: incomplete and complete. Complete algorithms can guarantee to find the optimal solution explore the entire solution space by employing distributed searches and dynamic programming. However, since DCOPs are NP-Hard, complete algorithms exhibit an exponentially increasing communication or computation to ensure the optimal solution. Thais makes them impracticable for large real-world problems. Thus, considerable research efforts have been made to develop incomplete algorithms which can find a sub-optimal solution at much smaller coordination overheads and be well applied to large-scale practical distributed applications.

Our study focuses on local search algorithms. In typical local search algorithms, agents keep exchanging assignments with their neighbors (two agents can communicate only if they are neighbors), and then decide whether to replace their value assignments to optimize their individual benefits in terms of their constraints and the received values from their neighbors. The general design of most local search algorithms for DCOPs is synchronous. Per step of the algorithm an agent sends its value assignment to all its neighbors in the constraint network and receives the value assignment of all its neighbors.

The rest of the project is organized as follows. In Sect. II, we briefly review related work. In Sect. III, the DCOP description, including other relevant definitions, are introduced. Section IV illustrates the details of our proposed message exchange algorithm (i.e., the SM agent). In Sect. V, an empirical evaluation is presented to evaluate the heuristics in comparison with classic existing methods. Finally, Sect. VI concludes this paper and gives the future work.

II Related Work

DCOPs help solve optimization problems where the relevant problem-solving knowledge (typically represented in the form of constraints) is distributed across a collection of agents [18]. DCOP techniques are particularly useful in settings in which business-sensitive knowledge cannot be centralized or where privacy con-

cerns prevent agents from sharing their knowledge with a central optimization solver. Many application problems in multi-agent systems can be formalized as DCOPs, in particular, distributed resource allocation problems including meeting scheduling [7], sensor networks [5], and synchronization of traffic lights [4]. Some of the better known DCOP techniques include ADOPT [9], DPOP [11], ODPOP [12] among others.

Since DCOPs are NP-hard, many recent studies apply local search algorithms [20, 21, 10, 14], in which the goal is finding a complete assignment to the decision variables of interest that not only satisfies the problem constraints but also aspire to optimize the relevant objective(s) [17]. Per iteration, an agent considers whether to change the assignment of its decision variable, however doing this affects not only its own utility but also the utilities of his neighbors. Asymmetric DCOP (ADCOP) [3] extends the basic framework to a more realistic scenario in which different agent evaluate differently various states of the world, allowing the cost of each constraint to be different for different agents.

Most studies investigating multi agent systems consider either **self-interested** agents ("*Egoists*") which are driven by their own goals and interest thus considered to be purely rational when they take actions [2] or **fully-cooperative** agents ("*Altruists*") which are fully compliant in aiming to maximize the public-welfare (even on the expense of their own) [9, 15]. A notable exception to this two-class division is a **partially-cooperative** model [22] which represents the willingness of agents to cooperate by defining thresholds of minimum requirements for cooperation, i.e., agents are willing to perform actions that may result in more utility for the group and less utility for themselves, as long as some minimum on their personal utility is preserved.

The recent paper [19], which is a precursor to this study, is able to generalize upon this approach, introducing a model in which agents' cooperation is based on the amount of utility they gain or lose, thus adjusting the partial cooperative model to realistic social behavior with a more dynamic nature of intentions rather than a fixed reference point.

We view all of these previous approaches to be extremely valuable and consider our proposed heuristics complimentary to these efforts.

III Background

In this section, we formally define the problem we aim to address and other relevant definitions:

Problem Setup

Distributed Constraint Optimization Problems

DCOP is a tuple $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{R} \rangle$ where:

- $\mathcal{A}$ is a finite set of agents $\{a_1, \ldots, a_n\}$.

- $\mathcal{X}$ is a finite set of variables $\{x_1, \ldots, x_m\}$, each variable is held by a single agent (an agent may hold more than one variable).
- $\mathcal{D}$ is a set of domains $\{D_1, \ldots, D_m\}$, each domain D_i contains a finite set of values which can be assigned to variable x_i.
- $\mathcal{R}$ is a set of relations (constraints). Each constraint $U \in \mathcal{R}$ defines a non-negative utility for every possible value combination of a set of variables, and is of the form $U\colon D_{i1} \times D_{i2} \times \ldots \times D_{ik} \rightarrow \mathbb{R}^+ \cup \{0\}$.

For simplicity, we also assume that each agent holds exactly one variable (unless stated otherwise). This assumption is commonly made in DCOP studies, e.g., [9]. Without loss of generality, a solution to a DCOP is an assignment to all variables that maximizes the total utility. Thus, it can be formalized as:

$$\mathcal{X}^* = \underset{v_i \in D_i v_j \in D_j}{\arg\max} \sum_{u_{ij} \in \mathcal{U}} u_{ij}(x_i = v_i, x_j = v_j) \tag{1}$$

Of note, here the term "agent" and "variable" can be used interchangeably.

Asymmetric DCOP

ADCOPs generalize DCOPs in the following manner: instead of assuming equal gains for constrained agents, the ADCOP constraint explicitly defines the exact gain of each participant. That is, domain values are mapped to a tuple of utilities, one for each constrained agent. More formally, an ADCOP is defined by the following tuple $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{R} \rangle$ where $\mathcal{A}, \mathcal{X}$ and $\mathcal{D}$ are defined in exactly the same manner as in DCOPs. Each constraint $U \in \mathcal{R}$ of an asymmetric DCOP defines a set of non-negative utility for every possible value combination of a set of variables, and takes the following form:

$$U\colon D_{i1} \times D_{i2} \times \ldots \times D_{ik} \rightarrow \prod_{j=1}^{k} \mathbb{R}^+ \cup \{0\} \tag{2}$$

Partial Cooperation

Setting a minimal threshold defines the extent in which agents are willing to cooperate. The product of the following determines that threshold:

- λ_i - The *bound* of agent i. Limits the lose it is willing to undertake in order to contribute to the global objective.
- μ_i - The *baseline* of agent i. A scalar valued function of previous utilities obtained by agent i, operates as a point of reference for future expected utility, the utility the agent assumes it "deserves".)

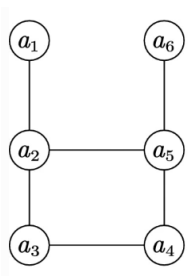
The Socially-Motivated Agent

A Socially-Motivated (SM) agent is an implementation of the Partial-Cooperation framework which also possess the following attributes:

- Vote -Allows an agent to direct its neighbors towards a specific value from their domain that it wishes them to choose.
- Taboo - Upon realizing the threshold might potently be breached, Taboo messages allow an agent to forbid its neighbor(s) from taking a specific assignment in the next iteration.

Basic Definitions

Definition 1. DCOP can be visualized as a constraint graph where the nodes represent the agents and the edges denote the constraints. Figure 1 is an example of a DCOP whose constraint graph and constraint matrixes are shown in Fig. 1a, b, respectively.



(a) DCOP graph

x_1 \ x_2	0	1
0	4	7
1	7	2

x_3 \ x_4	0	1
0	3	4
1	4	3

x_2 \ x_3	0	1
0	3	5
1	5	2

x_4 \ x_5	0	1
0	5	7
1	7	2

x_5 \ x_6	0	1
0	4	8
1	8	3

x_2 \ x_5	0	1
0	3	5
1	5	1

(b) payoff tables

Figure 1: DCOP instance.

Definition 2. The ***graph density*** of simple graphs is defined to be the ratio of the number of edges $|E|$ with respect to the maximum possible edges:

$$D = \frac{|E|}{\binom{|V|}{2}} = \frac{2|E|}{|V|(|V|-1)} \tag{3}$$

Definition 3. An important feature of most local search algorithms is that they hold the best assignment that was found throughout the search. This makes them ***anytime algorithms***, i.e., the quality of the solution can only increase or remain the same if more iterations of the algorithm are performed.

Definition 4. A ***Pareto optimal solution*** is a state where it is not possible to improve the situation of an agent without deteriorating that of at least one other.

Definition 5. Measurements for statistical dispersion in a distribution:

- The Lorenz Curve - A graphical representation of the distribution of wealth. It is often used to represent income distribution, where it shows for the bottom x% of population, what percentage (y%) of the total income they have.
- The Gini index - Based on the Lorenz curve, it is the ratio of the area that lies between the line of equality, and the Lorenz curve over the total area under the line of equality. The Gini index measures the inequality in a population.

IV Method

Algorithm

Set in the above backdrop, the algorithm proposed hereafter distills insights from [19] and aspires to contribute for a more thorough understanding of partially-cooperative agent-based systems. As in many computational intelligence algorithms inspired by biology and natural systems, this work is greatly inspired by human communication and information sharing in society, mimicking a realistic decision-making process.

Figure 2: The decision-making process per phase.

Depicted in figure 2 is a series of transitions between "listen" and "reply" states, where prior to each shift there's a "thinking" stage where the agent processes the input and makes a calculated decision regarding its next step This cycle is constantly repeating throughout the run. The "thinking" stage is different for each heuristic, i.e., although possibly given the same input, each may still reach a different conclusion as to what action should be taken next. The three heuristics that will be compared are: the Altruistic agent, Egoistic agent and SM agent. Our basic algorithm consists of 4 phases, between each of whom each agent passes the following information to its neighbors: (1) between the first phase and the second phase **preferences messages** are delivered, containing requests for the neighbors' next assignment. (2) between the second phase and the third phase, **alternative-value messages** are delivered, comprising of the agent's intentions regarding its own next assignment. (3) between the third phase and the forth, **taboo messages** are delivered, containing the agent's restrictions regarding its neighbors' suggestions for their next assignment. (4) between the final phase and the first one of the next iteration, **value messages** are delivered

with the agent's chosen new current assignment.

Initialization

In this step each agent: (1) learns who are its neighbors, (2) attains its constraints with each neighbor, (3) starts with a random initialization of its assignment and (4) sends it to the neighbors with a *value message*. (5) the SM agent will also receives its starting bound, which defines its "characteristics" - this bound is represented by a percentage in the range of 5%-50%, the smaller the bound the less cooperative the agent is. All the phases below are described form a single agent point of view, but apply to all (depending on their type):

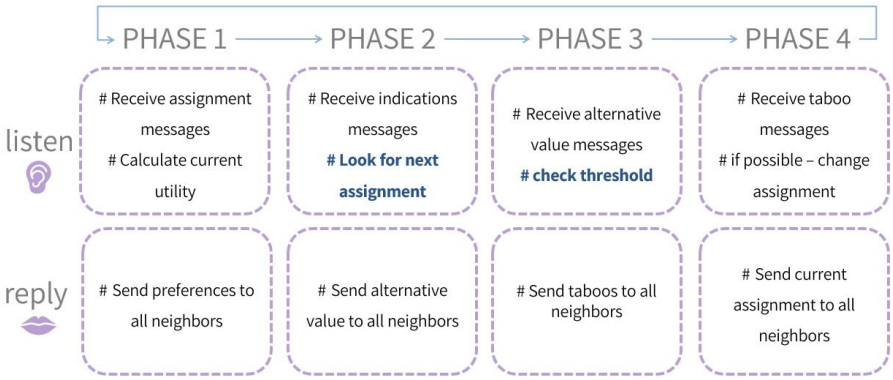


Figure 3: Algorithm's general workflow

Phase 1

All agents update their "local-view": the value of the assignment of all their neighbors, along with a calculation of their own utility accordingly. Only the SM agent has another step here - it updates its baseline, calculated as the average of the previous utility and the current utility. Following these updates each agent communicates its preferences to its peers (*preference messages*), possibly requesting them to change assignment so its own utility is maxed, in accordance to the constraints matrices.

Phase 2

After receiving *preference messages* from all neighbors, each agent conducts a different judgement call as to its next step - specifically, which request from its peers regarding its next assignment, should be accepted.

- Egoists - Ignores all the received messages, acting solely to increase its own utility in a pure greedy way.
- Altruists - The Altruist agent uses the indications received from its neighbors to calculate the change in the group's utility, i.e. the "social gain", for each possible value in the domain. This is calculated by summing its neighbors' gains and loses caused by each potential change. The agent then chooses the option with the maximal social gain.
- Socially Motivated - The SM agent will convert the indications to votes i.e. which value in the domain each neighbor would prefer the most, and multiply it with the "social gain", including its own preferences in the process. It then chooses the option with the highest rate.

The phase ends with each agent sending its *alternative value messages* to its peers.

Phase 3

Consequently *alternative value messages* are received.

- Socially Motivated - will go through their neighbors' suggestions for alternative value, and check if their thresholds will be breached by that move. reassured that the neighbors will respect the restriction. The threshold is calculated by: $\mu_i(1-\lambda_i)$ if the potential utility is lower than this calculation, a taboo message is sent.
- Egoists - will not send taboo messages, as they are selfish and will not comply to their neighbors' restrictions.
- Altruists - will not send taboo messages either, as they are willing to undertake any lose.

Phase 4

Starting with receiving all the taboo messages (if sent by neighbors), the agent is now changing assignments (given none has sent any on its chosen next assignment. Should there is at least one, no change is made). Only the SM agent has another step here - it gradually decrease its bound by 5×10^{-3} per iteration. Since a total of 1000 iterations are repeated, the bound will eventually reach zero (that is inevitably true for all agents as the starting bound varies between 5×10^{-1} to 5×10^{-2} for all). This phase concludes with the new current assignment sent by the *value messages*, and starts from Phase 1 all over again.

V Results

Experimental Setup

In order to evaluate the performance of the different heuristics, we compared their performance on a total of 100 random DCOPs problems:

- Size - Each problem consists of 50 agents (n = 50), each holding a single variable with 10 values in its domain ($|D| = 10$).
- Constraints - each cell in the constraints matrices was sampled from a uniform distribution 0, 100 using a stochastic generator.
- density - on par with the common literature, results were compared on both sparse and dense graphs, of density equals to 0.2 and 0.7 respectively. Hence, each heuristics was run on 50 problems with average of 10 neighbors per agent, and on 50 problems with average of 35 neighbors per agent.
- running time - duration was capped by 1000 iterations.

Variations in bound values

Compared to [19], different bound values were tested in order to reach the optimal taboo message strategy. Of note, the smaller the λ gets, the less an agent is willing to compromise its own utility to the benefit of the greater good, allowing less exploration to the MAS. The following λs were considered: (1) $\lambda = 1$, i.e. no-taboo messages at all, becoming more "generous". (2) $\lambda = 0$, here agents are not willing to compromise their utility at all, making them more "selfish", thus sending many taboos. Under this bound, every improvement in the system is necessarily a Pareto improvement. (3) λ is gradually decreasing by 1×10^{-3} per iteration, making the SM agents less and less cooperative. Since the initial bound varies between 5% and 50%, some agents indeed reach a bound of 0 but some are still willing to cooperate to some extent.

Evaluation

Altogether six types of simulations were curried out: 3 heuristics $\times$ 2 density levels, each of 50 problems. Results were averaged on every group of simulations. Two main performance measurements were used for evaluation: (1) the global sum of utilities, representing the overall social welfare of the system, (2) and the equality level in it as measured by the Gini index. The process of choosing bound value:

- $\lambda = \mathbf{1}$ - Resulted in a non stable progress of exploration, while reaching inferior solution to the other two heuristics, i.e. less social welfare.
- $\lambda = \mathbf{0}$ - presented a broken line, showing success of improving the global utility over the other two heuristics, however converges very early on, not

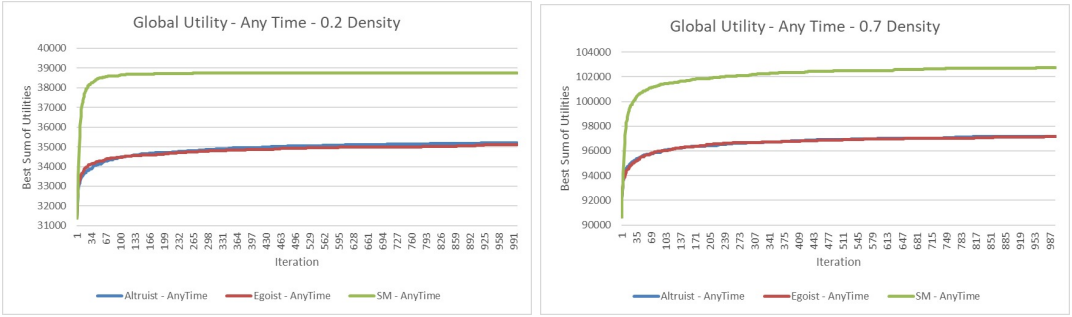
able to further investigate the solution's space, getting "stuck" with a not so impressive result.

- λ **is gradually decreasing by** 1×10^{-3} **per iteration** - presenting significant improvement on the algorithm performance, gaining advantage over the other two heuristics.
- λ **is gradually decreasing by** 5×10^{-3} **per iteration** - Resulted with the highest social welfare solution, showing more exploration in early stages and converging towards the end.

Depicted in table 1 is a comparison of results for the 3 heuristics tested. We direct the reader to the fact that dense graphs exhibit better results than sparse graph, for all 3 heuristics, both for the equality measurement and for the global utility gain (the final global utility compared to the starting point of the run), this may be the result of a "richer" environment (more connections on average result in more "resources to allocate"). Furthermore, more neighbors per agent allow a more equal distribution of the load.

	Sparse Graph		Dense Graph	
	Gini Index	Golbal Utility Gain	Gini Index	Golbal Utility Gain
Egoist	0.207	3,715	0.123	6,500
Altruist	0.224	3,813	0.132	6,531
SM	0.202	7,351	0.125	12,090

Table 1: Equality measurements and changes in global utilities.

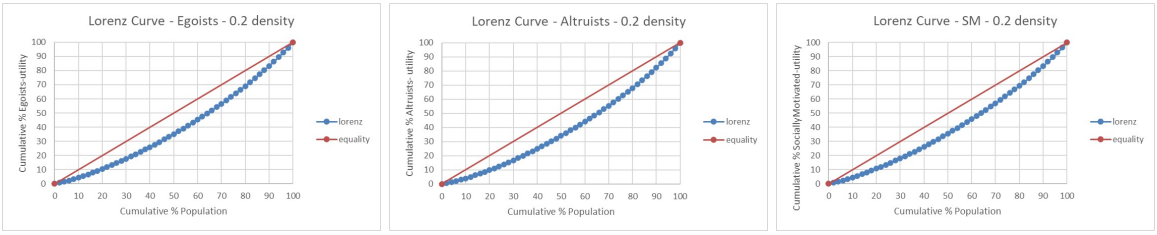


(a) Sparse Graphs (b) Dense Graphs

Figure 4: Anytime global utility comparison for both sparse and dense graphs.

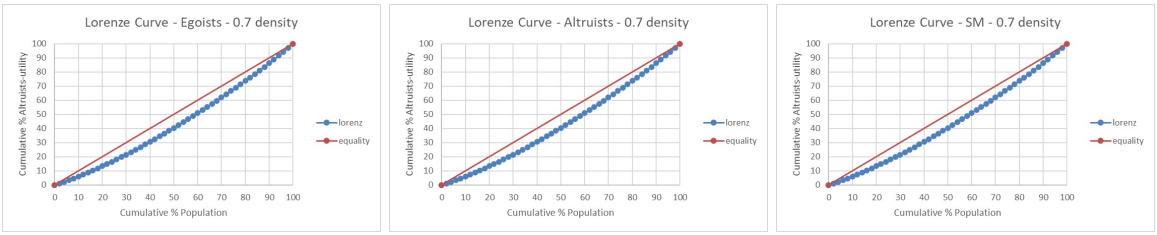
Discussion of Findings

Our results indicate a better heuristic than the classic approaches is obtained in conjunction with ensuring the final solution is in consensus by all agents. Our prior assumption regarding the SM agents' performance was that in exchange for a more "considerate" solution, the social welfare will decrease. However, in contrast to our prior speculation, the SM based local search did not result in a lower global utility nor in a higher equality measures compared to the other models. As for the reason the equality measures, i.e, the *Gini-Index*, results show similar inequality levels among all types of agents, even-though we initially presumed the SM model will involve less inequality (since the agents confirm they are all satisfied with every transition). This seems to be because each agent's approval regards only to the change in utility and not the utility as a whole. Moreover, each agent can compere its state only to its previous states, as it is not aware of his neighbors' utilities. In other words, the agent's point of view is dependent on its starting point, causing similar division of utilities compared to the other two heuristics.



(a) Egoist Agents' Lorenz curve (b) Altruist Agents' Lorenz curve (c) Socially Motivated Agents' Lorenz curve

Figure 5: A graphical representation of the distribution of utilities within each population. The closer the curve gets to the diagonal, the more equality resides. Results for sparse graphs.



(a) Egoist Agents' Lorenz curve (b) Altruist Agents' Lorenz curve (c) Socially Motivated Agents' Lorenz curve

Figure 6: Same description as figure 5. Results for dense graphs.

VI Conclusions

We study distributed constraint optimization problems in partially-cooperative multi-agent systems. We introduced a novel implementation of the Socially-Motivated agent, a local search algorithm inspired by human communication and information sharing in society, mimicking a realistic decision-making process. The whole project is divided into two parts: a theoretical part where the distributed constraint optimization problem was discussed and empirical section where our new method was presented. To the best of our knowledge we are the first to address the issue of system equality in utilities comparing it in commonly used heuristics. Our method demonstrates superb performance through an extensive set of problems with higher utility gains and equally measured utility dispersion compared to the conventional approaches. Our main contributions are as follows: (1) changing bound values as opposed to fixed bounds result in better performance (2) decreasing λ to 0 causes Pareto improvement towards the end of the simulation (3) taking under consideration expectations of gains as well as votes. (4) The algorithm is not monotonic allowing two neighbors to simultaneously change assignments.

We believe this work opens a new door to a broad range of research directions to be pursued, amongst many we advocate generalizing the algorithm to simultaneously account for both the equality of utilities and maximizing it, in the form Multi-Objectives Optimization. Furthermore, as our experiments involved homogeneous environments alone more types could be introduced in future work including interactions between different types of agents.

References

[1] Amir Gershman, Amnon Meisels, and Roie Zivan. Asynchronous forward bounding for distributed cops. *Journal of Artificial Intelligence Research*, 34:61–88, 2009.

[2] John Grant, Sarit Kraus, Michael John Wooldridge, and Inon Zuckerman. Manipulating boolean games through communication. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[3] Tal Grinshpoun, Alon Grubshtein, Roie Zivan, Arnon Netzer, and Amnon Meisels. Asymmetric distributed constraint optimization problems. *Journal of Artificial Intelligence Research*, 47:613–647, 2013.

[4] Robert Junges and Ana LC Bazzan. Evaluating the performance of dcop algorithms in a real world, dynamic problem. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pages 599–606, 2008.

[5] Victor Lesser, Charles L Ortiz Jr, and Milind Tambe. *Distributed sensor networks: A multiagent perspective*, volume 9. Springer Science & Business Media, 2003.

[6] Nancy A Lynch. *Distributed algorithms*. Elsevier, 1996.

[7] Rajiv Maheswaran, Milind Tambe, Emma Bowring, Jonathan Pearce, and Pradeep Varakantham. Taking dcop to the real world: Efficient complete solutions for distributed event scheduling. 2004.

[8] Roger Mailler and Victor R Lesser. Asynchronous partial overlay: A new algorithm for solving distributed constraint satisfaction problems. *Journal of Artificial Intelligence Research*, 25:529–576, 2006.

[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] Jonathan P Pearce and Milind Tambe. Quality guarantees on k-optimal solutions for distributed constraint optimization problems. In *IJCAI*, pages 1446–1451, 2007.

[11] Adrian Petcu and Boi Faltings. Dpop: A scalable method for multiagent constraint optimization. In *IJCAI 05*, number CONF, pages 266–271, 2005.

[12] Adrian Petcu and Boi Faltings. Odpop: An algorithm for open/distributed constraint optimization. In *AAAI*, volume 6, pages 703–708, 2006.

[13] Marc Pujol-Gonzalez. Multi-agent coordination: Dcops and beyond. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[14] Ruben Stranders, Alessandro Farinelli, Alex Rogers, and Nick Jennings. Decentralised control of continuously valued control parameters using the max-sum algorithm. 2009.

[15] Ruben Stranders, Alessandro Farinelli, Alex Rogers, and Nick Jennings. Decentralised coordination of mobile sensors using the max-sum algorithm. 2009.

[16] 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.

[17] Makoto Yokoo and Katsutoshi Hirayama. Algorithms for distributed constraint satisfaction: A review. *Autonomous Agents and Multi-Agent Systems*, 3(2):185–207, 2000.

[18] Makoto Yokoo, Toru Ishida, Edmund H Durfee, and Kazuhiro Kuwabara. Distributed constraint satisfaction for formalizing distributed problem solving. In *1992 12th International Conference on Distributed Computing System*, pages 614–615. IEEE Computer Society, 1992.

[19] Tal Ze'evi, Roie Zivan, and Omer Lev. Socially motivated partial cooperation in multi-agent local search. In *IJCAI*, pages 583–589, 2018.

[20] 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.

[21] Roie Zivan. Anytime local search for distributed constraint optimization. In *AAAI*, pages 393–398, 2008.

[22] Roie Zivan, Alon Grubshtein, Michal Friedman, and Amnon Meisels. Partial cooperation in multi-agent search. In *AAMAS*, pages 1267–1268, 2012.