Search on Asymmetric DCOPs by Strategic Agents

1st Yair Vaknin
Ben Gurion University
Be'er Sheva, Israel
yairva@post.bgu.ac.il

2nd Amnon Meisels
Ben Gurion University
Be'er Sheva, Israel
am@bgu.ac.il

***Abstract*—Asymmetric Distributed Constraint Optimization Problems (ADCOPs) are a useful model for representing real-life problems of distributed nature. Constraining agents in ADCOPs have different gains (or costs) for the constraints that involve them. All former ADCOP search algorithms assume cooperation among the agents and do not capture the possibility of strategic behavior by the searching agents. The present paper extends a recent approach that uses side payments among constraining agents in ADCOP local search, and proposes an improved such algorithm for strategic agents. Enabling search for strategic agents is especially suitable for asymmetric DCOPs, where the agents gain differently from the constraints and would naturally pursue personal gains.**

The proposed method uses a specially designed mechanism that enforces truthful behavior for agents placing bids of side payments during search. This in turn guarantees that the (strategic) agents' bids will form bids of maximal payoffs. The resulting search algorithm is an anytime algorithm that converges to stable solutions of higher social welfare that are local optima of the global social welfare, and computes the payments (contracts) that stabilize its outcome as a pure Nash equilibrium (PNE). The experimental evaluation shows that the payments charged by the mechanism in order to enforce truthful behavior are small compared to the total increase in the social welfare, and that the vast majority of the agents have improved personal gains when the algorithm terminates.

***Index Terms*—distributed constraints optimization, side payments, efficient PNEs**

I. Introduction

The Distributed Constraint Optimization Problem (DCOP [18], [20], [29], [30]) is a general model for solving real-life problems that are distributed by nature and cannot or should not be solved centrally [8], [35]. DCOPs are composed of agents, each holding one or more variables with a finite domain of possible value assignments. Constraints among variables (possibly held by different agents) assign costs to combinations of value assignments. Agents assign values to their variables and communicate with each other, attempting to generate a solution that is globally optimal with respect to the costs of the constraints.

The model of Asymmetric DCOPs (ADCOP [9]) generalises DCOPs and accommodates the common situation where agents have different costs for mutual assignments profile. Instead of assuming equal payoffs for constrained agents, every ADCOP constraint explicitly defines the exact payoff for each participant [9], [11]. That is, assignment profiles are mapped to a tuple of costs, one for each constrained agent. ADCOPs are naturally represented by graphs in which agents are represented by nodes and each constraint between two agents is represented by an edge on the graph. Since a constraint among two agents is a bi-matrix assigning the cost (or gain) of each of the constrained agents for every assignment combination, one can think of a constraint as a normal form game played by the constrained agents. This family of games has been termed *graphical games* [13].

Since ADCOPs are NP-hard, different incomplete algorithms were proposed for solving them ([9], [27], [34]). Recently, an innovative approach to local search on ADCOPs was proposed, that uses the (multi-agent) game-like nature of these problems [15]. This new family of search algorithms uses side payments contracted and exchanged among constraining agents. As a result of using these incentives, the algorithms can guarantee finding solutions of higher global social welfare. The Bidding Enhanced Efficiency Contracts (BEECon [26]) algorithm is the most recent member of this family, and was experimentally shown to be the best performing local search ADCOP algorithm. Of particular interest is the fact that since ADCOPs are analogous to games on networks, the BEECon algorithm guarantees the finding of *stable solutions* (i.e., pure Nash equilibria - PNEs) of higher efficiency. However, BEECon and all former ADCOP algorithms assume cooperative behavior of the participating agents and ignore the possibility of strategic (i.e., selfish) agents that may prefer their own utility over the global social welfare. Such an agent may misreport its private information and undermine the stability and efficiency guarantees of algorithms that use side payments among agents [15], [26].

The present paper proposes the integration of a *truthful bidding mechanism* into the search process of BEECon. The resulting search is robust to simple strategic manipulations by the participating agents, and guarantees all the advantages of BEECon. Among other guarantees, it converges to a local maximum of the social welfare.

Section II describes shortly the use of side payments in ADCOP search and focuses on the BEECon algorithm. The required preliminaries on mechanisms are introduced in Section III and Section IV presents the integration of a bidding mechanism into the BEECon algorithm. A preliminary experimental evaluation of the proposed method and algorithm is in Section V. The experimental results show that the payments taxed under the proposed algorithm in order to stabilize its outcome are small compared to the increase in global social welfare, and that most agents are better off upon termination of the algorithm.

II. Local Search with Side Payments on ADCOPs

All search algorithms on DCOPs and ADCOPs assume cooperation by the participating agents [16], [18]–[20], [29], [30] and for local search algorithms this takes the standard form of synchronous steps in which agents form a decision about their next assignment by consulting their neighbors (e.g., their constraining agents). In the MGM (Maximum Gain Message) algorithm agents exchange messages with their neighbors, report their expected gains from potential assignments among neighbors, and select the largest improvement among all possibilities [17]. In the DSA (Distributed Stochastic Algorithm) algorithm the agents select their assignment randomly [32]. Other local search algorithms use different selection criteria, but all of them share information completely [12]. The same cooperation is assumed also when the problems become asymmetric (e.g., ADCOPs) [7], [9], [34].

Recently, an innovative class of local search algorithms for ADCOPs was proposed, that uses side payments among neighboring agents [15], [26]. By using side payments agents can incentivise their neighbors to select some preferred assignment. Since asymmetric DCOPs have constraints among agents that are analogous to normal form games, one can think of ADCOPs as multi-agent games [9], [11], [15]. This new class of local search algorithms can be looked at as an extension of the best response mechanism [21] in multi-agents games, that guarantees convergence to stable solutions for general games and not just potential games [21]. The next subsection presents the currently best ADCOP local search algorithm, its use of side payments among the agents and its efficiency and stability guarantees.

A. Side payments in ADCOP search - the BEECon algorithm

The agents run BEECon algorithm in a fixed order - each agent in its turn (see Algorithm 1). The agent whose turn it is to act is termed the *current agent*. On its turn, the current agent i chooses a set of possible value assignment $V_i^s \subseteq V_i$ for its variable (line 2. For the rest of the paper it is assumed that $V_i^s = V_i$), and publishes V_i to its neighbors N_i (line 3). In response, every neighboring agent $j \in N_i$ calculates and offers a side payment $T_{j,i}^{v_i}$ that it offers the *current agent* for each possible assignment $v_i \in V_i$ (procedure **reply()**). The side payments $\{T_{j,i}^{v_i} \mid j \in N_i\}$ offered by the neighboring agents for an assignment v_i are considered *irreversible contracts* - to be paid only in case the current agent i chooses v_i as the assignment for its variable **upon termination** of the search process. If, however, the current agent changes its value assignment in any stage during the run of Algorithm 1 - the corresponding contract is nullified. The current agent considers all possible value assignments in V_i (line 7), sums up its own utility and the contracts proposed by its neighboring agents for each option, and selects the assignment that maximizes its own utility (lines 8-12). Finally, the current agent updates its neighbors about the chosen assignment, and it is now the turn of the next agent to act as the current agent.

Algorithm 1 Bidding Enhanced Efficiency Contracts

onTurn(i)
1: **let** $v_i \leftarrow$ the current strategy of agent i
2: choose $V_i^s \subseteq V_i$
3: **send**(**choice**, $V_i^s \cup \{v_i\}$) to all $j \in N_i$
4: **receive** $T_{j,i}^s$ **foreach** $s \in V_i^s \cup \{v_i\}$ from all $j \in N_i$
5: **let** $\mathbb{T} \leftarrow \sum_{j \in N_i} T_{j,i}^{v_i}$
6: $v_i' \leftarrow v_i$
7: **for all** $s \in V_i^s$ **do**
8:     **let** $T^s \leftarrow \sum_{j \in N_i} T_{j,i}^s$
9:     **if** $u_i(s, v_{-i}) + T^s > u_i(v_i, v_{-i}) + \mathbb{T}$ **then**
10:         $v_i' \leftarrow s$
11:         $\mathbb{T} \leftarrow T^s$
12:     **end if**
13: **end for**
14: **send**(**update**,v_i') to all $j \in N_i$

reply(choice, V_i^s**)**
1: **for all** $v_k \in V_i^s$ **do**
2:     **let** $T_{j,i}^{v_k} \leftarrow$ calculate_bid()
3: **end for**
4: **let** bids $\leftarrow [T_{j,i}^{v_1}, \ldots, T_{j,i}^{v_{|V_i^s|}}]$
5: **return** bids

Function **calculate_bid()** in procedure **reply** of Algorithm 1 is left undefined here. This function determines the bids (side payment) that a neighboring agent j places for each assignment proposed by the current agent i. In [26], the bids are *non-strategic* and are termed *maximal payoffs contracts*. In order to compute maximal payoffs the following method is used:

1) A neighboring agent j computes its utility $s_{v_k}^j = u_j(v_i, v_{-i})$ for each $v_i \in V_i$,
2) selects the "worst" value assignment $s_{min}^j = min\{s_{v_i} \mid v_i \in V_i\}$, that produces its lowest gain.
3) uses s_{min}^j as a reference point, and bids the difference $s_{v_i}^j - s_{min}^j$ for each $v_i \in V_i$.

The pseudo-code of computing maximal payoffs in [26] is given in 2. This type of bids is termed maximal payoffs contracts because a neighboring agent j that wants the current agent i to select some assignment v_i, is willing to sacrifice its entire relative benefit $s_{v_i}^j - s_{min}^j$ from the assignment. Obviously, a selfish (i.e., rational) agent j that tries to maximize its utility will attempt to gain something by placing a lower bid in most cases.

In the next subsection, the most important properties of the BEECon algorithm under maximal payoffs are listed from [26]. The challenge tackled by Section IV is the design and integration a bidding mechanism that enforces maximal payoffs on strategic agents and thereby achieves the guarantees of the BEECon algorithm for strategic agents.

Algorithm 2 Computing Maximal Payoffs Contracts

reply(choice, V_i^s**)**
for all $v_k \in V_i^s$ **do**
  let $s_{v_k} \leftarrow u_j(v_1, \dots v_k, \dots v_n)$
end for
let $s_{min} \leftarrow min\{s_{v_1}, \dots, s_{v_{|V_i|}}\}$
let bids $\leftarrow [s_{v_1} - s_{min}, \dots, s_{v_{|V_i|}} - s_{min}]$
return bids

B. Properties and guarantees of the BEECon algorithm

The central feature of BEECon (assuming maximal payoffs) is that it transforms a general ADCOP into an *exact potential game*. Maximal payoffs summed-up with the utility of the current agent (line 9) guarantee that when the current agent chooses strategy v_i, the change in its own utility - including all contracts - equals the change in the global social welfare (this and the following properties are fully proved in [26]). This fact implies that a current agent that chooses an assignment that maximizes the potential function, causes an increase in the social welfare. That is, BEECon is an *anytime* [33] local search algorithm. Moreover, maximal payoffs contracts guarantee that whenever there is an assignment-deviation of the current agent that can increase the social welfare, this assignment will be selected such that the algorithm converges to a local optimum of the social welfare. In other words, BEECon uniquely guarantees that upon convergence no single assignment deviation of any agent can improve the social welfare:

$$\forall i \in N : \nexists v' \in V_i \; s.t. \sum_{j \in N} u_j(v', v_{-i}) > \sum_{j \in N} u_j(v_i, v_{-i})$$

The above sample of properties and guarantees of the BEECon algorithm make it the best performing local search ADCOP algorithm [26]. Since these guarantees apply only under maximal payoffs, the next section turns to the task of designing a mechanism that will force strategic agents to propose maximal payoff bids during search.

III. A Bidding Mechanism for side payments during search

It is best to describe the goals of mechanisms by using the term of *social choice*. A social choice is an aggregation of the preferences of the different agents toward a single joint decision. *Mechanism design* attempts to implement a desired social choice in a strategic setting - assuming that the different members of society each act *rationally* in a game theoretic sense [22].

A main concern when designing a bidding mechanism is the possibility of *strategic bidding* by the agents. A strategic bid is one that tries to take advantage of the participants in the social choice action. Take for example a government that considers undertaking a public project of cost C (e.g., building a bridge), and let the government policy be: each citizen i declares the payment v_i it is willing to pay for the completion of the project. The project is accomplished if $\sum_i v_i \geq C$, and each citizen i is charged v_i. Such a straight forward policy might lead some citizens to free-ride and declare a payment of 0, hoping for the rest of the payments to exceed the cost of the project. In this simple example, free-riding can result in a social choice that is undesired by the government (and the citizens).

Given n players with $B_1, \dots, B_n$ bids for each of the alternatives in A, a *mechanism* is a function $f : B_1 \times \cdots \times B_n \rightarrow A$ and a vector of payment functions $p_1, \dots, p_n$, where $p_i : B_1 \times \cdots \times B_n \rightarrow \mathbb{R}$. Function f considers the bids of all agents and determines the social choice $a \in A$, and the payment functions $\{p_i \mid i \in N\}$ determine the payment of each agent.

In the context of an ADCOP local search algorithm, the players are the current agent i who proposes a set of assignments V_i for its variable, and its neighboring agents N_i. The alternatives are $A = \{(v, v_{-i}) \mid v \in V_i\}$, and the utility of a neighboring agent j from each alternative (v, v_{-i}) is $u_j(v, v_{-i})$. As BEECon uses bids for side payments (see Section II), strategic bidding for payments may harm the guarantees of either the efficiency or the stability of the algorithm. Consequently, a mechanism is needed to guarantee *truthful bidding* for payments. In other words, enforce the bidding agents to offer side payments similarly to the maximal payoffs bids of section II. Mechanisms that are designed to be truthful impose rules under which it is a dominant strategy for each agent to bid its true utility for every alternative. A well known example of a mechanism class that satisfies this requirement is the class of Vickrey-Clarke-Groves [5], [10], [28] (VCG) mechanisms.

A mechanism $(f, p_1, \dots, p_n)$ is a VCG Mechanism if it satisfies the following conditions:

- $f(b_1, \dots, b_n) \in argmax_{a \in A} \Sigma_i b_i(a)$; that is, f maximizes the (reported) social welfare.
- For some functions $h_1, \dots, h_n$, where $h_i : B_{-i} \rightarrow \mathbb{R}$, it holds that $p_i(b_1, \dots b_n) = h_i(b_{-i}) - \sum_{j \neq i} b_j(f(b_1, \dots, b_n))$. The main implication of this feature is that the payment a player is charged is independent of its own bid.

A common choice of h_i is $h_i = max_{a' \in A} \sum_{j \neq i} b_i(a')$. Such functions lead to the **Clarke pivot payments** [5] $p_i(a) = max_{a' \in A} \sum_{j \neq i} b_i(a') - \sum_{j \neq i} b_j(f(b_1, \dots, b_n))$. Clarke pivot payments are composed of two terms: $max_{a' \in A} \sum_{j \neq i} b_i(a')$ denotes the social welfare of all players except i if player i does not submit its bid, and the term $\sum_{j \neq i} b_j(f(b_1, \dots, b_n))$ denotes the utilities of the other players from the chosen alternative given the bid of agent i. Intuitively, player i pays the total damage it exerts on the other players [23], [25].

IV. Distributed Search by Strategic Agents

The pseudo-code for the proposed algorithm that uses a mechanism to incorporate side payments during search by strategic agents is presented in Algorithm 3. As in the BEECon algorithm, agents run in a fixed order. The agent whose turn it is to act is denoted i. In each turn, a set of possible value assignment $V_i^s \subseteq V_i$ is explored by a mechanism (line 2.

Algorithm 3 Distributed Search by Strategic Agents

onTurn(i)

1: **let** $v_i \leftarrow$ the current strategy of agent i
2: **choose** $V_i^s \subseteq V_i$
3: $\mathbb{M}$ queries agent i for its utilities $\{u_i(s, v_{-i}) \mid s \in V_i^s\}$
4: $\mathbb{M}$ queries all $j \in N_i$ for their utilities $\{u_j(s, v_{-i}) \mid s \in V_i^s\}$
5: $\mathbb{M}$ spots the assignment v_i' that maximizes the reported utilities $u_i(v_i', v_{-i}) + \sum_{j \in N_i} u_j(v_i', v_{-i})$
6: $\mathbb{M}$ computes Clarke pivot payments for agent i and for all $j \in N_i$
7: **send**(**update**,v_i') to all $j \in N_i$

Assume $V_i^s = V_i$). The mechanism $\mathbb{M}$ queries agent i (line 3) and all agents $j \in N_i$ (line 4) for their utilities for each alternative $\{(s, v_{-i}) \mid s \in V_i\}$. **The mechanism selects** the assignment that maximizes the (reported) social welfare (line 5), computes the side payment contracts (line 6), and publishes the social choice (line 7). The contracts may change during the run of the algorithm, but are charged only once - **upon termination**.

Proposition 1. If the agents bid truthfully, the search converges to a local optimum of the social welfare.

Proof. With truthful bidding (i.e., maximal payoffs), the algorithm acts exactly like BEECon - it iterates over the agents and for each agent i the assignment v_i that maximizes $u_i(v_i, v_{-i}) + \sum_{j \in N_i} u_j(v_i, v_{-i})$ is selected. From the correctness of BEECon, this algorithm also converges to a local optimum. □

Proposition 2. A bidding rational agent that only considers the current bid and ignores future turns will bid truthfully.

Proof. Follows directly from the truthfulness of VCG mechanisms. □

It is important to understand that Proposition 2 does not guarantee for agents that **consider future turns**. This issue is discussed at Section A of the Appendix. 2 simple 2-agents ADCOPs show that a strategic manipulation that considers future turns might work, although it is very likely to fail. However, while computing a strategic bid is computationally easy on a small 2-agents ADCOP, it is practically infeasible on a general ADCOP with hundreds of agents that hold private information and can place any bid. Especially among computational agents, it has already been argued that the complexity of a manipulation can rule it out in practice [6].

Ruling out strategic bidding that uses future steps considerations leaves one with guaranteed convergence (by Proposition 1). That is, the algorithm converges to a local optimum of the social welfare and computes the contracts that stabilize the outcome as a pure Nash equilibrium (PNE) from which no selfish agent deviates.

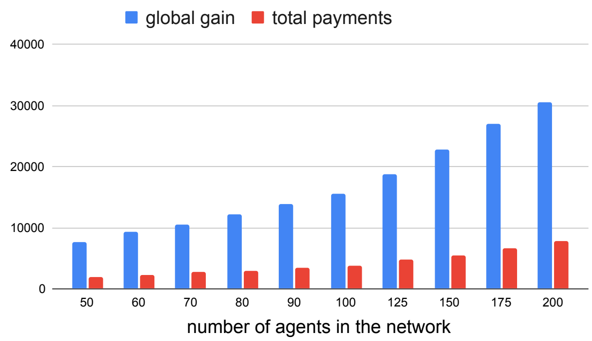

Fig. 1: Random networks with 10 neighbors per agent: increase in global social welfare, and total amount of payments

V. Experimental Evaluation

Experiments were conducted on two families of randomly generated ADCOPs. One family of uniformly distributed densities and the other of scale-free networks (e.g., Barabasi-Albert [2]). Both types of networks were of various sizes and densities. Barabasi-Albert networks can be categorized by a parameter m - the number of nodes that connect each new node to the already existing network during its construction [24] and are considered to resemble closely the structure of social networks [1]. The domain size of the agents in all of our experiments is 10, and the utilities in the game matrices were randomly and uniformly selected in the range [0,100]. For each configuration (type of network, size, and density), the results are averaged over 30 runs on randomly generated networks.

One important parameter that was evaluated experimentally is the sum of all payments charged by the mechanism compared to the improvement in global gain. The other interesting parameter is the fraction of agents that are better off when the algorithm terminates, taking their payments into account.

Figures 1,2 presents the results of running Algorithm 3 on random networks of various sizes. It is easy to see (Figure 1) that the payments charged by the mechanism in order to enforce truthfulness are small (around 25%) compared to the global gain. Most of the agents (83-84%) are better of upon termination (Figure 2).

The results of running Algorithm 3 on scale-free networks (of 500 agents) are presented in Figure 3. Interestingly, the proportion of the total payments to the global gain decreases with the density of the network, from 34% for networks with $m = 2$ to 23% when $m = 10$. Additionally, the proportion of agents that are better off upon termination increases with the density, from 77% when $m = 2$ to 83% when $m = 10$ (not in figure).

VI. Conclusion

Although asymmetric constraints have different personal gains for the searching agents, all former ADCOP search algorithms routinely assume full cooperation among all agents [9],

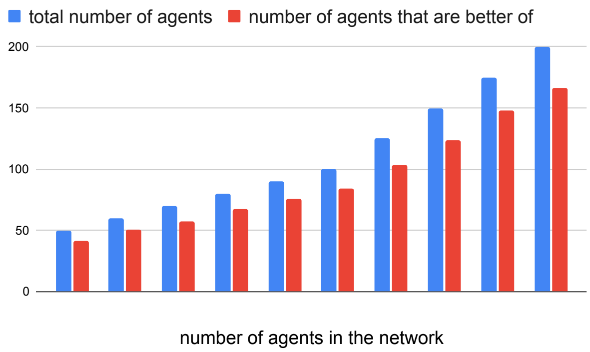

Fig. 2: Random networks with 10 neighbors per agent: number of agents that are better off upon termination

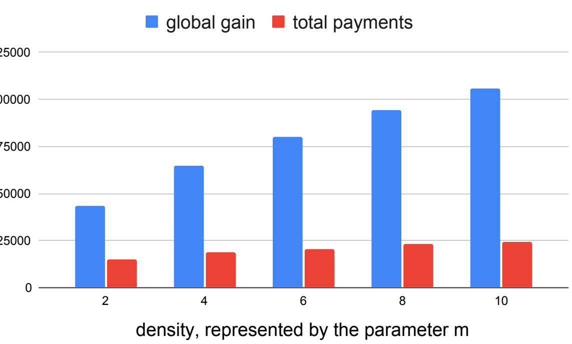

Fig. 3: Social welfare and total amount of payments on scale-free networks (Barabasi-Albert) of 500 agents and different densities

[16], [34]. Recently proposed ADCOP search algorithms exploit the game-like nature of ADCOPs and use side payments among agents during search to guarantee convergence to efficient and stable solutions [15], [26], enhancing even further the need to address strategic behavior of the searching agents. The present study proposes a local search algorithm for strategic agents that uses a mechanism of a well known family to guarantee truthful bids of side payments during search. The proposed algorithm extends the former best performing local search algorithm, preserves its guarantees and obtains two objectives - it converges to a local optimum of the social welfare, and computes the contracts that stabilize this outcome as a pure Nash equilibrium. The experimental evaluation demonstrates that the increase in the global social welfare when integrating a bidding mechanism is large compared to the payments that the mechanism charges in order to guarantee truthfulness, and that most of the agents are found experimentally to be better off when the algorithm terminates. An incentive-based multi agent search algorithm has recently been proposed for the Public Goods Game (PGG) [14]. An interesting next step will be to design a similar mechanism for PGGs. Based on the importance of PGGs [3], [4], [31], the present approach forms a meaningful step in obtaining multi agent (distributed) solutions to PGGs and more scenarios on networks.

References

[1] Reda Alhajj and Jon G. Rokne (Eds.). 2018. Encyclopedia of Social Network Analysis and Mining, 2nd Edition. Springer.

[2] Albert-Laszlo Barabasi and Reka Albert. 1999. Emergence of scaling in random networks. Science 286, 5439 (1999), 509–512.

[3] Leonardo Boncinelli and Paolo Pin. 2012. Stochastic stability in best shot network games. Games and Economic Behavior 75, 2 (2012), 538–554.

[4] Yann Bramoulle and Rachel Kranton. 2007. Public Goods in Networks. Journal of Economic Theory 135 (2007), 478–94.

[5] Edward H. Clarke. 1971. Multipart pricing of public goods. Public Choice 11 (1971), 17–33. Issue 1.

[6] Vincent Conitzer and Tuomas Sandholm. 2002. Complexity of Manipulating Elections with Few Candidates.. In AAAI/IAAI, Rina Dechter, Michael J. Kearns, and Richard S. Sutton (Eds.). AAAI Press / The MIT Press, 314–319.

[7] Ferdinando Fioretto, Enrico Pontelli, and William Yeoh. 2018. Distributed Constraint Optimization Problems and Applications: A Survey. J. Artif. Intell. Res. 61 (2018), 623–698.

[8] Ferdinando Fioretto, William Yeoh, Enrico Pontelli, Ye Ma, and Satishkumar J. Ranade. 2017. A Distributed Constraint Optimization (DCOP) Approach to the Economic Dispatch with Demand Response.. In AAMAS, Kate Larson, Michael Winikoff, Sanmay Das, and Edmund Durfee (Eds.). ACM, 999–1007.

[9] Tal Grinshpoun, Alon Grubshtein, Roie Zivan, Arnon Netzer, and Amnon Meisels. 2013. Asymmetric Distributed Constraint Optimization Problems. J. Artif. Intell. Res. 47 (2013), 613–647.

[10] T. Groves. 1973. Incentives in Teams. Econometrica 41 (1973), 617–631.

[11] Alon Grubshtein, Roie Zivan, and Amnon Meisels. 2012. Partial Cooperation in Multi-agent Local Search. In Proc. 4th Europ. Conf. Artif. Intell. (ECAI-2012), Vol. 242. 378–383.

[12] Katsutoshi Hirayama and Makoto Yokoo. 2005. The distributed breakout algorithms. Artif. Intell. 161, 1-2 (2005), 89–115.

[13] Michael J. Kearns. 2007. Graphical games. In Vazirani, Vijay V.; Nisan, Noam; Roughgarden, Tim; Tardos, Éva (2007). Algorithmic Game Theory. Cambridge University Press, 159–178.

[14] Vadim Levit, Zohar Komarovsky, Tal Grinshpoun, and Amnon Meisels. 2018. Incentive-based search for efficient equilibria of the public goods game. Artif. Intell. 262 (2018), 142–162.

[15] Vadim Levit and Amnon Meisels. 2018. Distributed Constrained Search by Selfish Agents for Efficient Equilibria. In Proc. 24th Intern. Conf. Princ. Pract. Const. Prog. (CP-18). Lille, France, 707–24.

[16] Omer Litov and Amnon Meisels. 2017. Forward bounding on pseudo-trees for DCOPs and ADCOPs. Artif. Intell. 252 (2017), 83–99.

[17] Rajiv T. Maheswaran, Jonathan P. Pearce, and Milind Tambe. 2004. Distributed Algorithms for DCOP: A Graphical-Game-Based Approach.. In ISCA PDCS. 432–439.

[18] Amnon Meisels. 2008. Distributed Search by Constrained Agents - Algorithms, Performance, Communication. Springer. 1–216 pages.

[19] A. Meisels and R. Zivan. 2007. Asynchronous Forward-checking for Distributed CSPs. Constraints 12 (2007), 131 – 150.

[20] Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo. 2005. Adopt: asynchronous distributed constraint optimization with quality guarantees. Artif. Intell. 161, 1-2 (2005), 149–180.

[21] D. Monderer and L. S. Sharpley. 1996. Potential Games. Games and Economic Behavior 14 (1996), 124–143.

[22] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani. 2007. Algorithmic Game Theory. Cambridge University Press.

[23] Alvaro Perez-Diaz, Enrico Gerding, and Frank McGroarty. 2018. Coordination and payment mechanisms for electric vehicle aggregators. Applied energy 212 (2018), 185–195.

[24] Francisco C. Santos and Jorge M. Pacheco. 2005. Scale-free networks provide a unifying framework for the emergence of cooperation. Physical Review Letters 95, 9 (2005), 098104.

[25] Georgios Tsaousoglou, Konstantinos Steriotis, Nikolaos Efthymiopoulos, Prodromos Makris, and Emmanouel Varvarigos. 2020. Truthful, practical and privacy-aware demand response in the smart grid via a distributed and optimal mechanism. IEEE Transactions on Smart Grid 11, 4 (2020), 3119–3130.

[26] Yair Vaknin and Amnon Meisels. December, 2021. Local Maxima in ADCOPs via Side Payments. In Proc. 20th IEEE/WIC/ACM Intern. Conf. Web Intell. Intellig. Agent Tech. (WI-IAT 21). ACM, Melbourne, Australia, 376–381.

[27] Cornelis Jan van Leeuwen and Przemyslaw Pawelczak. 2017. CoCoA: A Non-Iterative Approach to a Local Search (A)DCOP Solver. In 31st Conf. on Artificial Intelligence (AAAI-17), Satinder P. Singh and Shaul Markovitch (Eds.). AAAI Press, 3944–3950.

[28] William Vickrey. 1961. Counterspeculation, Auctions, and Competitive Sealed Tenders. In Journal of Finance. 16: 8 – 37.

[29] William Yeoh, Ariel Felner, and Sven Koenig. 2010. BnB-ADOPT: An Asynchronous Branch-and-Bound DCOP Algorithm. Jou. Artif. Intell. Res. (JAIR) 38 (2010), 85–133.

[30] William Yeoh and Makoto Yokoo. 2012. Distributed Problem Solving. AI Magazine 33, 3 (2012), 53–65.

[31] Sixie Yu, David Kempe, and Yevgeniy Vorobeychik. 2021. Altruism Design in Networked Public Goods Games. In Proc. 13th Intern. Joint Conf. Artif. Intell. (IJCAI-21). 493–99.

[32] 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. Artif. Intell. 161, 1-2 (2005), 55–87.

[33] Roie Zivan, Steven Okamoto, and Hilla Peled. 2014. Explorative anytime local search for distributed constraint optimization. Artif. Intell. 212 (2014), 1–26.

[34] Roie Zivan, Tomer Parash, Liel Cohen-Lavi, and Yarden Naveh. 2020. Applying Max-sum to asymmetric distributed constraint optimization problems. Auton. Agents Multi Agent Syst. 34, 1 (2020), 13.

[35] Roie Zivan, Harel Yedidsion, Steven Okamoto, Robin Glinton, and Katia P. Sycara. 2015. Distributed constraint optimization for teams of mobile sensing agents. Auton. Agents Multi Agent Syst. 29, 3 (2015), 495–536.

Appendix

The proposed mechanism for strategic agents guarantees truthfulness when a single step of bidding is considered (see Sections III, IV), in other words - when the agents in the neighborhood of the current agent consider only the bid in the present step. In order to start some considerations about strategic agents that attempt to take into account also future steps, let us look at the following two very simple examples.

Example 1 uses the constraints in table Ia and example 2 uses the constraints in table Ib. In both examples agent **A** is a truthful agent that plays the rows, and agent **B** is a strategic agent that plays the columns. The initial assignment profile is **(a,x)**.

From Proposition 2, only *multi-step manipulations* (strategic bids that consider also future turns) may succeed against the proposed mechanism. Consequently, let us consider that agent **B** places its bids strategically such that assignment profile **(b,y)** will be selected (while truthful bids result in assignment profile **(a,x)** in both examples). Example 1 shows the run of the proposed algorithm on the ADCOP represented in Table Ia.

Example 1.

The algorithm begins exploring the two possible assignment **a,b** of agent **A**. Agent **B** realizes that its greatest utility comes from assignment profile **(b,y)** and overbids the option of assignment **b** for the variable of agent **A**. The damage exerted on agent **A** is 10. Now the algorithm explores the possible assignments **x,y** of agent **B**, assignment profile **(b,y)** is reached and agent **B** gains 50. The overall gain of agent **B** is 50 - 10 = 40. In contrast - was agent **B** to bid truthfully all the way throughout the run of the algorithm, its gain would have been only 30, i.e., the strategic manipulation succeeds.

Although Example 1 shows that agent **B** can profitably manipulate the mechanism, it is enough to slightly modify the example into Example 2 in which the same strategic manipulation is not profitable. Example 2 shows the run of the proposed algorithm on the ADCOP represented by Table 2.

Example 2.

The algorithm begins exploring the two possible assignment **a,b** of agent **A**. Agent **B** realises that its greatest utility comes from assignment profile **(b,y)** and overbids the option of assignment **b** for the variable of agent **A**. The damage exerted on agent **A** is 40. Now the algorithms explores the possible assignments **x,y** of agent **B**, assignment profile **(b,y)** is reached and agent **B** gains 50. The overall gain of agent **B** is 50 - 40 = 10. In contrast - was agent **B** to bid truthfully all the way throughout the run of the algorithm, its gain would have been 30, i.e., the strategic manipulation failed.

The above two very simple examples demonstrate that although a multi-step manipulation is in principle possible, its success depends on the details of the potential future steps and in particular on private information of the other agents involved in future steps (not to mention that the future steps may involve totally unrelated agents in a different region of the network). It is easy to conclude that such manipulations are computationally too complex to succeed.

(a) Table 1

A\B	x	y
a	10,30	0,0
b	0,0	0,50

(b) Table 2

A\B	x	y
a	40,30	0,0
b	0,0	0,50