Socially Motivated Partial Cooperation in Multi-agent Local Search

Tal Ze'evi, Roie Zivan and **Omer Lev**
Ben-Gurion University of the Negev
Beer-Sheva, Israel
{talzee,zivanr,omerlev}@bgu.ac.il

Abstract

Partial Cooperation is a paradigm and a corresponding model, proposed to represent multi-agent systems in which agents are willing to cooperate to achieve a global goal, as long as some minimal threshold on their personal utility is satisfied. Distributed local search algorithms were proposed in order to solve asymmetric distributed constraint optimization problems (ADCOPs) in which agents are partially cooperative.

We contribute by: 1) extending the partial cooperative model to allow it to represent dynamic cooperation intentions, affected by changes in agents' wealth, in accordance with social studies literature. 2) proposing a novel local search algorithm in which agents receive indications of others' preferences on their actions and thus, can perform actions that are socially beneficial. Our empirical study reveals the advantage of the proposed algorithm in multiple benchmarks. Specifically, on realistic meeting scheduling problems it overcomes limitations of standard local search algorithms.

1 Introduction

Multi-agent systems commonly seek to reach an optimal state. One approach considers fully cooperative agents that perform actions in order to achieve a common global goal (cf. [Stranders *et al.*, 2009; Modi *et al.*, 2005; Gershman *et al.*, 2009]), while another explores agents that are self-interested, which take rational actions that increase their personal gains (cf. [Grant *et al.*, 2011]).

A partially cooperative model that handles scenarios, which do not fall into these two extreme classes was proposed in [Zivan *et al.*, 2012; Grubshtein *et al.*, 2012]. In such settings, agents act cooperatively – motivated by a desire to increase global (group) utility – as long as a minimum condition on their personal utility is satisfied. Such scenarios are common in many real-world settings, e.g., in a working environment where workers are expected to hold meetings, a worker may be willing to participate in a meeting at a time that is inconvenient for her, as long as she is able to hold meetings she considers urgent. A professor asked to teach an additional course for the benefit of the department in which she works may agree in one case, but if she is asked to teach yet another course, she refuses as she would not have enough time to conduct her own research. Car navigation applications are another relevant example, where in order to avoid generating traffic jams, some vehicles would be directed to slower routs. However, the driver would not follow the system's instructions if the delay is over some tolerance threshold. The partial cooperation model 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. Such willingness has support in social science theory [Wiepking and Breeze, 2012].

However, previous attempts to design partially cooperative models assumed people have a fixed reference point according to which they determine their cooperation intentions, and they are so altruistic as to give away any profit they gain, even if they themselves brought it about. In real life, people's intentions for cooperation are affected by changes in their wealth and by who brought it about [Kahneman and Tversky, 1979; Wiepking and Breeze, 2012]. This dynamic nature of intentions cannot be expressed by the existing partial cooperation model. Instead, we introduce a model in which agents' cooperation is based on the amount of utility they gain or lose. For example, if the professor is allocated a personal assistant that performs many of her everyday tasks, she might be willing to contribute some additional free time by teaching additional hours. In the navigating system example, a removal of a road block, which shortens the driving time may result in the driver willingness to tolerate a detour.

The adjustment of the partial cooperative model to realistic social behavior of humans allowed us to analyze the results produced by the two local search algorithms proposed in [Grubshtein *et al.*, 2012] when solving problems that include additional types of agents, i.e., agents whose cooperation willingness changes in different patterns following changes in their personal utility. This investigation revealed a weakness of these distributed local search algorithms: the agents in these algorithms attempt to find feasible solutions (solutions that satisfy the minimum personal utility requirements of all agents) that maximize their own gains according to their own knowledge. Hence, the willingness of agents to

cooperate under some conditions was used only to maintain the validity of the solutions rather than reach the desired high social welfare solution. The optimization process was identical to standard distributed local search algorithms as long as the minimum requirements were satisfied. Thus, when agents performed operations that decreased the utility of other agents, they were oblivious to the damage they incurred as long as the utility did not drop beneath the minimal threshold. This limited the ability of agents to perform socially beneficial actions.

Following these insights regarding the existing partial cooperative distributed local search algorithms, we propose a novel approach towards partial cooperative local search in which agents indicate to their neighbors which value assignments are preferred by them. These indications allow agents to make socially beneficial selections of value assignments.

Our empirical results demonstrate the advantage of the proposed algorithm over previously proposed partially cooperative local search algorithms in solving structured, unstructured and realistic DCOPs (Distributed Constraint Optimization Problem) including agents of novel partial cooperative types. Specifically, for realistic meeting scheduling problems, the proposed algorithm produces high quality results on scenarios where standard local search algorithms are known (and empirically found) to be ineffective. Moreover, we show that the proposed algorithm can be performed with privacy loss not significantly different than the privacy loss of other (existing) algorithms.

2 Background

2.1 Distributed Constraint Optimization

Without loss of generality and unless stated otherwise, we will assume that all problems are minimization problems.

A *DCOP* is a tuple $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{R} \rangle$. $\mathcal{A}$ is a finite set of agents $\{A_1, A_2, ..., A_n\}$. $\mathcal{X}$ is a finite set of variables $\{x_1, x_2, ..., x_m\}$. Each variable is held by a single agent. $\mathcal{D}$ is a set of domains $\{D_1, D_2, ..., D_m\}$. Each domain D_i contains the finite set of values that can be assigned to variable x_i. An assignment of value $d \in D_i$ to x_i is denoted by an ordered pair $\langle x_i, d \rangle$. $\mathcal{R}$ is a set of relations (constraints). Each constraint $C \in \mathcal{R}$ defines a non-negative *cost* for every possible value combination of a set of variables, and is of the form $C : D_{i_1} \times D_{i_2} \times ... \times D_{i_k} \to \mathbb{R}^+ \cup \{0\}$. A *binary constraint* refers to exactly two variables and is of the form $C_{ij} : D_i \times D_j \to \mathbb{R}^+ \cup \{0\}$.[1] A *binary DCOP* is a DCOP in which all constraints are binary. A *partial assignment* (PA) is a set of value assignments to variables, in which each variable appears at most once. $vars(PA)$ is the set of all variables that appear in PA. A constraint $C \in \mathcal{R}$ of the form $C : D_{i_1} \times D_{i_2} \times ... \times D_{i_k} \to \mathbb{R}^+ \cup \{0\}$ is *applicable* to PA if $x_{i_1}, x_{i_2}, ..., x_{i_k} \in vars(PA)$. The *cost of a PA* is the sum of all applicable constraints to PA over the assignments in PA. A *complete assignment* (or a *solution*) is a partial assignment that includes all the DCOP's variables ($vars(PA) = \mathcal{X}$). An *optimal solution* is a complete assignment with minimal cost.

For simplicity, we make standard assumptions that all DCOPs are binary DCOPs in which each agent holds exactly one variable (unless stated otherwise). These assumptions are commonly made in DCOP studies, e.g., [Modi *et al.*, 2005].

2.2 Asymmetric DCOP

ADCOPs generalize DCOPs by explicitly defining for each combination of assignments of constrained agents, the cost for each participant in the constraint [Grinshpoun *et al.*, 2013].

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 the same as in DCOP. Each constraint $C \in \mathcal{R}$ of an asymmetric DCOP defines a set of non-negative *costs* for every possible value combination of a set of variables, and takes the following form: $C : D_{i_1} \times D_{i_2} \times \cdots D_{i_k} \to \mathbb{R}^k_+$.

Notice that here $\mathbb{R}^k_+$ is a vector that includes for each agent A_j, $1 \le j \le k$ her cost for each combination of value assignments, so in practice each agent $1 \le j \le k$ holds her part of the constraint C_j, $C_j : D_{i_1} \times D_{i_2} \times \cdots D_{i_k} \to \mathbb{R}_+$. As before, an *optimal solution* is a complete assignment of all variables with minimal cost.

2.3 Partial Cooperation

In contrast to early studies of ADCOPs, which assumed full cooperation by the agents [Brito *et al.*, 2009; Grubshtein *et al.*, 2010], partial cooperation models represent agents that cooperate only under some conditions. The level of cooperation (which is represented by λ) determines the reference point according to which agents intentions are modeled. In order to allow the agents to consider solutions with high global quality, which may reduce their personal utility, the parameter λ bounds the losses that an agent is willing to undertake in order to contribute to the global objective, i.e., agents perform actions only if they do not result in a cost that exceeds the maximum cost they are willing to endure. Formally, the following parameters are used by the model:

Definition 1. *We note by μ_i the* **base-line cost** *of agent i (i.e., the cost for agent i that she assumes she will pay if she acts selfishly).*

Definition 2. *The* **cooperation intention parameter** $\lambda_i \ge 0$ *defines the maximal increase in the value of μ_i that is acceptable by agent i.*

These cooperation bounds can significantly decrease the number of feasible outcomes for a distributed incomplete algorithm, as can be seen in the next definition.

Definition 3. *A* **feasible outcome** *for a distributed algorithm is defined to be any outcome (solution) o in the set of all possible outcomes O, that satisfies the following condition.*

$$O^{feasible} = \{o \in O \mid \forall A_i \in \mathcal{A}, \quad c_i(o) \le \mu_i + \lambda_i\}$$

where $c_i(o)$ is the cost for agent A_i in outcome o.[2]

---

[1] We say that a variable is *involved* in a constraint if it is one of the variables the constraint refers to.

[2] The rest of our definitions are multiplicative in λ_i, and this definition can also be reformatted this way with a simple change, but we chose to present this as defined by Grubshtein et al. [Grubshtein *et al.*, 2012].

Two distributed local search algorithms, based on the Max Gain Messages (MGM) algorithm [Maheswaran *et al.*, 2004], were proposed in [Grubshtein *et al.*, 2012]. In the first, unsatisfied agents (agents whose current costs exceed the limit) send *nogood* messages to indicate to their neighbors that an assignment replacement is required, and *good* messages that remove a restriction imposed by a *nogood* when it does not apply anymore. Thus, this algorithm is called *Goods-MGM*.

In the second algorithm, Asymmetric Gain Coordination (AGC), agents exploit possible improvements until they converge to some local optimum, which cannot be further improved without breaching the cooperation bound of one of the agents. Before replacing a value assignment, an agent requests her neighbors' approval, which is given only if this value assignment replacement does not cause a breach of the cooperative bound for the neighbor. Only if all neighbors approve, the agent replaces her value assignment.

2.4 Privacy Loss

When agents inform others of their preferences – directly or implicitly (e.g., by changing an assignment) – they are revealing information that they may be reluctant to share. Therefore, distributed algorithms are often evaluated not only by the cost/utility of their solutions, their speed and use of the communication network, but also by the amount of private information they make agents divulge. The common approach towards evaluating privacy loss in algorithms solving distributed constraint problems is by measuring the entropy with respect to the knowledge of agents regarding the preferences of other agents [Greenstadt *et al.*, 2006; Brito *et al.*, 2009; Grinshpoun *et al.*, 2013]. Implementing this approach in the scenarios we consider, would mean evaluating privacy loss regarding preferences of an agent over the possible value assignment alternatives of her neighbors. This would be calculated by dividing the number of possible ways to order the alternatives given the revealed information, by all possible ways to order these alternatives. Multiplying one minus the result of this division by a hundred, will give the percentage of privacy lost.

3 Reference Dependent Partial Cooperation

Economics literature indicates that peoples' intentions change with respect to changes in their wealth (i.e., "reference-dependence"). Reference-dependent theories indicate that people are more sensitive to changes in wealth rather than to absolute wealth level [Kahneman and Tversky, 1979; Wiepking and Breeze, 2012]..

In order to allow the partial cooperative model to represent dynamic reference points, we redefine some parameters of the model:

Definition 4. *Let $\mu_{i,t}$ be the reference cost of agent A_i at iteration t of the algorithm (where $\mu_{i,0}$ is the baseline cost as defined above).*

Definition 5. *Let $\lambda_{i,t}$ be the cooperation intention parameter for agent A_i at iteration t of the algorithm, which as before defines the maximal increase in the cost acceptable by agent A_i, only with respect to $\mu_{i,t}$.*

Definition 6. *A complete assignment S is feasible in iteration t if it satisfies the following condition:*

$$\forall i \in A, c_i(S) \leq \mu_{i,t} \cdot (1 + \lambda_{i,t})$$

The outcome of a distributed algorithm that runs for m iterations is the complete assignment at the end of the m'th iteration (S_m), and it is feasible if the definition above holds for $\mu_{i,m}$ and $\lambda_{i,m}$.

Next, we present a number of examples of types of agents that can be represented by the extended model:[3]

Type 1 Fixed reference and cooperation parameters, i.e., for each agent A_i, for each iteration t, $\mu_{i,t} = \mu_{i,0}$ and $\lambda_{i,t} = \lambda_{i,0}$. This type of agents is identical to the types described in [Grubshtein *et al.*, 2012].

Type 2 Fixed λ, i.e., $\lambda_{i,t} = \lambda_{i,0}$ and a calculation of μ in each iteration as follows:

$$\mu_{i,t} = \mu_{i,t-1} + Min\{0, \frac{c_i(S_t) - c_i(S_{t-1})}{1 + \lambda_{i,0}}\}$$

This type resembles people or organizations that allocate specific budget for charity [Staples, 2004], or people that maintain and manage 'mental budgets' for philanthropic giving (based on *mental accounting* mechanisms cf. [LaBarge and Stinson, 2014]).

Type 3 This type is inspired by *reciprocal altruism*, in which an individual is willing to cooperate and give up personal wealth for others, with the expectation that they will act in a similar manner in the future [Trivers, 1971]. Formally this behavior is represented by a fixed λ, i.e., $\lambda_{i,t} = \lambda_{i,0}$ and a calculation of μ in each iteration as follows:

$$\mu_{i,t} = \mu_{i,t-1} + Min\{0, \Phi_{i,t-1}\left(\frac{c_i(S_t) - c_i(S_{t-1})}{1 + \lambda_{i,0}}\right)\}$$

where $\Phi_{i,t} = 0$ if the change in agent A_i's cost (i.e., $c(S_t) - c(S_{t-1})$) was caused by an action performed by their neighbor; and $\Phi_{i,t} = 1$ if the change was brought about only by agent A_i's own actions. Thus, only a decrease in cost caused by actions of neighbor agents affects the cooperation budget.[4]

Dominance

Next, we establish domination properties among the cooperation intentions of the agent types listed above. We will show that the type 1 agents are the most altruistic, type 2 agents the most selfish, and type 3 agents between them.

Let $B_{i,t}$ be the cost that agent A_i is willing to incur for the benefit of other agents at iteration t.

---

[3]While both $\mu_{i,t}$ and $\lambda_{i,t}$ are dynamic in the proposed extended model, for the example types of agents we fixed $\lambda_{i,t}$ and varied $\mu_{i,t}$. The representation of other types of agents may be more convenient by varying both or just $\lambda_{i,t}$.

[4]Although in the algorithms addressed in this paper, concurrent assignment replacements by neighboring agents are avoided, our definition is general so the model can be applied to other algorithms, which allow such actions.

Proposition 1. *If in some iteration t', an agent A_i of type 1 and an agent A_j of type 2 or type 3 have similar cooperation thresholds, i.e. $\lambda_{i,t'} = \lambda_{j,t'}$ and $\mu_{i,t'} = \mu_{j,t'}$, then, for any further iteration $t'' > t'$, if $c(S_{i,t''}) = c(S_{j,t''})$, then, $B_{j,t''} \leq B_{i,t''}$.*

Proof. $B_{\ell,t''} = \mu_{\ell,t''}(1 + \lambda_{\ell,t''}) - c_\ell(S_{t''})$ (for $\ell \in \{i,j\}$) and since $c(S_{i,t''}) = c(S_{j,t''})$ and by definition $\lambda_{\ell,0} = \lambda_{\ell,t}$ for any t, we only need to examine the change in $\mu_{\ell,t''}$. As $\mu_{i,t'} = \mu_{i,t''}$, we focus on agent A_j. For both type 2 and type 3 agents' $\mu_{j,t}$ is monotonically decreasing in t (because of $\min(0,\ldots)$ component of the $\mu_{j,t}$ definition). Hence, $\mu_{j,t''} \leq \mu_{j,t'}$, and therefore, $B_{j,t''} \leq B_{j,t'} = B_{i,t'} = B_{i,t''}$. $\square$

Proposition 2. *Let agent A_i be of type 2 and agent A_j be of type 3. If they have similar cooperation thresholds, i.e. $\lambda_{i,t'} = \lambda_{j,t'}$ and $\mu_{i,t'} = \mu_{j,t'}$ and have a similar history – $c_i(S_t) = c_j(S_t)$ for $t' \leq t$ – then, for any further iteration $t'' > t'$, $B_{i,t''} \leq B_{j,t''}$.*

Proof. As in the previous proof, $B_{\ell,t''} = \mu_{\ell,t''}(1 + \lambda_{\ell,t''}) - c_\ell(S_{t''})$ (for $\ell \in \{i,j\}$) and since $c(S_{i,t''}) = c(S_{j,t''})$ and by definition $\lambda_{\ell,0} = \lambda_{\ell,t''}$ for any t'', we only need to examine the change in $\mu_{\ell,t''}$. Agents of type 2 will decrease their $\mu_{i,t''}$ value when $c_i(S_{t''}) - c_i(S_{t''-1}) < 0$, while agents of type 3 will decrease their $\mu_{j,t''}$ values by the same amount, but only in a subset of these cases. Hence, $\mu_{i,t''} \leq \mu_{j,t''}$ for $t' \leq t''$, and therefore, $B_{i,t''} \leq B_{j,t''}$. $\square$

4 Socially-Motivated Local Search

In the partial cooperative local search algorithms described above, agents, besides exploiting their local knowledge, cooperate in order to preserve a level of personal utility that is acceptable by all agents.

In order to allow agents to exploit the cooperative intentions of their neighboring agents, and so to improve the solution's quality (social welfare), we propose a novel approach towards partial cooperative local search, in which agents take an extra step in the interaction process before selecting an assignment. In this new stage each agent shares with their neighbors some information regarding their preferences over their assignment selection After exchanging this information, each agent attempts to find a variable assignment, taking into consideration their own preferences as well as the indications received from their neighbors. We combine this approach with the AGC algorithm (cf. [Grubshtein *et al.*, 2012]) and propose Socially Motivated (SM) AGC.

Definition 7. *Let $\omega_{i,j} \in \Omega_i$ be the importance that agent A_i ascribes to agent A_j's preferences. s.t. :*
$0 \leq \omega_{i,j} \leq 1, and \sum_{j \in N(i) \cup \{A_i\}} \omega_{i,j} = 1$
where N(i) is the set of agent A_i's neighbors.

When $\omega_{i,i} = 0$, agent A_i will be completely altruistic, selecting her value assignment in accordance with the preferences of her neighbors and ignoring her personal interest. In contrary, when $\omega_{i,i} = 1$, A_i completely ignores the preferences of her neighbors and chooses her value assignment taking into consideration only her own interest.

Algorithm 1 includes the pseudo code of socially-motivated AGC.

Algorithm 1 SM_AGC

input: $baseLineAssignment_i$, $baseLineCost_i$, λ_i and Ω_i

$value \leftarrow baseLineAssignment_i$;
$\mu_{i,0} \leftarrow baseLineCost_i$;
$localView \leftarrow null$; // $localView$ is S_{t-1}
send($value$) to $N(i)$;
while stop condition not met **do**
  PHASE 1:
    Collect all $value$ messages and update $localView$
    for each $A_j \in N(i)$ **do**
      $\pi_{i,j} \leftarrow preferences(A_j)$;
      send($\pi_{i,j}$) to A_j;
  PHASE 2:
    Collect all π messages;
    $\Pi_i \leftarrow \pi_{j \in N(i)} \cup preferences(A_i)$;
    $alterVal_i \leftarrow socialImprovingAssignment(\Pi_i, \Omega_i)$;
    send($alterVal_i$, $socialGain_i$) to $N(i)$;
  PHASE 3:
    Collect all $\langle alterVal_j, socialGain_j \rangle$ messages;
    $a_j \leftarrow$ agent in $N(i) \cup A_i$ with maximal socialGain s.t.
      $c_i(v_j \leftarrow alterVal_j | S_t) \leq \mu_{i,t} \cdot (1 + \lambda_{i,t})$;
    $can\_improve \leftarrow socialGain_i > 0$ & $a_j = A_i$
    send($Neg!$) to $N(i) \setminus a_j$;
  PHASE 4:
    Collect $Neg!$ messages;
    if did not receive $Neg!$ & $can\_improve$ **then**
      $value \leftarrow alterVal_i$;
      send($value$) to $N(i)$;

In each round of the algorithm, each agent can send/receive a message for each of its constraints, and therefore, the overall number of messages per round is $2|\mathcal{R}|$. Since every agent can proceed concurrently with others, the running time of each round is bound by the agent with the maximal number of constraints – $\max_{i \in \mathcal{A}} |\{C_{ij} | j \in \mathcal{A}; C_{ij} \in \mathcal{C}\}|$.

For the selection of the alternative value assignment ($alterVal_i$) in Phase 2 we propose (and use in our experiments) the following heuristic for this selection: After collecting the indications π_j received from her neighbors, the agent calculates a sampling probability for each value in her domain, as follows:

$$p_{val} = \begin{cases} 0 & , \ \Gamma_{val} < 0 \\ \frac{\Gamma_{val}}{\sum_{val \in Domain_i, \Gamma_{val} > 0} \Gamma_{val}} & , \ otherwise \end{cases}$$

where:
$$\Gamma_{val} = \sum_{j \in N(i) \cup \{A_i\}} \omega_{i,j} \cdot \pi_j(val)$$

Afterwards, the agent randomly samples a value from the underlying distribution.

The main novelty of the proposed algorithm is in the sharing of indications regarding preferences on the selection assignment of neighboring agents. Thus, the information that agents share in this stage of the algorithm (Phase 1) is expected to have a dramatic effect on the performance of the algorithm. We propose five versions of the algorithm, which we compare in our experimental study.

We make a distinction between two categories of indications that agents share with their neighbors. The first, we call 'taboo' assignments, i.e., an agent informs her neighbor which of the neighbors' value selections will cause a breach of the current cooperation threshold. The second, which we

Version	Taboo	Vote
SmAGC_BI	-	Binary
SmAGC_CI	-	Cost
SmAGC_T	+	None
SmAGC_T_BI	+	Binary
SmAGC_T_CI	+	Cost

Table 1: Different Versions of the SmAGC algorithm

call a 'vote' allows agents to direct their neighbors to a specific value that they wish the neighbor will select. Such a vote can be binary or weighted. Table 1 summarizes the differences between the various versions of the SM_AGC algorithm that will be further discussed.

5 Experimental Evaluation

In order to evaluate the performance of the different versions of the socially motivated proposed local search algorithm, we compared their performance on five DCOP benchmarks:

1. Random uniform DCOPs: unstructured binary minimization DCOPs with density $p_1 = 0.1$, 100 agents ($n = 100$), each holding a single variable with 10 values in its domain ($d = 10$) (for more details see [Grubshtein *et al.*, 2012]).

2. K-regular graphs: similar problems with randomly generated graphs in which all agents had the same number of neighbors (5).

3. Scale Free networks: constructed by using the Barabasi-Albert model [Albert and Barabási, 2002], as used in [Zivan *et al.*, 2014].

4. Graph-Coloring problems: included 100 agents, each with 3 colors in the domain and a density parameter $p_1 = 0.05$ as used in [Zivan *et al.*, 2014].

5. Meetings-Scheduling DCOPs: modeling n agents trying to coordinate m meetings, where a particular agent A_i has probability $p_{i,k}$ to be invited to meet k ($1 \leq k \leq m$). The agents (unlike in the other benchmark setting) can hold more than one variable, each representing a meeting they hope to attend. The values in each domain represent the possible time-slots for the meeting to be scheduled and a value representing the option that the agent does not attend the meeting. Additional constraints express the unique time slot scheduling requirement, while local constraints express the priorities for an agent between her own meetings.

Following [Grubshtein *et al.*, 2012], for the first four benchmarks listed above, the baseline solutions were obtained by running a greedy distributed local search algorithm. For the meeting scheduling problems, such a procedure produces a very low quality solution, therefore, we produced the baseline solutions by ordering all meetings according to the number of participants from high to low, and assigning each meeting to a free time-slot according to this order.

The development was performed in Java using the 'AgentZero' framework – a dedicated framework for simulating and evaluating Multi-agent algorithms [Lutati *et al.*, 2014], with Eclipse Juno on a Windows operating system. Simulations were run on a single PC with two 2.4 GHz Intel processors, each with 6 cores, and 20 GB of RAM. The source code for both the various algorithms as

well as the different benchmarks we explored in this paper can be found at: `https://github.com/IEMAI/SMPC/tree/master`.

5.1 Results

For each benchmark we compared the algorithms in scenarios with the three different types of agents presented in Section 3. In all SM_AGC versions agents ascribe equal importance to their neighbors,[5] and any taboo values have zero sampling probability. Each algorithm ran for 1000 iterations on each problem. Results were averaged over 50 random instances.

Random uniform DCOPs: Figures 1a and 1b present the aggregated social cost in each iteration of the search, when the initial cooperation parameter is set to $\lambda = 0.1$, and $\lambda = 0.8$ respectively. In both settings, for all types of agents, both Goods-MGM and AGC yielded significantly inferior aggregated (social) cost ($p < 0.001$) than the various versions of SM_AGC. Goods-MGM failed to find any feasible solutions for smaller values of λ. Versions in which agents share 'taboo' indications, found significantly better solutions for $\lambda = 0.1$ ($p < 0.001$), however for $\lambda = 0.8$, the significant advantage of these versions was found mainly for agents of type 1, and type 2 who have shared their costs ($p < 0.01$). Under both settings, agents of type 2 were first to converge to solutions with low social cost across all algorithms. In contrary, agents of type 1, continued to explore the solution space for significantly better solutions ($p < 0.05$).[6] Surprisingly, for lower λ values and without the 'taboo' indications, sharing the precise costs performed worse than the binary option, i.e., the increased information hurt the search for social welfare. Similar results were obtained for K-regular graphs and for scale free nets, and are omitted for lack of space.

Graph-Coloring problems: Figure 2 presents the social cost of solutions produced for the graph coloring problems, using $\lambda = 0.1$. In contrast to the previous setups, Goods-MGM and AGC achieved results which are competitive with some versions of SM_AGC. While the version of SM_AGC in which agents share only 'taboo', found solutions with lower costs for problems with agents of type 2, and solutions of high costs for problems with agents of type 1. Similar results when using $\lambda = 0.8$ were omitted.

Meetings-Scheduling problems: Both AGC and Goods-MGM failed to exploit the willingness of agents to cooperate and produced solutions with very low social welfare. On the other hand, using the SM_AGC algorithm agents were able to improve their baseline attendance rate ($\sim$75%) by approximately 4%-20%. Agents who shared binary votes were able to achieve the highest social welfare and attendance rates ($p < 0.001$), while sharing only 'taboos' or both 'taboos' and 'votes' with exact costs achieved much lower social welfare ($p < 0.01$). Agents of type 3 achieved better results than agents of type 2 for all variants of SM_AGC ($p < 0.001$), and better than agents of type 1 when agents share only 'taboos' or both 'taboos' and 'votes' with exact costs ($p < 0.01$).

---

[5]We left the investigation of the algorithm's behavior in scenarios with various importance levels for future work.

[6]Similar results were obtained for denser problems with $p_1 = 0.7$ and were omitted for lack of space
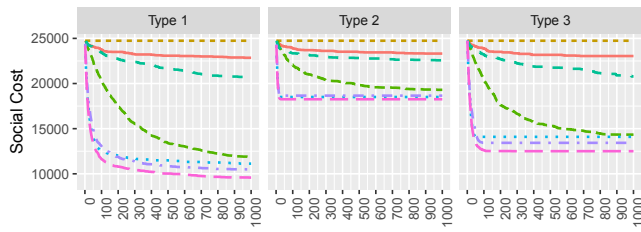
(a) $\lambda = 0.1$
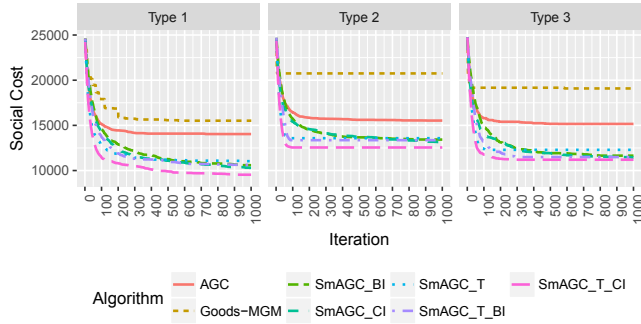
(b) $\lambda = 0.8$

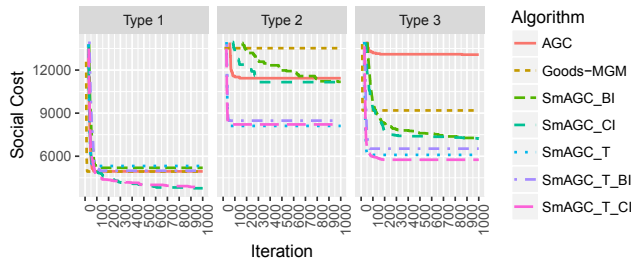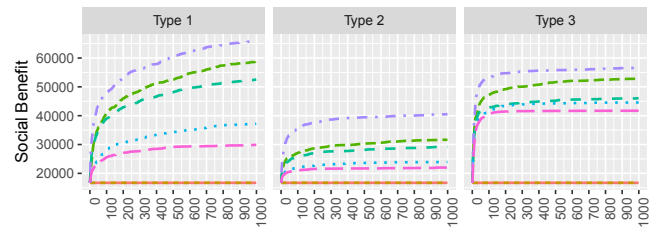Figure 1: Social cost for uniform random problems.

Figure 2: Social cost for Graph Coloring problems ($\lambda = 0.1$).

(a) Social benefit

(b) Attendance rate

Figure 3: Social benefit and Attendance rate for meeting scheduling problems (n=100, m=50, p=0.05, d=8, $\lambda = 0.1$).
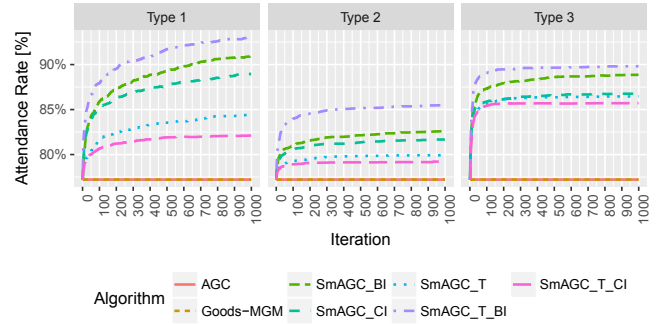
5.2 Analysis

Our results indicate in general, that the binary variants of the algorithm perform worse than those that reveal costs, except for meeting scheduling problems. This seems to be because, the binary values create a more egalitarian outcome, since each agent's unsatisfied constraints are treated the same. For the meeting scheduling problem, egalitarian solutions make sense as agents' loss from not attending meetings is the same. Adding information here – as with weighted 'vote' indication – actually adds noise to the algorithm, making it more difficult to reach an optimal solution. A similar phenomenon – more information leading to inferior end states – has been observed in many domains (e.g., in decision making processes, Meir et al. [Meir *et al.*, 2014]).

Turning to the other variant we have – adding 'taboo' to the different algorithms – we can see that, generally, 'taboo' variants perform better than the non-'taboo' versions. However, this advantage is stronger for $\lambda = 0.1$ than for $\lambda = 0.8$. We hypothesize that this is since the usage of 'taboo' guarantees a feasible assignment. Therefore, when the search space of feasible solutions is small (that is, when λ is small), using 'taboo' allows examining a much smaller space, and hence, reaching a good result fast. However, when λ is larger and

the search space is much larger, the convergence speed advantage of 'taboo' versions is less pronounced.

5.3 Privacy

For lack of space we omitted our privacy results and only discuss the two main insights they bring about. First, the vote sharing versions of SM_AGC, share information that is not shared by standard AGC, and therefore obviously they are less private. However, the version that only shares 'taboo' messages has similar privacy loss results to standard AGC. Yet, the social welfare results of this version are significantly better than standard AGC.

The second is that privacy loss for type 1 agents is far larger than for other agent types. This probably has to do with type 1 agents being far more willing to cooperate (as their threshold does not change as iterations progress), and therefore they have more chances to pass on information to other agents. In other words, there is a (natural) correlation between the cooperation intentions and the willingness to share private information.

6 Conclusion

We proposed an extension of the partial cooperative paradigm, which allows simulation of realistic scenarios, in which agents intentions for cooperation can change with respect to utility gains. Alongside it, we presented a local search algorithm in which the cooperative intentions of agents can be exploited, not only to ensure that the solution obtained is acceptable by all agents, but also in order to select a high quality solution. A significant advantage of the proposed algorithm over the existing partial cooperative algorithms was found even when only insatiability indications ('taboo') were

shared. The socially motivated local search algorithm produces high quality solutions on realistic problems with hard equality constraints (meeting scheduling) where standard local search fails.

This is a first step in a deeper exploration of agents' behaviors – more types can be introduced in future work, and *interactions between different types of agents* (i.e., a heterogeneous population of agents, instead of a single type) can be investigated.

References

[Albert and Barabási, 2002] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[Brito *et al.*, 2009] Ismel Brito, Amnon Meisels, Pedro Meseguer, and Roie Zivan. Distributed constraint satisfaction with partially known constraints. *Constraints*, 14(2):199–234, 2009.

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

[Grant *et al.*, 2011] J. Grant, S. Kraus, M. Wooldridge, and I. Zuckerman. Manipulating boolean games through communication. In *IJCAI*, pages 210–215, 2011.

[Greenstadt *et al.*, 2006] R. Greenstadt, J. Pearce, and M. Tambe. Analysis of privacy loss in distributed constraint optimization. In *AAAI-06*, pages 647–653, Boston, MA, USA, July 2006.

[Grinshpoun *et al.*, 2013] Tal Grinshpoun, Alon Grubshtein, Roie Zivan, Arnon Netzer, and Amnon Meisels. Asymmetric distributed constraint optimization problems. *J. Artif. Intell. Res. (JAIR)*, 47:613–647, 2013.

[Grubshtein *et al.*, 2010] A. Grubshtein, R. Zivan, T. Grinshpon, and A. Meisels. Local search for distributed asymmetric optimization. In *AAMAS 2010*, pages 1015–1022, May 2010.

[Grubshtein *et al.*, 2012] Alon Grubshtein, Roie Zivan, and Amnon Meisels. Partial cooperation in multi-agent local search. In *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, pages 378–383, 2012.

[Kahneman and Tversky, 1979] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the econometric society*, pages 263–291, 1979.

[LaBarge and Stinson, 2014] Monica C LaBarge and Jeffrey L Stinson. The role of mental budgeting in philanthropic decision-making. *Nonprofit and Voluntary Sector Quarterly*, 43(6):993–1013, 2014.

[Lutati *et al.*, 2014] Benny Lutati, Inna Gontmakher, Michael Lando, Arnon Netzer, Amnon Meisels, and Alon Grubshtein. Agentzero: A framework for simulating and evaluating multi-agent algorithms. In *Agent-Oriented Software Engineering - Reflections on Architectures, Methodologies, Languages, and Frameworks*, pages 309–327. 2014.

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

[Meir *et al.*, 2014] Reshef Meir, Omer Lev, and Jeffrey S. Rosenschein. A local-dominance theory of voting equilibria. In *Proceedings of the 15th ACM conference on Economics and Computation (EC)*, pages 313–330, Palo Alto, Califronia, June 2014.

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

[Staples, 2004] Chris Staples. What does corporate social responsibility mean for charitable fundraising in the uk? *International Journal of Nonprofit and Voluntary Sector Marketing*, 9(2):154–158, 2004.

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

[Trivers, 1971] Robert L Trivers. The evolution of reciprocal altruism. *The Quarterly review of biology*, 46(1):35–57, 1971.

[Wiepking and Breeze, 2012] Pamala Wiepking and Beth Breeze. Feeling poor, acting stingy: The effect of money perceptions on charitable giving. *International Journal of Nonprofit and Voluntary Sector Marketing*, 17(1):13–24, 2012.

[Zivan *et al.*, 2012] Roie Zivan, Alon Grubshtein, Michal Friedman, and Amnon Meisels. Partial cooperation in multi-agent search. In *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 2012 (3 Volumes)*, pages 1267–1268, 2012.

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