

Quality guarantees for region optimal DCOP algorithms

Meritxell Vinyals¹, Eric Shieh², Jesus Cerquides¹, Juan Antonio Rodriguez-Aguilar¹,
Zhengyu Yin², Milind Tambe², and Emma Bowring³

¹ Artificial Intelligence Research Institute (IIIA), Campus UAB, Bellaterra, Spain

{meritxell, cerquide,jar}@iiia.csic.es

² University of Southern California, Los Angeles, CA 90089

{eshieh, zhengyuy,tambe}@usc.edu

³ University of the Pacific, Stockton, CA 95211

ebowring@pacific.edu

ABSTRACT

k - and t -optimality algorithms [9, 6] provide solutions to DCOPs that are optimal in regions characterized by its size and distance respectively. Moreover, they provide quality guarantees on their solutions. Here we generalise the k - and t -optimal framework to introduce \mathcal{C} -optimality, a flexible framework that provides reward-independent quality guarantees for optima in regions characterised by any arbitrary criterion. Therefore, \mathcal{C} -optimality allows us to explore the space of criteria (beyond size and distance) looking for those that lead to better solution qualities. We benefit from this larger space of criteria to propose a new criterion, the so-called size-bounded-distance criterion, which outperforms k - and t -optimality.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed AI

General Terms

Algorithms, Design, Theory

Keywords

DCOP, approximate algorithm, bound, region optimality

1. INTRODUCTION

Distributed Constraint Optimization (DCOP) is a popular framework for cooperative multi-agent decision making. It has been applied to real-world domains such as sensor networks [14], traffic control [5], or meeting scheduling [10]. In real-world domains, and particularly in large-scale applications, DCOP techniques have to cope with limitations on resources and time available for reasoning. Because DCOP is NP-Hard [8], complete DCOP algorithms (e.g. Adopt [8], OptAPO [7], DPOP [10]) that guarantee global optimality are unaffordable for these domains due to their exponen-

tial costs. In contrast to complete algorithms, incomplete algorithms [14, 4, 12, 9, 6] provide better scalability.

Unfortunately, an important limitation for the application of incomplete algorithms is that they usually fail to provide quality guarantees on their solutions. The importance of quality guarantees is twofold. First, they help guarantee that agents do not converge to a solution whose quality is below a certain fraction of the optimal solution (which can have catastrophic effects in certain domains). Secondly, quality guarantees can aid in algorithm selection and network structure selection in situations where the algorithmic cost of coordination must be weighed up against solution quality (trade-off cost versus quality).

To the best of our knowledge, k -size and t -distance optimal algorithms [9, 6] are the only incomplete DCOP algorithms that can provide guarantees on the worst-case solution quality of their solutions at design time and exploiting different levels of knowledge of the particular problem instance(s). These quality guarantees exploit the available knowledge, if any, about the DCOP(s) to solve regarding their graph structure [9] and their reward structure [3]. Unlike other incomplete algorithms that focus on individual agent decisions [14, 4, 12], k -size [9] and t -distance [6] optimal algorithms are based on coordinating the decisions of local groups (neighbourhoods) of agents. Thus, given a DCOP, agents inside a neighbourhood coordinate to locally optimise their joint decision by considering any joint assignment that can improve their joint reward. The difference between k -size and t -distance optimal algorithms is the criterion employed to generate neighbourhoods: k -size-optimality creates neighbourhoods of a fixed size (k), whereas t -distance-optimality creates per each agent a neighbourhood that includes all other agents within a certain distance (t) in the constraint graph. In both cases, we can regard a collection of neighbourhoods as an *exploration region* for either a k - or t -optimal algorithm in a constraint graph.

Although k -size and t -distance are the criteria explored so far in the literature, it is reasonable to wonder whether there are further local optimality criteria that can lead to better solution qualities while providing quality guarantees. In this paper we provide the foundations to explore this fundamental research question. First of all, we generalise the k - and t -optimal framework to introduce \mathcal{C} -optimality, a flexible framework that provides quality guarantees for local

Cite as: Quality guarantees for region optimal DCOP algorithms, M. Vinyals et al., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. XXX-XXX. Copyright © 2011, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

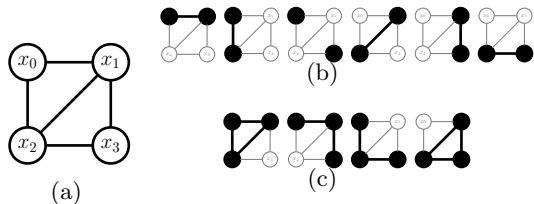


Figure 1: Example of (a) a DCOP graph, (b) its 2-size region and (c) its 3-size region.

optima in regions characterised by any arbitrary criterion. Therefore, \mathcal{C} -optimality allows us to explore the space of local optimality criteria (beyond size and distance) looking for those that lead to better solution qualities.

We benefit from this larger space of criteria to propose a novel criterion to define regions, the so-called size-bounded-distance criterion, which we design to overcome the main drawbacks of size and distance optimality. Finally, we extend the DALO algorithm proposed in [6] to compute \mathcal{C} -optimal solutions. We empirically show that the size-bounded-distance criterion indeed leads to better solution qualities, outperforming k - and t -optimality. Therefore, the \mathcal{C} -optimality framework opens new research opportunities to study the design of new local optimality criteria.

Likewise k - and t -optimality, \mathcal{C} -optimality is algorithmic-independent, meaning that bounds are defined over solutions not over the algorithms to find them. Thus, although we propose \mathcal{C} -DALO as general purpose algorithm to find any \mathcal{C} -optimal for arbitrary criteria, it does not imply that there cannot exist different region-optimal algorithms. Indeed, in our previous work [11], we employ analogous guarantees to provide worst-case bounds on the solutions of the loopy belief propagation algorithm, a popular approximate algorithm for finding the Maximum a Posteriori assignment in Markov Random Fields. In contrast, this work focuses on DCOPs. Firstly, we generalise the bounds in [11] to provide a framework for regional DCOPs algorithms. Secondly, we analyse the benefits of exploring arbitrary region criteria.

The paper is organised as follows. Section 2 provides some background on DCOPs and on the k - and t -optimality frameworks. Section 3 introduces the notion of \mathcal{C} -optimality solution as a local solution for an arbitrary criterion and the mechanisms for computing quality guarantees for \mathcal{C} -optimal solutions. Moreover, it also proves that the \mathcal{C} -optimality framework generalises k - and t -optimality. Section 4 introduces a new local optimality criterion, size-bounded distance, and empirically compares the quality solutions obtained by the new criterion with respect to k - and t -optimal solutions. Finally, section 5 draws conclusions and sets paths to future research.

2. BACKGROUND

2.1 DCOP Definition

A Distributed Constraint Optimization Problem (DCOP) consists of a set of variables, each assigned to an agent which must assign a discrete value to the variable: these values correspond to individual actions that can be taken by agents. Constraints exist between subsets of these variables that determine rewards to the agent team based on the combinations of values chosen by their respective agents, namely relations. Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be a set of variables over

domains $\mathcal{D}_1, \dots, \mathcal{D}_n$. A relation on a set of variables $V \subseteq \mathcal{X}$ is expressed as a reward function $S_V : \mathcal{D}_V \rightarrow \mathbb{R}^+$, where \mathcal{D}_V is the joint domain over the variables in V . This function represents the reward generated by the relation over the variables in V when the variables take on an assignment in the joint domain \mathcal{D}_V . Whenever there is no need to identify the domain, we simply use S to note relations.

In a DCOP each agent knows all the relations that involve its variable(s). In this work we assume that each agent is assigned a single variable, so we will use the terms “agent” and “variable” interchangeably.

Formally, a DCOP is a tuple $\langle \mathcal{X}, \mathcal{D}, \mathcal{R} \rangle$, where: \mathcal{X} is a set of variables (each one assigned to a different agent); \mathcal{D} is the joint domain space for all variables; and \mathcal{R} is a set of reward relations. The solution quality for an assignment $d \in \mathcal{D}$ to the variables in \mathcal{X} is the sum of the rewards for the assignment over all the relations in the DCOP, namely:

$$R(d) = \sum_{S_V \in \mathcal{R}} S_V(d_V) \quad (1)$$

where $d_V \in \mathcal{D}_V$ contains the values assigned by d to the variables in V . With slight abuse of notation we allow to write equation 1 as $R(d) = \sum_{S \in \mathcal{R}} S(d)$.

Solving a DCOP amounts to choosing values for the variables in \mathcal{X} such that the solution quality is maximized. A binary DCOP (each relation involves a maximum of two variables) is typically represented by its constraint graph, whose vertices stand for variables and whose edges link variables that have some direct dependency (appear together in the domain of some relation). Examples of constraint graphs are depicted in figures 1(a) and 2(a).

2.2 Size and distance optimality

Since DCOP is NP-hard, an important line of work focuses on developing fast incomplete algorithms. Along this direction, a significant trend is to study approaches based on coordinating the decisions of local groups of agents, instead of having each agent make an individual choice.

Two important local optimality criteria that establish how to group agents to coordinate their decisions are k -size [9] and t -distance [6] optimality. According to k -size optimality agents form groups of k agents. For instance, figures 1(b) and 1(c) depict the groups of 2 agents and 3 agents respectively for the DCOP in figure 1(a) where boldfaced nodes stand for agents included in the group. Given an assignment x^* , it is a local optimum, k -optimum, when no group of k or fewer agents can improve its reward $R(x^*)$ by simultaneously changing their variable assignments. On the other hand, t -optimality defines locality based on a group of surrounding nodes within a fixed distance t of a central node. For instance, figures 2(b) and 2(c) depict the groups of agents at distance 1 and 2 respectively for each agent in the DCOP in figure 2(a). Likewise k -optimality, a t -optimum occurs when no group of agents can improve its reward.

k - and t -optimal algorithms represent an important class of incomplete algorithms that have agents dynamically form local groups to coordinate action choices. A significant feature of k - and t -optimal algorithms is that they provide guarantees on the solution quality of a DCOP as a fraction of the global optimum, prior to the execution of the algorithm. An algorithm has a quality guarantee δ (being $0 \leq \delta \leq 1$) if every solution provided by the algorithm has at least quality $\delta \cdot R(x^*)$ where x^* stands for the global optimum assignment.

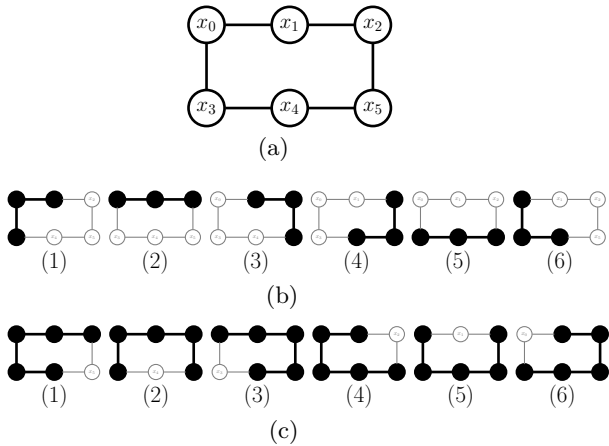


Figure 2: Example of (a) a DCOP graph, (b) its 1-distance region and (c) its 2-distance region.

The larger the quality guarantee, the closer the algorithm is to providing an optimal solution to the problem.

Both k - and t -optimality have explored mechanisms for computing bounds. Firstly, both k -size optimality and t -distance optimality provide means for computing bounds independently of the problem instance [9, 6], namely disregarding the graph structure and reward structure. Secondly, knowledge of a problem instance can be used to obtain tighter guarantees. One way is to exploit the knowledge about the graph structure of the DCOP (e.g. star, ring) [9]. Another way is to exploit the reward structure [3]. We can group such mechanisms based on their computational costs.

On the one hand, a tight bound on the quality of every k - or t -optimum can be computed using a linear program (LP) [9, 6]. In this method, rewards on the relations in the DCOP are treated as variables in a program whose goal is to minimise the quality guarantee. When the program is solved, the decision variables are instantiated with the values that, if used as relation rewards, would produce the DCOP whose local optimum has the lowest reward with respect to the global optimal solution. For example, for k -optimality and for a specific graph structure, after running the program we obtain (1) a quality guarantee δ for any k -optimal solution on any DCOP having the specific constraint graph and (2) a DCOP having the specific constraint graph and a k -optimal solution x^k whose quality is equal to the bound, namely $R(x^k) = \delta \cdot R(x^*)$

On the other hand, there are methods that are computationally cheaper and can compute bounds in constant time [9, 6]. Despite the computational savings of these methods, with respect to the LP-based approach, in general tightness is not guaranteed.

3. GENERALIZING SIZE AND DISTANCE OPTIMALITY

In this section we generalize the concept of size and distance optimality to \mathcal{C} -optimality, which allows us to characterize any local optimum in a region \mathcal{C} characterized by an arbitrary criterion.

Notice that given a DCOP both k - and t - local optimality criteria define a region, namely a family of neighbourhoods (subsets of variables) \mathcal{C} . For instance, in figure 2(b), we show the neighbourhoods in the 1-distance region of the

DCOP in figure 2(a), where boldfaced nodes in the constraint graph stand for variables included in the neighbourhood. Given some assignment x , we say that it is optimal in a neighbourhood $\mathcal{C}^\alpha \in \mathcal{C}$ if its reward cannot be improved by changing the values of some of the variables in the neighbourhood. For instance, the first graph on the left in figure 2(b) represents a neighbourhood. An assignment x is optimal in that neighbourhood if any other assignment that maintains the values of x_2, x_4 and x_5 receives at most the same reward as x . Then, we can claim optimality for x in a region \mathcal{C} (noted as x^c) whenever it is optimal in each neighbourhood in the region. For instance, an assignment x will be optimal in the region depicted in figure 2(c) if it is optimal in each of its neighbourhoods. Therefore, in general, for both k -size and t -distance based optimality, we observe that:

- each criterion is based on the definition of a region over the constraint graph; and
- given any assignment, checking for either k -size or t -distance optimality amounts to checking for optimality in that region.

Hereafter we propose a general notion of region optimality, the so-called \mathcal{C} -optimality, and describe how to calculate bounds for a \mathcal{C} -optimal assignment, namely an assignment that is optimal in an arbitrary region \mathcal{C} .

3.1 Region optimality

Next, we introduce the concepts of neighbourhood and region so that we can formally define \mathcal{C} -optimality. After that, we analyse the way in which neighbourhoods relate to each others by formalizing the idea that a larger neighbourhood covers a smaller one.

Formally, a neighbourhood is a subset of variables of \mathcal{X} . Given two assignments x and y , we define $D(x, y)$ as the set containing the variables whose values in x and y differ. Given a neighbourhood A , we say that x is a *neighbour of y in A* iff x differs from y only in variables that are contained in A .

A region \mathcal{C} is a multi-set¹ of subsets of \mathcal{X} , namely a multi-set of neighbourhoods of \mathcal{X} . Given a region \mathcal{C} , we say that x is *inside region \mathcal{C} of y* iff x differs from y only in variables that are contained in one of the neighbourhoods in \mathcal{C} , that is, if there is a neighborhood $\mathcal{C}^\alpha \in \mathcal{C}$ such that x is neighbour of y in \mathcal{C}^α .

An assignment x is \mathcal{C} -optimal if it cannot be improved by any other assignment inside region \mathcal{C} of x . That is, for every assignment y inside region \mathcal{C} of x , we have that $R(x) \geq R(y)$.

Relations among neighbourhoods

Given two neighbourhoods $A, B \subseteq \mathcal{X}$ we say that B completely covers A if $A \subseteq B$. We say that B does not cover A at all if $A \cap B = \emptyset$. Otherwise, we say that B covers A partially.

As an example of these relations, consider neighbourhoods (1) and (4) in figure 2(b), noted as $A = \{x_0, x_1, x_3\}$ and $B = \{x_2, x_4, x_5\}$ respectively, and neighbourhood (1) in figure 2(c), noted as $C = \{x_0, x_1, x_2, x_3, x_4\}$. Then, we have that A covers C partially (it contains some variables in C) whereas C covers A completely (C contains all variables in

¹A multi-set is a generalisation of a set that can hold multiple instances of the very same element.

A). Moreover, A does not cover B at all and vice versa because these neighbourhoods do not have any variable in common.

Then, we say that $A \subseteq \mathcal{X}$ is covered by \mathcal{C} if there is a neighbourhood $C^\alpha \in \mathcal{C}$ such that C^α completely covers A . For example, neighbourhood (1) in figure 2(b) is covered by the region of neighbourhoods in figure 2(c), because, among others, neighbourhood (1) in this region covers it completely.

For each neighbourhood C^α we can classify each relation S in a DCOP into one of three disjoint groups, depending on whether C^α covers S completely ($T(C^\alpha)$), partially ($P(C^\alpha)$), or not at all ($N(C^\alpha)$).

For each relation $S_V \in \mathcal{R}$ we define $cc(S_V, \mathcal{C}) = |\{C^\alpha \in \mathcal{C} \text{ s.t. } V \subseteq C^\alpha\}|$, that is, the number of neighbourhoods in \mathcal{C} that cover the domain of S_V completely. We also define $nc(S_V, \mathcal{C}) = |\{C^\alpha \in \mathcal{C} \text{ s.t. } V \cap C^\alpha = \emptyset\}|$, that is, the number of neighbourhoods in \mathcal{C} that do not cover the domain of S_V at all.

3.2 Quality guarantees for region optima

After its formal definition, we are interested in providing a bound on the quality of any \mathcal{C} -optimal assignment in a DCOP with non-negative rewards. We say that we have a bound δ when we can state that the quality of any \mathcal{C} -optimal assignment x^C is larger than δ times the quality of the optimal x^* . Hence, having a bound δ means that for every x^C we have that $\frac{R(x^C)}{R(x^*)} \geq \delta$. For a given set of relations \mathcal{R} , let x^C be the \mathcal{C} -optimal assignment with smallest reward, then $\frac{R(x^C)}{R(x^*)}$ provides a tight bound on the quality of any \mathcal{C} -optimal assignment for the specific rewards \mathcal{R} .

We are interested in defining bounds that are independent of the particular reward values of the DCOP. In that setting, a simple way to provide a bound on the quality is to directly search the space of reward values to find the set of rewards \mathcal{R}^* that minimizes $\frac{R(x^C)}{R(x^*)}$.

More formally, this can be encoded as:

Find \mathcal{R} , x^C and x^* that

$$\text{minimize } \frac{R(x^C)}{R(x^*)}$$

subject to x^C being a \mathcal{C} -optimal for \mathcal{R}

Applying some transformations detailed in [13], we can simplify this program into the following linear program (LP) with x and y being vectors of positive real numbers:

$$\text{minimize } \sum_{S \in \mathcal{R}} x_S$$

subject to

$$\sum_{S \in \mathcal{R}} y_S = 1$$

and for each neighbourhood C covered by \mathcal{C} subject to

$$\sum_{S \in \mathcal{R}} x_S \geq \sum_{S \in T(C)} y_S + \sum_{S \in N(C)} x_S$$

where $T(C)$ contains the relations completely covered by C and $N(C)$ the relations that are not covered by C at all.

After solving this LP, $\delta = \sum_{S \in \mathcal{R}} x_S$ provides a tight bound on the quality of a \mathcal{C} -optimal solution for the graph structure represented by \mathcal{R} . Let M be the number of variables of the largest neighbourhood in \mathcal{C} . The LP has $2 \cdot |\mathcal{R}|$ variables and $\mathcal{O}(2^M \cdot |\mathcal{C}|)$ constraints, and hence it is solvable in time polynomial in $|\mathcal{R}|$ and in $2^M \cdot |\mathcal{C}|$.

3.3 Faster quality guarantees

The computational complexity of the previous LP can be high as the number of relations $|\mathcal{R}|$, the number of neighbourhoods $|\mathcal{C}|$ or its size M grows. In this section we show that we can compute a bound in time $\mathcal{O}(|\mathcal{R}||\mathcal{C}|)$. Further-

more, the result will prove as a very valuable tool for future theoretical developments. As a counterpart, we lose the tightness of the bound.

PROPOSITION 1. Let $\langle \mathcal{X}, \mathcal{D}, \mathcal{R} \rangle$ be a DCOP with non-negative rewards and \mathcal{C} a region. If x^C is a \mathcal{C} -optimal assignment then

$$R(x^C) \geq \frac{cc_*}{|\mathcal{C}| - nc_*} R(x^*) \quad (2)$$

where $cc_* = \min_{S \in \mathcal{R}} cc(S, \mathcal{C})$, $nc_* = \min_{S \in \mathcal{R}} nc(S, \mathcal{C})$, and x^* is the optimal assignment.

Proposition 1 directly provides a simple algorithm to compute a bound. Given a region \mathcal{C} and a graph structure, we can directly assess cc_* and nc_* by computing $cc(S, \mathcal{C})$ and $nc(S, \mathcal{C})$ for each relation $S \in \mathcal{R}$ and taking the minimum. This will take time $\mathcal{O}(|\mathcal{R}||\mathcal{C}|)$, that is linear in the number of relations of the DCOP and linear in the number of neighbourhoods in the region.

As an example, now we turn back to figure 2 to assess the bounds for a \mathcal{C} -optimal assignment using equation 2. First, we assess the bound for the 1-distance region \mathcal{C}_1 in figure 2(b). Given the relation $S = R_{\{x_0, x_1\}}$, we assess the number of neighbourhoods that completely cover $\{x_0, x_1\}$ as $cc(S, \mathcal{C}_1) = 2$ (the two first neighbourhoods on the left-hand side) and the number of neighbourhoods that do not cover $\{x_0, x_1\}$ at all as $nc(S, \mathcal{C}_1) = 2$ (the fourth and fifth neighbourhoods). After repeating the process for the rest of relations in the constraint graph, we obtain that $cc_* = 2$ and $nc_* = 2$, and hence $\frac{cc_*}{|\mathcal{C}_1| - nc_*} = \frac{2}{6-2} = \frac{1}{2}$. Notice that this leads to a better bound than the one we obtain following the result in [6], since $\frac{m+t-1}{n} = \frac{1}{3}$. This is due to the fact that we are computing the bound specifically for this graph structure, whilst the bounds provided in [6] are independent of the graph structure. If now we consider the 2-distance region \mathcal{C}_2 in figure 2(c), we obtain that $\frac{cc_*}{|\mathcal{C}_2| - nc_*} = \frac{4}{6-0} = \frac{2}{3}$. Again, this leads to a better bound than the one reported in [6] since $\frac{m+t-1}{n} = \frac{1}{2}$. Note that the bounds provided as example are tight. However, despite these examples, the bound assessed by proposition 1 is not guaranteed to be tight and can return worse bounds than the ones provided by the LP-based mechanism.

Both the LP and proposition 1 assess bounds that depend on the graph structure but are independent of the specific reward values. We can always use them to assess bounds independently of the graph structure by assessing the bound for the complete graph, since any other structure is a particular case of the complete graph with some rewards set to zero.

The proof for proposition 1 is a generalization of the one in [9] for k -optimality.

PROOF. For every $C^\alpha \in \mathcal{C}$, consider an assignment x^α such that $x_i^\alpha = x_i^C$ if $x_i \notin C^\alpha$ and $x_i^\alpha = x_i^*$ if $x_i \in C^\alpha$. Since x^C is \mathcal{C} -optimal, for all $C^\alpha \in \mathcal{C}$, $R(x^C) \geq R(x^\alpha)$ holds, and hence

$$R(x^C) \geq \frac{\sum_{C^\alpha \in \mathcal{C}} R(x^\alpha)}{|\mathcal{C}|}. \quad (3)$$

Now for each x^α , we have that $R(x^\alpha) = \sum_{S \in \mathcal{R}} S(x^\alpha)$.

We can split the sum into completely covered ($T(C^\alpha)$), partially covered ($P(C^\alpha)$), or not covered at all ($N(C^\alpha)$) relations, having $R(x^\alpha) = \sum_{S \in T(C^\alpha)} S(x^\alpha) + \sum_{S \in P(C^\alpha)} S(x^\alpha) + \sum_{S \in N(C^\alpha)} S(x^\alpha)$.

Then, by setting partially covered relations to the minimum possible reward (0 assuming non-negative rewards),

$R(x^\alpha) \geq \sum_{S \in T(C^\alpha)} S(x^\alpha) + \sum_{S \in N(C^\alpha)} S(x^\alpha)$. Now, by definition of x^α , for every variable x_i in a relation completely covered by C^α we have that $x_i^\alpha = x_i^*$, and for every variable x_i in a relation not covered at all by C^α we have that $x_i^\alpha = x_i^c$. Hence, $R(x^\alpha) \geq \sum_{S \in T(C^\alpha)} S(x^*) + \sum_{S \in N(C^\alpha)} S(x^c)$. To assess a bound, after substituting this inequality in equation 3, we have that

$$R(x^c) \geq \frac{\sum_{C^\alpha \in \mathcal{C}} \sum_{S \in T(C^\alpha)} S(x^*) + \sum_{C^\alpha \in \mathcal{C}} \sum_{S \in N(C^\alpha)} S(x^c)}{|\mathcal{C}|}. \quad (4)$$

We need to express the numerator in terms of $R(x^c)$ and $R(x^*)$. Grouping the sum by relations and reminding that $cc_* = \min_{S \in \mathcal{R}} cc(S, \mathcal{C})$, the term on the left can be expressed as:

$$\begin{aligned} \sum_{C^\alpha \in \mathcal{C}} \sum_{S \in T(C^\alpha)} S(x^*) &= \sum_{S \in \mathcal{R}} cc(S, \mathcal{C}) \cdot S(x^*) \geq \\ &\geq \sum_{S \in \mathcal{R}} cc_* \cdot S(x^*) = cc_* \sum_{S \in \mathcal{R}} S(x^*) = cc_* \cdot R(x^*). \end{aligned}$$

Furthermore, recalling that $nc_* = \min_{S \in \mathcal{R}} nc(S, \mathcal{C})$, we can do the same with the right term:

$$\begin{aligned} \sum_{C^\alpha \in \mathcal{C}} \sum_{S \in N(C^\alpha)} S(x^c) &= \sum_{S \in \mathcal{R}} nc(S, \mathcal{C}) \cdot S(x^c) \geq \\ &\geq \sum_{S \in \mathcal{R}} nc_* \cdot S(x^c) = nc_* \sum_{S \in \mathcal{R}} S(x^c) = nc_* \cdot R(x^c). \end{aligned}$$

After substituting these two results in equation 3 and rearranging terms, we obtain equation 2. \square

In the next two sections we show that the constant-time reward-independent bounds provided for size and distance optimality in [6, 9] are particular cases of proposition 1.

3.4 Size-optimal bounds as a specific case of region-optimal bounds

Now we present the main result in [9] as a specific case of \mathcal{C} -optimality. An assignment is k -size-optimal if it can not be improved by changing the value of any group of size k or fewer variables.

PROPOSITION 2. *Let $\langle \mathcal{X}, \mathcal{D}, \mathcal{R} \rangle$ be a DCOP with non-negative rewards and m the maximum relation arity. Then, for any k -optimal assignment x^k :*

$$R(x^k) \geq \frac{\binom{|\mathcal{X}|-m}{k-m}}{\binom{|\mathcal{X}|}{k} - \binom{|\mathcal{X}|-m}{k}} R(x^*) \quad (5)$$

PROOF. This result is just a specific case of our general result where we take as region all subsets of size k , that is $\mathcal{C} = \{C^\alpha \subseteq \mathcal{X} \mid |C^\alpha| = k\}$. The number of neighbourhoods is $|\mathcal{C}| = \binom{|\mathcal{X}|}{k}$. The number of neighbourhoods that completely cover S is $cc(S, \mathcal{C}) = \binom{|\mathcal{X}|-|S|}{k-|S|}$, where $|S|$ stands for the cardinality of S (take the variables in S plus $k - |S|$ variables out of the remaining $|\mathcal{X}| - |S|$). Because $cc(S, \mathcal{C})$ reaches the minimum value with the maximum value of $|S|$, $cc_* = \binom{|\mathcal{X}|-m}{k-m}$. The number of neighbourhoods in \mathcal{C} that do not cover S at all is $nc(S, \mathcal{C}) = \binom{|\mathcal{X}|-|S|}{k}$ (take k variables out of the remaining $|\mathcal{X}| - |S|$ variables). Because $nc(S, \mathcal{C})$ reaches the minimum value with the maximum value of $|S|$, $nc_* = \binom{|\mathcal{X}|-m}{k}$. Finally, we obtain equation 5 by using $|\mathcal{X}|$, cc_* and nc_* in equation 2, and simplifying. \square

3.5 Distance-optimal bounds as a specific case of region optimal bounds

Now we present the main result in [6] as a specific case of \mathcal{C} -optimality. First, let us notice that the bound in [6] can be more easily proved if the graph is assumed to be connected. After that, we will see that the bound can be improved in the case that the graph is composed of a set of connected components. Consider a connected DCOP with n variables, minimum constraint arity m , non-negative rewards, and globally optimal assignment x^* . It is easy to see that whenever $m + t - 1 > n$, the length of the shortest path between any two nodes is smaller than t , and hence any t -distance optimal assignment will in fact be globally optimal.

PROPOSITION 3. *Let $\langle \mathcal{X}, \mathcal{D}, \mathcal{R} \rangle$ be a connected DCOP with non-negative rewards. Then, whenever $m + t - 1 \leq n$, we can bound the quality of any t -distance optimal assignment x^t as*

$$R(x^t) \geq \frac{(m+t-1)}{n} R(x^*) \quad (6)$$

PROOF. This result is just a specific case of our general result where we take as region the t -distance neighbourhoods for each variable $x \in \mathcal{X}$, that is $\mathcal{C} = \{\Omega_t(x) \mid x \in \mathcal{X}\}$. The number of neighbourhoods in the region is $|\mathcal{C}| = n$. Next, we show that for every relation S , we have that the number of neighbourhoods in \mathcal{C} that completely cover S , $cc(S, \mathcal{C})$ is at least $m+t-1$. The only variables that do not have S in their t -neighbourhood are those variables that are at distance t or more from every variable in S . If no such variables exist, then $cc(S, \mathcal{C}) = n > m+t-1$. Otherwise, let x' be one of these variables. There is a shortest path connecting x' to its closest variable in S (say x). The path must have length at least t , that is $x, x_1, \dots, x_{t-1}, \dots, x'$. Now, it is clear that S is in the t -neighbourhood of the $t-1$ variables $\{x_1, \dots, x_{t-1}\}$. Note that since we are taking the shortest path to any variable in S , no x_i can be in S . Since S is also in the t -neighbourhood of every variable in S and there can be no intersection between S and $\{x_1, \dots, x_{t-1}\}$, we have $cc(S, \mathcal{C}) = |S| + t - 1 \geq m + t - 1$. Hence $cc_* \geq m + t - 1$. By definition, $nc_* \geq 0$. Finally, we obtain equation 6 by using $|\mathcal{C}|$, cc_* and nc_* in equation 2, and simplifying. \square

If the DCOP is not connected, we can obtain a better bound by simply applying equation 6 to each connected component and taking the minimum. That is $R(x^t) \geq \frac{(m+t-1)}{n_*} R(x^*)$ where n_* is the number of elements of the largest connected component, which is always smaller than n .

4. EMPIRICAL EVALUATION

In this section we show how we can benefit from the larger space of criteria for defining regions provided by \mathcal{C} -optimality. We start by analyzing the regions generated by k -size and t -distance on DCOPs with different structures, to conclude that k -size generates a potentially huge number of neighborhoods of limited size and t -distance generates a limited number of potentially huge neighborhoods. To keep under control the amount and size of neighborhoods we introduce a new type of regions, namely size-bounded distance regions, that include a limited number of limited size neighborhoods. Finally, we empirically show that algorithms for approximate DCOP solving can benefit from using size-bounded distance regions.

We start by analyzing k -size and t -distance regions in section 4.1, to motivate the introduction of size-bounded distance regions in section 4.2. The DALO algorithm was proposed in [6] to find either k - or t - optimal solutions. In section 4.3 we show how we can extend it to find an optimal in any region \mathcal{C} . Finally, in section 4.4 we compare the performance of size, distance and size-bounded distance regions on DCOPs with different graph structures using DALO.

4.1 Analysis of size and distance regions

We are interested in analyzing the regions generated by k -size and t -distance on DCOPs with different structures. More concretely, we want to assess the number of different neighbourhoods as well as the size (number of variables) for each neighbourhood, since both parameters strongly influence the amount of computation needed to obtain a k , t -optimum. The worst case time for checking optimality in a neighbourhood is exponential in its number of variables. Furthermore, if an agent has to consider a large number of neighbourhoods, it will have to share its time among them. Hence, in terms of computational effort, it is of interest to find regions that have a limited number of neighbourhoods of limited size. In k -size optimality the size is limited by k but the number of neighbourhoods grows as $\binom{|\mathcal{X}|}{k}$, which can turn out prohibitively large. In t -distance optimality the number of neighbourhoods is $\mathcal{O}(|\mathcal{X}|)$ but the size of the neighbourhoods is not limited. For example, the 1-distance region of a complete graph contains a single neighbourhood with all the variables, and hence finding a 1-distance optimal in a complete graph is as hard as finding a global optimum.

For a more detailed empirical analysis, we have computed statistics of the maximum neighbourhood size in a region (MaxS) and the number of neighbourhoods per agent (#) over randomly generated constraint graphs. We have used three different types of graph structures: $G(n, M)$ random graphs [2], Barabasi-Albert (BA) scale-free graphs [1], and non-linear preferential attachment (NLPA) graphs based on the BA model, but with a larger emphasis on many nodes having fewer connections. All the graphs have 100 nodes with a density of four meaning that on average each node has four neighbours. We compare the results of three different criteria: 5-size (K5)², 1-distance (T1) and 2-distance (T2). The first three rows in table 1 present the averages over 50 DCOPs of MaxS and # for each criteria and each type of graphs. From these statistics we observe that T1 and T2 distance criteria result in very large neighbourhoods, especially on scale-free and NLPA graphs due to the presence of *hub* agents with a large number of neighbours. We also observe that K5 criterion generates a large number of neighbourhoods, specially in scale-free and NLPA due to the presence of hub nodes (e.g. the average number of neighbourhoods per agent in NLPA graphs is 11366).

From this analysis we can conclude that k -size generates a potentially huge number of neighborhoods of limited size and t -distance generates a limited number of potentially huge neighborhoods. To overcome this, we introduce a new type of regions, namely size-bounded distance regions, which include a limited number of bounded size neighborhoods.

4.2 Size-bounded distance optimality

Our aim at formulating the size-bounded distance crite-

²As in [6, 9] neighbourhoods of 5 variables that are not connected in the graph are discarded.

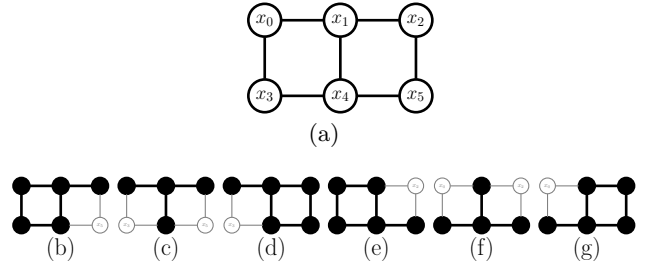


Figure 3: Example of (a) a DCOP graph, and (b)-(g) the set neighborhoods for the 5-size-distance bounded region.

tion is to provide an alternative trade-off to size and distance, being more aware of the complexity of the regions they generate.

Let $T(x_i, x_j)$ be the distance between two variables in the constraint graph. Let $\Omega_t(x_i) = \{x_j | T(x_i, x_j) \leq t\}$ be the t -distance neighbourhood centered on variable x_i . Then, the s -size-bounded-distance neighbourhood is the largest t -distance region whose number of variables does not exceed the limit s . Formally, let $t(x_i) = \max \{t \text{ s.t. } |\Omega_t(x_i)| \leq s\}$ be the largest value for t such that $|\Omega_t(x_i)| \leq s$. The s -size-bounded-distance neighbourhood centered on variable x_i is defined as $\Phi_s(x_i) = \Omega_{t(x_i)}(x_i)$.

Figure 3 (b)-(g) depicts 5-size-bounded distance neighbourhoods for agents x_0 to x_5 for the DCOP in figure 3 (a). Observe that agents can end up exploring different distance levels in their neighbourhoods as a result of bounding their size to s . In our example, agents x_0 , x_2 , x_3 and x_5 explore their 2-distance neighbourhood with size 5 (figures 3 (b)(d)(e)(g)), whereas agents x_1 and x_4 are restricted to 1-distance neighbourhood with size 4 (figures 3 (c)(f)).

Now, the s -size-bounded distance region includes the s -size-bounded-distance neighbourhood of each agent $x_i \in \mathcal{X}$. Moreover, in order to ensure that all relations are covered, the s -size-bounded-distance region also includes a neighbourhood for every edge in the graph.

Note that in size-bounded distance optimality both the number of neighbourhoods and their size are limited. Now we can go back to table 4, to compare the number of regions and its size with the state-of-the-art criteria. In the last row we show the averages over 50 constraint graphs of MaxS and # for 5-size-bounded-distance optimality (S5) for each type of graph. We can see that S5 is the only criterion that manages to keep the size of the region limited (to 5 agents) together with a reasonable number of neighbourhoods per agent (between 3 and 10 depending on the graph structure).

4.3 DALO for region optimality

The DALO algorithm is an asynchronous algorithm that starts with a random initial assignment and monotonically increases the solution quality by independently optimizing in each of the neighbourhoods that are created. As described in [6], DALO has three phases: initialization, optimization, and implementation.

During the initialization phase agents distributedly create a set of neighbourhoods and assign each neighbourhood to a leader agent (the central node to minimize communication) that will be in charge of its optimization. After initialization, agents run in parallel the optimization and implementation phases for each assigned neighbourhood until stabilization. During the optimization phase, each leader agent optimises by searching for a joint assignment of the variables in its

neighborhoods that improve their reward. After optimizing, the leader agent runs the implementation phase trying to implement the new joint assignment found. Because neighbourhoods are optimised in parallel and a variable can appear in multiple neighbourhoods, DALO implementation phase uses an asynchronous protocol based on a standard lock/commit pattern to ensure stability.

To use DALO with an arbitrary region, we focused on the initialization phase to modify how agents create the groups over which they optimise. Concretely to allow DALO to search for a \mathcal{C} -optimal, agents will distributedly generate the neighbourhoods in region \mathcal{C} . For example, to use DALO in the s -size-bounded distance region each agent will iterate through various t -distance neighbourhoods, by broadcasting at distance t , to determine the largest t -distance neighbourhood whose size does not exceed s . After initialization, for the specific region, optimization and implementation phases are ran as specified in [6], independently of the used region.

4.4 Empirical results

In this section we compare the results obtained by DALO using four different criteria: 5-size ($K5$), 1-distance ($T1$), 2-distance ($T2$), and 5-size bounded distance ($S5$) criteria.

We ran similar experimental settings to Kiekintveld et al. [6]. We measured the performance of the extension of the DALO algorithm³ described in section 4.3 when running over each one of the regions generated by the four criteria described above. Thus, we tested DALO for the four criteria over the different types of graphs described in section 4.1. All the graphs have 100 nodes, each one with density 4, meaning that on average each node has 4 neighbours. Moreover, variables' domain size is 10, and rewards are integers sampled from a distribution $U[0, 10000]$.

Besides graph types, we also considered different *Computation/Communication Ratios* (CCR) [6]. The CCR setting defines the number of constraint assignments that may be evaluated at each communication step. For example, $CCR = 0.01$ allows each node to process up to 100 checks in a time step. We vary the setting of CCR in our experiments to test DALO in two settings with different relative cost for sending messages and computation, namely $CCR = 0.01$ and $CCR = 0.1$. In general, the larger the value of CCR , the higher the computation cost. Notice that, with respect to the experimental settings in [6], we discarded using $CCR = 0$. The rationale is that if $CCR = 0$, communication is infinitely more costly than computation and hence the best strategy is computing the optimal by means of a fully centralized algorithm.

Figures 4 (a)-(f) plot the normalized solution quality of each algorithm along global time for each graph structure and CCR metric. The normalized solution quality is computed by: (1) subtracting the initial reward, as assessed by DALO for a given criterion, from the reward at a given global time; and (2) dividing the result by the best known reward obtained by DALO out of the four criteria. All results are averaged over 25 sample instances. In what follows, we compare the four criteria along two dimensions: (1) the final normalised solution quality; and (2) the convergence speed required to reach a good solution quality.

Regarding solution quality, the results vary depending the value of CCR and graph structure. On the one hand, in

	Random		Scale-free		NPLA	
	MaxS	#	MaxS	#	MaxS	#
K5	5	167	5	963	5	11366
T1	10	1	27	1	63	1
T2	38	1	82	1	99	1
S5	5	3	5	3	5	10

Table 1: Statistics for regions generated by k5, t1, t2 and s5 criteria for 100 agents. MaxS stands for the maximum size of a neighbourhood and # for the average number of neighbourhoods per agent.

scenarios where computation is more costly ($CCR = 0.1$), overall $S5$ outperforms the rest of criteria. Although $T1$ is very competitive and its solution quality comes very close to that of $S5$ over random and scale free graphs, $S5$ significantly outperforms $T1$ on NLPA graphs. Moreover, both $S5$ and $T1$ largely outperform $K5$. The reason of the poor performance of $K5$ is that it generated neighbourhoods of fixed size. On the other hand, in scenarios where computation is cheaper ($CCR = 0.01$), the differences of final solution qualities between $S5$, $T1$, and $K5$ are not significant. There is an aspect though that deserves special attention. Notice that for all the test cases, the performance of DALO over $T2$ regions is much worse than the performance over the regions generated by the rest of criteria. We can explain this result by analysing the complexity of $T2$ regions as shown in table 1. Thus, we observe that $T2$ generates very large neighbourhoods that can not be optimised within the maximum global time (1000 global time steps). The solution quality degradation when handling $T2$ regions is particularly significant on scale-free and NLPA graphs because the criterion generates neighbourhoods whose size is close to the size of the original problem (99 variables on average in NLPA graphs).

Regarding convergence speed, $S5$ regions help DALO converge to a high solution quality faster. Likewise our analysis about solution quality above, $T1$ is again competitive with respect to $S5$, though $S5$ largely outperforms $T1$ on NLPA graphs. This is because, as observed in [6], NLPA graphs are characterized by large *hub* nodes with many connections that results in large neighbourhoods that take long for agents to optimise. Regarding $K5$, convergence speed is slower than that of $S5$ and $T1$ because each leader in DALO coordinates a neighbourhood of size 5, whereas the neighbourhoods for $S5$ and $T1$ may be smaller.

To summarise, our experimental results show that criteria that produce regions with large number of neighbourhoods or/and large neighbourhood sizes are not guaranteed to outperform criteria that produce less complex regions. In fact, overall the size-bounded distance criterion proposed in section 4.2 was able to outperform the rest of criteria by limiting the complexity of the regions that it generates.

5. CONCLUSIONS

In this paper we generalise the k - and t -optimal frameworks [9, 6] to introduce \mathcal{C} -optimality, a flexible framework that provides quality guarantees for local optima in regions characterised by any arbitrary criterion. With this aim, we provide: (1) a formal definition of \mathcal{C} -optimality, namely of local optimality in some arbitrary region; and (2) quality guarantees for region optimal solutions that exploit the knowledge about the graph structure. Regarding quality guarantees, we defined two methods with different computational costs: (1) a first one, based on solving an LP, that guarantees tightness; and (2) a second one that requires lin-

³We used the DALO code provided by the authors at <http://teamcore.usc.edu/dcop/>.

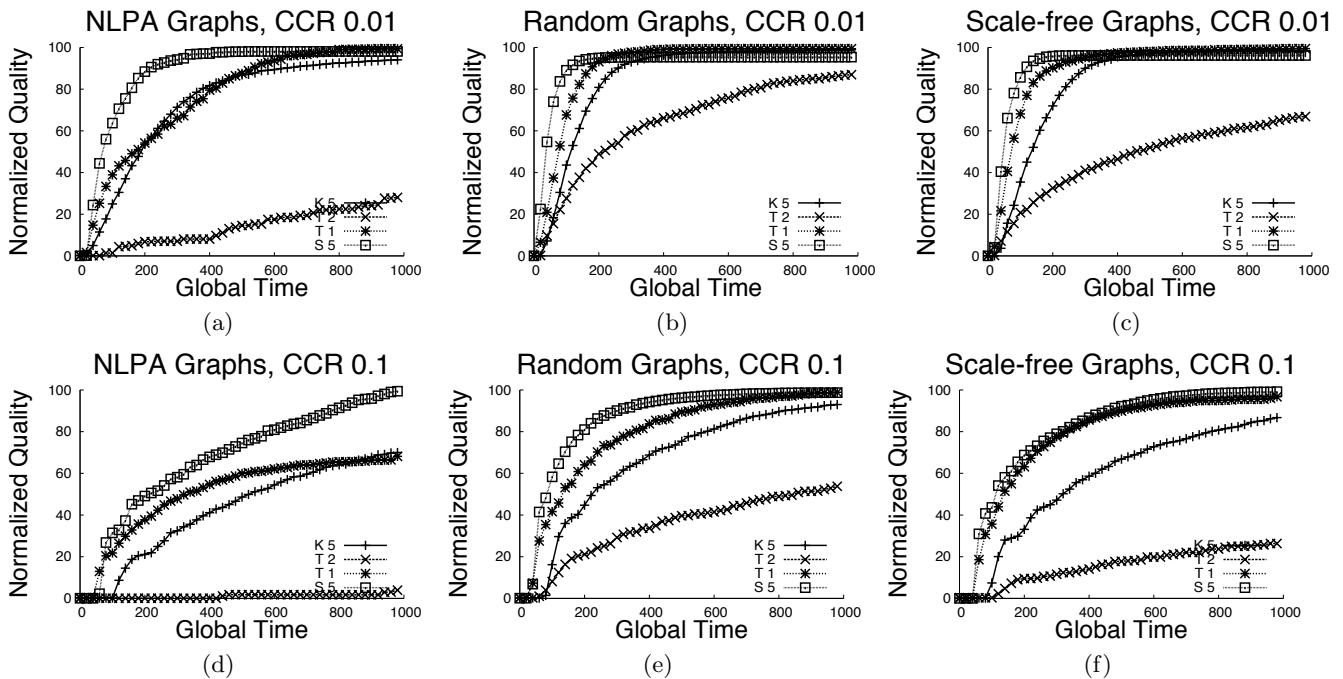


Figure 4: Experimental results comparing DALO for $K5$, $T1$, $T2$ and $S5$ regions.

ear time but does not ensure tightness. Moreover, we prove that the bounds provided for size and distance optimality are particular instances of the \mathcal{C} -optimal bounds.

To illustrate how the \mathcal{C} -optimality framework allows us to explore the space for arbitrary criteria, we proposed a novel criterion to generate regions, the so-called size-bounded-distance criterion. This new criterion has been designed to overcome the main drawbacks of size and distance optimality. Moreover, we extend the DALO algorithm [6] to compute \mathcal{C} -optimal solutions. Our empirical analysis of the size-bounded-distance criterion shows that it outperforms both size and distance criteria by providing a more fine-grained control of the complexity of the regions to explore.

As future work, we plan to extend the \mathcal{C} -optimal bounds to exploit some a-priori knowledge of the reward structure of the problem, if available, along the lines of [3]. Furthermore, since a critical issue in the design of any \mathcal{C} -optimal algorithm is the choice of regions, we will focus on defining techniques that allow us to explore the space of regions in search for regions with limited complexity and high quality guarantees.

Acknowledgments. Members of the Teamcore Research Group thanks Perceptionics Inc for their support of the research. Work funded by projects EVE (TIN2009-14702-C02-01 and 02), AT(CONSOLIDER CSD2007-0022), and Generalitat de Catalunya (2009-SGR-1434). Vinyals is supported by the Ministry of Education of Spain (FPU grant AP2006-04636).

6. REFERENCES

- [1] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
- [2] B. Bollobas. *Random Graphs*. Cambridge Press, 2001.
- [3] E. Bowring, J. P. Pearce, C. Portway, M. Jain, and M. Tambe. On k -optimal distributed constraint optimization algorithms: new bounds and algorithms. In *AAMAS (2)*, pages 607–614, 2008.
- [4] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *AAMAS (2)*, pages 639–646, 2008.
- [5] R. Junges and A. L. C. Bazzan. Evaluating the performance of DCOP algorithms in a real world, dynamic problem. In *AAMAS*, pages 599–606, 2008.
- [6] C. Kiekintveld, Z. Yin, A. Kumar, and M. Tambe. Asynchronous algorithms for approximate distributed constraint optimization with quality bounds. In *AAMAS*, pages 133–140, 2010.
- [7] R. Mailler and V. R. Lesser. Solving distributed constraint optimization problems using cooperative mediation. In *AAMAS*, pages 438–445, 2004.
- [8] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. Adopt: asynchronous distributed constraint optimization with quality guarantees. *Artif. Intell.*, 161(1-2):149–180, 2005.
- [9] J. P. Pearce and M. Tambe. Quality guarantees on k -optimal solutions for distributed constraint optimization problems. In *IJCAI*, pages 1446–1451, 2007.
- [10] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *IJCAI*, pages 266–271, 2005.
- [11] M. Vinyals, J. Cerquides, A. Farinelli, and J. A. Rodriguez-Aguilar. Worst-case bounds on the quality of max-product fixed-points. In *NIPS*, December 2010.
- [12] M. Vinyals, M. Pujol, J. Rodriguez-Aguilar, and J. Cerquides. Divide and Coordinate: solving DCOPs by agreement. In *AAMAS*, pages 150–156, 2010.
- [13] M. Vinyals, E. Shieh, J. Cerquides, J. A. Rodriguez-Aguilar, Z. Yin, M. Tambe, and E. Bowring. LP formulation of regional-optimal bounds. Technical Report TR-III-A-2011-01, 2011. At: <http://www.iiia.csic.es/files/pdfs/TR201101.pdf>.
- [14] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg. Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artif. Intell.*, 161(1-2):55–87, 2005.