Quality Guarantees on *k*-Optimal Solutions for Distributed Constraint Optimization Problems

Jonathan P. Pearce and Milind Tambe*
University of Southern California
Computer Science Department
{jppearce@usc.edu, tambe@usc.edu}

Abstract

A distributed constraint optimization problem (DCOP) is a formalism that captures the rewards and costs of local interactions within a team of agents. Because complete algorithms to solve DCOPs are unsuitable for some dynamic or anytime domains, researchers have explored incomplete DCOP algorithms that result in locally optimal solutions. One type of categorization of such algorithms, and the solutions they produce, is *k*-optimality; a *k*-optimal solution is one that cannot be improved by any deviation by *k* or fewer agents. This paper presents the first known guarantees on solution quality for *k*-optimal solutions. The guarantees are independent of the costs and rewards in the DCOP, and once computed can be used for any DCOP of a given constraint graph structure.

1 Introduction

In a large class of multi-agent scenarios, a set of agents chooses a joint action as a combination of individual actions. Often, the locality of agents' interactions means that the utility generated by each agent's action depends only on the actions of a subset of the other agents. In this case, the outcomes of possible joint actions can be compactly represented in cooperative domains by a distributed constraint optimization problem (DCOP) [Modi *et al.*, 2005; Zhang *et al.*, 2005a]. A DCOP can take the form of a graph in which each node is an agent and each edge denotes a subset of agents whose actions, taken together, incur costs or rewards to the agent team. Applications of DCOP include sensor networks [Modi *et al.*, 2005], meeting scheduling [Petcu and Faltings, 2005] and RoboCup soccer [Vlassis *et al.*, 2004].

Globally optimal DCOP algorithms can incur large computation or communication costs for domains where the number of agents is large or where time is limited. However, incomplete algorithms in which agents react on the basis of local knowledge of neighbors and constraint utilities can lead to a system that scales up easily and is more robust to dynamic environments. Researchers have introduced *k*-optimal algorithms in which small groups of agents optimize based on their local constraints, resulting in a *k*-optimal DCOP assignment, in which no subset of *k* or fewer agents can improve the overall solution. Some examples include the 1-optimal algorithms DBA [Yokoo and Hirayama, 1996] and DSA [Fitzpatrick and Meertens, 2003] for distributed constraint satisfaction problems (DisCSPs), which were later extended to DCOPs [Zhang *et al.*, 2005a], as well as the 2-optimal algorithms in [Maheswaran *et al.*, 2004], in which optimization was done by agents acting in pairs. Previous work has focused on upper bounds on the number of *k*-optima in DCOPs [Pearce *et al.*, 2006], as well as experimental analysis of *k*-optimal algorithms [Zhang *et al.*, 2005a; Maheswaran *et al.*, 2004].

Unfortunately, the lack of theoretical guarantees on the quality of solutions obtained by *k*-optimal algorithms was a fundamental limitation; until now, we could not guarantee a lower bound on the quality of the solution obtained with respect to the quality of the global optimum. In this paper, we introduce such guarantees. These guarantees can help determine an appropriate *k*-optimal algorithm, or possibly an appropriate constraint graph structure, for agents to use in situations where the cost of coordination between agents must be weighed against the quality of the solution reached. If increasing the value of *k* will provide a large increase in guaranteed solution quality, it may be worth the extra computation or communication required to reach a higher *k*-optimal solution. For example, consider a team of autonomous underwater vehicles (AUVs) [Zhang *et al.*, 2005b] that must quickly choose a joint action in order to observe some transitory underwater phenomenon. The combination of individual actions by nearby AUVs may generate costs or rewards to the team, and the overall utility of the joint action is determined by their sum. If this problem were represented as a DCOP, nearby AUVs would share constraints in the graph, while far-away AUVs would not. However, the actual rewards on these constraints may not be known until the AUVs are deployed, and in addition, due to time constraints, an incomplete, *k*-optimal algorithm, rather than a complete algorithm, must be used to find a solution. In this case, worst-case quality guarantees for *k*-optimal solutions for a given *k*, that are independent of the actual costs and rewards in the DCOP, are useful to help

*This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010.

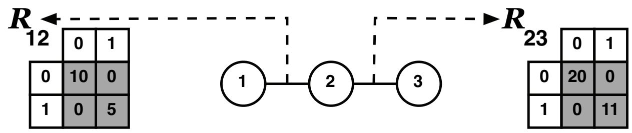

Figure 1: DCOP example

decide which algorithm to use. Alternatively, the guarantees can help to choose between different AUV formations, i. e. different constraint graphs.

We present two distinct types of guarantees for k-optima. The first, in Sections 3 and 4, is a lower bound on the quality of any k-optimum, expressed as a fraction of the quality of the optimal solution. The second, in Section 5, is a lower bound on the proportion of all DCOP assignments that a k-optimum must dominate in terms of quality. This type is useful in approximating the difficulty of finding a better solution than a given k-optimum. For both, we provide general bounds that apply to all constraint graph structures, as well as tighter bounds made possible if the graph is known in advance.

2 DCOP and k-optima

We consider a DCOP in which each agent controls a variable to which it must assign a value. Constraints exist on subsets of these variables; each constraint generates a cost or reward to the team based on the values assigned to each variable in the corresponding subset. Although we assume in this paper that each agent controls a single variable, all results are valid for cases in which agents control more than one variable.

Formally, a DCOP is a set of variables (one per agent) $N := \{1, \ldots, n\}$ and a set of domains $\mathcal{A} := \{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$, where the i^{th} variable takes value $a_i \in \mathcal{A}_i$. We denote the assignment of the multi-agent team by $a = [a_1 \cdots a_n]$. Valued constraints exist on various subsets $S \subset N$ of these variables. A constraint on S is expressed as a reward function $R_S(a)$. This function represents the reward generated by the constraint on S when the agents take assignment a; costs are expressed as negative rewards. θ is the set of all such subsets S on which a constraint exists, and no $S \in \theta$ is a subset of any other $S \in \theta$. For convenience, we will refer to these subsets S as "constraints" and the functions $R_S(\cdot)$ as "constraint reward functions." The solution quality for a particular complete assignment a is the sum of the rewards for that assignment from all constraints in the DCOP: $R(a) = \sum_{S \in \theta} R_S(a)$.

In [Pearce *et al.*, 2006], the *deviating group* between two assignments, a and $\tilde{a}$, was defined as $D(a, \tilde{a}) := \{i \in N : a_i \neq \tilde{a}_i\}$, i.e. the set of variables whose values in $\tilde{a}$ differ from their values in a. The *distance* between two assignments was defined as $d(a, \tilde{a}) := |D(a, \tilde{a})|$ where $|\cdot|$ denotes the size of the set. An assignment a is classified as a *k-optimum* if $R(a) - R(\tilde{a}) \geq 0 \; \forall \tilde{a}$ such that $d(a, \tilde{a}) \leq k$. Equivalently, at a k-optimum, no subset of k or fewer agents can improve the overall reward by choosing different values; every such subset is acting optimally given the values of the others.

Example 1 *Figure 1 is a binary DCOP in which agents choose values from* $\{0, 1\}$*, with constraints* $S_1 = \{1, 2\}$ *and* $S_2 = \{2, 3\}$ *with rewards shown. The assignment* $a = [1\ 1\ 1]$ *is* 1*-optimal because any single agent that deviates reduces the team reward. However,* $[1\ 1\ 1]$ *is not* 2*-optimal because if the group* $\{2, 3\}$ *deviated, making the assignment* $\tilde{a} = [1\ 0\ 0]$*, team reward would increase from 16 to 20. The globally optimal solution,* $a^* = [0\ 0\ 0]$ *is k-optimal for all* $k \in \{1, 2, 3\}$*.*□

In addition to categorizing local optima, k-optimality provides a natural classification for DCOP algorithms. Many algorithms are guaranteed to converge to k-optima, including DBA [Zhang *et al.*, 2005a], DSA [Fitzpatrick and Meertens, 2003], and coordinate ascent [Vlassis *et al.*, 2004] for $k = 1$, and MGM-2 and SCA-2 [Maheswaran *et al.*, 2004] for $k = 2$. Globally optimal algorithms such as Adopt [Modi *et al.*, 2005], OptAPO [Mailler and Lesser, 2004] and DPOP [Petcu and Faltings, 2005] converge to a k-optimum for $k = n$.

3 Quality guarantees on k-optima

This section provides reward-independent guarantees on solution quality for any k-optimal DCOP assignment. If we must choose a k-optimal algorithm for agents to use, it is useful to see how much reward will be gained or lost in the worst case by choosing a higher or lower value for k. We assume the actual costs and rewards on the DCOP are not known *a priori* (otherwise the DCOP could be solved centrally ahead of time). We provide a guarantee for a k-optimal solution as a fraction of the reward of the optimal solution, assuming that all rewards in the DCOP are non-negative (the reward structure of any DCOP can be normalized to one with all non-negative rewards as long as no infinitely large costs exist).

Proposition 1 *For any DCOP of n agents, with maximum constraint arity of m, where all constraint rewards are non-negative, and where* a^* *is the globally optimal solution, then, for any k-optimal assignment, a, where* $m \leq k < n$*,*

$$R(a) \geq \frac{\binom{n-m}{k-m}}{\binom{n}{k} - \binom{n-m}{k}} R(a^*). \tag{1}$$

Proof: By the definition of k-optimality, any assignment $\tilde{a}$ such that $d(a, \tilde{a}) \leq k$ must have reward $R(\tilde{a}) \leq R(a)$. We call this set of assignments $\tilde{A}$. Now consider any non-null subset $\hat{A} \subset \tilde{A}$. For any assignment $\hat{a} \in \hat{A}$, the constraints θ in the DCOP can be divided into three discrete sets, given a and $\hat{a}$:

- $\theta_1(a, \hat{a}) \subset \theta$ such that $\forall S \in \theta_1(a, \hat{a}), S \subset D(a, \hat{a})$.
- $\theta_2(a, \hat{a}) \subset \theta$ s.t. $\forall S \in \theta_2(a, \hat{a}), S \cap D(a, \hat{a}) = \emptyset$.
- $\theta_3(a, \hat{a}) \subset \theta$ s.t. $\forall S \in \theta_3(a, \hat{a}), S \notin \theta_1(a, \hat{a}) \cup \theta_2(a, \hat{a})$.

$\theta_1(a, \hat{a})$ contains the constraints that include only the variables in $\hat{a}$ which have deviated from their values in a; $\theta_2(a, \hat{a})$ contains the constraints that include only the variables in $\hat{a}$ which have not deviated from a; and $\theta_3(a, \hat{a})$ contains the constraints that include at least one of each. Thus:

$$R(\hat{a}) = \sum_{S \in \theta_1(a,\hat{a})} R_S(\hat{a}) + \sum_{S \in \theta_2(a,\hat{a})} R_S(\hat{a}) + \sum_{S \in \theta_3(a,\hat{a})} R_S(\hat{a}).$$

And, the sum of rewards of all assignments $\hat{a}$ in $\hat{A}$ is:

$$\sum_{\hat{a} \in \hat{A}} R(\hat{a}) = \sum_{\hat{a} \in \hat{A}} \Big(\sum_{S \in \theta_1(a,\hat{a})} R_S(\hat{a}) + \sum_{S \in \theta_2(a,\hat{a})} R_S(\hat{a}) + \sum_{S \in \theta_3(a,\hat{a})} R_S(\hat{a}) \Big)$$
$$\geq \sum_{\hat{a} \in \hat{A}} \sum_{S \in \theta_1(a,\hat{a})} R_S(\hat{a}) + \sum_{\hat{a} \in \hat{A}} \sum_{S \in \theta_2(a,\hat{a})} R_S(\hat{a}).$$

Since $R(a) > R(\hat{a}), \forall \hat{a} \in \hat{A}$,

$$R(a) \geq \frac{\sum_{\hat{a}\in\hat{A}} \sum_{S\in\theta_1(a,\hat{a})} R_S(\hat{a}) + \sum_{\hat{a}\in\hat{A}} \sum_{S\in\theta_2(a,\hat{a})} R_S(\hat{a})}{|\hat{A}|}. \quad (2)$$

Now, if the two numerator terms and the denominator can be expressed in terms of $R(a^*)$ and $R(a)$, then we have a bound on $R(a)$ in terms of $R(a^*)$. To do this, we consider the particular $\hat{A}$ which contains all assignments $\hat{a}$ such that $d(a, \hat{a}) = k$, and $\forall \hat{a} \in \hat{A}, \forall \hat{a}_i \in D(a, \hat{a}), \hat{a}_i = a^*_i$. This means that exactly k variables in $\hat{a}$ have deviated from their value in a, and these variables are taking the same values that they had in a^*.

There are $\binom{d(a,a^*)}{k}$ assignments $\hat{a} \in \hat{A}$. For *every* constraint $S \in \theta$, there are exactly $\binom{d(a,a^*)-|S|}{k-|S|}$ different assignments $\hat{a} \in \hat{A}$ for which $S \in \theta_1(a, \hat{a})$. This is because there exists a unique $\hat{a} \in \hat{A}$ for every subset of k variables in $D(a, a^*)$. If $S \subset D(a, \hat{a})$, as stipulated by the definition of $\theta_1(a, \hat{a})$, then there are $d(a, a^*) - |S|$ remaining variables from which $k - |S|$ must be chosen to complete $D(a, \hat{a})$, and so there are $\binom{d(a,a^*)-|S|}{k-|S|}$ possible assignments $\hat{a}$ for which this is true. For all $\hat{a}$, for all $S \in \theta_1(a, \hat{a})$, $R_S(\hat{a}) = R_S(a^*)$, so $\sum_{\hat{a}\in\hat{A}} \sum_{S\in\theta_1(a,\hat{a})} R_S(\hat{a}) = \sum_{S\in\theta} \binom{d(a,a^*)-|S|}{k-|S|} R_S(a^*) \geq \binom{d(a,a^*)-m}{k-m} R(a^*)$.

Similarly, for every constraint $S \in \theta$, there are $\binom{d(a,a^*)-|S|}{k}$ different assignments $\hat{a} \in \hat{A}$ for which $S \in \theta_2(a, \hat{a})$. If $S \cap D(a, \hat{a}) = \emptyset$, as stipulated by the definition of $\theta_2(a, \hat{a})$, then there are $d(a, a^*) - |S|$ remaining variables from which k must be chosen to complete $D(a, \hat{a})$, and so there are $\binom{d(a,a^*)-|S|}{k}$ possible assignments $\hat{a}$ for which this is true. For all $\hat{a}$, for all $S \in \theta_2(a, \hat{a}), R_S(\hat{a}) = R_S(a)$, and, so $\sum_{\hat{a}\in\hat{A}} \sum_{S\in\theta_2(a,\hat{a})} R_S(\hat{a}) = \sum_{S\in\theta} \binom{d(a,a^*)-|S|}{k} R_S(a) \geq \binom{d(a,a^*)-m}{k} R(a)$.

Therefore, from Equation 2,

$$R(a) \geq \frac{\binom{d(a,a^*)-m}{k-m} R(a^*) + \binom{d(a,a^*)-m}{k} R(a)}{\binom{d(a,a^*)}{k}} \geq \frac{\binom{d(a,a^*)-m}{k-m}}{\binom{d(a,a^*)}{k} - \binom{d(a,a^*)-m}{k}} R(a^*)$$

which is minimized when $d(a, a^*) = n$, so Equation 1 holds as a guarantee for a k-optimum in any DCOP. It is possible that $k > n - m$; in this case we take $\binom{n-m}{k}$ to be 0. ■

For binary DCOPs ($m = 2$), Equation 1 simplifies to:

$$R(a) \geq \frac{(k-1)}{(2n-k-1)} R(a^*).$$

The following example illustrates Proposition 1:

Example 2 *Consider a DCOP with five variables numbered 1 to 5, with domains of* {0,1}*, fully connected with binary constraints between all variable pairs. Suppose that* $a =$ [0 0 0 0 0] *is a 3-optimum, and that* $a^* =$ [1 1 1 1 1] *is the global optimum. Then* $d(a, a^*) = 5$*, and* $\hat{A}$ *contains* $\binom{d(a,a^*)}{k} = 10$ *assignments:* [1 1 1 0 0], [1 1 0 1 0], [1 1 0 0 1], [1 0 1 1 0], [1 0 1 0 1], [1 0 0 1 1], [0 1 1 1 0], [0 1 1 0 1], [0 1 0 1 1], [0 0 1 1 1]*. Whatever the values of the rewards are, every constraint reward* $R_S(a^*)$ *will equal* $R_S(\hat{a})$ *for* $\binom{n-2}{k-2} = 3$ *assignments in* $\hat{A}$ *(e.g.* $R_{\{1,2\}}(a^*) = R_{\{1,2\}}(\hat{a})$ *for* $\hat{a} =$ [1 1 1 0 0], [1 1 0 1 0]*, and* [1 1 0 0 1]*) and similarly, every constraint reward* $R_S(a)$ *equals* $R_S(\hat{a})$ *for* $\binom{n-2}{k} = 1$ *assignment in* $\hat{A}$*. Thus,* $R(a) \geq \frac{3}{10-1} R(a^*) = \frac{1}{3} R(a^*)$.□

We now show that Proposition 1 is tight, i.e. that there exist DCOPs with k-optima of quality equal to the bound.

Proposition 2 $\forall n, m, k$ *such that* $m \leq k < n$*, there exists some DCOP with* n *variables, with maximum constraint arity* m *with a* k*-optimal assignment,* a*, such that, if* a^* *is the globally optimal solution,*

$$R(a) = \frac{\binom{n-m}{k-m}}{\binom{n}{k} - \binom{n-m}{k}} R(a^*). \quad (3)$$

Proof: Consider a fully-connected m-ary DCOP where the domain of each variable contains at least two values {0,1} and every constraint R_S contains the following reward function:

$$R_S(a) = \begin{cases} \frac{\binom{n-m}{k-m}}{\binom{n}{k} - \binom{n-m}{k}} & , \forall i \in S, a_i = 0 \\ 1 & , \forall i \in S, a_i = 1 \\ 0 & , \text{otherwise} \end{cases}$$

The optimal solution a^* is $a^*_i = 1, \forall i$. If a is defined such that $a_i = 0, \forall i$, then Equation 3 is true. Now we show that a is k-optimal. For any assignment $\hat{a}$, such that $d(a, \hat{a}) = k$,

$$\begin{aligned} R(\hat{a}) &= \sum_{S\in\theta_1(a,\hat{a})} R(\hat{a}_S) + \sum_{S\in\theta_2(a,\hat{a})} R(\hat{a}_S) + \sum_{S\in\theta_3(a,\hat{a})} R(\hat{a}_S). \\ &\leq \binom{k}{m} + \binom{n-k}{m} \frac{\binom{n-m}{k-m}}{\binom{n}{k} - \binom{n-m}{k}} + 0 = \frac{\binom{k}{m}\left[\binom{n}{k} - \binom{n-m}{k}\right] + \binom{n-k}{m}\binom{n-m}{k-m}}{\binom{n}{k} - \binom{n-m}{k}} \\ &= \frac{n!}{m!(k-m)!(n-k)!} \div \frac{n!(n-m-k)! - (n-m)!(n-k)!}{k!(n-k)!(n-m-k)!} \\ &= \frac{n!k!(n-m-k)!}{m!(k-m)![n!(n-m-k)! - (n-k)!(n-m)!]} \\ &= \binom{n}{m} \frac{(n-m)!k!(n-m-k)!}{(k-m)![n!(n-m-k)! - (n-m)!(n-k)!]} \\ &= \binom{n}{m} \frac{\binom{n-m}{k-m} k!(n-m-k)!}{\frac{n!(n-m-k)!}{(n-k)!} - (n-m)!} = \binom{n}{m} \frac{\binom{n-m}{k-m}}{\binom{n}{k} - \binom{n-m}{k}} = R(a) \end{aligned}$$

because in a, each of the $\binom{n}{m}$ constraints in the DCOP are producing the same reward. Since this can be shown for $d(a, \hat{a}) = j, \forall j$ such that $1 \leq j \leq k$, a is k-optimal. ■

4 Graph-based quality guarantees

The guarantee for k-optima in Section 3 applies to all possible DCOP graph structures. However, knowledge of the structure of constraint graphs can be used to obtain tighter guarantees. This is done by again expressing the two numerator terms in Equation 2 as multiples of $R(a^*)$ and $R(a)$. However, for a sparse graph, if $\hat{A}$ is chosen as defined in Proposition 1, there may be many assignments $\hat{a} \in \hat{A}$ that have few or no constraints S in $\theta_1(a, \hat{a})$ because the variables in $D(a, \hat{a})$ may not share any constraints. Instead, exploiting the graph structure by choosing a smaller $\hat{A}$ can lead to a tighter bound. We can take $\hat{A}$ from Proposition 1, i.e. $\hat{A}$ which contains all $\hat{a}$ such that $d(a, \hat{a}) = k$ and $\forall \hat{a} \in \hat{A}, \forall \hat{a}_i \in D(a, \hat{a}), \hat{a}_i = a^*_i$. Then, we restrict this $\hat{A}$ further, so that $\forall \hat{a} \in \hat{A}$, the variables in $D(a, \hat{a})$ form a connected subgraph of the DCOP graph (or hypergraph), meaning that any two variables in $D(a, \hat{a})$ must be connected by some chain of constraints. This allows us to again transform Equation 2 to express $R(a)$ in terms of $R(a^*)$;

this new method can produce tighter guarantees for k-optima in sparse graphs. As an illustration, provably tight guarantees for binary DCOPs on ring graphs (each variable has two constraints) and star graphs (each variable has one constraint except the central variable, which has $n-1$) are given below.

Proposition 3 *For any binary DCOP of n agents with a ring graph structure, where all constraint rewards are non-negative, and a^* is the globally optimal solution, then, for any k-optimal assignment, a, where $k < n$,*

$$R(a) \geq \frac{k-1}{k+1} R(a^*). \tag{4}$$

Proof: Returning to Equation 2, $|\hat{A}| = n$ because $D(a, \hat{a})$ could consist of any of the n connected subgraphs of k variables in a ring. For any constraint $S \in \theta$, there are $k-1$ assignments $\hat{a} \in \hat{A}$ for which $S \in \theta_1(a, \hat{a})$ because there are $k-1$ connected subgraphs of k variables in a ring that contain S. Therefore, $\sum_{\hat{a} \in \hat{A}} \sum_{S \in \theta_1(a,\hat{a})} R_S(\hat{a}) = (k-1)R(a^*)$. Also, there are $n-k-1$ assignments $\hat{a} \in \hat{A}$ for which $S \in \theta_2(a, \hat{a})$ because there are $n-k-1$ ways to choose S in a ring so that it does not include any variable in a given connected subgraph of k variables. Therefore, $\sum_{\hat{a} \in \hat{A}} \sum_{S \in \theta_2(a,\hat{a})} R_S(\hat{a}) = (n-k-1)R(a)$. So, from Equation 2,

$$R(a) \geq \frac{(k-1)R(a^*) + (n-k-1)R(a)}{n}$$

and therefore Equation 4 holds. ■

Proposition 4 *For any binary DCOP of n agents with a star graph structure, where all constraint rewards are non-negative, and a^* is the globally optimal solution, then, for any k-optimal assignment, a, where $k < n$,*

$$R(a) \geq \frac{k-1}{n-1} R(a^*). \tag{5}$$

Proof: The proof is similar to the previous proof. In a star graph, there are $\binom{n-1}{k-1}$ subgraphs of k variables, and therefore $|\hat{A}| = \binom{n-1}{k-1}$. Every constraint $S \in \theta$ includes the central variable and one other variable, and thus there are $\binom{n-2}{k-2}$ connected subgraphs of k variables that contain S, and therefore $\sum_{\hat{a} \in \hat{A}} \sum_{S \in \theta_1(a,\hat{a})} R_S(\hat{a}) = \binom{n-2}{k-2} R(a^*)$. Finally, there are no ways to choose S so that it does not include any variable in a given connected subgraph of k variables. Therefore, $\sum_{\hat{a} \in \hat{A}} \sum_{S \in \theta_2(a,\hat{a})} R_S(\hat{a}) = 0R(a)$. So, from Equation 2,

$$R(a) \geq \frac{\binom{n-2}{k-2} R(a^*) + 0R(a)}{\binom{n-1}{k-1}}$$

and therefore Equation 5 holds. ■

Tightness can be proven by constructing DCOPs on ring and chain graphs with the same rewards as in Proposition 2; proofs are omitted for space. The bound for rings can also be applied to chains, since any chain can be expressed as a ring where all rewards on one constraint are zero.

Finally, bounds for DCOPs with arbitrary graphs and non-negative constraint rewards can be found using a linear-fractional program (LFP). This method gives a tight bound for any graph, since it instantiates the rewards for all constraints, but requires a globally optimal solution to the LFP, in contrast to the constant-time guarantees of Equations 1, 4 and 5. An LFP such as this is reducible to a linear program (LP) [Boyd and Vandenberghe, 2004]. The objective is to minimize $\frac{R(a)}{R(a^*)}$ such that $\forall \tilde{a} \in \tilde{A}, R(a) - R(\tilde{a}) \geq 0$, given $\tilde{A}$ as defined in Proposition 1. Note that $R(a^*)$ and $R(a)$ can be expressed as $\sum_{S \in \theta} R_S(a^*)$ and $\sum_{S \in \theta} R_S(a^*)$. We can now transform the DCOP so that every $R(\tilde{a})$ can also be expressed in terms of sums of $R_S(a^*)$ and $R_S(a)$, without changing or invalidating the guarantee on $R(a)$. Therefore, the LFP will contain only two variables for each $S \in \theta$, one for $R_S(a^*)$ and one for $R_S(a)$, where the domain of each one is the set of non-negative real numbers. The transformation is to set all reward functions $R_S(\cdot)$ for all $S \in \theta$ to 0, except for two cases: when all variables $i \in S$ have the same value as in a^*, or when all $i \in S$ have the same value as in a. This has no effect on $R(a^*)$ or $R(a)$, because $R_S(a^*)$ and $R_S(a)$ will be unchanged for all $S \in \theta$. It also has no effect on the optimality of a^* or the k-optimality of a, since the only change is to reduce the global reward for assignments other than a^* and a. Thus, the tight lower bound on $\frac{R(a)}{R(a^*)}$ still applies to the original DCOP.

5 Domination analysis of k-optima

In this section we now provide a different type of guarantee: lower bounds on the proportion of all possible DCOP assignments which any k-optimum must dominate in terms of solution quality. This proportion, called a *domination ratio*, provides a guide for how difficult it may be to find a solution of higher quality than a k-optimum; this metric is commonly used to evaluate heuristics for combinatorial optimization problems [Gutin and Yeo, 2005].

For example, suppose for some k, the solution quality guarantee from Section 4 for any k-optimum was 50% of optimal, but, additionally, it was known that any k-optimum was guaranteed to dominate 95% of all possible assignments to the DCOP. Then, at most only 5% of the other assignments could be of higher quality, indicating that it would likely be computationally expensive to find a better assignment, either with a higher k algorithm, or by some other method, and so a k-optimal algorithm should be used despite the low guarantee of 50% of the optimal solution quality. Now suppose instead for the same problem, the k-optimum was guaranteed to dominate only 20% of all assignments. Then it becomes more likely that a better solution could be found quickly, and so the k-optimal algorithm might not be recommended.

To find the domination ratio, observe that any k-optimum a must be of the same or higher quality than all $\tilde{a} \in \tilde{A}$ as defined in Proposition 1. So, the ratio is:

$$\frac{1 + |\tilde{A}|}{\prod_{i \in N} |\mathcal{A}_n|}. \tag{6}$$

If the constraint graph is fully connected (or not known, and so must be assumed to be fully connected), and each variable has q values, then $|\tilde{A}| = \sum_{j=1}^{k} \binom{n}{j} (q-1)^j$ and $\prod_{i \in N} |\mathcal{A}_n| = q^n$.

If the graph is known to be not fully connected, then the set $\tilde{A}$ from Equation 6 can be expanded to include assignments of distance greater than k from a, providing a stronger guarantee on the ratio of the assignment space that must be dominated by any k-optimum. Specifically, if a is k-optimal, then any assignment where *any number* of disjoint subsets of size $\leq k$ have deviated from a must be of the same or lower quality as a, as long as no constraint includes any two agents in different such subsets; this idea is illustrated below:

Example 3 *Consider a binary DCOP of five variables, numbered 1 to 5, with domains of two values, with unknown constraint graph. Any 3-optimum must be of equal or greater quality than* $\left(1+\binom{5}{1}+\binom{5}{2}+\binom{5}{3}\right)/2^5 = 81.25\%$ *of all possible assignments, i.e. where 0, 1, 2, or 3 agents have deviated.*

Now, suppose the graph is known to be a chain with variables ordered by number. Since a deviation by either the variables {1,2} or {4,5} cannot increase global reward, and no constraint exists across these subsets, then neither can a deviation by {1,2,4,5}, even though four variables are deviating. The same applies to {1,3,4,5} and {1,2,3,5}, since both are made up of subsets of three or fewer variables that do not share constraints. So, a 3-optimum is now of equal or greater quality than $\left(1+\binom{5}{1}+\binom{5}{2}+\binom{5}{3}+3\right)/2^5 = 90.63\%$ *of all assignments.* □

An improved guarantee can be found by enumerating the set $\tilde{A}$ of assignments $\tilde{a}$ with equal or lower reward than a; this set is expanded due to the DCOP graph structure as in the above example. The following proposition makes this possible; we introduce new notation for it: If we define n different subsets of agents as D_i for $i = 1 \ldots n$, we use $D^m = \cup_{i=1}^m D_i$, i.e. D^m is the union of the first m subsets. The proof is by induction over each subset D_i for $i = 1 \ldots n$.

Proposition 5 *Let a be some k-optimal assignment. Let* $\tilde{a}^n$ *be another assignment for which* $D(a, \tilde{a}^n)$ *can be expressed as* $D^n = \cup_{i=1}^n D_i$ *where:*

- $\forall D_i, |D_i| \leq k$. (subsets contain k or fewer agents)
- $\forall D_i, D_j, D_i \cap D_j = \emptyset$. (subsets are disjoint)
- $\forall D_i, D_j, \nexists i \in D_i, j \in D_j$ *such that* $i, j \in S$, *for any* $S \in \theta$. (no constraint exists between agents in different subsets)

Then, $R(a) \geq R(\tilde{a}^n)$.

Proof:

Base case: If $n = 1$ then $D^n = D_1$ and $R(a) \geq R(\tilde{a}^n)$ by definition of k-optimality.

Inductive step: $R(a) \geq R(\tilde{a}^{n-1}) \Rightarrow R(a) \geq R(\tilde{a}^n)$.

The set of all agents can be divided into the set of agents in D^{n-1}, the set of agents in D_n, and the set of agents not in D^n. Also, by inductive hypothesis, $R(a) \geq R(\tilde{a}^{n-1})$. Therefore,

$$R(a) = \sum_{S\in\theta:S\cap D^{n-1}\neq\emptyset} R_S(a) + \sum_{S\in\theta:S\cap D_n\neq\emptyset} R_S(a) + \sum_{S\in\theta:S\cap D^n=\emptyset} R_S(a)$$
$$\geq \sum_{S\in\theta:S\cap D^{n-1}\neq\emptyset} R_S(\tilde{a}_S^{n-1}) + \sum_{S\in\theta:S\cap D_n\neq\emptyset} R_S(\tilde{a}^{n-1}) + \sum_{S\in\theta:S\cap D^n=\emptyset} R_S(\tilde{a}^{n-1})$$

so $\sum_{S\in\theta:S\cap D^{n-1}\neq\emptyset} R_S(a) \geq \sum_{S\in\theta:S\cap D^{n-1}\neq\emptyset} R_S(\tilde{a}^{n-1})$ because the rewards from agents outside D^{n-1} are the same for a and $\tilde{a}^{n-1}$.

Let a' be an assignment such that $D(a, a') = D_n = D(\tilde{a}^{n-1}, \tilde{a}^n)$. Because a is k-optimal, $R(a) \geq R(a')$; therefore,

$$R(a) = \sum_{S\in\theta:S\cap D^{n-1}\neq\emptyset} R_S(a) + \sum_{S\in\theta:S\cap D_n\neq\emptyset} R_S(a) + \sum_{S\in\theta:S\cap D^n=\emptyset} R_S(a)$$
$$\geq \sum_{S\in\theta:S\cap D^{n-1}\neq\emptyset} R_S(a') + \sum_{S\in\theta:S\cap D_n\neq\emptyset} R_S(a') + \sum_{S\in\theta:S\cap D^n=\emptyset} R_S(a').$$

and so $\sum_{S\in\theta:S\cap D_n\neq\emptyset} R_S(a) \geq \sum_{S\in\theta:S\cap D_n\neq\emptyset} R_S(a')$ because the rewards from agents outside D_n are the same for a and a'.

We also know that $\sum_{S\in\theta:S\cap D^n=\emptyset} R_S(a) = \sum_{S\in\theta:S\cap D^n=\emptyset} R_S(\tilde{a}^n)$ because the rewards from agents outside D^n are the same for a and $\tilde{a}^n$; therefore,

$$R(a) \geq \sum_{S\in\theta:S\cap D^{n-1}\neq\emptyset} R_S(\tilde{a}^{n-1}) + \sum_{S\in\theta:S\cap D_n\neq\emptyset} R_S(a') + \sum_{S\in\theta:S\cap D^n=\emptyset} R_S(\tilde{a}^n)$$
$$= \sum_{S\in\theta:S\cap D^{n-1}\neq\emptyset} R_S(\tilde{a}^n) + \sum_{S\in\theta:S\cap D_n\neq\emptyset} R_S(\tilde{a}^n) + \sum_{S\in\theta:S\cap D^n=\emptyset} R_S(\tilde{a}^n)$$

because the rewards from D^{n-1} are the same for $\tilde{a}^{n-1}$ and $\tilde{a}^n$, and the rewards from D_n are the same for a' and $\tilde{a}^n$. Therefore, $R(a) \geq R(\tilde{a}^n)$. ■

6 Experimental results

While the main thrust of this paper is on theoretical guarantees for k-optima, this section gives an illustration of the guarantees in action, and how they are affected by constraint graph structure. Figures 2a, 2b, and 2c show quality guarantees for binary DCOPs with fully connected graphs, ring graphs, and star graphs, calculated directly from Equations 1, 4 and 5. Figure 2d shows quality guarantees for binary-tree DCOPs, obtained using the LFP from Section 4. The x-axis plots the value chosen for k, and the y-axis plots the lower bound for k-optima as a percentage of the optimal solution quality for systems of 5, 10, 15, and 20 agents. These results show how the worst-case benefit of increasing k varies depending on graph structure. For example, in a five-agent DCOP, a 3-optimum is guaranteed to be 50% of optimal whether the graph is a star or a ring. However, moving to $k = 4$ means that worst-case solution quality will improve to 75% for a star, but only to 60% for a ring. For fully connected graphs, the benefit of increasing k goes up as k increases; whereas for stars it stays constant, and for chains it decreases, except for when $k = n$. Results for binary trees are mixed.

Figure 3 shows the domination ratio guarantees for k-optima from Section 5, for DCOPs where variables have do-

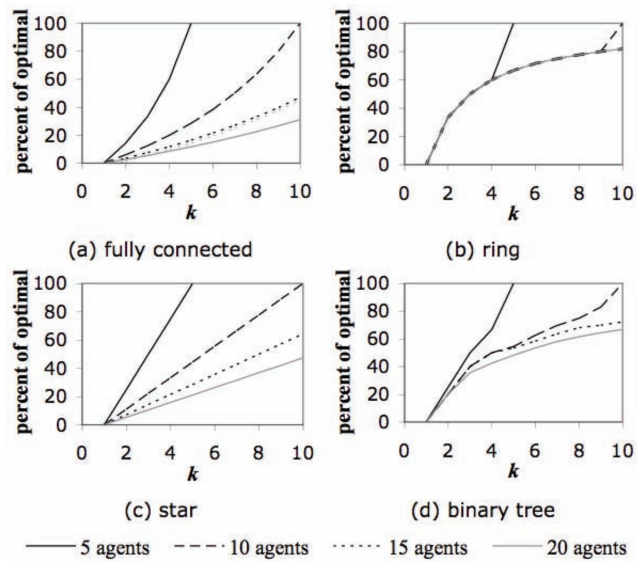

Figure 2: Quality guarantees for k-optima with respect to the global optimum for DCOPs of various graph structures.

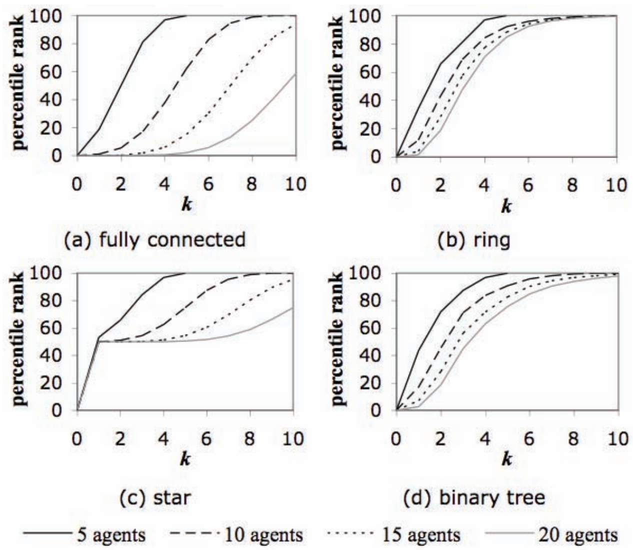

Figure 3: Domination ratio guarantees for k-optima for various graph structures.

mains of two values. This figure, when considered with Figure 2, provides insight into the difficulty of finding a solution of higher quality than a k-optimum. For example, a 7-optimum in a fully connected graph of 10 agents (Figure 2a) is only guaranteed to be 50% of optimal; however this 7-optimum is guaranteed to be of higher quality than 94.5% of all possible assignments to that DCOP (Figure 3a), which suggests that finding a better solution may be difficult. In contrast, a 3-optimum in a ring of 10 agents (Figure 2b) has the same guarantee of 50% of optimal solution, but this 3-optimum is only guaranteed to be of higher quality than 69% of all possible assignments, which suggests that finding a better solution may be easier.

7 Related work and conclusion

This paper contains the first guarantees on solution quality for k-optimal DCOP assignments. The performance of any local DCOP algorithms can now be compared in terms of worst case guaranteed solution quality, either on a particular constraint graph, or over all possible graphs. In addition, since the guarantees are reward-independent, they can be used for any DCOP of a given graph structure, once computed.

In [Pearce *et al.*, 2006], upper bounds on the number of possible k-optima that could exist in a given DCOP graph were presented. The work in this paper focuses instead on lower bounds on solution quality for k-optima for a given DCOP graph. This paper provides a complement to the experimental analysis of local optima (1-optima) arising from the execution of incomplete DCOP algorithms [Zhang *et al.*, 2005a; Maheswaran *et al.*, 2004]. However, in this paper, the emphasis is on the worst case rather than the average case.

The results in this paper can help illuminate the relationship between local and global optimality in many types of multi-agent systems, e. g. networked distributed POMDPs [Nair *et al.*, 2005]. All results in this paper also apply to centralized constraint reasoning. However, examining properties of solutions that arise from coordinated value changes of small groups of variables is especially useful in distributed settings, given the computational and communication expense of large-scale coordination.

References

[Boyd and Vandenberghe, 2004] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge U. Press, 2004.

[Fitzpatrick and Meertens, 2003] S. Fitzpatrick and L. Meertens. Distributed coordination through anarchic optimization. In V. Lesser, C. L. Ortiz, and M. Tambe, editors, *Distributed Sensor Networks: A Multiagent Perspective*, pages 257–295. Kluwer, 2003.

[Gutin and Yeo, 2005] G. Gutin and A. Yeo. Domination analysis of combinatorial optimization algorithms and problems. In M. Golumbic and I. Hartman, editors, *Graph Theory, Combinatorics and Algorithms: Interdisciplinary Applications*. Kluwer, 2005.

[Maheswaran *et al.*, 2004] R. T. Maheswaran, J. P. Pearce, and M. Tambe. Distributed algorithms for DCOP: A graphical-game-based approach. In *PDCS*, 2004.

[Mailler and Lesser, 2004] R. Mailler and V. Lesser. Solving distributed constraint optimization problems using cooperative mediation. In *AAMAS*, 2004.

[Modi *et al.*, 2005] P. J. Modi, W. Shen, M. Tambe, and M. Yokoo. Adopt: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1-2):149–180, 2005.

[Nair *et al.*, 2005] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *AAAI*, 2005.

[Pearce *et al.*, 2006] J. P. Pearce, R. T. Maheswaran, and M. Tambe. Solution sets for DCOPs and graphical games. In *AAMAS*, 2006.

[Petcu and Faltings, 2005] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *IJCAI*, 2005.

[Vlassis *et al.*, 2004] N. Vlassis, R. Elhorst, and J. R. Kok. Anytime algorithms for multiagent decision making using coordination graphs. In *Proc. Intl. Conf. on Systems, Man and Cybernetics*, 2004.

[Yokoo and Hirayama, 1996] M. Yokoo and K. Hirayama. Distributed breakout algorithm for solving distributed constraint satisfaction and optimization problems. In *ICMAS*, 1996.

[Zhang *et al.*, 2005a] W. Zhang, G. Wang, Z. Xing, and L. Wittenberg. Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1-2):55–87, 2005.

[Zhang *et al.*, 2005b] Y. Zhang, J. G. Bellingham, R. E. Davis, and Y. Chao. Optimizing autonomous underwater vehicles' survey for reconstruction of an ocean field that varies in space and time. In *American Geophysical Union, Fall meeting*, 2005.