Solution Sets in DCOPs and Graphical Games

Jonathan P. Pearce
Computer Science Dept.
Univ. of Southern California
Los Angeles, CA 90089
jppearce@usc.edu

Rajiv T. Maheswaran
Information Sciences Institute
Univ. of Southern California
Los Angeles, CA 90292
maheswar@isi.edu

Milind Tambe
Computer Science Dept.
Univ. of Southern California
Los Angeles, CA 90089
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, each of whom is choosing an individual action. When rapidly selecting a single joint action for a team, we typically solve DCOPs (often using locally optimal algorithms) to generate a single solution. However, in scenarios where a set of joint actions (i.e. a set of assignments to a DCOP) is to be generated, metrics are needed to help appropriately select this set and efficiently allocate resources for the joint actions in the set. To address this need, we introduce *k*-optimality, a metric that captures the desirable properties of diversity and relative quality of a set of locally-optimal solutions using a parameter that can be tuned based on the level of these properties required. To achieve effective resource allocation for this set, we introduce several upper bounds on the cardinalities of *k*-optimal joint action sets. These bounds are computable in constant time if we ignore the graph structure, but tighter, graph-based bounds are feasible with higher computation cost. Bounds help choose the appropriate level of *k*-optimality for settings with fixed resources and help determine appropriate resource allocation for settings where a fixed level of *k*-optimality is desired. In addition, our bounds for a 1-optimal joint action set for a DCOP also apply to the number of pure-strategy Nash equilibria in a graphical game of noncooperative agents.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Design, Theory

Keywords

constraint reasoning, DCOP, multiagent systems, k-optimality

1. INTRODUCTION

In a large class of multi-agent scenarios, a set of agents chooses a joint action (JA) 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 JAs can be compactly represented in cooperative domains by a distributed constraint optimization problem (DCOP)[12, 20] or, in noncooperative domains, by a graphical game [6]. Each of these models can take the form of a graph (or hypergraph) in which each node is an agent and each edge (or hyperedge) denotes a subset of agents whose actions, when taken together, incur costs or rewards, either to the agent team (in DCOPs) or to individual agents (in graphical games). In the case of DCOP, if each agent controls a single variable, then a single JA is a complete assignment of values to variables (i.e. every agent chooses an individual action for itself). We focus primarily on the team setting, using DCOP, whose applications include multi-agent plan coordination [4], sensor networks [12], and RoboCup soccer [17].

Traditionally, researchers have focused on obtaining a single JA, expressed as a single assignment of actions to agents in a DCOP. However, in this paper, we consider a multi-agent system that generates a *set* of JAs, i.e. multiple assignments to the same DCOP. Generating sets of JAs is useful in domains such as disaster rescue (to provide multiple rescue options to a human commander) [15], patrolling (to execute multiple patrols in the same area) [14], training simulations (to provide several options to a student) and others [16]. We provide three key contributions to address such domains. The first contribution is to determine the appropriate metric for evaluating a set of JAs. While high absolute reward is an appropriate metric in single-solution domains, reward alone is a poor metric for these multiple-JA domains, as it often yields clusters of very similar JAs, as shown in Section 2 of this paper. Clustering is undesirable, as diversity, the difference among JAs, is a key property for a JA set in many domains [16]. Diversity alone is undesirable, because it leads to low-quality solutions. Hence, this paper introduces a new metric, *k*-optimality, that naturally captures the diversity and relative quality of a JA set. A *k*-optimal JA has the highest reward within a neighborhood of other JAs that differ from it by at most *k* individual actions; i.e. no *k*-optimal JA can be improved if *k* or fewer agents change actions. Therefore, *k*-optimality quantifies the neighborhood in which a JA is optimal. In a *k*-optimal JA set, defined as a set of JAs, each of which is itself *k*-optimal, all JAs in the set are guaranteed a particular level of relative quality (best in their neighborhoods), as well as diversity (every two JAs must be separated by at least $k + 1$ individual actions).

Domains requiring repeated patrols in an area by a team of UAVs (unmanned air vehicles), UGVs (unmanned ground vehicles), or robots, for peacekeeping or law enforcement after a disaster, provide one key illustration of the utility of *k*-optimality. For example, given a team of patrol robots in charge of executing multiple joint

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

patrols in an area as in [14], each robot may be assigned a region within the area. Each robot is controlled by a single agent, and hence, for one joint patrol, each agent must choose one of several possible routes to patrol within its region. A joint patrol is a JA, where each agent's action is the route it has chosen to patrol, and rewards and costs arise from the combination of routes patrolled by agents in adjacent or overlapping regions. For example, if two nearby agents choose routes that largely overlap on a low-activity street, the constraint between those agents would incur a cost, while routes that overlap on a high-activity street would generate a reward. Agents in distant regions would not share a constraint. If reward alone is used as a metric to select joint patrols, then all selected joint patrols could be the same, except for the action of one agent. This set of patrols would be repetitive and predictable to adversaries. If we pick some diverse joint patrols at random, they may be very low-quality patrols. Using k-optimality directly addresses such circumstances; k-optimality ensures that all joint patrols differ by at least $k+1$ agents' actions, as well as ensuring that this diversity would not come at the expense of obviously bad joint patrols, as each is optimal within a radius of at least k agents' actions.

After introducing k-optimality, our second key contribution in this paper is addressing efficient resource allocation for the multiple JAs in a k-optimal set, by defining tight upper bounds on the number of k-optimal JAs that can exist for a given DCOP. These bounds are necessitated by two key features of the typical domains where a k-optimal set is applicable. First, each JA in the set consumes some resources that must be allocated in advance. Such resource consumption arises because: (i) a team actually executes each JA in the set, as in our patrolling example above, or (ii) the JA set is presented to a human user (or another agent) as a list of options to choose from, requiring time. In each case, resources are consumed based on the JA set size. Second, while the existence of the constraints between agents is known *a priori*, the actual rewards and costs on the constraints depend on conditions that are not known until runtime, and so resources must be allocated before the rewards and costs are known and before the agents generate the k-optimal JA set. In the patrolling domain, constraints are known to exist between patrol robots assigned to adjacent or overlapping regions. However, their costs and rewards depend on recent field reports of adversarial activity that are not known until the robots are deployed. At this point the robots must already be fueled in order for them to immediately generate and execute a set of k-optimal patrols. The resource to be allocated to the robots is the amount of fuel required to execute each patrol; thus it is critical to ensure that enough fuel is given to each robot so that each JA found can be executed, without burdening the robots with wasted fuel that will go unused. Consider another domain involving a team of disaster rescue agents that must generate a set of k-optimal JAs in order to present a set of diverse options to a human commander, where each option represents the best JA within a neighborhood of similar JAs. The commander will choose one JA for the team to actually execute. Constraints exist between agents whose actions must be coordinated (i.e. members of subteams) but their costs and rewards depend on conditions on the ground that are unknown until the time when the agents must be deployed. Here, the resource is the time the commander has to make the decision. Presenting too many options will cause the commander to run out of time before considering them all, and presenting too few may cause high-quality options to be omitted.

Because each JA consumes resources, knowing the maximal number of k-optimal JAs that could exist for a given DCOP would allow us to allocate sufficient resources for a given level of k. Unfortunately, we cannot predict this number because the costs and rewards for the DCOP are not known in advance. Despite this uncertainty, reward-independent bounds can be obtained on the size of a k-optimal JA set, i.e. to safely allocate enough resources for a given value of k for any DCOP with a particular graph structure. We first identify a mapping to coding theory, yielding bounds independent of both reward and graph structure. We then provide a method to use the structure of the DCOP graph (or hypergraph of arbitrary arity) to obtain significantly tighter bounds.

The third key contribution in this paper is to establish a connection to noncooperative settings by proving that our bounds for 1-optima also apply to the number of pure-strategy Nash equilibria in any graphical game on a given graph, which remains an open problem. In addition to their uses in resource allocation, these bounds provide insight into the problem landscapes that can exist in both cooperative and noncooperative settings.

2. *k*-OPTIMALITY

We introduce the notion of *k-optimality* as a metric that captures both relative quality and diversity when selecting a set of JAs. We begin with our model of the multi-agent team problem, which is a DCOP in which each agent controls a single variable to which it must assign a value. These values correspond to individual actions that can be taken by the agents. Subgroups of agents, whose combined actions generate a cost or reward to the team, define the constraints between agents. Because we assume that each agent controls a single variable, we will use the terms "agent" and "variable" interchangeably. More formally, for a set of agents $\mathcal{I} := \{1, \ldots, I\}$, the i^{th} agent takes action $a_i \in \mathcal{A}_i$. We denote the joint action of a subgroup of agents $S \subset \mathcal{I}$ by $a_S := \times_{i \in S} a_i \in \mathcal{A}_S$ where $\mathcal{A}_S := \bigtimes_{i \in S} \mathcal{A}_i$ and the joint actions (JAs) of the entire multi-agent team by $a = [a_1 \cdots a_I] \in \mathcal{A}$ where $\mathcal{A} := \bigtimes_{i \in \mathcal{I}} \mathcal{A}_i$. The team reward for taking a particular JA, a, is an aggregation of the rewards obtained by subgroups in the team: $R(a) = \sum_{S \in \theta} R_S(a) = \sum_{S \in \theta} R_S(a_S)$ where S is a minimal subgroup that generates a reward (or incurs a cost) in an n-ary DCOP (i.e. a constraint), θ is the collection of all such minimal subgroups for a given problem and $R_S(\cdot)$ denotes a function that maps $\mathcal{A}_S$ to $\mathbb{R}$. By minimality, we mean that the reward component R_S cannot be decomposed further. Mathematically: $\forall S \in \theta, R_S(a_S) \neq R_{S_1}(a_{S_1}) + R_{S_2}(a_{S_2})$ for any $R_{S_1}(\cdot) : \mathcal{A}_{S_1} \to \mathbb{R}, R_{S_2}(\cdot) : \mathcal{A}_{S_2} \to \mathbb{R}, S_1, S_2 \subset \mathcal{I}$ such that $S_1 \cup S_2 = S, S_1, S_2 \neq \emptyset$. It is important to express the constraints minimally to accurately represent dependencies among agents.

To evaluate JA sets, specifically JAs with respect to each other, we need notions of neighborhood and distance among JAs. For two JAs, a and $\tilde{a}$, we define the following terms. The *deviating group* is $D(a, \tilde{a}) := \{i \in \mathcal{I} : a_i \neq \tilde{a}_i\}$, i.e. the set of agents whose actions in JA $\tilde{a}$ differ from their actions in JA a. The *distance* is $d(a, \tilde{a}) := |D(a, \tilde{a})|$ where $|\cdot|$ denotes the cardinality of the set. The *relative reward* of a JA a with respect to another JA $\tilde{a}$ is $\Delta(a, \tilde{a}) := R(a) - R(\tilde{a}) = \sum_{S \in \theta : S \cap D(a,\tilde{a}) \neq \emptyset} [R_S(a_S) - R_S(\tilde{a}_S)]$. In this summation, only the rewards on constraints incident on deviating agents are considered, since the other rewards remain the same. We assume every subgroup of agents G has a unique optimal subgroup joint action a_G^* for any context, where a context consists of a_{G^C}, the actions of the agents not in G, i.e. the complement of G. Mathematically, if $G \subset \mathcal{I}$ where $G \neq \emptyset$ and $G \neq \mathcal{I}$, then $\exists\, a_G^* \in \mathcal{A}_G$ *s.t.* $R(a_G^*; a_{G^C}) > R(a_G; a_{G^C}) \forall a_G \neq a_G^*$. Here the notation $R(a_G; a_{G^C})$ is used to indicate the overall team reward generated when subgroup G takes the JA a_G with respect to a fixed context of a_{G^C}. The above assumption is natural for domains where rewards come from precise measurements, and is common in work on bounds and estimates for numbers of local optima [2] and Nash equilibria [11]. Given this assumption, we now classify a as a k-

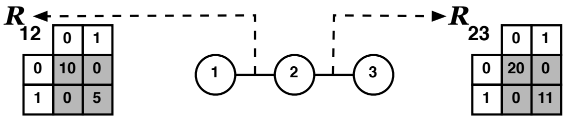


Figure 1: DCOP example

optimal JA or *k-optimum* if $\Delta(a,\tilde{a}) > 0\ \forall\tilde{a}\quad \text{s.t}\quad d(a,\tilde{a}) \leq k$. Equivalently, if the set of agents have reached a k-optimum, then no subgroup of cardinality $\leq k$ can improve the overall reward by choosing different actions; every such subgroup is acting optimally with respect to its context.

A collection of k-optimal JAs must be mutually separated by a distance $\geq k+1$ as they each have the highest reward within a radius of k. Thus, a higher k-optimality of a collection implies a greater level of relative reward and diversity. Let $A_q(I,k) = \{a \in \mathcal{A} : \Delta(a,\tilde{a}) > 0\ \forall\tilde{a}\ \text{ s.t }\ d(a,\tilde{a}) \leq k\}$ be the set of all k-optimal JAs for a team of I agents with domains of cardinality q. It is straightforward to show $A_q(I,k+1) \subseteq A_q(I,k)$.

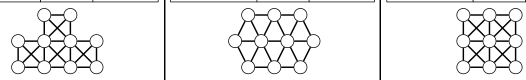
EXAMPLE 1. *Figure 1 is a binary DCOP in which agents choose actions from* $\{0,1\}$*, with rewards shown for the two constraints (minimal subgroups)* $S_1 = \{1,2\}$ *and* $S_2 = \{2,3\}$. *The assignment* $a = [1\ 1\ 1]$ *is* 1*-optimal because any single agent that deviates reduces the team reward. 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* $k \in \{1,2,3\}$.□

We now show, in an experiment, the advantages of k-optimal JA sets as capturing both diversity and high reward compared with JA sets chosen by other metrics. The lower half of Figure 2(a) shows a DCOP graph representing a team of 10 patrol robots, each of whom must choose one of two routes to patrol in its region. The nodes are agents and the edges represent binary constraints between agents assigned to overlapping regions. The actions (i.e. the chosen routes) of these agents combine to produce a cost or reward to the team. For each of 20 runs, the edges were initialized with rewards from a uniform random distribution. The set of all 1-optima was enumerated. Then, for the same DCOP, we found equal-sized sets of JAs using two other metrics. In one metric, the set of JAs with highest reward are included, and in the next, JAs were selected purely for diversity by the following method. We repeatedly cycled through all possible JAs in lexicographic order, and included a JA in the set if the distance between it and every JA already in the set was not less than a specified distance; in this case 2. The average reward and the diversity (expressed as the minimum distance between any pair of JAs in the set) for the JA sets chosen using each of the three metrics over all 20 runs is shown in the upper half of Figure 2(a). While the sets of 1-optima come close to the reward level of the sets chosen purely according to reward, they are clearly more diverse (T-tests for this claim showed a significance within .0001%). If a minimum distance of 2 is required in order to guarantee diversity, then using reward alone as a metric is insufficient; in fact the JA sets generated using that metric had an average minimum distance of 1.21, compared with 2.25 for 1-optimal JA sets (which guarantee a minimum distance of $k+1 = 2$). The 1-optimal JA set also provides significantly higher average reward than the sets chosen to maintain a given minimum distance, which had an average reward of 0.037 (T-test significance within .0001%.). Similar results with equal significance were observed for the 10-agent graph in Figure 2(b) and the nine-agent graph in Figure 2(c). Note also that this experiment used $k = 1$, the lowest possible k. Increasing k would, by definition, increase the diversity of the k-optimal JA set as well as the neighborhood size for which each JA is optimal.

(a)	avg. reward	avg. min. distance
1-optima	.850	2.25
reward only	.950	1.21
dist. of 2	.037	2.00

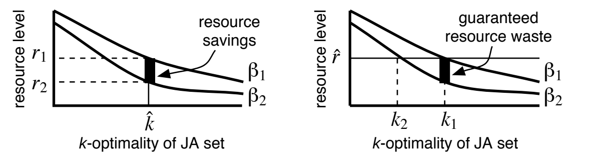
(b)	avg. reward	avg. min. distance
1-optima	.809	2.39
reward only	.930	1.00
dist. of 2	-.101	2.00

(c)	avg. reward	avg. min. distance
1-optima	.832	2.63
reward only	.911	1.21
dist. of 2	.033	2.00



(a) (b) (c)

Figure 2: 1-optima vs. JA sets chosen using other metrics



(a) (b)

Figure 3: Hypothetical example illustrating the advantages of tighter bounds

In addition to categorizing local optima in a DCOP, k-optimality provides a natural classification for DCOP algorithms. Many known algorithms are guaranteed to converge to k-optima for some $k > 0$, including DBA [20], DSA [5], and coordinate ascent [17] for $k = 1$, MGM-2 and SCA-2 [9] for $k = 2$, and Adopt [12], OptAPO [10] and DPOP [13] for $k = I$. For $k < I$, random restarts of these "k-optimal algorithms" can be used to find sets of k-optimal JAs.

3. UPPER BOUNDS ON *k*-OPTIMA

Upper bounds on the number of possible k-optimal JAs, $|A_q(I,k)|$ are useful for two reasons: they yield resource savings in domains where a particular level of k-optimality is desired, and help determine the appropriate level of k-optimality to prevent guaranteed waste of resources (fuel, time, etc.) in settings with fixed resources.

First, a particular level of k-optimality may be desired for a JA set: a high k will include JAs that are more diverse, and optimal within a larger radius, but high-k algorithms have significantly higher coordination/communication overheads [12, 10, 9]; hence lower k is preferable under time pressure. Lower k may also be preferable if an agent team or a human user wants a more detailed set of JAs, for example, more joint patrols, more rescue options, etc. For a given level of k-optimality, bounds indicate the maximum resource requirement for any k-optimal JA set. Thus, tighter bounds provide savings by allowing fewer resources to be allocated *a priori* while ensuring enough will be available for all k-optimal JAs, regardless of the rewards and costs on the constraints. Figure 3 is a hypothetical example, with k on the x-axis and the number of resources to be allocated on the y-axis. β_1 and β_2 are two different upper bounds on the number of k-optimal JAs that can exist for a given DCOP. Part (a) shows how the tighter bound β_2 indicates that a resource level of r_2 is sufficient for all $\hat{k}$-optimal JAs, if each JA consumes one resource, yielding a savings of $r_1 - r_2$ over using β_1.

Second, if resource availability is fixed, tighter bounds help us choose an appropriate level of k-optimality. If k is too low, we may exhaust our resources on bad JAs (similar JAs with poor relative quality). In contrast, fewer k-optimal JAs can exist as k increases, and so if k is too high, available resources that could be spent on additional JAs are guaranteed to go unused. Tighter bounds provide a more accurate measure of this kind of guaranteed waste and thus, allow a more appropriate k to be chosen. In Figure 3(b), under fixed resource level $\hat{r}$, the looser bound β_1 hides the resources guaranteed to go unused when k_1 is used. This waste is revealed by β_2, with the thick line indicating the resources that, if allocated, will never be used, as there cannot exist enough k-optima to use them all; instead, we now see that using k_2 will reduce this guaranteed waste.

To find the first upper bounds on the number of k-optima for a given DCOP graph, we discovered a correspondence to coding theory [8]. In error-correcting codes, a set of codewords must be chosen from the space of all possible words, where each word is a string of characters from an alphabet. All codewords are sufficiently different from one another so that transmission errors will not cause one to be confused for another. Finding the maximum possible number of k-optima can be mapped to finding the maximum number of codewords in a space of q^I words where the minimum distance between any two codewords is $d = k + 1$. We can map JAs (complete DCOP assignments) to words and k-optima to codewords as follows: A JA a taken by I agents each with a domain of cardinality q is analogous to a word of length I from an alphabet of cardinality q. The distance $d(a, \tilde{a})$ can then be interpreted as a Hamming distance between two words. Then, if a is k-optimal, and $d(a, \tilde{a}) \leq k$, then $\tilde{a}$ cannot also be k-optimal by definition. Thus, any two k-optima must be separated by distance $\geq k + 1$.

Three well-known bounds [8] on codewords are Hamming: $\beta_H = q^I / \left(\sum_{j=0}^{\lfloor k/2 \rfloor} \binom{I}{j} (q-1)^j\right)$, Singleton: $\beta_S = q^{I-k}$, and Plotkin: $\beta_P = \left\lfloor \frac{k+1}{k+1-(1-q^{-1})I} \right\rfloor$, which is only valid when $(1 - 1/q)n < k + 1$. Note that for the special case of $q = 2$, it is possible to use the relation $\beta_H(I, k, q) = \beta_H(I - 1, k - 1, q)$[8] to obtain a tighter bound for odd k using the Hamming bound. Now, to find a reward-independent bound on the number of 1-optima for three agents with $q = 2$, (e.g., the system in Example 1), we obtain $\min\{\beta_H, \beta_S, \beta_P\} = \beta_H = 4$, without knowing R_{12} and R_{23} explicitly.

Unfortunately, problems of even d (odd k), are not of interest for error-correcting codes, and β_H, the Hamming bound, is very loose or useless for DCOP when $q > 2$, e.g., for 1-optima (solutions reached by DSA) the bound is equal to the number of possible assignments in this case. Hence, for DCOP, we pursue an improved bound for $q > 2$ and odd k. β_H is derived by using a sphere-packing argument stating that the total number of words q^I must be greater than the number of codewords $A_q(I, k)$ multiplied by the size of a sphere of radius $\lfloor k/2 \rfloor$ centered around each codeword. A sphere $S_A(a^*, r)$ with center a^* and radius r is the set of JAs $\tilde{a}$ such that $d(a^*, \tilde{a}) \leq r$, and represents words that cannot be codewords (except for its center). It can be shown that $S_A(a^*, \lfloor k/2 \rfloor)$ contains exactly $\sum_{j=0}^{\lfloor k/2 \rfloor} \binom{I}{j} (q-1)^j$ words. If k is even, the tightest packing occurs with spheres of radius $k/2$ and each word can be uniquely assigned to the sphere of its closest codeword. If k is odd, it is possible for a word to be equidistant from two codewords and it is unclear how to assign it to a sphere. The Hamming bound addresses this issue by using the bound for $k-1$ when k is odd, which leads to smaller spheres and a bound larger than necessary. This ignores the contribution of a word that lies on the "boundary" between several spheres. These boundary assignments can be appropriately partitioned to achieve a tighter bound on the number of k-optima for odd k, called the *Modified Hamming bound*.

Proposition 1. *For odd k, $A_q(I, k) \leq \min\{A_1, A_2\}$ where*

$$A_1 = \frac{q^I - \binom{I}{(k+1)/2}(q-1)^{(k+1)/2}}{\sum_{j=0}^{\lfloor k/2 \rfloor} \binom{I}{j}(q-1)^j}$$

$$A_2 = \frac{q^I}{\sum_{j=0}^{\lfloor k/2 \rfloor} \binom{I}{j}(q-1)^j + \binom{I}{(k+1)/2}(q-1)^{(k+1)/2}(I^{-1})}$$

Proof. Any word that has Hamming distance $\lfloor k/2 \rfloor$ or less from a codeword belongs in that codeword's sphere, because belonging to more than one sphere would violate the code's distance requirement. Given an odd value of k, each codeword will have $\binom{I}{(k+1)/2}(q-1)^{(k+1)/2}$ words that are a distance of $(k + 1)/2$ away from it. It cannot claim all these words for its sphere exclusively, as they may be equidistant from other codewords. We do know however that each of these words can be on the boundary of at most I spheres (i.e. can be equidistant from at most I codewords) because they are of length I. Furthermore, each of these words can be equidistant from at most $A_q(I, k)$ codewords, i.e. the total number of codewords in the space. Thus, each codeword can safely incorporate $1/\min\{I, A_q(I, k)\}$ of each of these boundary words into its sphere without any portion being claimed by more than one sphere. Aggregating over all the words on the boundary, we can increase the volume of the sphere by $\binom{I}{(k+1)/2}(q-1)^{(k+1)/2}/\min\{I, A_q(I, k)\}$. Using the sphere-packing argument with the portions of the boundary words added to each sphere, if $A_q(I, k) \leq I$, we have

$$q^I \geq A_q(I,k)\left[\sum_{j=0}^{\lfloor k/2 \rfloor} \binom{I}{j}(q-1)^j + \frac{\binom{I}{(k+1)/2}(q-1)^{(k+1)/2}}{A_q(I,k)}\right]$$

$$\Rightarrow A_q(I,k) \leq \frac{q^I - \binom{I}{(k+1)/2}(q-1)^{(k+1)/2}}{\sum_{j=0}^{\lfloor k/2 \rfloor} \binom{I}{j}(q-1)^j} \equiv A_1,$$

and if $A_q(I, k) \geq I$, we have

$$q^I \geq A_q(I,k)\left[\sum_{j=0}^{\lfloor k/2 \rfloor} \binom{I}{j}(q-1)^j + \frac{\binom{I}{(k+1)/2}(q-1)^{(k+1)/2}}{I}\right]$$

$$\Rightarrow A_q(I,k) \leq \frac{q^I}{\sum_{j=0}^{\lfloor k/2 \rfloor} \binom{I}{j}(q-1)^j + \binom{I}{(k+1)/2}(q-1)^{(k+1)/2}(I^{-1})} \equiv A_2.$$

We have $A_q(I, k) \leq I \Rightarrow A_q(I, k) \leq A_1$ and $A_q(I, k) \geq I \Rightarrow A_q(I, k) \leq A_2$. We can show that $A_1 \odot I \Leftrightarrow A_2 \odot I$, $\forall \odot \in \{<, >, =\}$. Furthermore, $A_1 \odot I, A_2 \odot I \Leftrightarrow A_1 \odot A_2$. Thus, when $A_1 \leq I$, then $A_2 \leq I$ and $A_1 \leq A_2$. So, $A_q(I, k) \leq A_1 = \min\{A_1, A_2\}$ when $A_1 \leq I$. And, when $A_1 > I$, then $A_2 > I$ and $A_1 > A_2$. So, $A_q(I, k) \leq A_2 = \min\{A_1, A_2\}$ when $A_1 > I$. Therefore, $A_q(I, k) \leq \min\{A_1, A_2\}$.■

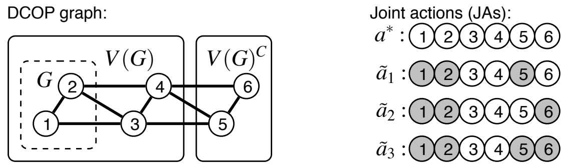
We call the Modified Hamming bound β_{MH} and define $\beta_{HSP} = \min\{\beta_H, \beta_S, \beta_P, \beta_{MH}\}$, including the relation for β_H for $q = 2$; i.e. β_{HSP} gives the best of all the (graph-independent) bounds so far.

4. GRAPH-BASED ANALYSIS: *k*-OPTIMA

The β_{HSP} bound and its components depend only on I, k and q, regardless of how the team reward is decomposed onto constraints; i.e., the bounds are the same for all θ. For instance, the bound on 1-optima for Example 1 (found to be 4 in the previous section) ignored the fact that agents 1 and 3 do not share a constraint, and yields the same result independent of the DCOP graph structure. However, exploiting this structure (as captured by θ) can significantly tighten the bounds on $\{|A_q(I, k)|\}_{k=1}^{I}$. In particular, in obtaining the bounds in Section 3, pairs of JAs were mutually exclusive as k-optima (only one of the two could be k-optimal) if they were separated by a distance $\leq k$. We now show how some JAs separated by a distance $\geq k + 1$ must also be mutually exclusive as k-optima.

We define $D_G(a, \tilde{a}) := \{i \in G : a_i \neq \tilde{a}_i\}$ and $V(G) := \cup_{S \in \theta : G \cap S \neq \emptyset} S$. Intuitively, $D_G(a, \tilde{a})$ is the set of agents within the subgroup G who have chosen different actions between a and $\tilde{a}$, and $V(G)$ is the set of agents (including those in G) who are a member of some constraint $S \in \theta$ incident on a member of G (e.g., G and the agents who share a constraint with some member of G). Then, $V(G)^C$ is the set of all agents whose contribution to the team reward is independent of the values taken by G.

Proposition 2. *Let there be a JA $a^* \in A_q(I, k)$ and let $\tilde{a} \in A$ be another JA for which $d(a^*, \tilde{a}) > k$. If $\exists\, G \subset \mathcal{I}$, $G \neq \emptyset$ for which $|G| \leq k$ and $D_{V(G)}(a^*, \tilde{a}) = G$, then $\tilde{a} \notin A_q(I, k)$.*



(a) (b)

Figure 4: A visual representation of the effect of Proposition 2.

Proof. Given a^*, $\tilde{a}$, and G with the properties stated above, we have that $\forall a : d(a^*, a) \le k$, $\Delta(a^*, a) > 0$. If a is defined such that $a_i = \tilde{a}_i$ for $i \in V(G)$ and $a_i = a_i^*$ for $i \notin V(G)$, then $D(a^*, a) = G$ and $d(a^*, a) \le k$ which implies

$$\Delta(a^*, a) = \sum_{S \in \theta: S \cap D(a^*, a) \neq \emptyset} R_S(a_S^*) - R_S(a_S) = \sum_{S \in \theta: S \cap G \neq \emptyset} R_S(a_S^*) - R_S(a_S)$$
$$= \sum_{S \in \theta: S \cap G \neq \emptyset} R_S(a_S^*) - R_S(\tilde{a}_S) > 0.$$

If $\hat{a}$ is defined such that $\hat{a}_i = a_i^*$ for $i \in V(G)$ and $\hat{a}_i = \tilde{a}_i$ for $i \notin V(G)$, then $D(\tilde{a}, \hat{a}) = G$ and $d(\tilde{a}, \hat{a}) \le k$, and

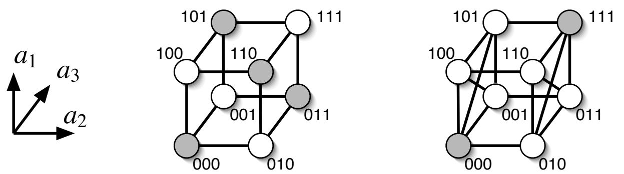
$$\Delta(\tilde{a}, \hat{a}) = \sum_{S \in \theta: S \cap D(\tilde{a}, \hat{a}) \neq \emptyset} R_S(\tilde{a}_S) - R_S(\hat{a}_S) = \sum_{S \in \theta: S \cap G \neq \emptyset} R_S(\tilde{a}_S) - R_S(\hat{a}_S)$$
$$= \sum_{S \in \theta: S \cap G \neq \emptyset} R_S(\tilde{a}_S) - R_S(a_S^*) < 0,$$

thus, $\tilde{a} \notin A_q(I, k)$ because $\Delta(\tilde{a}, \hat{a}) < 0$ and $d(\tilde{a}, \hat{a}) \le k$.■

Proposition 2 provides conditions where if a^* is k-optimal then $\tilde{a}$, which may be separated from a^* by a distance greater than $k+1$ may not be k-optimal, thus tightening bounds on k-optimal JA sets. With Proposition 2, since agents are typically not fully connected to all other agents, the *relevant context* a subgroup faces is not the entire set of other agents. Thus, the subgroup and its relevant context form a view (captured by $V(G)$) that is not the entire team. We consider the case where a JA $\tilde{a}$ has $d(a^*, \tilde{a}) > k$. We also have group G of size k within whose view $V(G)$, G are the only deviators between a^* and $\tilde{a}$ (although agents outside the view must also have deviated, because $d(a^*, \tilde{a}) > k$). We then know that $\tilde{a}$ contains a group G of size k or less that has taken a suboptimal subgroup joint action with respect to its relevant context and thus $\tilde{a}$ cannot be k-optimal, i.e. if the group G chose a_G^* instead of $\tilde{a}_G$ under its relevant context $V(G) \setminus G$ for $\tilde{a}$, then team reward would increase.

Figure 4(a) shows G, $V(G)$, and $V(G)^C$ for a sample DCOP of six agents with a domain of two actions, white and gray. Without Proposition 2, $\tilde{a}_1$, $\tilde{a}_2$, and $\tilde{a}_3$ could all potentially be 2-optimal. However, Proposition 2 guarantees that they are not, leading to a tighter bound on the number of 2-optima that could exist. To see the effect, note that if a^* is 2-optimal, then $G = \{1, 2\}$, a subgroup of size 2, must have taken an optimal subgroup joint action (all white) given its relevant context (all white). Even though $\tilde{a}_1$, $\tilde{a}_2$, and $\tilde{a}_3$ are a distance greater than 2 from a^*, they cannot be 2-optimal, since in each of them, G faces the same relevant context (all white) but is now taking a suboptimal subgroup joint action (all gray).

To explain the significance of Proposition 2 to bounds, we introduce the notion of an *exclusivity relation* $E \subset \mathcal{I}$ which captures the restriction that if deviating group $D(a, \tilde{a}) = E$, then at most one of a and $\tilde{a}$ can be k-optimal. An *exclusivity relation set* for k-optimality, $\mathcal{E}_k \subset \mathcal{P}(\mathcal{I})$, is a collection of such relations that limits $|A_q(I, k)|$, the number of JAs that can be k-optimal in a reward-independent setting (otherwise every JA could potentially be k-optimal). In particular, the set $\mathcal{E}_k$ defines an *exclusivity graph* H_k where each node corresponds uniquely to one of all q^I possible JAs. Edges are defined between pairs of JAs, a and $\tilde{a}$, if $D(a, \tilde{a}) \in \mathcal{E}_k$. The size of the maximum independent set (MIS) of H_k, the largest subset of nodes such that no pair defines an edge, gives an upper bound on $|A_q(I, k)|$. Naturally, an expanded $\mathcal{E}_k$ implies a more connected exclusivity graph and thus a tighter bound on $|A_q(I, k)|$.



(a) (b)

Figure 5: Exclusivity graphs for 1-optima for Example 1 with MIS shown in gray, (a) not using Proposition 2 and (b) using it.

Without introducing graph-based analysis, β_{HSP} for each k provides a bound on the MIS of H_k when $\mathcal{E}_k = \bigcup_{E \subset \mathcal{I}: 1 \le |E| \le k} E$. This set $\mathcal{E}_k$ captures only the restriction that no two JAs within a distance of k can both be k-optimal. Consider Example 1, but with unknown rewards on the links. Here, the exclusivity relation set for 1-optima without considering the DCOP graph is $\mathcal{E}_1 = \{\{1\}, \{2\}, \{3\}\}$, meaning that no two JAs differing only by the action taken by either agent 1, 2, or 3, can both be 1-optimal. This leads to the exclusivity graph in Figure 5(a) whose MIS implies a bound of 4.

The significance of Proposition 2 is that it provides additional exclusivity relations for solutions separated by distance $\ge k+1$, which arise only because we considered the structure of the DCOP graph, which will allow a tighter bound to be computed. This graph-based exclusivity relation set is $\widetilde{\mathcal{E}}_k = \bigcup_{E \subset \mathcal{I}: 1 \le |E| \le k} \bigcup_{F \in \mathcal{P}(V(E)^C)} [E \cup F]$ which is a superset of $\mathcal{E}_k$. Additional relations exist because multiple exclusivity relations ($\bigcup_{F \in \mathcal{P}(V(E)^C)} [E \cup F]$) appear the same to the subgroup E because of its reduced view $V(E)$. Now, for Example 1, the exclusivity relation set for 1-optima when considering the DCOP graph is $\widetilde{\mathcal{E}}_1 = \{\{1\}\{2\}, \{3\}, \{1, 3\}\}$, which now has the additional relation {1,3}. This relation, included because of the realization that agents 1 and 3 are not connected, says that no two JAs can both be 1-optimal if they differ only in the actions of both agent 1 and agent 3. This leads to the exclusivity graph in Figure 5(b) whose MIS implies a bound of 2. Algorithms for obtaining bounds using $\widetilde{\mathcal{E}}_k$ will be discussed in Section 6.

5. APPLICATION TO NASH EQUILIBRIA

Our graph-based bounds can be extended beyond agent teams to noncooperative settings. It is possible to employ the same exclusivity relations for 1-optimal DCOP assignments to bound the number of pure-strategy Nash equilibria in a graphical game (of the same graph structure) using any of our bounds for $|A_q(I, 1)|$. Bounds on Nash equilibria [11] are useful for design and analysis of mechanisms as they predict the maximum number of outcomes of a game.

We begin with a set of noncooperative agents $\mathcal{I} = \{1, \ldots I\}$, where the i^{th} agent's utility is $U^i(a_i; a_{\{\mathcal{I} \setminus i\}}) = \sum_{S_i \in \theta_i} U^i_{S_i}(a_i; a_{\{S_i \setminus i\}})$ which is a decomposition into an aggregation of component utilities generated from minimal subgroups. Note that the combination of actions taken by any subgroup of agents may generate utility for any agent i, therefore the subgroups are denoted as S_i rather than S, as in the cooperative case, where the utility went to the entire team. The notation a_i and $a_{\{G \setminus i\}}$ refers to the i^{th} agent's action and the actions of the group G with i removed, respectively. We refer to a as a joint action (JA), with the understanding that it is composed of actions motivated by individual utilities. Let the *view* of the i^{th} agent in a noncooperative setting to be $V(i) = \cup_{S_i \in \theta_i} S_i$. The deviating group with respect to G is: $D_G(a, \tilde{a}) := \{i \in G : a_i \neq \tilde{a}_i\}$. Assuming every player has a unique optimal response to its context, then if a^* is a pure-strategy Nash equilibrium, and $d(a^*, a) = 1$,

$i = D(a^*, a)$, we know that $U^i(a_i^*; a^*_{\{\mathcal{I}\setminus i\}}) > U^i(a_i; a^*_{\{\mathcal{I}\setminus i\}})$ and a is not a pure-strategy Nash equilibrium. However, applying the graph (or hypergraph) structure of the game, captured by the sets $\{\theta_i\}$, we get exclusivity relations between JAs with distance > 1 as follows.

Proposition 3. *If a^* is a pure-strategy Nash equilibrium, $\tilde{a} \in A$ such that $d(a^*, \tilde{a}) > 1$, and $\exists i \in \mathcal{I}$ such that $D_{V(i)}(a^*, \tilde{a}) = i$, then $\tilde{a}$ is not a pure-strategy Nash equilibrium.*

Proof. We have $U^i(\tilde{a}_i; \tilde{a}_{\{\mathcal{I}\setminus i\}})$

$$= \sum_{S_i \in \theta_i} U^i_{S_i}(\tilde{a}_i; \tilde{a}_{\{S_i\setminus i\}}) = \sum_{S_i \in \theta_i} U^i_{S_i}(\tilde{a}_i; a^*_{\{S_i\setminus i\}})$$
$$< \sum_{S_i \in \theta_i} U^i_{S_i}(a^*_i; a^*_{\{S_i\setminus i\}}) = \sum_{S_i \in \theta_i} U^i_{S_i}(a^*_i; \tilde{a}_{\{S_i\setminus i\}}) = U^i(a^*_i; \tilde{a}_{\{\mathcal{I}\setminus i\}}).$$

The first and last equalities are by definition. The second and third equalities are because $D_{V(i)}(a^*, \tilde{a}) = i$. The inequality is because a^* is a pure-strategy Nash equilibrium. The result is that $\tilde{a}_i$ is not an optimal response to $\tilde{a}_{\{\mathcal{I}\setminus i\}}$ and thus cannot be a Nash equilibrium. ■

Proposition 3 states that a^* and $\tilde{a}$ cannot both be Nash equilibria if $\exists i,\ D_{V(i)}(a^*, \tilde{a}) = i$, which is identical to the condition that prevents two JAs (in a team setting) from being 1-optimal. The commonality is that in both the cooperative and noncooperative settings, agents have optimal actions for any given context, and in both settings there is a notion of relevant context, $V(i) \setminus i$, which can be a subset of other agents $\{\mathcal{I} \setminus i\}$. The difference is that the views are generated in different manners: $V(i) = \cup_{S\in\theta: i\cap S\neq\emptyset} S$ in a cooperative setting, while $V(i) = \cup_{S_i\in\theta_i} S_i$ in a noncooperative setting. Given the views, we can generate the exclusivity relation set in the same manner, $\mathcal{E}_1 = \bigcup_{i\in\mathcal{I}} \bigcup_{F\in\mathcal{P}(V(i)^C)} [i \cup F]$. Given the exclusivity relation set, we can create an exclusivity graph for a noncooperative setting in a fashion similar to the one in Section 4. Thus, the bound on the number of Nash equilibria for a noncooperative graphical game is identical to the bound on 1-optimal JAs for a cooperative DCOP, if both share the same exclusivity relation set $\mathcal{E}_1$.

6. GRAPH-BASED BOUNDS

As seen earlier, the graph structure expands the exclusivity relation set for k-optimality in cooperative (DCOP) settings and Nash equilibria in noncooperative (graphical-game) settings. This set defines exclusivity graph H_k whose maximum independent set (MIS) provides a bound for the number of k-optimal JAs (or alternatively, for the number of Nash equilibria). Finding the size of the MIS is NP-complete in the general case [1], so we investigated heuristic techniques to obtain an upper bound on $|A_q(I, k)|$. We observe that any fully-connected subset (clique) of H_k can contain at most one k-optimum. Thus, the number of cliques in any clique partitioning of H_k also provides an upper bound on $|A_q(I, k)|$, where a partitioning yielding fewer cliques will provide a tighter bound. Hence, our first approach is the polynomial-time F_{CLIQUE} clique-partitioning algorithm, shown in [7] to outperform several competitors.

Our second heuristic technique to find a graph-based bound is Algorithm 1, the *Symmetric Region Packing bound*, β_{SRP}, which uses a packing method analogous to Proposition 1, where each k-optimum claims a region of the space of all possible JAs (the nodes of H_k). Because these regions are constructed to be disjoint and have identical volumes, dividing the space of all JAs by this volume yields a bound. Figure 6 shows β_{SRP} computed for 1-optima for Example 1. We choose an arbitrary JA $a \in A$ which we assume to be k-optimal (a = [0 0 0] in Figure 6), around which we will construct a region claimed by a.

Applying the exclusivity relations from $\widetilde{\mathcal{E}}_k$, we generate a set $B(a) = \cup_{E\in\widetilde{\mathcal{E}}_k} f(a, E)$ where $f(a, E)$ yields the JA that is excluded from being k-optimal by a and E. The first two rows of Figure 6 show $\widetilde{\mathcal{E}}_1$ and the set B([0 0 0]). Applying the exclusivity relations again for each $b \in B(a)$, and discarding JAs already included in a or $B(a)$, we generate a set $\overline{B}(b) = \cup_{E\in\widetilde{\mathcal{E}}_k} f(b, E)$ which contains all JAs that potentially exclude b from being k-optimal. In Figure 6, we apply $\widetilde{\mathcal{E}}_1$ to find $\overline{B}(b)$ for all $b \in B(a)$ = {[1 0 0], [0 1 0], [0 0 1], [1 0 1]} where the grayed out JAs are those discarded for being in $\{a\} \cup B(a)$. To ensure that the region that a claims is disjoint from the regions claimed by other k-optima, a should only claim a fraction of each $b \in B(a)$. This can be achieved if a shares each b equally with all other k-optima that might exclude b. These additional k-optima are contained within $\overline{B}(b)$. However, not all $\overline{b} \in \overline{B}(b)$ can actually be k-optimal as they might exclude each other. If we construct a graph $\overline{H}_k(b)$ with nodes for all $\overline{b} \in \overline{B}(b)$ and edges formed using $\widetilde{\mathcal{E}}_k$, and we find M_b, the size of the MIS, then a can safely claim $1/(1 + M_b)$ of b. We again use clique partitioning to safely estimate M_b. In Figure 6, for b = [0 1 0], $\overline{B}$([0 1 0]) leads to a three-node, three-edge exclusivity graph $\overline{H}_k$([0 1 0]). By adding the values of $1/(1 + M_b)$ for all $b \in B(a)$ (plus one for itself), we obtain that a can safely claim a region of size 3, which implies $\beta_{SRP} = \lfloor 2^3/3 \rfloor = 2$. Algorithm 1's runtime is polynomial in the number of possible JAs, which is a comparatively small cost for a bound that applies to every possible instantiation of rewards to actions. An exhaustive search for the MIS of H_k would be exponential in this number (doubly exponential in the number of agents).

Algorithm 1 for Symmetric Region Packing (SRP) bound

$\widetilde{\mathcal{E}}_k = \bigcup_{E\subset\mathcal{I}:1\le|E|\le k} \bigcup_{F\in\mathcal{P}(V(E)^C)} [E \cup F]$
a = [0 0 0]
$|A_k| = 1$
$B(a) = \cup_{E\in\widetilde{\mathcal{E}}_k} f(a, E)$
for all b $\in B(a)$ **do**
  $\overline{B}(b) = (\cup_{E\in\widetilde{\mathcal{E}}_k} f(b, E)) \setminus (a \cup B(a))$
  $\overline{H}_k(b)$.addNodes($\overline{B}(b)$)
  for all $\overline{b_1}, \overline{b_2} \in \overline{B}(b)$ **do**
    if $D(\overline{b_1}, \overline{b_2}) \in \widetilde{\mathcal{E}}_k$ **then**
      $\overline{H}_k(b)$.addEdge($\overline{b_1}, \overline{b_2}$)
  M_b = |cliquePartition($\overline{H}_k(b)$)|
  $|A_k| = |A_k| + 1/(1 + M_b)$
$\beta_{SRP} = (q^I)/|A_k|$

$\widetilde{\mathcal{E}}_1$ = {	{1},	{2},	{3},	{1,3} }
B([0 0 0]) = {	[1 0 0],	[0 1 0],	[0 0 1],	[1 0 1] }
$\widetilde{\mathcal{E}}_1$ = {	$\overline{B}$([1 0 0])	$\overline{B}$([0 1 0])	$\overline{B}$([0 0 1])	$\overline{B}$([1 0 1])
{1},	[0 0 0]	[1 1 0]	[1 0 1]	[0 0 1]
{2},	[1 1 0]	[0 0 0]	[0 1 1]	[1 1 1]
{3},	[1 0 1]	[0 1 1]	[0 0 0]	[1 0 0]
{1,3}}	[0 0 1]	[1 1 1]	[1 0 0]	[1 1 1]
$\overline{H}_1(b)$ (exclusivity subgraph)	110	110 {1,3} {3} 011 111 {1}	011	111
M_b =	1	1	1	1
$1/(1+M_b)$ =	1/2	1/2	1/2	1/2

Figure 6: Computation of β_{SRP} for Example 1

7. EXPERIMENTAL RESULTS

We performed five evaluations in addition to the experiment described in Section 2. The first evaluates the impact of k-optimality for higher values of k. For each of the three DCOP graphs from Figure 2(a-c), Figure 7(a-c) shows key properties for 1-, 2- and 3-optima. The first column of each table shows $|\tilde{A}|$, the size of the

(a)

	$\lvert\tilde{A}\rvert$	avg. reward
1-opt.	10	.850
2-opt.	55	.964
3-opt.	175	.993

(b)

	$\lvert\tilde{A}\rvert$	avg. reward
1-opt.	10	.809
2-opt.	55	.961
3-opt.	175	.986

(c)

	$\lvert\tilde{A}\rvert$	avg. reward
1-opt.	9	.832
2-opt.	45	.977
3-opt.	129	.982

Figure 7: 1-optima vs. JA sets chosen using other metrics

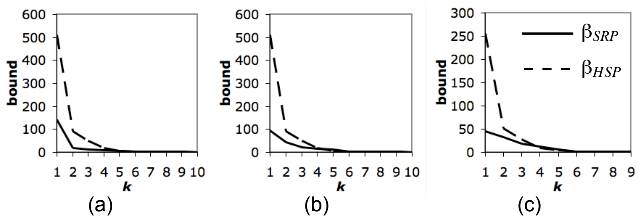


Figure 8: β_{SRP} vs. β_{HSP} for DCOP graphs from Figure 2

neighborhood containing all JAs within a distance of k from a k-optimal JA a, and hence of lower reward than a. For example, in the joint patrol domain described in Section 2, Figure 7(a) shows that, if agents are arranged as in the DCOP graph from Figure 2 (a), any 1-optimal joint patrol must have a higher reward than at least 10 other joint patrols. We see that as k increases, the k-optimal set contains JAs that each individually dominate a larger and larger neighborhood. The second column shows, for each of the three graphs, the average reward of each k-optimal JA set found over 20 problem instances, generated by assigning rewards to the links from a uniform random distribution. We define the reward of a k-optimal JA set as the mean reward of all k-optimal JAs that exist for a particular problem instance; each figure in the second column is therefore a mean of means. As k was increased, leading to a larger neighborhood of dominated JAs, the average reward of the k-optimal JA sets show a significant increase (T-tests showed the increase in average reward as k increased was significant within 5%.)

However as k increases, the number of possible k-optimal JAs decreases, and hence the next four evaluations explore the effectiveness of the different bounds on the number of k-optima. For the three DCOP graphs shown in Figure 2, Figure 8 provides a concrete demonstration of the gains in resource allocation due to the tighter bounds made possible with graph-based analysis. The x axis in Figure 8 shows k, and the y axis shows the β_{HSP} and β_{SRP} bounds on the number of k-optima that can exist. To understand the implications of these results on resource allocation, consider a patrolling problem where the constraints between agents are shown in the 10-agent DCOP graph from Figure 2(a), and all agents consume one unit of fuel for each JA taken. Suppose that $k = 2$ has been chosen, and so at runtime, the agents will use MGM-2 [9], repeatedly, to find and execute a set of 2-optimal JAs. We must allocate enough fuel to the agents *a priori* so they can execute up to all possible 2-optimal JAs. Figure 8(a) shows that if β_{HSP} is used, the agents would be loaded with 93 units of fuel to ensure enough for all 2-optimal JAs. However, β_{SRP} reveals that only 18 units of fuel are sufficient, a five-fold savings. (For clarity we note that on all three graphs, both bounds are 1 when $k = I$ and 2 when $I - 3 \leq k < I$.)

To systematically investigate the impact of graph structure on bounds, we generated a large number of DCOP graphs of varying size and density. We started with complete binary graphs (all pairs of agents are connected) where each node (agent) had a unique ID. To gradually make each graph sparser, edges were repeatedly removed according to the following two-step process: (1) Find the lowest-ID node that has more than one incident edge. (2) If such a node exists, find the lowest-ID node that shares an edge with it, and remove this edge. Figure 9 shows the β_{HSP} and β_{SRP} bounds

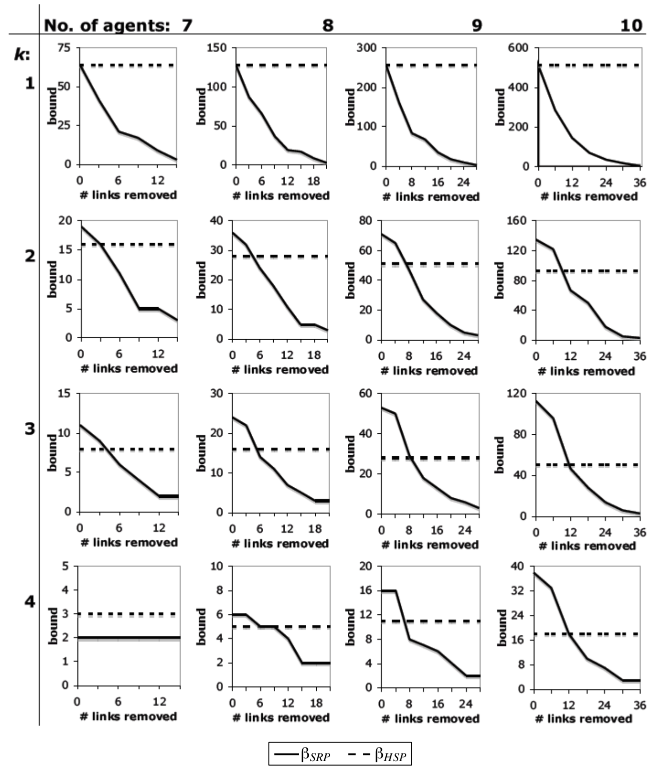


Figure 9: Comparisons of β_{SRP} vs. β_{HSP}

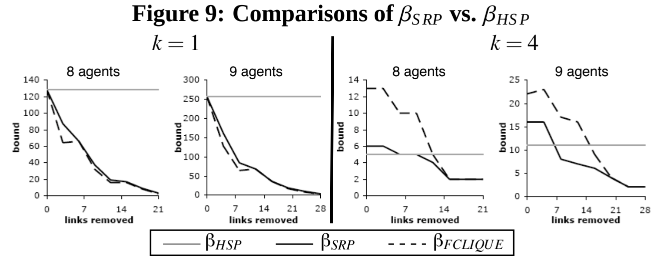


Figure 10: Comparisons of $\beta_{SRP}, \beta_{HSP}, \beta_{FCLIQUE}$

for k-optima for $k \in \{1, 2, 3, 4\}$ and $I \in \{7, 8, 9, 10\}$. For each of the 16 plots shown, the y axis shows the bounds and the x-axis shows the number of links removed from the graph according to the above method. While $\beta_{HSP} < \beta_{SRP}$ for very dense graphs, β_{SRP} provides significant gains for the vast majority of cases. For example, for the graph with 10 agents, and 24 links removed, and a fixed $k = 1$, β_{HSP} implies that we must equip the agents with 512 resources to ensure that all resources are not exhausted before all 1-optimal actions are executed. However, β_{SRP} indicates that a 15-fold reduction to 34 resources will suffice, yielding a savings of 478 due to the use of graph structure when computing bounds.

A fourth experiment compared β_{HSP} and β_{SRP} to the bound obtained by applying F_{CLIQUE}, $\beta_{FCLIQUE}$ to DCOP graphs from the previous experiment. Selected results are shown in Figure 10 for graphs of 8 and 9 agents. While $\beta_{FCLIQUE}$ is marginally better for $k = 1$, β_{SRP} has clear gains for $k = 4$. Identifying the relative effectiveness of various algorithms that exploit our exclusivity relation sets is clearly an area for future work.

Finally, Figure 11 compares the constant-time-computable graph-independent bounds from Section 3, in particular, showing the improvement of β_{MH} over $\min\{\beta_H, \beta_S, \beta_P\}$ for selected odd values of k, given three possible actions for each agent ($q = 3$). The x-axis shows I, the number of agents and the y-axis show s $100 \cdot (\min\{\beta_H, \beta_S, \beta_P\} - \beta_{MH}) / \min\{\beta_H, \beta_S, \beta_P\}$. For odd values of $k > 1$,

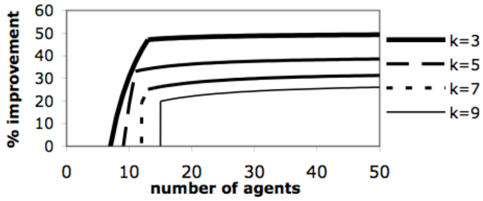


Figure 11: Improvement of β_{MH} on $\min\{\beta_H, \beta_S, \beta_P\}$

as I increased, β_{MH} provided a tighter bound on the number of k-optima. The most improvement was for $k = 3$; as I increased, β_{MH} gave a bound 50% tighter than the others.

8. RELATED WORK AND CONCLUSION

This paper provides a theoretical complement to the experimental analysis of local minima (1-optima) and landscapes in centralized constraint satisfaction problems (CSPs) [19] as well as incomplete DCOP algorithms [20, 9]. In contrast, we provide categorization and theoretical justification for k-optimality, bounds on the number of k-optima (both graph-independent and dependent), algorithms to compute graph-based bounds, and experimental analysis. We note that k-optimality can also apply to centralized constraint reasoning as a measure of the relative quality and diversity of solutions in a set. 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.

Our research on bounding k-optimal solution sets for DCOPs is related to estimating numbers of local optima in centralized local search and evolutionary computing [2, 18]. The key difference is in the exploitation of constraint graph structure, not harnessed in previous work, to bound the number of optima.

Given that counting the number of Nash equilibria in a game with known payoffs is $\#\mathcal{P}$-hard [3], bounds have been investigated for particular types of games [11]. Graph structure is utilized in algorithms to expedite finding Nash equilibria for a given graphical game with known payoffs [6]. However, finding tight bounds on Nash equilibria over all possible games on a given graph (i.e., reward-independent bounds) remained an open problem.

In summary, in this paper, (1) we have introduced k-optimality as a metric that captures diversity and relative quality, properties desirable for evaluating sets of DCOP assignments, where each assignment represents a JA to be considered or executed. Finding bounds on k-optimal JA sets is useful for resource allocation problems associated with executing JA sets in sequence or presenting a JA set as a set of options. (2) We discover a correspondence to coding theory that yields bounds β_H, β_S, and β_P, independent of reward and graph structure and (3) introduce a tighter bound β_{MH} for odd k, all of which are computable in constant time. We introduced (4) a method to exploit DCOP graph structure to obtain tighter reward-independent bounds on the number of k-optima that can exist. (5) We also show that our method extends to noncooperative settings, as our bound for 1-optima in a DCOP can be used as a bound on the number of pure-strategy Nash equilibria in a graphical game of arbitary payoffs. Finally, (6) we develop techniques for computing bounds (β_{SRP}, $\beta_{FCLIQUE}$) using the graph-based exclusivity relation sets and (7) illustrate their utility on a diverse collection of graphs.

9. ACKNOWLEDGMENTS

Thanks to Gal Kaminka, Pragnesh Jay Modi, Praveen Paruchuri, and Paul Scerri for their helpful comments. 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.

10. REFERENCES

[1] N. Alon and N. Kahale. Approximating the independence number via the theta-function. *Mathematical Programming*, 80:253–264, 1998.

[2] R. Caruana and M. Mullin. Estimating the number of local minima in complex search spaces. In *IJCAI Workshop on Optimization*, 1999.

[3] V. Conitzer and T. Sandholm. Complexity results about Nash equilibria. In *IJCAI*, 2003.

[4] J. Cox, E. Durfee, and T. Bartold. A distributed framework for solving the multiagent plan coordination problem. In *AAMAS*, 2005.

[5] 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.

[6] M. Kearns, M. Littman, and S. Singh. Graphical models for game theory. In *Proc. UAI*, 2001.

[7] J. T. Kim and D. R. Shin. New efficient clique partitioning algorithms for register-transfer synthesis of data paths. *Journal of the Korean Phys. Soc.*, 40(4):754–758, 2002.

[8] S. Ling and C. Xing. *Coding Theory: A First Course*. Cambridge U. Press, 2004.

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

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

[11] A. McLennan and I. Park. Generic 4 x 4 two person games have at most 15 Nash equilibria. *Games and Economic Behavior*, 26(1):111–130, 1999.

[12] 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.

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

[14] S. Ruan, C. Meirina, F. Yu, K. R. Pattipati, and R. L. Popp. Patrolling in a stochastic environment. In *10th Intl. Command and Control Research Symp.*, 2005.

[15] N. Schurr, J. Marecki, P. Scerri, J. Lewis, and M. Tambe. The DEFACTO system: Training tool for incident commanders. In *IAAI*, 2005.

[16] A. Tate, J. Dalton, and J. Levine. Generation of multiple qualitatively different plan options. In *Proc. AIPS*, 1998.

[17] 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.

[18] D. Whitley, S. Rana, and R. B. Heckendorn. Representation issues in neighborhood search and evolutionary algorithms. In D. Quagliarella, et al., editor, *Genetic Algs. and Evolution Strategies in Eng. and Comp. Sci.*, pages 39–57. Wiley, 1998.

[19] M. Yokoo. How adding more constraints makes a problem easier for hill-climbing algorithms: Analyzing landscapes of CSPs. In *Int'l Conf. on Constraint Programming*, 1997.

[20] W. Zhang, Z. Xing, G. Wang, and L. Wittenburg. An analysis and application of distributed constraint satisfaction and optimization algorithms in sensor networks. In *AAMAS*, 2003.