# Locally Optimal Algorithms and Solutions for Distributed Constraint Optimization

**Jonathan P. Pearce**
Computer Science Dept., University of Southern California
3737 Watt Way, PHE 210, Los Angeles, CA 90089
+1 213 740 7231
http://www-scf.usc.edu/ jppearce
jppearce@usc.edu
paper type: research overview

**Abstract**

This paper summarizes the author's recent work in distributed constraint optimization (DCOP). New local algorithms, as well as theoretical results about the types of solutions that these algorithms can reach, are discussed.

## Introduction

My research deals with understanding and managing interactions among autonomous agents. The focus is primarily on cooperative agents who act as a team to achieve a goal, although some of the work applies to systems of noncooperative agents as well. In a large class of multi-agent scenarios, a group of agents must perform 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 by graphical models. The Distributed Constraint Optimization Problem (DCOP) (Modi *et al.* 2005) is one such model that has recently emerged as popular, general paradigm for representing many cooperative multi-agent scenarios, including include multi-agent plan coordination, sensor networks, and RoboCup soccer.

Traditionally, researchers have focused on complete algorithms for obtaining a single, globally optimal solution to DCOPs. However, as the scale of these domains become large, current complete algorithms can incur large computation or communication costs. For example, a large-scale network of personal assistant agents might require global optimization over hundreds of agents and thousands of variables, which is currently very expensive. Alternatively, some domains may require that a solution be reached quickly; for example, a team of patrol units may need to quickly decide on a route for a joint patrol to efficiently survey an area before conditions change on the ground (Pearce, Maheswaran, & Tambe 2006). Though heuristics that significantly speed up convergence have been developed (Maheswaran *et al.* 2004), the complexity is still prohibitive in large-scale domains. On the other hand, incomplete algorithms in which individual agents, or small groups of agents, react on the basis of local knowledge of neighbors and constraint utilities to reach a local optimum, lead to a system that scales up easily and is more robust to dynamic environments. Our work includes the development of efficient incomplete algorithms for DCOP, a rich categorization of new solution concepts for agents in a DCOP, and theoretical guarantees on several properties of these solutions. Connections are also made to networks of noncooperative agents; in particular, certain properties of local optima in DCOPs are shown to also apply to Nash equilibria in graphical games.


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.

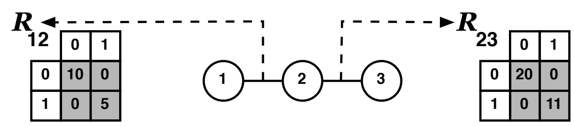

Figure 1: DCOP example

## Current Work

We begin with our model of the multi-agent team problem, 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.

Recognizing the importance of local algorithms for DCOP, researchers initially introduced DBA(Yokoo & Hirayama 1996) and DSA(Fitzpatrick & Meertens 2003) for Distributed CSPs, which were later extended to DCOPs with weighted constraints (Zhang *et al.* 2003). In both algorithms, individual agents react based on their local constraints and the actions chosen by other relevant agents until a local optimum is reached, where no single agent can improve the solution by choosing a different action.

To better understand the space of local optima, we introduced a metric called $k$-optimality. A $k$-optimal DCOP solution is one in which no subset of $k$ or fewer agents can improve the overall solution quality by choosing different actions. Therefore, $k$-optimality quantifies the neighborhood in which a local optimum is optimal. Figure 1 is a DCOP in which agents choose actions from $\{0, 1\}$, with rewards shown for the two constraints $\{1, 2\}$ and $\{2, 3\}$. The assignment [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 $\{2, 3\}$ deviates, making the assignment [1 0 0], team reward increases from 16 to 20. The global optimum, $a^* = [0\ 0\ 0]$ is $k$-optimal for all $k \in \{1, 2, 3\}$.

In a *set* of $k$-optimal solutions, each is guaranteed a particular level of relative quality (best within a certain neighborhood), as well as diversity (every two solutions must be separated by at least $k+1$ individual actions). Due to these properties, finding a single $k$-optimum may be attractive when time or computation is too limited to find the global optimum. Additionally, finding a set of $k$-optimal solutions may be useful for providing options to a supervisor. If we define an algorithm as $k$-optimal if it is guaranteed to reach a $k$-optimum, then algorithms such as DSA are 1-optimal. We developed two new 2-optimal algorithms, MGM-2 and SCA-2 (Maheswaran, Pearce, & Tambe 2004), in which agents repeatedly form pairs and react to the constraints on both agents. An experimental analysis of these algorithms, as well as DSA and DBA, on various graph coloring problems, showed cases where each was the preferred choice.

In an effort to determine which values of $k$ are most appropriate to use when searching for local optima in various situations (including constraints on time and communication), we have proven several interesting properties of $k$-optimal DCOP solutions. These properties are independent of the actual costs and rewards in the DCOP, and are especially useful in domains in which the rewards are unknown or subject to repeated change. The first such results are guarantees on the solution quality of any $k$-optimal DCOP solution. Because of reward-independence, the guarantees are expressed as the proportion of all possible assignments to the DCOP that must be of equal or lesser quality to any $k$-optimal assignment (i.e., a percentile). We provide guarantees that are computable in constant time; however our key contribution is the development of tighter guarantees made possible by considering the graph structure of the DCOP. Such guarantees could be used to help determine an appropriate $k$-optimal algorithm for agents to use, in situations where the cost of coordination between multiple agents must be weighed against the quality of the solution reached. Our analysis showed that some graph types allowed for much stronger guarantees on the solution quality of $k$-optima than other graph types, and experiments with randomly generated DCOPs suggested that the guarantees were provably tight.

A second contribution has been the development of upper bounds on the number of possible $k$-optima that can exist in a DCOP graph (Pearce, Maheswaran, & Tambe 2006). These bounds are necessitated by two key features of typical domains where finding a set of $k$-optima is useful. First, each solution in the set consumes some resources that must be allocated in advance. Such resource consumption arises because: (i) a team actually executes each solution in the set, or (ii) the solution 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 size of the set. Second, while the existence of the constraints between agents is known *a priori*, the actual rewards on the constraints depend on conditions unknown until runtime, and so resources must be allocated before the rewards are known and before the agents generate the $k$-optimal set. Because each solution consumes resources, knowing the maximal number of $k$-optimal solutions that could exist for a given DCOP would allow us to allocate sufficient resources for a given level of $k$. We provide bounds, computable in constant time, that ignore the graph structure, as well as tighter, graph-based bounds with higher computation cost. In addition, our bounds on the number of 1-optima in a DCOP also apply to the number of pure-strategy Nash equilibria in a graphical game of noncooperative agents if the graph structure in both models is the same. Bounds on Nash equilibria are useful for design and analysis of mechanisms as they predict the maximum number of outcomes of a game.

## Future Work

I plan to pursue several ideas. The first is to define a family of efficient $k$-optimal algorithms for $k > 2$. This will provide a greater choice of approaches to take when it is not feasible or desirable to find a globally optimal solution. This choice can be informed by the properties of $k$-optima that have already been proven, which all exist independent of the rewards on the constraints in the DCOP. Developing tighter bounds on potential solution quality and number of $k$-optima in a DCOP may be possible if some information about the rewards is known ahead of time. Examining the effect of this information and designing efficient algorithms to take it into account when computing such guarantees is a second idea I plan to explore. Finally, I would like to explore properties of locally optimal solutions in domains that may not be representable as a standard DCOP, and contain uncertainty, dynamism, and self-interest.

## References


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

Maheswaran, R. T.; Tambe, M.; Bowring, E.; Pearce, J. P.; and Varakantham, P. 2004. Taking DCOP to the real world: efficient complete solutions for distributed multi-event scheduling. In *AAMAS*.

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

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

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

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

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