Artificial Intelligence 212 (2014) 1–26

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Explorative anytime local search for distributed constraint optimization ☆

Roie Zivan *, Steven Okamoto, Hilla Peled

Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel

ARTICLE INFO

Article history:
Received 29 October 2012
Received in revised form 2 March 2014
Accepted 3 March 2014
Available online 17 March 2014

Keywords:
Distributed constraint optimization
Incomplete search
Exploration heuristics

ABSTRACT

Distributed Constraint Optimization Problems (DCOPs) are an elegant model for representing and solving many realistic combinatorial problems that are distributed by nature. DCOPs are NP-hard and therefore many recent studies consider incomplete algorithms for solving them. Distributed local search algorithms, in which agents in the system hold value assignments to their variables and iteratively make decisions on whether to replace them, can be used for solving DCOPs. However, because of the differences between the global evaluation of a system's state and the private evaluation of states by agents, agents are unaware of the global best state that is explored by the algorithm. Previous attempts to use local search algorithms for solving DCOPs reported the state held by the system at the termination of the algorithm, which was not necessarily the (global) best state explored.

A general framework that enhances distributed local search algorithms for DCOPs with the *anytime* property is proposed. The proposed framework makes use of a BFS-tree in order to accumulate the costs of the system's state during the algorithm's iterative performance and to propagate the detection of a new best state when it is found. The proposed framework does not require additional network load. Agents are required to hold a small (linear) additional space (beside the requirements of the algorithm in use).

We further propose a set of increased exploration heuristics that exploit the proposed anytime framework. These exploration methods implement different approaches towards exploration. Our empirical study considers various scenarios including random, realistic, and structured problems. It reveals the advantage of the use of the proposed heuristics in the anytime framework over state-of-the-art local search algorithms.

1. Introduction

The Distributed Constraint Optimization Problem (*DCOP*) is a general model for distributed problem solving that has a wide range of applications in Multi-Agent Systems and has generated significant interest from researchers [2–9]. DCOPs are composed of agents, each holding one or more variables. Each variable has a domain of possible value assignments. Constraints among variables (possibly held by different agents) assign costs to combinations of value assignments. Agents assign values to their variables and communicate with each other, attempting to generate a solution that is globally optimal with respect to the costs of the constraints [4,10].

☆ This paper is an extension of [1]. Besides an extended description and examples, it proposes new exploration heuristics that exploit the anytime framework and an intensive empirical study that reveals the advantages in using the proposed framework.

* Corresponding author.
E-mail addresses: zivanr@bgu.ac.il (R. Zivan), okamotos@bgu.ac.il (S. Okamoto), hillapel@bgu.ac.il (H. Peled).

http://dx.doi.org/10.1016/j.artint.2014.03.002

There is a wide scope in the motivation for research on DCOPs, since they can be used to model many everyday combinatorial problems that are distributed by nature. Some examples are the *Nurse Shift Assignment problem* [11,12], the *Sensor Network Tracking problem* [7], and the *Log-Based Reconciliation problem* [13].

DCOPs represent real life problems that cannot or should not be solved centrally for any of several reasons, among them lack of autonomy, single point of failure, and privacy of agents.

A number of studies on DCOPs presented complete algorithms [4,5,14]. However, since DCOPs are NP-hard, there has been growing interest in the last few years in local (incomplete) DCOP algorithms [15,7,1,16,17]. Although local search does not guarantee that the obtained solution is optimal, it can be applied to large problems and is compatible with real-time applications.

The general design of most state-of-the-art local search algorithms for DCOPs is synchronous [18,7,19,20] (a notable exception was presented in [21]). In each step (or iteration) of the algorithm an agent sends its assignment to all its neighbors in the constraint graph and receives the assignments of all its neighbors. They differ in the method agents use to decide whether to replace their current value assignments to their variables, e.g., in the max gain messages (MGM) algorithm [15], the agent that can improve its state the most in its neighborhood replaces its assignment. A stochastic decision whether to replace an assignment is made by agents in the distributed stochastic algorithm (DSA) [7].

In the case of centralized optimization problems, local search techniques are used when the problems are too large to perform a complete search. Traditionally, local search algorithms maintain a complete assignment for the problem and use a goal function in order to evaluate this assignment. Different methods that balance between exploration and exploitation are used to improve the current assignment of the algorithm [22–24]. An important feature of most centralized local search algorithms is that they hold the best assignment that was found throughout the search. This makes them *anytime* algorithms, i.e., the quality of the solution can only remain the same or increase if more steps of the algorithm are performed [25]. This property cannot be guaranteed as easily in a distributed environment where agents are only aware of the cost of their own assignment (and maybe that of their neighbors too), but no one actually knows when a good global solution is obtained.

In [7], DSA and DBA are evaluated solving sensor network DCOPs. Apparently these algorithms perform well on this application even without a pure anytime property. The algorithms were compared by evaluating the state held by agents at the end of the run. However, Zhang et al. [7] do not offer a way to report the best state explored by the algorithms as proposed in this study. This limits the chances of local search algorithms implementing an exploring heuristic to be successful.

In order to implement anytime local search algorithms that follow the same synchronous structure of DSA and DBA for distributed optimization problems, the global result of every synchronous step must be calculated and the best solution must be stored. A trivial approach would be to centralize in every step the costs calculated by all agents to a single agent. This agent would then inform the other agents each time a solution that improves the results on all previous solutions is obtained. However, this method has drawbacks both in the increase in the number of messages and in the violation of privacy caused from the need to inform a single agent (not necessarily a neighbor of all agents) of the quality of all other agents' states in each step of the algorithm.

The present paper proposes a general framework for enhancing local search algorithms for DCOPs that follow the general synchronous structure with the anytime property. In the proposed framework the quality of each state is accumulated via a spanning tree of the constraint graph. Agents receive information about the quality of the recent states of the algorithm from their *children* in the spanning tree, calculate the resulting quality including their own contribution according to the goal function, and pass it to their parents. The *root agent* makes the final calculation of the cost of the system's state in each step and propagates down the tree the index of the step in which the system was in the most successful state. When the search is terminated, all agents hold the assignment of the best state according to the global goal function.

The proposed framework can be combined with any synchronous incomplete algorithm such as MGM, DSA, DBA, or Max-Sum. The combination allows any such algorithm to report the best solution it traversed during its run, i.e. it makes it an anytime algorithm.

In order to produce the best state out of m steps, the algorithm must run $m + 2h$ synchronous steps where h is the height of the tree used. Since the only requirement of the tree is that it is a spanning tree on the constraint graph, i.e., that it maintains a parent route from every agent to the root agent, the tree can be a BFS-tree and its height h is expected to be small (in the worst case h equals the number of agents n). Our experimental study reveals that starting from very low density parameters, the height of the BFS-tree is indeed very small (logarithmic in the number of agents). Previous studies on distributed systems have used BFS trees, e.g., for maintaining shortest paths in a communication network [26]. However, to the best of our knowledge we are the first to use it for aggregating local state information in order to make a decision on the global best solution.

The proposed framework does not require agents to send any messages in addition to the messages sent by the original algorithm. The additional space requirement for each agent is $O(h)$.

We study the potential of the proposed framework by proposing a set of exploration methods (heuristics) that exploit the anytime property by introducing extreme exploration to exploitive algorithms. We present an extensive empirical evaluation of the proposed methods on three different benchmarks for DCOPs. The proposed methods find solutions of higher quality than state-of-the-art algorithms when implemented within the anytime local search framework.

The rest of the paper is organized as follows: Related work is presented in Section 2. Section 3 describes the distributed constraint optimization problem (DCOP). State-of-the-art local search algorithms for solving DCOPs are presented in