Engineering Applications of Artificial Intelligence 71 (2018) 87–99

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Applying max-sum to teams of mobile sensing agents

Harel Yedidsion [a,*], Roie Zivan [a], Alessandro Farinelli [b]

[a] Industrial Engineering and Management Department, Ben-Gurion University of the Negev, Beer Sheva, Israel
[b] Computer Science Department, University of Verona, Verona, Italy

ARTICLE INFO

Keywords:
Distributed constraint optimization
Incomplete algorithms
GDL
Max-sum
Multi-agent systems
Exploration
Mobile sensor networks

ABSTRACT

Multi-agent applications that include teams of mobile sensing agents are challenging since they are inherently dynamic and a single movement of a mobile sensor can change the problem that the entire team is facing. A variation of the Distributed Constraint Optimization model for Mobile Sensor Teams (DCOP_MST) was previously adjusted to represent such problems along with local search algorithms that were enhanced with exploration methods. This paper considers the use of the Max-sum algorithm for solving problems of deploying a mobile sensor team in an unknown environment to track and monitor points of interest (targets), represented by the DCOP_MST model.

The DCOP_MST model allows the representation of different functions for aggregating the joint coverage of targets by multiple sensors. The use of different functions has a dramatic effect on the complexity of the Max-sum algorithm. When using cardinality functions, Max-sum can be performed efficiently regardless of the arity of constraints. When Max-sum is used to solve applications that require other (more complex) aggregation functions, its complexity is exponential in the arity of the constraints and thus, its usefulness is limited.

In this paper we investigate the performance of the Max-sum algorithm on two implementations of the DCOP_MST model. Each implementation considers a different *joint credibility function* for determining the coverage for each target, with respect to the locations and the credibility of agents. In the first, the coverage is calculated according to the number of agents that are located within sensing range from the target. This function can be calculated efficiently. The second takes the angle between the lines of sight of different agents to a target into consideration. The larger the difference in the angle between the lines of sight, the higher the coverage efficiency.

We analyze the challenges in adjusting the Max-sum algorithm in both scenarios and propose enhancements of the algorithm that make it more efficient. We provide empirical evidence of the advantages resulting from these enhancements in comparison to the naive algorithm.

1. Introduction

As development of autonomous robots rapidly expands, alongside sensor, actuation and communication technology, it is likely that soon, teams of mobile sensing agents would be commonly used to perform various collective tasks. Some challenging applications of Mobile Sensor Teams (MSTs) include tracking and monitoring points of interest in an unknown environment (Lesser et al., 2012; Zivan et al., 2015), measuring a scalar field (La and Sheng, 2013), maintaining wireless sensor networks (Hermelin et al., 2017), and creating a communication network (Jain et al., 2009). In other applications, MST's form rescue teams operating in disaster areas (Macarthur et al., 2011; Pujol-Gonzalez et al., 2013). Examples of underwater data collection using autonomous underwater robotic vehicles include monitoring of algal blooms (Smith et al., 2010), seismic activity (Nooner and Chadwick, 2009), measurement of ocean currents (Hollinger et al., 2016) and schools of robotic fish monitoring pollution in waterways (Hu et al., 2011). Moreover, the advancement of the internet-of-things (IoT) technology provides the necessary infrastructure for mobile sensors to become smart agents, which can share information and coordinate their actions (Rust et al., 2016). In such a setting, where a large number of mobile agents need to cooperate, it is important to have efficient protocols for communication, task allocation, deployment and decision-making.

MSTs are inherently decentralized as each agent has exclusive control of its own location and has limited computational and communication resources. As the number of agents increases, these limitations necessitate that computation and communication be distributed over the

* Corresponding author.
E-mail addresses: yedidsio@bgu.ac.il (H. Yedidsion), zivanr@bgu.ac.il (R. Zivan), alessandro.farinelli@univr.it (A. Farinelli).

https://doi.org/10.1016/j.engappai.2018.02.017
Received 11 January 2017; Received in revised form 5 February 2018; Accepted 19 February 2018

entire team to avoid a single point of failure, communication bottlenecks or unacceptably long delays.

Modeling distributed multi agent systems is often done using the Distributed Constraint Optimization (DCOP) framework (Maheswaran et al., 2004b; Meisels, 2008; Yeoh et al., 2008; Le et al., 2016) (Section 3.1 offers a formal description of the framework).

Recently, Zivan et al. (2015) proposed a model and corresponding local search algorithms for representing and solving such scenarios, particularly focusing on teams of mobile sensing agents that need to select a deployment for the sensors in order to cover a partially unknown environment—*DCOP_MST*. The DCOP_MST model is an extension of the DCOP model that allows agents to adjust their location in order to adapt to dynamically changing environments. The local distributed search algorithms that were proposed for solving DCOP_MST, were adjustments of standard local search techniques (such as Maximum Gain Message (MGM) Maheswaran et al., 2004a and Distributed Stochastic Algorithm (DSA) Zhang et al., 2005) to the model, enhanced by specifically-designed exploration methods (Zivan et al., 2015). The need for reasonable response times drives agent to only consider alternative positions in their local environment. This locality in turn, generates the need to enhance the algorithms with exploration methods that enable agents to consider suboptimal positions in order to escape local minima and find targets outside of their local environment.

The Max-sum algorithm has been the subject of intensive study in DCOP problems and has been applied to many realistic applications including mobile sensor networks (Stranders et al., 2009; Vargo et al., 2013), supply chain management (Chli and Winsper, 2015) and teams of rescue agents (Ramchurn et al., 2010). Max-sum is an incomplete inference algorithm, which propagates costs/utilities, unlike incomplete local search algorithms in which agents share their selected assignments with their neighbors (Zivan et al., 2014). While on random synthetic problems, Max-sum is outperformed by local search algorithms, in many realistic scenarios, such as sensor network scenarios, Max-sum was found to have an advantage (Farinelli et al., 2008, 2013; Stranders et al., 2009; Voice et al., 2010). This motivates the efforts to apply Max-sum to DCOP_MST and evaluate its performance in realistic mobile sensor scenarios, which can be modeled by DCOP_MST.

The need for exploration can be reduced by extending the local environments of the agents and allowing them to consider more distant tasks/targets. However, this would increase the number of agents that can be assigned to each task. Since the computation performed by Max-sum is exponential in the number of agents involved in a constraint, constraints that involve many agents (*k-ary*) represent a computational bottleneck. While a number of techniques were proposed to reduce such complexity (Stranders et al., 2009; Macarthur et al., 2011), they are not applicable to every implementation.

Thus, in this work we apply the Max-sum algorithm to two implementations of the DCOP_MST model. Each implementation uses a different function for calculating the joint coverage of a target by the agents that are located in sensing range from it. The first (F_{Sum}), simply adds the credibilities of agents in range. Thus, the optimal joint coverage for this target can be calculated efficiently (Tarlow et al., 2010; Pujol-Gonzalez et al., 2013). The second, (F_{PP}), takes into consideration the angle between the line of sights of agents to the target, assuming that sensing a target from the same angle produces the same information and the larger the difference in the angle between their lines of sight, the more unique information each sensor can provide. This assumption is most common when vision sensors (e.g., cameras) are used (Vazquez et al., 2003; Erdem and Sclaroff, 2006). For this scenario, targets computation is exponential in the arity of the constraint (number of neighboring agents) as in the general case.

We contribute to the state of the art first by applying the Max-sum algorithm to a complex scenario in which it encounters symmetry problems and in which standard runtime enhancement techniques fail to work. We then offer novel solutions to both the symmetry problem and to the runtime enhancement.

The application of Max-sum to F_{PP} necessitates solving the symmetry problem generated by the exploitive nature of Max-sum. We solve this problem by suggesting an efficient local version of the Ordered Value Propagation technique (Zivan and Peled, 2012).

Next, we propose a novel exploration method, specifically designed for Max-sum, based on meta-reasoning: agents select for each target a subset of the sensors that can be effective for covering it. The size of the subset is equal to the maximal number of sensors required for covering the target. This target is ignored in the process for selecting the locations of other sensors. As a result, such sensors that were not selected for coverage of targets are free to explore for new targets.

The proposed function meta reasoning method (FMR) breaks the relation between the size of the local environment of agents and the arity of the constraints, i.e., the arity of the constraint is not defined by the number of sensors that can be within sensing range of a target t after the next assignment selection (i.e., the "neighbors" of t) but rather by the required number of sensors for covering t. Thus, even if we enlarge the local environment of agents and the number of neighbors of t grows, the number of neighbors for t in the reconstructed factor-graph is bounded from above by the number of sensors required for covering it. Our empirical study reveals that a greedy heuristic for selecting the subset of the neighboring sensors for coverage improves the performance of the method further.

We empirically compare the proposed exploration methods and the adjusted iterative version of standard Max-sum to existing local search methods for DCOP_MST.

Our results demonstrate that standard Max-sum is superior to standard local search algorithms (in terms of iterations to reach convergence and solution quality) but it is outperformed by local search algorithms that include exploration methods. However, when Max-sum is combined with any of the exploration methods described, it outperforms the explorative local search algorithms, and the combination of Max-sum with FMR dominates all other approaches. Moreover, we demonstrate that an increase in the size of the local environments of agents does not affect the runtime required for completing an iteration for agents performing the FMR method while the runtime required for agents to complete an iteration in all other methods based on Max-sum grows exponentially.

The rest of the paper is organized as follows: Section 2 discusses previous work, while Section 3 describes the DCOP_MST model and the existing leading solution algorithms. Section 4 presents the adjustment of Max-sum for solving DCOP_MSTs (i.e., Max-sum_MST). Section 4.5 explains the symmetry problem in Max-sum and our proposed solution. Section 4.6 describes the exploration methods we propose. Finally, Section 5 describes our experimental study and Section 6 concludes the paper.

2. Related work

The problem of coordinating distributed sensor networks has been solved using a wide range of techniques ranging from bio-inspired behaviors (Xiang and Lee, 2008; Leitão et al., 2012; Das et al., 2014) and machine learning techniques (Wang and de Silva, 2008), to economic and game-theoretic mechanisms (Hsieh, 2009). Other modeling approaches, geared towards software agents, utilize agent-based technology (Aiello et al., 2009; Fortino and Galzarano, 2013). A number of papers proposed the DCOP model for representing and solving coordination problems related to sensor networks (Farinelli et al., 2013; Nguyen et al., 2014) and mobile sensor networks (Stranders et al., 2009).

DCOP is a general model for distributed problem solving that has been widely used to coordinate the activities of cooperative agents (Maheswaran et al., 2004a; Zhang et al., 2005; Rogers et al., 2011; Li et al., 2016). The DCOP literature offers a rich wealth of solution techniques, ranging from complete approaches (Modi et al., 2005), which are guaranteed to find the optimal solution, to heuristic methods (Zhang