Expert Systems with Applications 41 (2014) 5139–5157

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Review

Distributed Constraint Optimization Problems: Review and perspectives

Allan R. Leite [a,*], Fabrício Enembreck [a], Jean-Paul A. Barthès [b]

[a] Pontifical Catholic University of Parana, PUCPR, Brazil
[b] Université de Technologie de Compiègne, UTC, France

ARTICLE INFO

Keywords:
Multi-agent systems
Autonomous agents
Distributed Constraint Optimization Problems

ABSTRACT

Intelligent agents is a research area of the Artificial Intelligence intensely studied since the 1980s. Multi-agent systems represent a powerful paradigm of analyzing, projecting, and developing complex systems. One of the main difficulties in modeling a multi-agent system is defining the coordination model, due to the autonomous behavior of the agents. Distributed Constraint Optimization Problems (DCOP) have emerged as one of most important formalisms for coordination and distributed problem solving in multi-agent systems and are capable of modeling a large class of real world problems naturally. This work aims to provide an overview and critical review of DCOP, addressing the most popular methods and techniques, the evolution and comparison of algorithms, and future perspectives on this promising research area.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Intelligent agents is a research area of the Artificial Intelligence intensely studied since the 1980s (Wooldridge & Jennings, 1995). The development of agent-based applications has drawn attention due to the distributed and autonomous nature of the software entities. By definition, a software agent has intelligence regarding a specific domain and a local view of the problem, which are used to reason and evaluate the impact of their actions on the environment. In addition, an agent is able to take action autonomously. Meanwhile, because agents have a partial view of the problem, they must cooperate to solve their local problems. One of the difficulties involved during the conception of multi-agent systems is related to the ability of agents to cooperate and coordinate their actions to achieve a global solution.

Therefore, the decentralized coordination is a key issue in a multi-agent system. Defining the coordination model is usually not a trivial task due to the autonomous behavior and interest of the agents. Some challenges of the decentralized coordination in multi-agent systems involve communication model, decentralized control, partial knowledge, and self-interested agents (Shoham & Leyton-Brown, 2009). To address these challenges of the decentralized coordination, there are several approaches in literature, such as reactive coordination (Ferber, 1999; Jennings & Bussmann, 2003), learning (Sutton & Barto, 1998; Wooldridge, 2009), coalition formation (Sandholm & Lesser, 1995; Weiss, 1999), planning (O'Hare & Jennings, 1996; Scerri, Vincent, & Mailler, 2010), and negotiation protocols (Rosenschein & Zlotkin, 1994; Shoham & Leyton-Brown, 2009). One of the difficulties in these approaches is to enable an efficient adaptive behavior of the agents during their interactions.

Distributed Constraint Optimization Problems (DCOP) have emerged as one of the main coordination techniques in multi-agent systems due to their ability to optimize the global objective function of the problem described as the aggregation of distributed constraint cost functions (Scerri et al., 2010). The main motivation to employ this formalism as a decentralized coordination model in multi-agent systems is that DCOP algorithms are distributed, scalable, and robust.

By definition, DCOP consists of a set of constraints and variables distributed among agents, where each variable has a finite and discrete domain. Each value in the domain represents a possible state of the agent. Constraints are denoted by a cost or reward relation between a pair of variable assignments. Therefore, DCOP aims to find a sequence of actions that minimizes the global cost of the problem modeled as a set of constraints.

Because combinatorial problems such as satisfaction and optimization problems have been shown to be an NP-Hard problem (Garey & Johnson, 1979), DCOP requires efficient strategies to reduce the search space of the problem. In general, the main research focus is on developing new DCOP algorithms. As a result, several DCOP algorithms have been proposed so far; however, each technique has some limitations. Thus, the choice of DCOP algorithm

* Corresponding author. Address: Pontifical Catholic University of Parana (PUC-PR), Post-graduate Program in Informatics (PPGIA), Imaculada Conceição street, 1155 Prado Velho, Curitiba, PR, Brazil. Tel.: +55 (47) 9914 7596.
E-mail address: allan@ppgia.pucpr.br (A.R. Leite).

http://dx.doi.org/10.1016/j.eswa.2014.02.039
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

is an important task in obtaining a suitable performance on a specific problem. In other cases, these limitations may be prohibitive in certain real world domains. This work aims to present an overview and critical review of DCOP formalism, addressing the main techniques, evolution and comparison of algorithms, and future prospects of this promising research area.

2. DCOP background

DCOP provides a generic framework that can model a large class of real world coordination tasks in multi-agent systems. Some possible domains include sensor networks (Lesser et al., 2003), traffic control (Junges & Bazzan, 2008), meeting scheduling (Maheswaran, Tambe, Bowring, Pearce, & Varakantham, 2004a), and wireless network configuration (Monteiro, Pujolle, Pellenz, Penna, & Souza, 2012). DCOP involves a number of agents handling variables with finite and discrete domains. Variables may have constraints, which represent a cost function for each possible assignment. DCOP seeks assignments that minimize the global cost of the problem. Thus, DCOP is modeled through constraint programming techniques.

2.1. Constraint Programming

Constraint Programming (CP) is the study of computational systems based on constraint, one of several problem-solving methods in Artificial Intelligence. In general, constraint programming aims to solve complex problems, particularly combinatorial ones, by constraints defined on an application domain. A constraint is a logical relation between one or more variables, where each variable has a domain. The goal of the logical relations is to restrict the possible values that variables can assume. Constraints specify part of the information on a given problem, may be heterogeneous (i.e., constraints among variables of different domains), are declarative, and are rarely independent (Barták, 2001).

Therefore, CP provides techniques for solving a wide class of complex problems, especially problems such as resource scheduling, planning, coordination and combinatorial optimization. One of the most popular CP approaches is the Constraint Satisfaction Problem (Tsang, 1993). In this approach, the goal is to find a set of assignments for each variable such that no constraint is violated.

2.2. Constraint Satisfaction Problem

A Constraint Satisfaction Problem (CSP) consists of finding a sequence of actions to be taken to lead to a valid solution, which is initially unknown. However, intelligent methods can be applied to solve these problems to optimize the search to focus on promising states. CSP may be described as a problem whose purpose is to find a combination of values for all variables such that all constraints are satisfied. A CSP consists of a set of variables, a finite and discrete domain for each variable, and a set of constraints (Tsang, 1993). A solution to a CSP is an assignment of a value from the domain to each variable, such that all constraints are satisfied.

2.3. Distributed Constraint Satisfaction Problem

CSP defines a formalism to solve complex problems in a centralized way. Nevertheless, a CSP can be extended to operate as a distributed solving process. Several domains of the Distributed Artificial Intelligence can be modeled as Distributed Constraint Satisfaction Problems (DisCSP). In DisCSP, the variables of the problem are distributed among agents whose goal is to coordinate their actions to avoid local optima (Yokoo, Durfee, Ishida, & Kuwabara, 1998).

DisCSP can naturally be represented by a constraint graph. Each agent corresponds to a node and the edges represent the constraints between two connected nodes. The solution is achieved by local computations performed at each node of the graph. The execution order of these local computations may be arbitrary or highly flexible and may be synchronous or asynchronous (Yokoo et al., 1998).

2.4. Constraint Optimization Problem

Methods based on constraint satisfaction can find fulfilling solutions, when they exist. However, this formalism cannot model problems where solutions have degrees of quality or cost. In DisCSP, a solution can be characterized only as satisfactory or unsatisfactory.

A Constraint Optimization Problem (COP) is considered a generalization of CSP, in which the constraints have costs or rewards. In other words, each constraint of the problem is defined as an optimization function, also known as a cost function. The purpose of a COP is to find a set of assignments for all variables to optimize the global objective function of the problem (Modi, Shen, Tambe, & Yokoo, 2005). Solving a COP is much more complex than solving a CSP.

2.5. Distributed Constraint Optimization Problem

A Distributed Constraint Optimization Problem (DCOP), although an extension of DisCSP, offers techniques that go beyond satisfaction problem-solving or simple optimization methods (Lesser et al., 2003). In a DCOP, as in DisCSP, the variables are distributed among agents, but the local computations in a DCOP aim to optimize the cost of the global function defined for the problem.

Another feature of a DCOP is to provide bounds on solution quality (Modi et al., 2005). This bounded-error method is useful when the domain has time restriction or hardware limitations, but the solution quality is still within a previously defined bound. In general, large-scale or real-time applications are the most suitable for employing such incomplete methods.

2.6. DCOP definition

In general, the definition of DCOP is similar to DisCSP. A DCOP consists of n variables $V = \{v_1, v_2, \ldots, v_n\}$, where the values of the variables are taken from finite and discrete domains $D_1, D_2, \ldots, D_n$, respectively. Only agent x_i can assign values to v_i and knows D_i. Each agent x_i chooses a value d_i, where $d_i \in D_i$. Therefore, agents must coordinate their actions to choose values of their variables to optimize the cost of the global function of the problem (Mailler & Lesser, 2004).

In fact, the main difference between DCOP and the DisCSP formalism is the definition of constraint. In DCOP, constraints are cost or reward functions. A cost function for a pair of agents x_i and x_j is defined by $f_{ij} : D_i \text{ x } D_j \rightarrow N$. Two agents are neighbors if a constraint exists between them. Thus, a DCOP must find a set of A^*

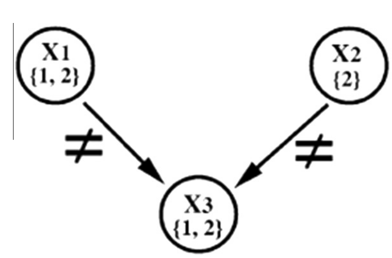

Fig. 1. DisCSP represented as a constraint graph (Yokoo et al., 1998).

Download English Version:

https://daneshyari.com/en/article/383735

Download Persian Version:

https://daneshyari.com/article/383735

Daneshyari.com