Distributed Constraint Optimization and its Application to Multiagent Resource Allocation

Pragnesh Jay Modi
University of Southern California/Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292, USA
{modi}@isi.edu

Abstract

Distributed optimization requires the optimization of a global objective function that is distributed among a set of autonomous, communicating agents and is unknown by any individual agent. The problem is inherently distributed and the solution strategy has no control over the given distribution. Constraint based techniques offer a promising approach for these types of problems. However, previous work has either limited representation to binary good/nogood constraints or relied on synchronous sequential computation to find optimal solutions. Asynchronous methods have previously lacked any guarantees of optimality. This work proposes *Adopt*, an asynchronous, distributed, complete method for solving distributed constraint optimization problems. The fundamental ideas in Adopt are to represent constraints as discrete functions (or valuations) — instead of binary good/nogood values — and to use the evaluation of these constraints to measure progress towards optimality. In addition, Adopt uses a sound and complete partial solution combination method to allow non-sequential, asychronous computation. Adopt is applied to a real-world distributed resource allocation problem. Distributed resource allocation is a general problem in which a set of agents must optimally assign their resources to a set of tasks with respect to certain criteria. It arises in many real-world domains such as distributed sensor networks, disaster rescue, hospital scheduling, and others.

Thesis Proposal

Distributed Constraint Optimization Problem consists of n variables $V = \{x_1, x_2, ...x_n\}$, each assigned to an agent, where the values of the variables are taken from finite, discrete domains $D_1, D_2, ..., D_n$, respectively. Only the agent who is assigned a variable has control of its value and knowledge of its domain. The objective is to choose values for variables such that some criterion function over all possible assignments is at an extremum. In general optimization problems, one can imagine any arbitrarily complex criterion function. In this work, we restrict ourselves to functions that can be decomposed into the sum of a set of binary (and/or unary) functions. Thus, for each pair of variables x_i, x_j, we are given a *cost function* $f_{ij} : D_i \times D_j \rightarrow N \cup \infty$. Intuitively, one can think of the cost function as quantifing the degree to which a particular assignment of values to a pair of variables is "deficient", or less than optimal. The objective is to find a complete assignment $\mathcal{A}^*$ of values to variables such that the total deficiency is minimized. (An assignment is *complete* if all variables in V are assigned some value.) More formally, let $\mathcal{C} = \{\mathcal{A} \mid \mathcal{A}$ is a complete assignment of values to variables in V $\}$. We wish to find $\mathcal{A}^*$ such that $\mathcal{A}^* = \arg\min_{\mathcal{A}\in\mathcal{C}} F(\mathcal{A})$, where

$$F(\mathcal{A}) = \sum_{x_i, x_j \in V} f_{ij}(d_i, d_j) \quad , where\ x_i = d_i,\ x_j = d_j\ in\ \mathcal{A}$$

A general distributed resource allocation problem consists of a set of agents that can each perform some set of operations and a set of weighted tasks to be completed. In order to be completed, a task requires some subset of agents to perform the necessary operations. Thus, we can define tasks by the operations that agents must perform in order to complete them. The problem to be solved is an allocation of operations to tasks such that all tasks are performed, or if resources are limited, the sum of the performed tasks is maximized. More formally, a Distributed Resource Allocation Problem is a structure $<\mathcal{A}g, \mathcal{O}, \mathcal{T}, w>$ where

- $\mathcal{A}g$ is a set of agents, $\mathcal{A}g = \{A_1, A_2, ..., A_n\}$.
- $\mathcal{O} = \{O_1^1, O_2^1, ..., O_p^i, ..., O_q^n\}$ is a set of operations, where operation O_p^i denotes the p'th operation of agent A_i. An agent can only perform one operation at a time.
- $\mathcal{T}$ is a set of tasks, where a task is a collection of sets of operations. Let T be a task in $\mathcal{T}(T \subseteq$ power set of $\mathcal{O})$. $t_r \in T$ is a set of operations called a *minimal set* because it represents the minimal resources necessary to complete the task. There may be alternative minimal sets that can be used to complete a given task. Minimal sets from two different tasks *conflict* if they contain operations belonging to the same agent.
- w: $\mathcal{T} \rightarrow N \cup \infty$ is a *weight function* that quantifies the cost of not completing a given task.

A *solution* to a resource allocation problem involves choosing non-conflicting minimal sets for tasks such that all task are completed or the cost of ignored tasks is minimized when resources are limited. This work addresses the resource allocation problem by modelling it as a distributed constraint optimization problem and new algorithms are presented for solving distributed constraint optimization.