Late-Breaking Developments in the Field of Artificial Intelligence
Papers Presented at the Twenty-Seventh AAAI Conference on Artificial Intelligence

Virtual Structure Reduction for Distributed Constraint Problem Solving

Nathaniel Gemelli and **Jeffrey Hudack**
{nagemell, jwhudack}@syr.edu
Syracuse University, Syracuse, NY, USA
Air Force Research Laboratory, Rome, NY, USA

Jae C. Oh
jcoh@ecs.syr.edu
Syracuse University
Syracuse, NY, USA

Abstract

Distributed Constraint Problem solving represents a fundamental research area in distributed artificial intelligence and multi-agent systems. The constraint density, or the ratio of the number of constraints to the number of variables, determines the difficulty of either finding a solution or minimizing the set of variable assignment conflicts. Reducing density typically reduces difficulty. We present a fully distributed technique for reducing the effective density of constraint graphs, called Virtual Structure Reduction (VSR). The VSR technique leverages the occurrence of variables that must be assigned the same value based on shared constraints and can improve solver performance using existing algorithms. We discuss our Distributed Constraint Optimization Problem (DCOP) solver, integrated with the Distributed Stochastic Algorithm (DSA), called VSR-DSA. The VSR-DSA algorithm demonstrates performance gains vs DSA in both solution quality and time on 3-coloring problems.

Introduction

Constraint density, or the ratio of number of constraints to number of variables, has a significant effect on the difficulty of distributed problem solving. As the density of problem instances increases, a rapid progression from underconstrainedness to over-constrainedness occurs. As density increases Distributed Constraint Problems tend to become more difficult to solve. This paper presents a fully distributed method for reducing constraint density via the concept of *frozen pairs* (Culberson and Gent 2001).

This work draws upon theoretical results in Constraint Satisfaction Problem (CSP) analysis. In (Cheeseman, Kanefsky, and Taylor 1991) and (Culberson and Gent 2001), methods for reducing the average density of a CSP are suggested but not algorithmically defined. The operations provided, or discussed, are limited to a centralized approach, or CSP. We know of no distributed approach performing density reduction for constraint graphs. Our work codifies the ideas from (Cheeseman, Kanefsky, and Taylor 1991) and captures the phenomenon of *frozen pairs* from (Culberson and Gent 2001) to provide a fully distributed density reduction capability. Low constraint density problems are more likely to yield a consistent solution from an initial random assignment as opposed to high constraint densities where it becomes more likely that an algorithm will find portions of the problem space that are unsatisfiable given the number of consistent values available to the variables. For both Distributed Constraint Satisfaction Problems (DisCSP) and Optimization (DCOP), lower densities typically lead to problem instances that are easier to solve.

Virtual Structure Reduction

The Virtual Structure Reduction (VSR) technique relies on the detection of frozen pairs. We define *frozen pairs* in terms of a constraint graph with: a set of variables $V = \{x_1, \ldots, x_n\}$, a set of domains $D = \{D_1, \ldots, D_n\}$ where each D_i has exactly k values, and a set of m binary constraints $C = \{C_1, C_2, ..., C_m\}$ where each $C_i(d_p, d_q)$ is a predicate on the Cartesian product $D_p \times D_q$ that returns 0 iff the values of the variables satisfy the constraint, otherwise returns 1. Let N be the neighbor function, such that $N(x_i) = \{x_p | \exists C(d_i, d_p) \in C\}$ and S is the shared neighbor function, such that $S(x_i, x_j) = N(x_i) \cap N(x_j)$. We define x_i to be frozen with x_j iff $S(x_i, x_j)$ contains a maximal (k-1)-clique.

When frozen, x_i and x_j must have the same variable assignment. Variables that are frozen to one another have a value that can be induced by the assignment of the core. This property ensures that any agent who is frozen can negotiate a local assignment with the members of the core and share that assignment with the other frozen agent. We term this negotiating agent as the *surrogate*. Surrogate agent's hold all agents whom they are negotiating for in a *spine list*. Once the surrogate finds a suitable variable assignment, it informs its spine list.

The VSR technique is an integrated solution with the Distributed Stochastic Algorithm (DSA), called VSR-DSA. The algorithm consists of two main phases: *reduction* and *assignment*. The Reduction phase is a 3-state cyclic process, discussed below, for discovering and assigning surrogates that terminates when no further reductions are found. The Assignment phase assigns a value to variables and leverages the surrogate assignments provided by the reduction phase.

The **reduction phase** is performed locally by agents that are currently *active*, with all agents begin initialized with *active* = *true*. When an agent is not active, it is because that agent has relinquished control of its variable assignment.

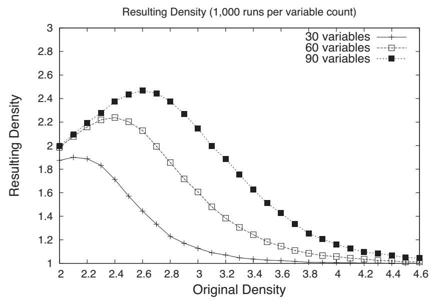

Figure 1: Resulting density after running VSR on 3-coloring constraint graphs.

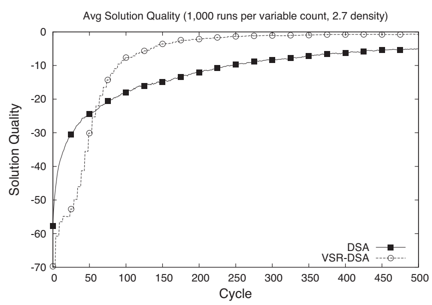

Figure 2: Average solution quality as minimization of conflict in k=3 graph coloring with 39 variables.

(1) Local Structure Sharing. All agents share their local view (structure) with all their neighbors. An agent's local view consists of their variable(s) domain and set of constraints with other variables.

(2) Frozen Pair Discovery. Each agent processes the local views provided by neighbors to determine if it is frozen with another. Agents that are frozen with each other will determine this independently and locally. In the event an agent discovers it is frozen with another agent(s) it must also determine whether or not it will potentially become the *surrogate* agent, determined via lexicographical identifier. Agents that will relinquish control to a surrogate send a message to their neighbors indicating such. The agent's neighbors update their own local view accordingly to ensure consistency of local structural knowledge.

(3) Collapse. Non-surrogate agents send a message to their surrogate consisting of intent to relinquish control, local view, spine list (if any). The new surrogate will update its local view and combine the two spine lists. Sending the spine list is necessary as it is possible that the collapsing agent is itself acting as a surrogate.

In the **Assignment phase**, variable assignment is attempted every cycle when no local structure changes have occurred by calling a minimally modified version of DSA, a variation on the Fixed Probability algorithm (referred to as DSA-B in (Zhang, Wang, and Wittenburg 2002)) outlined in (Fitzpatrick and Meertens 2001). Our DSA implementation accounts for passing variable assignment down an agent's potential spine list.

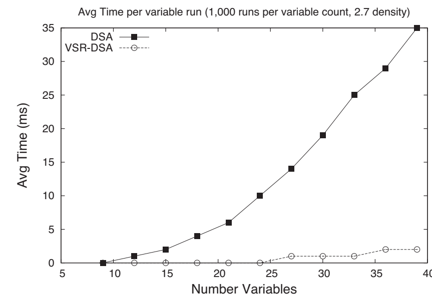

Figure 3: Average time in msec to result or termination.

Evaluation

Our first evaluation is with respect to the reduction in density of the constraint graphs using only the reduction capabilities of VSR. We present the resulting density measures after running VSR on constraint graphs of different sizes and starting densities in Figure 1. Our results show that using VSR on problems with original densities that are relatively low (i.e. 2.0) we see very little gain. This is due to the lack of core structures that create frozen pairs in the problem. Without a sufficient number of frozen pairs, significant reduction can not be found. In Figure 1 we see that as density rises, more reduction occurs, ultimately converging to 1, a k-clique.

To test performance, we evaluate the VSR-DSA algorithm on the *MaxSAT 3-colorability* DCOP using randomized graph instances. Given a set of cost functions $f = \{f_1, \ldots f_m\}$ where each $f_i(d_i, d_j)$ returns 0 iff $d_i \neq d_j$, 1 otherwise, the goal is to find a set of assignments $A = \{d_1, \ldots, d_n\}$ such that global cost $G(A) = \sum_{i=1}^{m} f_i(A)$ is minimized.

We show the experimental results regarding solution quality (Figure 2) and time (Figure 3) required to solve 2.7 density DCOPs. The two algorithms shown are DSA and VSR-DSA. In Figure 2, the VSR-DSA algorithm convergence slightly slower in the beginning (in cycles), but ultimately converges to a better minimization of the cost functions in f. The convergence speeds up once structural collapsing is nearing complete. In Figure 3, we see that VSR-DSA drastically outperforms the basic DSA algorithm in terms of real computation time. This performance gain is due to the reduced number of agents that the system has to wait on for negotiation to occur. Due to agents that have relinquished control of their variable assignments, there is less communication and a hastened turnaround time until the set of agents in the system get to re-negotiate.

Conclusions

We have introduced a fully distributed approach for reducing problem density in constraint graphs called Virtual Structure Reduction (VSR). We have shown empirical results that the resulting reduction of density can lead to increased performance using the VSR-DSA algorithm. Our preliminary results suggest that other existing solvers might benefit from the VSR technique as either an integration or pre-processing step.

References

Cheeseman, P.; Kanefsky, B.; and Taylor, W. 1991. Where the really hard problems are. In *Proceedings of the 12th IJCAI*, 331–337.

Culberson, J., and Gent, I. 2001. Frozen development in graph coloring. *Theoretical Computer Science* 265(1):227–264.

Fitzpatrick, S., and Meertens, L. 2001. An experimental assessment of a stochastic, anytime, decentralized, soft colourer for sparse graphs. *Stochastic Algorithms: Foundations and Applications* 345–347.

Zhang, W.; Wang, G.; and Wittenburg, L. 2002. Distributed stochastic search for constraint satisfaction and optimization: Parallelism, phase transitions and performance. In *Proceedings of AAAI Workshop on Probabilistic Approaches in Search*.