Dynamic Prioritization of Complex Agents in Distributed Constraint Satisfaction Problems

Aaron A. Armstrong
Edmund H. Durfee

Dept. of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109 USA
{armst, durfee}@umich.edu

Cooperative distributed problem solving (CDPS) is often modeled as being done by a group of loosely-coupled computational agents involved in extensive local computations. A useful way of viewing this is as a distributed constraint satisfaction problem (DCSP), where there are constraints between the local solutions of the different agents. The agents want to exchange enough information to identify violations of constraints and to rectify them.

One way of ensuring systematic exchange of partial solutions and identification of constraint violations is to order the agents, such that some agents make commitments to particular solutions that others then need to work around. If a work-around cannot be found, the system backtracks by asking agents up the pecking order to try different commitments. While instituting an ordering over the agents leads to systematic exploration, in the worst case there could still be an exhaustive search over the space of combinations of local solutions. To make this approach more effective, therefore, it can help if the ordering over the agents tends to focus search in more promising areas first.

Mapping this back to the CSP framework, it seems that ordering the agents is analogous to ordering the variables. In the past, this strategy has been employed, along with the typical assumption that each agent has one variable. However, in CDPS, to use communication bandwidth efficiently, the problem is distributed into a relatively small number of complex local problems. Realistically then, the agents must be thought of as having multiple variables. Now, even if variable ordering information is available, agents cannot be ordered strictly based on single variable ordering, because it is unclear how best to combine variable priorities to obtain a ranking of the agents. Further, even if good methods for generating agent priorities were found, they could still be inferior to algorithms allowing dynamic priority assignment, since dynamic prioritization allows the CSP search process to discover and take into account information particular to the current problem.

This work has been supported, in part, by the National Science Foundation under PYI award 91-58473.

Our algorithm draws upon the Asynchronous Backtracking algorithm (ABT) of Yokoo, Durfee, et al. (1992) and upon the Weak-Commitment algorithm of Yokoo (1994). Given a prior problem division among the agents, it attempts to solve the problem in a series of epoches. At the start of an epoch, each agent calculates a local priority estimate and then broadcasts the estimate to the others. Each collects these priorities and determines its position in a total (linear) order. During the epoch, the agents use ABT to search for solutions and detect no-goods. An epoch ends when a solution is discovered, the last possible solution has been disallowed, or criteria for reprioritization are satisfied.

Our investigations have identified two particularly useful approximations of the "least-constrained agent" heuristic for computing agent priorities. The first uses measures based on the number of no-goods discovered by an agent. The number of no-goods that an agent encounters should approximate the degree of constraint upon that agent (Yokoo 1994). The other uses various estimates of the total number of local solutions for an agent. With problems from a path planning domain, we have found that no-good information substantially reduces resource expenditures (time, messages sent, etc.) during search. Directly calculating the number of local solutions, while very expensive locally, provides even better performance.

We have used a genetic algorithm to investigate combinational approaches. Experimentally, we have found that a combination of no-good information and estimates of the number of local solutions provides the best performance. In future work, we would like more closely to investigate the tradeoff between local computation time and number of messages sent among the agents.

References

Yokoo, M., Durfee, E., Ishida, T., and Kuwabara, K. 1992. Distributed Constraint Satisfaction for Formalizing Distributed Problem Solving. *Proc. of 12th IEEE International Conference on Distributed Computing Systems*, 614-621.

Yokoo, M. 1994. Weak-Commitment Search for Solving Constraint Satisfaction Problems. *Proc. of the 12th National Conference on Artificial Int.*, 313-318.