Experimental Evaluation of Distributed Maximal Constraint Satisfaction Algorithm

Masahiko Ando and Masato Noto

Department of Electrical, Electronics and Information Engineering,
Faculty of Engineering, Kanagawa University
3-27-1, Rokkakubashi, Kanagawa-ku, Yokohama, 221-8686, Japan
Phone: +81-45-481-5661 ext. 3807, Fax.: +81-45-491-7915
E-mail: ando@cs.ee.kanagawa-u.ac.jp, noto@cc.kanagawa-u.ac.jp

Abstract: A constraint satisfaction problem (CSP) is a general framework that can formalize various application problems in artificial intelligence. In this paper, we will focus on an important subclass of distributed partial CSP called the distributed maximal CSP that can be applied to more practical kinds of problems. Specifically, we propose a method of solving distributed maximal CSPs using a combination of approximate and exact algorithms that yields faster optimal solutions than otherwise possible using conventional methods. Experimental results are presented that demonstrate the effectiveness of the proposed new approach.

1. Introduction

In recent years, considerable research is being done on multi-agent systems in which multiple autonomous calculating processes (i.e., agents) solve problems through distributed cooperation in the field of artificial intelligence (AI). Considering that so many types of problems in AI can be formulated as constraint satisfaction problems (CSPs), extended distributed CSPs[2] were proposed enabling CSPs to be applied to distributed environments as a general framework for formally dealing with problems that involve multiple agents. However, more practical real-world problems tend to be over-constrained, and the descriptive power of the CSP is not always sufficient in formulating the problem because of the various constraints involved. Relatively little research thus far has considered the application of distributed CSPs to actual systems dealing with real-world problems. To solve such problems, it is necessary to relax the constraints of over-constrained CSPs as much as possible, then solve the relaxed CSP. Besides distributed CSPs, researchers have proposed a number of other formulations extending the framework of the CSP studies including partial CSPs, distributed partial CSPs[5][8], and so on.

In this work, we will focus on an important subclass of distributed partial CSP called the distributed maximal CSP that can be applied to more practical kinds of problems. Specifically, we propose a method of solving distributed maximal CSPs using a combination of approximate and exact algorithms that yields faster optimal solutions than otherwise possible using conventional methods. Experimental results are presented that demonstrate the effectiveness of the proposed new approach.

2. Constraint Satisfaction Problems

CSPs are search problems that involve finding one set of assignments to variables that satisfies all the given constraints from among sets of variables, each with a finite and discrete domain. Solving a CSP involves finding one set or all sets of assignments to variables that satisfies all constraints[3]. CSPs are also generally NP-complete, and one could anticipate in the worst case that the amount of computation for the number of variables could become exponential. More specifically, CSPs consist of variables, ranges of variable values, and inter-variable constraints. By range of variable values here we mean the range of values that can be assigned to a variable, and inter-variable constraints are sets of values that are permitted between multiple variables.

The variables and constraints in CSPs are allocated to multiple agents, and extended distributed CSPs[9] permit CSPs to be applied to distributed environments. The goal of distributed CSPs is for the values assigned to all agent variables to satisfy all the constraints, but distributed CSPs tend to be over-constrained[4] when they are applied to more realistic real-world problems. One framework for addressing these kinds of over-constrained distributed CSPs is the distributed partial CSP in which constraints are relaxed as much as possible, and the relaxed-constraint problems are then solved. An important subclass of distributed partial CSPs is the distributed maximal CSP which searches for the solution that minimizes the number of constraint violations.

3. Distributed Maximal Constraint Satisfaction Algorithms

As in other areas of AI, there are two competing kinds of solutions to distributed maximal CSPs[10]: exact versus approximate solutions. In this study too we employ the exact synchronous branch and bound method and the approximate iterative distributed breakout approach[1][6][7] as distributed algorithms for solving problems in a distributed state.

3.1 Synchronous Branch and Bound

Synchronous branch and bound is an exact algorithm that simulates the conventional branch and bound method in a distributed environment with multiple agents. The optimal solution and completeness of the algorithm are guaranteed since the synchronous branch and bound is an exact solution. But the solution may not be found in real time when the search space is large.

3.2 Iterative Distributed Breakout

Iterative distributed breakout applies the distributed breakout method to the distributed maximal CSP repeatedly, and then enjoys all the same advantages of the distributed breakout approach. The iterative distributed breakout derives a quasi optimal solution very quickly even when the search space is large, but completeness of the algorithm is not guaranteed. Another drawback is that, even when an optimal solution to a distributed maximal CSP is found, the iterative distributed breakout cannot determine whether the solution is optimal or not.

3.3 Problem of Algorithms

A primary objective of the present study is to derive optimal solutions to distributed maximal CSPs, so we have basically relied on the synchronous branch and bound algorithm. The difficulty with this approach as we observed earlier is that, although the conventional synchronous branch and bound guarantees completeness of the algorithm, it may not be able to find the solution in real time if the search space is large.

Clearly the key here is to somehow narrow down the search space so that solutions can be found quickly in real time.

4. Proposed Algorithm

In distributed maximal CSPs, the necessary value N and the sufficient value S are given as initial values. Assuming here that P represents CSPs that generally do not have solutions because they are over-constrained and PS is the set of CSPs in which P's have somehow been relaxed, the solution to a distributed maximal CSP is the CSP in PS and its solution that is unconstrained and its distance from P is less than the necessary value N. When a solution is found where the distance from P is less than S, the search can be terminated. The solution minimizing the distance from P is the optimal solution to the distributed maximal CSP.

Since our primary objective in this work is to quickly find the optimal solution to a distributed maximal CSP, we define the sufficient value as 0. In synchronous branch and bound, the algorithm backtracks if the necessary value of the number of a partial solution's constraint violations exceeds N. If the necessary value is less than N and the partial solution extends to the end, the solution is memorized and that number of constraint violations becomes the new N.

The main features of the proposed algorithm are as follows:

1. Reliance on synchronous branch and bound after preprocessing by iterative distributed breakout.
2. Iterative distributed breakout is run for a set, period of time to derive a quasi optimal solution to the distributed maximal CSP.
3. The number of constraint violations of the quasi optimal solution that is thus derived is regarded as the initial value of the necessary value N_0.
4. The synchronous branch and bound is run using the initial value N_0 to derive the optimal solution.

Note that in this procedure the exact algorithm is implemented last, so completeness of the algorithm, a primary virtue of exact solutions, is assured.

5. Experimental Results and Considerations

In this work we devised a system for quickly solving distributed maximal CSPs and applied it to the distributed graph coloring problem. The objective of the problem is for the agents corresponding to nodes determine their own colors that are different from the colors of other agents to which they are linked. The graph coloring problem can be easily formulated as a CSP, and for that reason it commonly used as a standard CSP example. A typical real-world problem that could be solved by applying the graph coloring problem is the problem of allocating frequencies, which can be solved by expressing the various requirements as parameters as a generalized instance of the graph coloring problem.

For our present purposes, we used a map of the United States as a distributed graph coloring problem, and ran simulations for the parameters listed in Table 1 comparing the synchronous branch and bound algorithm alone against our proposed method. The results are presented in Table 2. The initial values for the variables were determined at random, and the number of cycles and execution times are averages for all the trial runs.

One can see from Table 2 that our proposed method yielded the optimal solution in fewer cycles and shorter execution time than the stand-alone synchronous branch and bound method, so our approach is superior by both criteria. Our approach exploits the advantages of both the synchronous branch and bound and iterative distributed breakout methods since it is both fast and also guarantees completeness of the algorithm.

6. Conclusions

Our objective in this study was not to derive quasi optimal solutions to distributed maximal CSPs, but rather to obtain optimal solutions more quickly. Ultimately, one cannot guarantee the completeness of an algorithm unless an exact solution is used. But observing the extreme speed with which quasi optimal solutions are derived by iterative distributed breakout, we have demonstrated that greater search efficiency is achieved by using a synchronous branch and bound approach for completeness following preprocessing by iterative dis-

Table 1. : Simulation Conditions

Problem	Distributed graph coloring problem
Number of colors	3
Number of agents	48
Number of links	103
Number of trial runs	200 runs for each algorithm

Table 2. : Experimental Results

	Number of cycle	Execution time (ms)
Synchronous branch and bound	212	24935
Proposed method	169	22540

tributed breakout. The proposed method should therefore be quite effective in dealing with problems involving a large search space that appear insoluble in real time using the synchronous branch and bound algorithm. Let us also point out that in cases where the search space is relatively small, our method may require longer search time than the conventional approach. In future work, we plan to identify the specific ranges and types search space problems that could be solved more efficiently using our proposed method.

References

[1] K. Chandy and L. Lamport, "Distributed Snapshots: Determining Global States of Distributed Systems," *ACM Transaction on Computer Systems*, No. 3, pp. 63–75, 1985.

[2] E. C. Freuder, "Partial Constraint Satisfaction," *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 278–283, 1989.

[3] E. C. Freuder and R. J. Wallace, "Partial Constraint Satisfaction," Proceedings Artificial Intelligence, Vol. 58, No. 1–3, pp. 21–70, 1992.

[4] J. Larrosa and P. Meseguer, "Phase Transition in max–CSP," *Proceedings of the Twelfth European Conference on Artificial Intelligence*, pp. 190–194, 1996.

[5] V. R. Lesser and D. D. Corkill, "The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distibuted Problem Solving Networks," *AI Magazine*, Vol. 4, No. 3, pp. 15–33, 1983.

[6] P. Morris, "The Breakout Method for Escaping From Local Minimum," *Proceedings of the Eleventh National Conference on Artificial Intelligence,* Vol. 21, No. 6, pp. 1446–1461, 1991.

[7] M. Yokoo and K. Hirayama, "Distributed Breakout Algorithm for Solving Distributed Constraint Satisfaction Problems, "*Proceedings of the Second International Conference on Multi-Agent Systems*, pp. 401–408, 1996.

[8] M. Yokoo and K. Hirayama, "Distributed Partial Constraint Satisfaction Problem, In G. Smolka, editor," *Principles and Practice of Constraint Programming –CP97*, volume 1330 of *Lecture Notes in Computer Science*, pp. 222–236. Springer-Verlag, 1997.

[9] M. Yokoo, E. H. Durfee, T. Ishida, and K. Kuwabara, "Distributed Constraint Satisfaction for Formalizing Distributed Problem Solving," *Proceedings of the Twelfth IEEE International Conference on Distributed Computing Systems*, pp. 614–621, 1992.

[10] Y. Zhang and A. Mackworth, "Parallel and Distributed Algorithms for Finite Constraint Satisfaction Problems," *Proceedings of the Third IEEE Symposium on Parallel and Distributed Processing*, pp. 394–397, 1991.