Distributed Constraint Satisfaction with Cooperating Asynchronous Solvers

Georg Ringwelski

GMD FIRST, German National Research Center for Information Technology
Kekuléstraße 7, 12489 Berlin, Germany, georg.ringwelski@gmd.de

A Constraint Satisfaction Problem (CSP) is to find an assignment to a set of variables that is consistent wrt. a set of constraints over these variables. CSPs frequently arise in applications of distributed artificial intelligence [3] and may often not be solved by a centralized constraint solver for privacy or security reasons. In this distributed case (DCSP) constraints and variables are distributed among multiple automated agents.

To solve a CSP it has turned out to be effective to provide information gained from constraints to other constraints via common variables as soon as it is available. With this constraint propagation large parts of the search space are cut very early. Constraint propagation defines a confluent transition system if the constraints are interpreted as inflationary and monotonic functions reducing the variables' domains [1]. In Asynchronous Constraint Solving (ACS) [2] we make use of this fact by invoking the propagation algorithms of posted constraints from a buffer with an internal scheduling. In addition we offer the possibility to asynchronously retract constraints. If a previously posted constraint is retracted, all variables and constraints will have the same state as if it had never been posted. Using the asynchronously executed methods 'post' and 'retract' on variable assignments (that *constrain* a variable to have a certain value) search algorithms can be defined to solve any CSP if implementations for all used constraints are provided.

ACS is well suited for the use in DCSP because no global information is necessary for history management or other CSP tasks. Constraint postings and retractions are buffered in the priority-queue of their corresponding solver such that no synchronization has to be made. In our current Java implementation every agent holds a solver, where constraints over any physically reachable variable can be posted or retracted from any other agent or local application. As application we implement a distributed time schedulung tool where appointments can be organized and reorganized automatically with constraint satisfaction.

References

[1] Krzysztof R. Apt. The essence of constraint propagation. *Theoretical Computer Science*, 221(1-2):179–210, 1998.

[2] Georg Ringwelski. A new execution model for constraint processing in object-oriented software. In *Proceedings of the WFLP01*, 2001.

[3] Makoto Yokoo and Edmund H. Durfee. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 10(5), 1998.

T. Walsh (Ed.): CP 2001, LNCS 2239, p. 777, 2001.