

Parallel Execution Models for Constraint Propagation [★]

Alvaro Ruiz-Andino, Lourdes Araujo, Fernando Sáenz, and Jose Ruz

Department of Computer Science, University Complutense of Madrid
e-mail: alvaro@sip.ucm.es

Constraint propagation algorithms present inherent parallelism. Each constraint behaves as a concurrent process triggered by changes in the store of variables, updating the store in its turn. There is an inherent sequentiality, as well, since a constraint must be executed only as the consequence of a previous execution of another constraint. We have developed different parallel execution models of constraint propagation for MIMD distributed memory machines. We have adopted the *indexical scheme*, an adequate approach to achieve consistency for n-ary constraints. The proposed models arise from two techniques, dynamic and static, for scheduling constraint executions (assignment of constraint executions to processing elements). In the static scheduling models the constraint graph is divided into N partitions, which are executed in parallel on N processors. We have investigated an important issue affecting performance, the criterion to establish the graph partition in order to balance the run-time workload. In the dynamic scheduling models, any processor can execute any constraint, improving the workload balance. However, a coordination mechanism is required to ensure a sound order in the execution of constraints. We have designed coordination mechanisms for both centralised and distributed control schemes. Several parallel processing methods for solving Constraint Satisfaction Problems have been proposed. [1] and [3] must be remarked in relation with our work.

These execution models have been implemented in C, developed and tested on a CRAY T3E multiprocessor. Benchmarks considered so far exhibit a speedup between three and five, though better results may be expected for larger problems. The study of the distribution of constraints among processors has shown that a strongly connected partitioning is worse than a heuristic distribution that balances the ready set. Tests on broadcast frequency revealed the convenience of an immediate update. The dynamic model exhibits lower speedups than the static model.

References

1. Baudot, B., Deville, Y.: Analysis of Distributed Arc-Consistency Algorithms. Technical Report 97-07. University of Louvain. Belgium (1997).
2. Ruiz-Andino, A., Araujo, L., Ruz, J.: Parallel constraint satisfaction and optimisation. The PCSO system. Technical Report 71.98. Department of Computer Science. Universidad Complutense de Madrid, Spain. (1998).
3. Zhang, Y., Mackworth, K.: Parallel and Distributed Finite Constraint Satisfaction: Complexity, Algorithms and Experiments. Parallel Processing for Artificial Intelligence. Elsevier (1993).

[★] Supported by project TIC95-0433.