From: AAAI-94 Proceedings. Copyright © 1994, AAAI (www.aaai.org). All rights reserved.

A Hybrid Parallel IDA* Search *

Shubha S. Nerur
Computer Science and Engineering
University of Texas at Arlington
Box 19015, Arlington, TX 76019
nerur@cse.uta.edu

ABSTRACT

Heuristic search is a fundamental problem-solving method in artificial intelligence. The main limitation of search is its computational complexity which can be overcome by parallel implementation of the algorithms. Distributed tree search and Parallel window search are two of the approaches to parallelizing search algorithms.

We are developing an algorithm called HyPS (Hybrid Parallel Search) which is a combination of distributed tree search (RKR87) and parallel window search (PK91). In the HyPS, the set of processors is divided into clusters. Each cluster searches the same space, but uses a unique cost threshold. Within each cluster, processors are given unique portions of the search space to expand. Each cluster, adopts the rule of parallel windows, but distributed tree search is performed within a cluster. First, the initial state is expanded generating distinct subtrees. The number of distinct subtrees generated is equal to the number of processors in a cluster. Each processor in a cluster receives a subtree on which IDA* search is performed. The first cluster is given the heuristic estimate to reach the goal. All remaining clusters are given incrementally larger thresholds. The algorithm finds a first solution or an optimal solution. Many of the processors will be idle when searching for an optimal solution. Load balancing within a cluster will overcome the idling of processors within a cluster. HyPS is implemented on Connection Machine 5. The parallel algorithm is tested for different cluster sizes to find the first solution and the optimal solution. We are interested in finding which cluster size yields the best solution.

Test to Find	Clusters	Speedup	Efficiency
First Sol.	1	19.07	0.30
	2	22.02	0.34
	4	45.69	0.71
First Sol. with operator ordering	1	14.65	0.23
	2	27.53	0.43
	4	108.03	1.69
Optimal Sol.	1	7.9	0.12
	2	6.7	0.10
	4	2.2	0.03
Optimal Sol. with load balancing	1	9.36	0.14
	2	6.16	0.10
	4	2.93	0.05
Optimal Sol. with Load balancing & operator ordering	1	13.20	0.21
	2	6.10	0.10
	4	5.27	0.08

Test to Find	Clusters	Speedup	Efficiency
Optimal Sol. with Load balancing & operator ordering	1	57.70	0.90
	2	106.82	1.67
	4	36.71	0.57

The domain of testing is the Fifteen Puzzle problem and the robot path planning problem. The average speedup and efficiency for the Fifteen Puzzle Problem instances are shown in the first table. Also the average speedup and efficiency for the Robot Planning Problem instances are shown in the second table. The initial testing shows marked improvement over serial and parallel algorithms.

References

Curt Powley and Richard E Korf. Single-agent parallel window search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5), 1991.

V N Rao, V Kumar, and K Ramesh. A parallel implementation of Iterative-Deepening-A*. In *Proceedings of AAAI*, pages 178–182, 1987.

*Supported by the National Science Foundation grant IRI-9308308 and by grant TRA93029N from the National Center for Supercomputing Applications.