Journal of Advanced Computing Research Vol. 1, Issue 2 (2016) 14-16

A New Efficient Algorithm based on Parallel Best-First Search Algorithm for Scheduling Concurrent Tasks on Grid Computing

Nur Hidayah binti Omar [#1], Mohamed Faidz Mohamed Said [#2]
[#] *Universiti Teknologi MARA*
70300 Seremban, Negeri Sembilan, MALAYSIA
[1] dayahomar94@gmail.com
[2] faidzms@ieee.org

***Abstract*—The explanation about parallel best-first search for scheduling concurrent tasks on Grid computing is given. Parallel algorithm can perform various instructions simultaneously like in concurrent processing that can divide a complex task, combine all the individual outputs and process multiple systems to get the final result in quick time.**

Keywords—parallel, best-first search, concurrent tasks, Grid computing

I. Introduction

The problem is split up into sub-problems and executed in parallel to get individual outputs. Subsequently, these individual outputs are combined to get the final desired output. Best-First Search is an algorithm that traverses a graph to reach a target in the shortest possible path [1]. Unlike BFS and DFS, Best-First Search follows an evaluation function to determine which node is the most appropriate to traverse next. The growth of Internet along with the easy availability of computers has changed the way to store and process data. Thus, people need concurrent processing that can separate a complex task and process multiple systems to produce the output quickly.

II. Theory

A. *Bicriterion Shortest Path*

To improve efficiency, the performance of three best-first algorithms are analyzed and compared that accept heuristic information. These algorithms are NAMOA*, MOA* and Tung & Chew's algorithm (TC) [2]. The experiment investigates the effect of heuristic information. All of the algorithms considered follow a best-first search strategy.

B. *A* Orthogonal Matching Pursuit*

A developing field targeting at the reconstruction of distributed signals is known as compressed sensing, which make the recovery process under-determined. A* Orthogonal Matching Pursuit (A*OMP) is a novel semi-greedy recovery approach was proposed by [3].

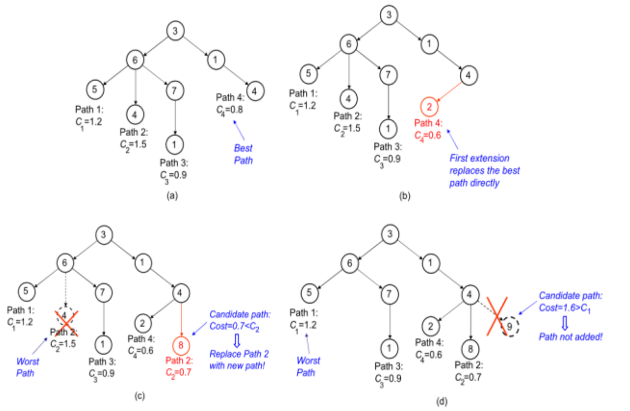

Figure 1. Evaluation of the search tree during a single iteration of the A*OMP algorithm

C. *Branch-and-Bound Algorithms*

The branch-and-bound (B&B) algorithms have been succeeding to find the exact solutions for a large array of optimization problems. According to [4], a tree search strategy has been used by B&B to implicitly enumerate all possible solutions to a given problem. The search strategy, the branching strategy, and the pruning rules are the components to improve the behavior of the algorithm. The reason of why best-first search offers a number of significant advantages over Depth First Search (DFS) is because it is able to discover right solutions earlier in the search process.

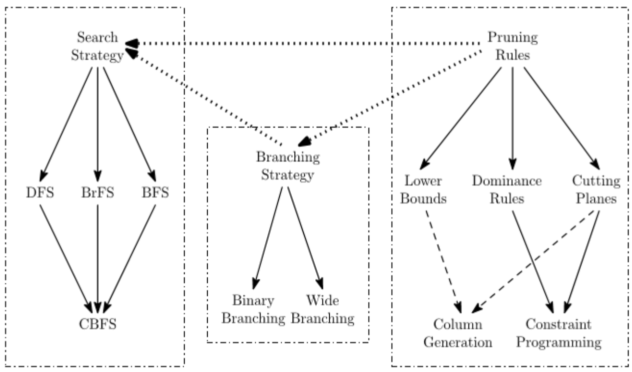

Figure 2. A diagram of relationships between various algorithm components

The best evaluation is not ever sufficient to identify the best node to select with best-first search strategy, it needs to define the breaking rules for cases when nodes have the same value as the FIFO, the LIFO and the consistent rules. The purpose is to carry through bounds of the parallel execution for breaking rule and a comparison of their efficiency to cope with these anomalies [5].

D. Best-First Search

Solving discrete optimization problems (DOP) is performed by using by parallel algorithms. BFS algorithms expand the most promising node currently ready on the search frontier [6]. Best-first branch and bound (BFBB) also referred by as A*.

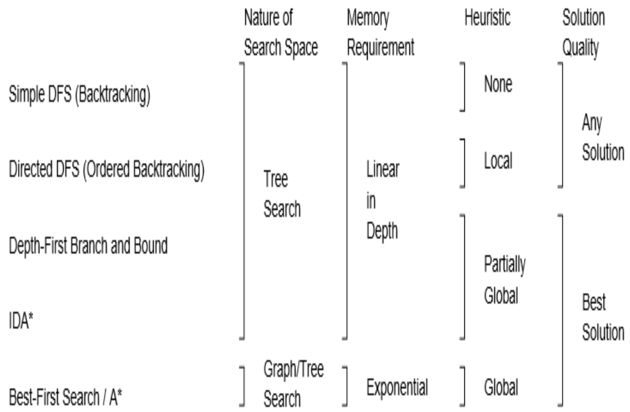

Figure 3. Various enumerative search procedures and their characteristics

A search strategy that combines Best First Search (BFS) and Depth First Search (DFS) is known as Hybrid Best First Search (HBFS) [7]. HBFS gives a global lower bound on the optimum like BFS, while also giving upper bounds like DFS. Integrating treewidth into Backtracking with Tree Decomposition (BTD) by using the lower bounds reported by HBFS.

Discover duplicate states without requiring frequent locking and abstraction to partition the state space and Parallel Best-N Block First (PBNF) uses [8]. PBNF gives speculative expansions when necessary to keep threads busy. This approach allows it to extend well to sub-optimal and anytime heuristic search. Paper [9] said that best-first heuristic search in a shared-memory setting and each thread attempts to expand the promising nodes.

A* is best-first search algorithms that need space exponential in length of the solution path. To overcome this restriction algorithms like Iterative Deepening A* (IDA*) and Recursive Best First Search (RBFS) renew parts of problem space trading space for time [10]. RBFS based algorithm which uses all available memory was presented.

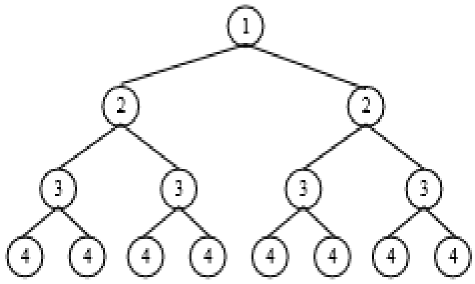

Figure 4. Example of search space with costs equal to depth

Greedy Best-First Search (GBFS) for satisfying planning was proposed, analyzed and tested in different exploration-based techniques. It shows the potential of exploration-based techniques by combining GBFS and random walk exploration locally. Uninformative Heuristic Regions (UHRs) and Early Mistakes (Ems) for GBFS are analyzed and illustrated on a number on International Planning Competition (IPC) benchmarks [11]. Corresponding solutions, namely Greedy Best-First Search with Local Exploration (GBFS-LE) and Type-based Greedy Best-First Search (Type-GBFS), are proposed and shown to outperform GBFS substantially. This also introduces the Diverse Anytime Search (DAS) framework, which reduces unproductive time and improves plan quality by randomized exploration.

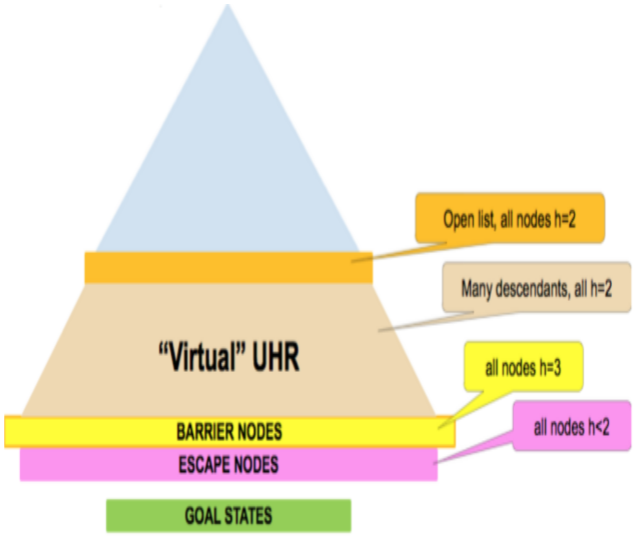

Figure 5. Abstract structure of the search tree of GBFS when it has stalled in heuristic value 2

A single-agent search algorithm that expands states of a graph same as A* does namely as Time-Bounded A* (TBA*). Time-Bounded Best-First Search (TB-BFS) is an idea of the time-bounded approach to any best-first search algorithm. TB-BFS was allowed by restarting strategies to solve search problems in dynamic environments [12].

Evaluates the power of best-first search over AND/OR search spaces for solving the Most Probable Explanation (MPE) task in Bayesian networks. The main goodness of the

AND/OR representation of the search space is its sensitivity to the structure of the problem [13].

E. Grid Computing

Provide non-trivial services to users and aggregate the power of widely distributed resources are the motivation of Grid computing [14].

III. CONCLUSION

More types of applications than just trivially parallel ones are using computational Grids. The bandwidth of wide-area networks increases rapidly.

REFERENCES

[1] tutorialspoint. (2016, 12 Oct 2016). *Parallel Algorithm - Quick Guide*. Available: https://www.tutorialspoint.com/parallel_algorithm/parallel_algorithm_quick_guide.htm

[2] E. Machuca, L. Mandow, J. L. Pérez de la Cruz, and A. Ruiz-Sepulveda, "A comparison of heuristic best-first algorithms for bicriterion shortest path problems," *European Journal of Operational Research,* vol. 217, pp. 44-53, 2012.

[3] N. B. Karahanoglu and H. Erdogan, "A* orthogonal matching pursuit: Best-first search for compressed sensing signal recovery," *Digital Signal Processing,* vol. 22, pp. 555-568, 2012.

[4] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning," *Discrete Optimization,* vol. 19, pp. 79-102, 2016.

[5] B. Mans and C. Roucairol, "Performances of parallel branch and bound algorithms with best-first search," *Discrete Applied Mathematics,* vol. 66, pp. 57 - 74, 1993.

[6] A. Grama and V. Kumar, "State of the art in parallel search techniques for discrete optimization problems," *IEEE Transactions on Knowledge and Data Engineering,* vol. 11, pp. 28-35, 1999.

[7] D. Allouche, S. De Givry, G. Katsirelos, T. Schiex, and M. Zytnicki, "Anytime hybrid best-first search with tree decomposition for weighted CSP," in *International Conference on Principles and Practice of Constraint Programming*, 2015, pp. 12-29.

[8] E. Burns, S. Lemons, W. Ruml, and R. Zhou, "Best-first heuristic search for multicore machines," *Journal of Artificial Intelligence Research,* vol. 39, pp. 689-743, 2010.

[9] E. Burns, S. Lemons, W. Ruml, and R. Zhou, "Parallel best-first search: Optimal and suboptimal solutions," in *Proceedings of the International Symposium on Combinatorial Search*, 2009.

[10] M. Robnik-Šikonja, "Effective use of memory in linear space best first search," FRI, Technical Report1996.

[11] F. Xie, "Exploration in Greedy Best-First Search for Satisficing Planning," University of Alberta, 2016.

[12] C. Hernández, R. Asin, and J. A. Baier, "Time-Bounded Best-First Search," in *Seventh Annual Symposium on Combinatorial Search*, 2014.

[13] R. Marinescu and R. Dechter, "Best-first AND/OR search for most probable explanations," *arXiv preprint arXiv:1206.5268,* 2012.

[14] F. Dong and S. G. Akl, "Scheduling Algorithms for Grid Computing: State of the Art and Open Problems," *Technical Report No. 2006-504,* 2006.