Block Architecture Problem with Depth First Search Solution and Its Application

Robbi Rahim1,6, Dahlan Abdullah2,6, Janner Simarmata3,6, Andri Pranolo4,6, Ansari Saleh Ahmar5,6, Rahmat Hidayat6,7, Darmawan Napitupulu6,8, Heri Nurdiyanto6,9, Bayu Febriadi10 and Z Zamzami10

1School of Computer Engineering and Communication, Universiti Malaysia Perlis, Malaysia
2Department of Informatics, Universitas Malikussaleh, Aceh, Indonesia
3Universitas Negeri Medan, Medan, Indonesia
4Department of Informatics, Universitas Ahmad Dahlan, Indonesia
5Department of Statistics, Universitas Negeri Makassar, Indonesia
6Komunitas Kolaborasi Publikasi Indonesia (KO2PI), Indonesia
7Department of Information Technology, Politeknik Negeri Padang, Indonesia
8Research Center for Quality System and Testing Technology, Indonesian Institute of Sciences, Indonesia
9Department of Informatics, STMIK Dharma Wacana, Indonesia
10Department of Information System, Universitas Lancang Kuning, Indonesia

*usurobbi85@zoho.com

Abstract. Searching is a common process performed by many computer users, Raita algorithm is one algorithm that can be used to match and find information in accordance with the patterns entered. Raita algorithm applied to the file search application using java programming language and the results obtained from the testing process of the file search quickly and with accurate results and support many data types.

1. Introduction

Block Architecture Problem is one of the classic problems in the field of Artificial Intelligence studies, this problem is illustrated in boxes arranged into 2 or 3 piles of beams as initial conditions. The goal of this problem is to get a pile of boxes as desired. An allowed operation in the process of completion is to decrease (x), which means that box x is derived from a certain stack and place it (x, y), meaning that the x box is placed above the y box, provided that the x and y boxes must be at the top of a pile.

Block Architecture Problem can be solved with some algorithms such as A* (A Star), Breadth First Search and Depth First Search[1]–[3]. The Depth First Search[1], [3], [4] algorithm is chosen for solving this problem because the search process of the solution will continue to achieve the desired goal[1], [5], and the Depth First Search algorithm is more efficient[6] for search space with multiple branches because it does not need to evaluate all the nodes at a certain level[6] and also the algorithm...
Depth First Search requires relatively small memory because only the nodes on the active path are stored[2]. [7].

Depth First Search algorithm used in the completion of Block Architecture Problem will be applied to the application created using Visual Basic.Net programming language to simulate the solution of the problem, the experiment is perform on 8 blocks with the number of piles of block as much as 3 pieces of piles, namely: stack-1, 2 and 3, the conclusion of Block Architecture Problem with applications could be one of solution form state and space problem.

2. Methodology
Searching[8]–[11] is one of the most important processes in searching for a solution, the block architecture problem in this research is solved by applying the Depth First Search algorithm and then implemented the application created by using Visual Basic.Net, the operation and action of the problem and the solution can be seen in the following example :

1. Initial condition, there are 3 pieces named C, A, B, C is the lowest pile and A is the top pile, see figure 1 below:

![Figure 1. Initial Condition](image)

2. Move block B from pole-1 to pole-3. The rule is that the block must be the topmost block of the 1st pole and occupy the topmost position on the 3rd pole.

![Figure 2. Moving block from pole-1 to pole-3](image)

3. Move block A from pole-1 to pole-2. The rule is that the block must be the topmost block of the 1st pole and the A beam occupies the topmost position on the 2nd pole.

![Figure 3. Moving block from pole-1 to pole-2](image)
4. Move block B from pole-3 to pole-2. The rule is that the block B must be the topmost block from pole-3 and the B block occupies the topmost position on the 2nd pole.

![Figure 4](image)

Figure 4. Moving block from pole-3 to pole-2

The above process is a manual process of how to move the block letter from pole-1 to pole-3 with existing rules, manual testing in figure 1 to figure 4 with 3 blocks and it can be more than 3 blocks, the result block after moving as follows:

![Figure 5](image)

Figure 5. Goal State

Figure 5 is a manual process of moving block letter from pole-1 to pole-3, from initial state to goal state requires many steps with different variations to completion process, and the depth first search algorithm could use to find any solution.

3. Result and Discussion
Depth First Search algorithm used in the accomplishment of Block Architecture Problem and applied to application could be used well to solve the problem, figure 6 is an application proposed with example of initial state and goal state

![Figure 6](image)

Figure 6. Solution Approach
Figure 6 shows each step for transfer blocks from each pole, the conclusion process will be shown in simulation with 1 second interval for each block when switching. the depth first search algorithm used can be applied well to the simulation created.

4. Conclusion
Block Architecture Problem can be solved well using the Depth First Search algorithm because the search process is perform almost in all possible nodes, the simulation can be used up to 9 blocks with possible solutions up to approximately 95000 process states and takes a very long time to get the fastest solution, future development allows to apply or compare algorithms such as BFS, Heuristics, Dynamic Programming to get the best results for completion Block Architecture Problem

References
[1] R. Marinescu, 2010. Best-first vs. depth-first AND/OR search for multi-objective constraint optimization, in Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI, 1, pp. 439–446.
[2] A. S. M. Lumenta, 2014. Perbandingan Metode Pencarian Depth-First Search, Breadth-First Search dan Best-First Search pada Permainan 8-Puzzle, e-Journal Tek. Elektro Komput.
[3] C. Mencía, M. R. Sierra, and R. Varela, 2013. Depth-first heuristic search for the job shop scheduling problem, Ann. Oper. Res., 206, 1, pp. 265–296.
[4] C. Rhee, Y. D. Liang, S. K. Dhall, and S. Lakshmivarahan, 1994. Efficient algorithms for finding depth-first and breadth-first search trees in permutation graphs, Inf. Process. Lett., 49, 1, pp. 45–50.
[5] U. A. Acar, A. Charguéraud, and M. Rainey, 2015. A Work-efficient Algorithm for Parallel Unordered Depth-first Search, Proc. Int. Conf. High Perform. Comput. Networking, Storage Anal., pp. 67:1–67:12.
[6] P. A. Dow and R. E. Korf, 2009. Duplicate avoidance in depth-first search with applications to treewidth, in IJCAI International Joint Conference on Artificial Intelligence, pp. 480–485.
[7] E. Zunic, A. Djedovic, and B. Zunic, 2016. Software solution for optimal planning of sales persons work based on Depth-First Search and Breadth-First Search algorithms, in 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2016 - Proceedings, pp. 1248–1253.
[8] R. Rahim, I. Zulkarnain, and H. Jaya, 2017. A review: search visualization with Knuth Morris Pratt algorithm,” in IOP Conference Series: Materials Science and Engineering, 237, 1, pp. 12026.
[9] R. Rahim, I. Zulkarnain, and H. Jaya, 2017. Double hashing technique in closed hashing search process,” IOP Conf. Ser. Mater. Sci. Eng., 237, 1, pp. 12027.
[10] R. Rahim, S. Nurarrif, M. Ramadhan, S. Aisyah, and W. Purba, 2017. Comparison Searching Process of Linear, Binary and Interpolation Algorithm, J. Phys. Conf. Ser., 930, 1, pp. 12007.
[11] Rahim, R., Ahmar, A.S., Ardyanti, A.P., and Nofriansyah, D., 2017. Visual Approach of Searching Process using Boyer-Moore Algorithm, J. Phys. Conf. Ser., 930, 1, pp. 012001.