Branch And Bound Algorithm Analysis For Solving Job Assignment Problems

Sriyani Violina[1], Helmy Faisal Muttaqin[2], Esa Fauzi[3], Yosi Malatta Madsu[4]

[1]Informatics Department, Engineering Faculty, Widyatama University Jalan Cikutra No 204A Bandung
[2]Informatics Department, Engineering Faculty, Widyatama University Jalan Cikutra No 204A Bandung
[3]Informatics Department, Engineering Faculty, Widyatama University Jalan Cikutra No 204A Bandung
[4]Informatics Department, Engineering Faculty, Widyatama University Jalan Cikutra No 204A Bandung
[1]riyani.violina@widyatama.ac.id

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

Abstract: The Branch & Bound algorithm is a method commonly used to solve optimization problems. Some examples of problems that the Branch & Bound algorithm can solve are Knapsack Problems, Traveling Salesman Problems, Scheduling Problems and many other optimization problems. Job Assignment Problem is one of the optimization problems of how to do n people doing n jobs where one person does one job. This paper will analyze the Branch and Bound Algorithm in solving Job Assignments Problem.

Keywords: Branch & Bound, Job Assignment Problem

1. Introduction

The Branch & Bound algorithm is a method commonly used to solve optimization problems. Many problems can be solved using these two methods. Some examples of problems that the Branch & Bound algorithm can solve are Knapsack Problems, Traveling Salesman Problems, Scheduling Problems and many other optimization problems.

The Branch and Bound algorithm is an algorithm that uses a state space tree to solve a problem, in this case it is similar to the backtracking algorithm. Job

Job Assignment Problem is one of the fundamental combinatorial optimization problems. In its most common form. Examples of problems have a number of people and a number of jobs. Each person can be assigned to do any job, which has different costs depending on the job. The goal is to do as many jobs as possible by assigning one person to each job and one job per person, in such a way that the total cost is minimized.

There are many methods that can be used to solve Job Assignment Problems. This paper will analyze the Branch and Bound Algorithm in solving Job Assignments Problem.

2. Method

2.1 Job Assignment Problem

Job Assignment Problem is a problem regarding the placement of an individual (object) in carrying out a task (job) where exactly one person corresponds to exactly one job according to that person's ability level. Assigning tasks for each individual results in different costs. For example, person P if doing job 1 requires a cost (cost) of c (P, 1), person Q if doing job 2 requires a cost (cost) of c (Q, 2), and so on. So that a job mapping is formed for each person in each job and the costs.

Suppose there are 4 people with 4 jobs, it will produce 3! Or 6 possibilities. From the results of the mapping, there will be the minimum cost. Therefore we need an algorithm that can solve complex problems such as Job Assignment Problems.

2.2 Branch and Bound Algorithm

The Branch and Bound (B&B) algorithm is a method or method of systematically searching the solution space. In this case the solution space will be organized into a status space tree. The formation of a status space tree in this algorithm is built with a broad search scheme or Breadth First Search which is abbreviated as BFS. However, the difference is that the expanded node is not based on the order of generation, but the node that has the smallest “cost” value among other lifecycle nodes.

Each node has a "cost" value, which is the estimated value or the estimated path of the smallest cost from the node to the destination node (goal node), and can be expressed in general terms as follows:

$$\hat{c}(i) = \hat{f}(i) + \hat{g}(i)$$

where

ĉ (i) = value for node i
ƒ (i) = value reaches vertex i from root
ĝ (i) = value reaching the destination node from the vertex.

The way the B&B algorithm works is based on the following two principles.

1. Recursively divide the status space into smaller spaces and minimize "costs" on these spaces. This process is called branching.
2. Branching will be equivalent to brute-force enumeration. To improve performance, bound is used to limit the space in the status that is generated, eliminating candidate solutions that are proven to not contain optimal solutions.

As the name implies, this algorithm has a bound or limiting function. This constraint function is useful for delimiting paths that are considered not leading to a solution node.

2.3 Testing Scenarios

The job assignment problem testing will be carried out in one example of the following cases, namely there are 4 jobs and 4 people, each of which has a cost as in table 1.

Table 1 Job Assignment Problem matrix (4 jobs and 4 people)

	Job 1	**Job 2**	**Job 3**	**Job 4**
Person 1 (A)	**11**	**4**	**9**	**10**
Person 2 (B)	**8**	**6**	**5**	**9**
Person 3 (C)	**7**	**10**	**3**	**10**
Person 4 (D)	**9**	**8**	**11**	**6**

3. Result and Analysis

Solving Job Assignment Problems using Branch and Bound is done by determining the lower limit by adding the minimum cost of each row in table 1. The result can be seen in table 2.

Table 2 Minimum Cost of each Row

	Job 1	**Job 2**	**Job 3**	**Job 4**
Person 1 (A)	**11**	**4**	**9**	**10**
Person 2 (B)	**8**	**6**	**5**	**9**
Person 3 (C)	**7**	**10**	**3**	**10**
Person 4 (D)	**9**	**8**	**11**	**6**

So that Lower Bound (LB) = 4+5+3+6 = 18

There are 4 possibilities, namely A doing Job 1, A doing Job 2, A doing Job 3 and A doing Job 4. Calculate the LB of each of these possibilities.

Possibility 1: Person 1 (A) does Job 1

Table 3 Person 1 (A) does Job 1

	Job 1	**Job 2**	**Job 3**	**Job 4**
Person 1 (A)	**11**	**4**	**9**	**10**
Person 2 (B)	~~**8**~~	**6**	**5**	**9**
Person 3 (C)	~~**7**~~	**10**	**3**	**10**
Person 4 (D)	~~**9**~~	**8**	**11**	**6**

Then LB for probability 1 is : 11+5+3+6 = 25

Possibility 2: Person 1 (A) does Job 2

Table 4 Person 1 (A) does Job 2

	Job 1	**Job 2**	**Job 3**	**Job 4**
Person 1 (A)	**11**	**4**	**9**	**10**
Person 2 (B)	**8**	**~~6~~**	**5**	**9**
Person 3 (C)	**7**	**~~10~~**	**3**	**10**
Person 4 (D)	**9**	**~~8~~**	**11**	**6**

Then LB for probability 2 is : 4+5+3+6 = 18

Possibility 3: Person 1 (A) does Job 3

Table 5 Person 1 (A) does Job 3

	Job 1	**Job 2**	**Job 3**	**Job 4**
Person 1 (A)	**11**	**4**	**9**	**10**
Person 2 (B)	**8**	**6**	**~~5~~**	**9**
Person 3 (C)	**7**	**10**	**~~3~~**	**10**
Person 4 (D)	**9**	**8**	**~~11~~**	**6**

Then LB for probability 3 is : 9+6+7+6 = 28

Possibility 4: Person 1 (A) does Job 4

Table 6 Person 1 (A) does Job 4

	Job 1	**Job 2**	**Job 3**	**Job 4**
Person 1 (A)	**11**	**4**	**9**	**10**
Person 2 (B)	**8**	**6**	**5**	**~~9~~**
Person 3 (C)	**7**	**10**	**3**	**~~10~~**
Person 4 (D)	**9**	**8**	**11**	**~~6~~**

Then LB for probability 4 is : 10+5+3+8 = 26

Then formed a tree like figure 1

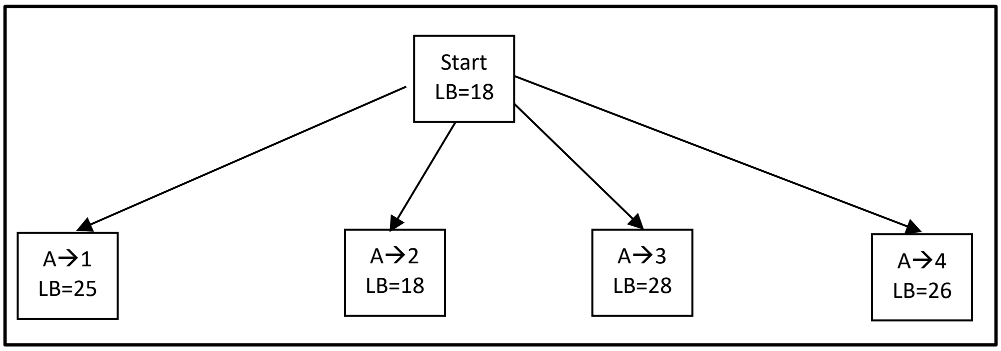


Fig 1 level 1 Job Assignment Problem solution using BnB

From Figure 2, the node with the minimum value to be expanded is selected, namely A◊2 with LB = 18. Then the second person (B) is chosen to do the job or assignment. There are 3 possibilities, namely B doing job 1, Job 3 or Job 4 (Job 2 is done by A)

Possibility 1: Person 2 (B) does Job 1

Table 6 Person 2 (B) does Job 1

	Job 1	**Job 2**	**Job 3**	**Job 4**
Person 1 (A)	**~~11~~**	**4**	**9**	**10**
Person 2 (B)	**8**	**~~6~~**	**5**	**9**
Person 3 (C)	**~~7~~**	**~~10~~**	**3**	**10**

Person 4 (D)	**~~9~~**	**~~8~~**	**11**	**6**

Then LB for probability 1 is 4+8+3+6 = 21

Possibility 2: Person 2 (B) does Job 3

Table 7 Person 2 (B) does Job 3

	Job 1	**Job 2**	**Job 3**	**Job 4**
Person 1 (A)	**11**	**4**	**~~9~~**	**10**
Person 2 (B)	**8**	**~~6~~**	**5**	**9**
Person 3 (C)	**7**	**~~10~~**	**~~3~~**	**10**
Person 4 (D)	**9**	**~~8~~**	**~~11~~**	**6**

Then LB for probability 2 is: 4+5+7+6 = 22

Possibility 3: Person 2 (B) does Job 4

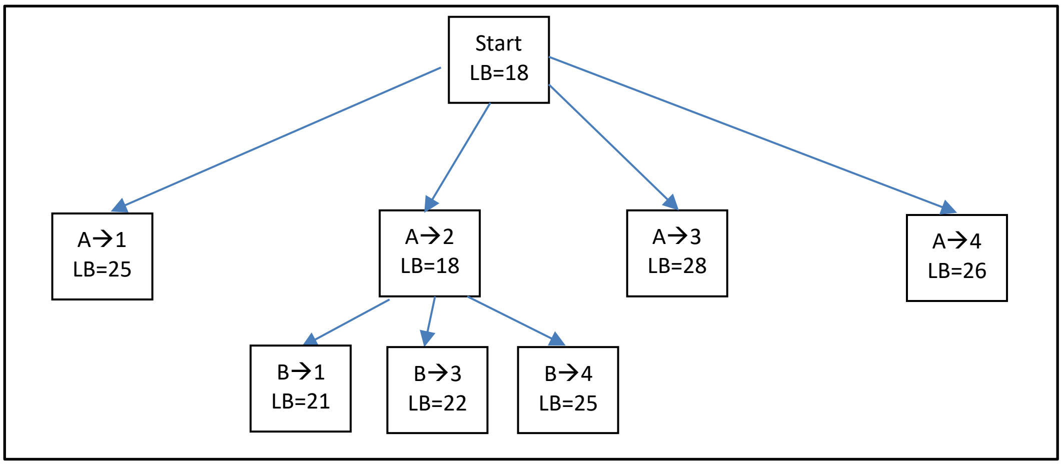
Table 7 Person 2 (B) does Job 4

	Job 1	**Job 2**	**Job 3**	**Job 4**
Person 1 (A)	**11**	**4**	**9**	**~~10~~**
Person 2 (B)	**8**	**~~6~~**	**5**	**9**
Person 3 (C)	**7**	**~~10~~**	**3**	**~~10~~**
Person 4 (D)	**9**	**~~8~~**	**11**	**~~6~~**

Then LB for probability 1 is 4+9+3+9 = 25

Then formed a tree like figure 2

Fig 2 level 2 Job Assignment Problem solution using BnB



Next, select the node with the minimum cost to expand, namely B→1 and do the calculations as before, for the next 2 possibilities.

Possibility 1: A does Job 2, B does Job 1, C does Job 3 and D does Job 4 with the cost is 4 + 8 + 3 + 6 = 21.

Possibility 2: A does Job 2, B does Job 1, C does Job 4 and D does Job 3 with the cost is 4 + 8 + 10 + 11 = 33.

Then the possibility of 1 is chosen with cost = 21, compared to all the remaining nodes, this cost is the minimum cost so that it is chosen as the solution.

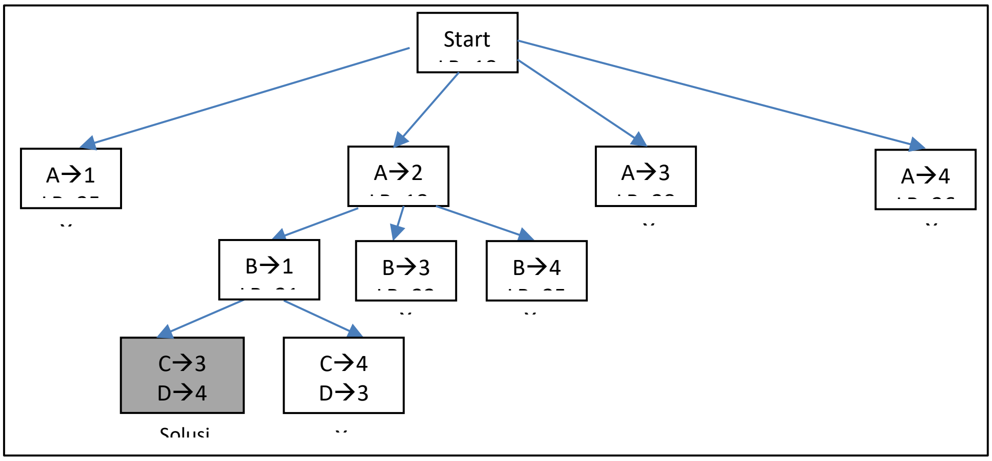


Fig 3 Job Assignment Problem solution using BnB

4. Conclusion

The use of branch and bound results in a shorter solution because the branch and bound algorithm performs calculations recursively, while still calculating the best value, that is what is known as branching. In addition, the best value is recorded for each calculation, so that it can improve the performance of the algorithm, what is known as bounding.

5. References

1. Neapolitan, Richard, “Foundations of Algorithms”, Jones & Bartlett Publishers, 2014
2. Steven S. Skiena, The Algorithm Design Manual, Springer, 1997
3. Wang, W., Tian, G., Zhang, T., Jabarullah, N. H., Li, F., Fathollahi-Fard, A. M., ... & Li, Z. (2021). Scheme selection of design for disassembly (DFD) based on sustainability: A novel hybrid of interval 2-tuple linguistic intuitionistic fuzzy numbers and regret theory. *Journal of Cleaner Production*, 281, 124724.