Designing University Transportation Route with Branch and Bound Algorithm

Nikolas Wangsaputra / 13514048
Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514048@stei.std.itb.ac.id

***Abstract* — Efficient transportation inside a campus area would increase student's productivity by reducing travel expense and time. Unfortunately there are some universities which haven't provided sufficient transportation inside the campus area. To actually design a good transportation system, a decent route planning is essential. Using travelling salesman approach and branch and bound algorithm, one can design an efficient route time-wise or distance-wise. The result is guaranteed to be the best/optimum in a certain parameter although might not be the best applicable solution. However, this method is viable to implement since it is easy to modify the parameter required, thus making it more applicable.**

Keywords — optimization; transportation; Travelling Salesman; Path Finding; Branch and Bound;

I. Introduction

A. Path Finding Problem

In some aspect of science and technology, scientists sometimes face certain problem where they have to make a choice of path based on certain parameters. These parameters often include the expense of a set of solution or profit made from the solution. Usually, when the decision choice is insignificant, people can easily find a best path choice with bare hands quickly. But more often than not, those problems became quite complicated and repetitive that solving those manually would quickly become a tedious work, even with a help from a graph illustration.

Certain sectors of science usually have a different approach to these kinds of problem. For example, industrial technology researcher would often use the linear optimization on solving decision making problem. This approach is usually coupled with the Simplex method to solve huge problems. On the other hand, those studying management or planning would rather solve these problems with hands rather than using a specific algorithm, since their decision-making process often involves many indefinite factors that should be considered carefully while doesn't really care about solving speed. Some sectors of science also use manual approach since their decision choice is not substantial.

Information technology scientist or computer scientist would often face the path finding problem in some cases, such as finding path in video games, maps, etc. The difference from most other scientific approach to these kinds of problem, however, lies on how the speed of solving a certain problem becomes the most important matter of thought. Developers often took a long time deciding which algorithm to use and implement, just to save milliseconds of processing time. This explains why Dijkstra's algorithm is still widely used and researched as a part of artificial intelligence sector.

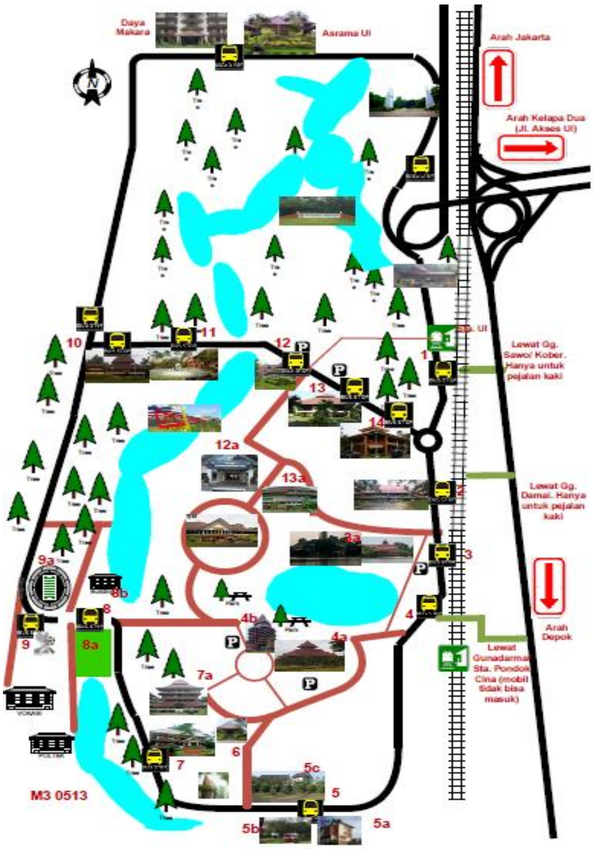
Lately, there are some more algorithm choices available to solve those path finding problems, such as branch and bound, which will be discussed later in the paper.

B. Mass Transportation Designing

Transportation is one of the most vital issues in urban planning. It is known to take a big role in an area's productivity since it connects every other element (health care, environment, and economics). A good transportation system easily increases productivity of an area, be it a whole country or even just a small district, while a poorly designed transportation system would lead to various problems such as traffic jams or pollution.

In most large scale university around the world, there are usually transportation methods to travel around the buildings. In University of Indonesia, for example, the students travel around with buses called *Bikun* (A.K.A. Yellow bus). This immensely reduced the cost and time needed to travel, effectively increasing productivity of the students.

Picture 1: Transportation in University of Indonesia
Source: http://www.anakui.com/[5]



Picture 2: University of Indonesia's bus stops
Source: http://rumah-metri.blogspot.com/[6]

However, there are some new university areas that haven't had the local transportation means yet, such as ITB Jatinangor. The reason is that the area is still being developed such that the transportation system was neglected. As a result, the students would usually use their own personal transportation such as car or motorcycle. While the students have got used to this transportation system, this system will cause a problem later on, such as parking or traffic jam. Therefore, it is beneficial to design a good mass transportation system to accommodate the students, so that transportation issue would not hinder a long term development to the infrastructure of the campus building.

This paper will attempt to demonstrate a usage of branch and bound algorithm to take part in solving transportation problem, specifically in university. Hopefully this research will open up more possibilities and be insightful to students and planner alike.

II. Background Concepts and Theories

A. Graph

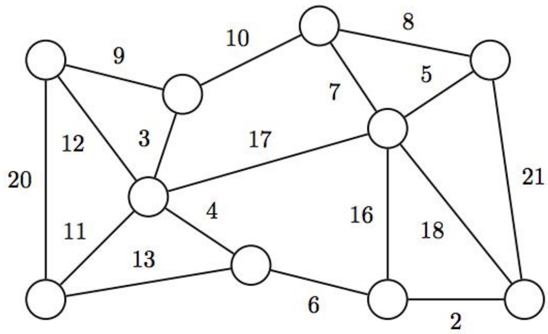
Graph is a structure consists of vertices (singular: vertex, also often termed nodes) and edges[1] of which each of them connects 2 vertices (although it is possible for an edge to connect a vertex to itself). It is usually defined as set (V, E) where:

- V = a nonempty set of vertices {v1, v2, v3, … , vn}
- E = a set of edges that connect a pair of vertices {e1, e2, e3, … , en}. Each edge has either one or two vertices associated with it, called endpoints.

There exist many variations of graph, with each of them is able to represent certain type of data/condition.

1) Weighted Graph

A weighted graph is a variation of graph where every edge is assigned a number which is called weight. This number may represent distance, cost, or other measurable value.



Picture 3: Example of a weighted graph
Source: cs.brynmawr.edu[4]

2) Directed Graph

A directed graph is a variation of graph where every edge has a specific direction assigned to it. In application, most graphs usually are directed and weighted graph, as this kind of graph contains most information needed.

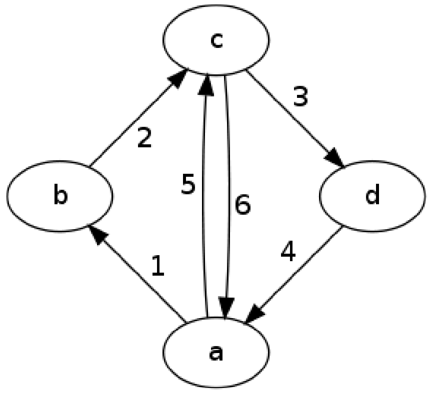


Figure 4: Example of directed, weighted graph. Source: stackoverflow.com [3]

B. Travelling Salesman Problem

In general, the travelling salesman problem (TSP) is a problem finding the best way to visit all vertices in a single graph from a vertex once and then return to the origin vertex. The best way itself can be measured by various parameter, be it the number of vertices (usually in a non-weighted graph), distance, cost, time, etc.

The travelling salesman problem is usually modeled with a directed graph, and solved using various algorithms, such as brute force, greedy, or branch and bound.

Outside of Information Technology, there are different approaches toward this problem. For instances, industrial engineering's operational research sometimes uses linear optimization because it is more adaptable to their other related problems. [2]

There are a variation of TSP problem called set TSP problem. Set TSP Problem is a variation (or more precisely, generalization) of the Travelling Salesman problem, in which every vertex must be visited at least once, as opposed to the ordinary TSP in which every vertex must be visited only once.

C. Branch and Bound Algorithm

Solving a TSP quickly and accurately usually requires a problem solving algorithm. One of the most common algorithm to solve a TSP problem is the branch and bound algorithm.

Branch and Bound (often shortened to BnB) is a systematic method for solving optimization problems. It is introduced by A.H. Land and A.G. Doig in 1960. Branch and bound algorithm is optimal, as it searches the whole space of solutions for a given problem to find the best solution. Furthermore, it is a rather general optimization technique that applies in most cases, even where the greedy method and dynamic programming fail.

However, despite the broad usage possibility, branch and bound algorithm is generally slower than greedy or dynamic programming. It is indeed faster than brute force algorithm, but requires a careful implementation to actually run fast on average.

The general idea of branch and bound is a search algorithm similar to BFS for the optimal solution, but without expanding all of the nodes (i.e., their children generated). Rather, a carefully selected criterion determines which node to expand and when to expand it, and another criterion tells the algorithm when an optimal solution has been found. [8]

There are two main steps in branch and bound algorithm, which are the "branch" part and the "bound part". The branch part is where the tree got expanded recursively, where bound is the part where the algorithm searches for the lower/upper bound and trim all nodes which are no longer viable as an answer. The bound itself is different for each problem, where unique problems may have their own heuristic to determine that certain bound. Even one problem will have some equally valid way to determine the boundary. The heuristic itself is usually written as $\hat{c}(i) = \hat{f}(i) + \hat{g}(i)$ where $\hat{c}(i)$ is a cost for a certain vertex i, $\hat{f}(i)$ is the cost to reach vertex i from the origin, and $\hat{g}(i)$ is the cost to reach the target vertex from vertex i.

The general steps in using branch and bound is:

1) Choose an origin vertex, and determine a bound value using heuristic if needed.

2) Recursively expand the graph while continously storing the bound value.

3) When a solution is found, kill all vertices which cost is already higher than the result (thus impossible to become the solution)

4) Continue from the least cost vertex recursively until there are no more possible expansion.

III. Problem Identification

A. Case Study : Jatinangor ITB

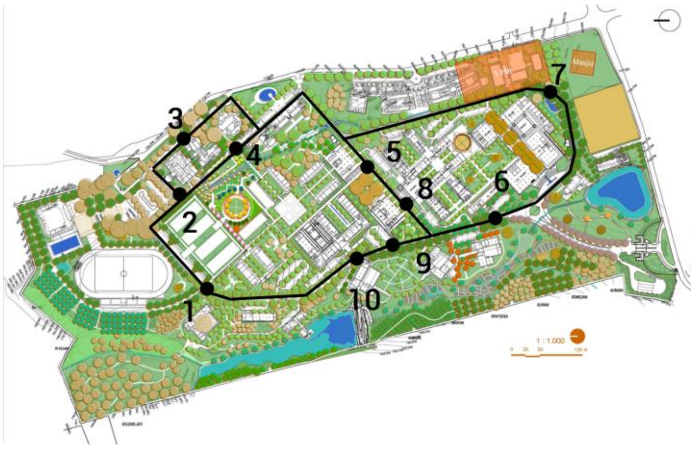
Picture 5: The map of Jatinangor ITB Campus Area Source: jatinangor.itb.ac.id [7]

Jatinangor is one of ITB's campus areas, the other one being in Ganeca. It is located quite far away from the downtown, and usually remarked as underdeveloped. According to some students who resides there, there are no local mass transportation in the area and the university still hasn't provide a sufficient transportation until now, while the area itself is big enough so that the students are forced to bring their own transportation. This may bring some problem later on, such as pollution and lack of parking space.

In this paper, the author is using Jatinangor ITB's campus area as a study case to design an inner transportation route.

B. Converting the map into a graph

After inspecting the map, which includes researching paths that are suitable for buses and the choke points of the campus area, an illustration was able to be made. The map below illustrates feasible roads which are accessible by buses, as well as some nodes which represents the possible bus stops, considering the crowd level and the traffic rate of that certain area.



Picture 6: The master plan of Jatinangor ITB campus area with bus routes and stops illustration
Source: jatinangor.itb.ac.id [7] with addition

The nodes in picture 6 indicate the crowded area which is feasible as a bus stop. According to one of the student, node number 1 is generally the busiest as it is a dormitory area where most of the students resides and visit. This will reflect later on as we input the graph to the algorithm where 1 would be the point of origin, although the result will most likely be the same.

Now that the route and stops are explicitly drawn, a weighted graph that represents the whole route can be depicted.

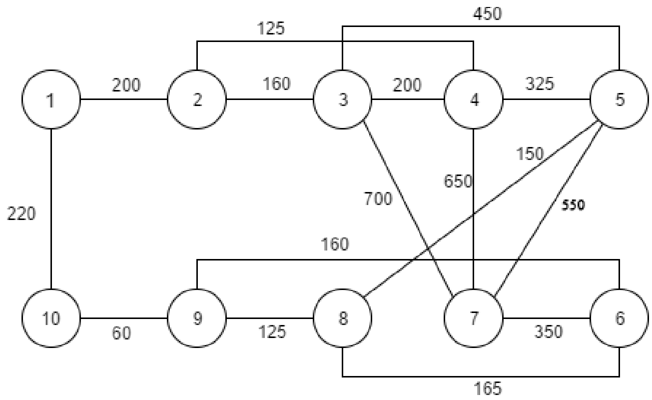


Figure 7: Weighted graph from the route in picture 6
Source: author

Note that in this illustration, every edge is bidirectional. This represents the real life situation where all roads in this campus area are two-way road.

Also, in this trial, the weight of each edge is set to be the distance between the two respective nodes. The distance of each edge is measured by meters and not precise, that is, rounded to the nearest 10. For convenience, other factors such as traffic, road quality and size, and other traffic hindrance were excluded from measurement.

Now that the graph is visualized in a simpler form, it can now be easily converted into an adjacency matrix. This matrix can then be used in the algorithm without hassle.

∞	200	∞	∞	∞	∞	∞	∞	∞	220
200	∞	160	125	∞	∞	∞	∞	∞	∞
∞	160	∞	200	450	∞	700	∞	∞	∞
∞	125	200	∞	325	∞	650	∞	∞	∞
∞	∞	∞	∞	∞	∞	550	150	∞	∞
∞	∞	∞	∞	∞	∞	350	165	160	∞
∞	∞	700	650	550	350	∞	∞	∞	∞
∞	∞	∞	∞	150	165	∞	∞	∞	∞
∞	∞	∞	∞	∞	160	∞	125	∞	60
220	∞	∞	∞	∞	∞	∞	∞	60	∞

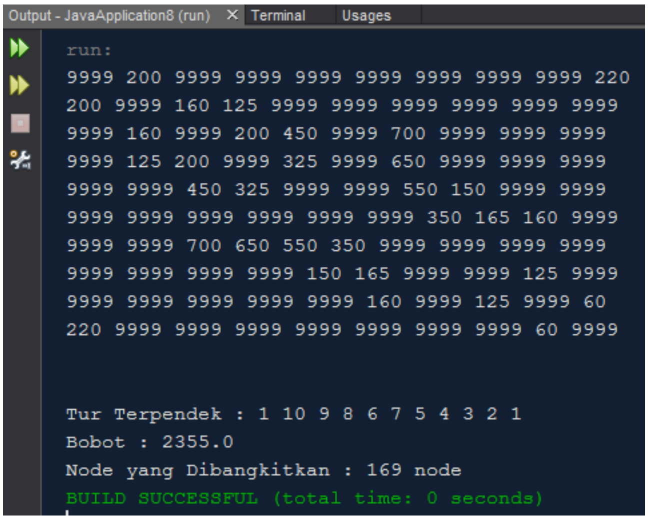
Figure 8: Adjacency matrix from the route in picture 6
Source: Author

IV. PROCESSING WITH BRANCH AND BOUND ALGORITHM

A. Using branch and bound Algorithm

In this trial, the matrix will be solved using branch and bound algorithm which is exactly the same as in the theory. Since the size of matrix is big, a branch and bound simulator (tucil 3 IF2211) is used as a solution finder.

This application will run a basic branch and bound algorithm as written in section 2 with the complete graph tour approach. User inputs the matrix, and then the application will write out a path which represents the shortest path tour available.



Picture 9: Result of the branch and bound algorithm applied to adjacency matrix from figure 8.
Source: Author

According to the application, the shortest tour around the campus area is 2355 meters through node:

1- 10 - 9 -8 - 6 - 7- 5 -4 -3 -2 -1

B. Result Analysis

Based on the solution generated by the application, the shortest route is simply going around the ring, which make a lot of sense for this kind of map. The solution is optimal; however, there are some considerations to this solution:

1) Technically, the algorithm ignores the possibility of going through the same vertex twice to get an optimum solution. Although this doesn't seem to happen very often, it may become a single cause of inaccuracy.

2) The result generated only considers the range of each nodes. In real life situation, it is probably better to consider the other aspects such as time taken, traffic condition, etc. By including those parameters, it is not impossible to design a system that not only find the shortest time taken to travel, but also the accurate prediction of the arrival of the bus and the duration of travel.

3) There might be an adaptation to find two or more ideal route so that there is a multiple bus route operating in the campus area.

The first consideration (visiting a vertex more than once) can actually be solved by augmenting the matrix. Suppose that it is more ideal to travel through <u>A->B->C</u>->B->A rather than directly from <u>A -> C</u> -> B -> A, but B have already been visited. We can actually add another edge to all of the vertices that contains the shortest path between the two vertices. When we run the algorithm, it will show the result like A->C->B->A. It is already known that the shortest path between A->C is actually A->B->C, so then the algorithm will replace those edge.

This method will probably work best in a complicated map like in the downtown rather than in a university complex, because the difference in result will be more significant. The simplicity of the map made the algorithm somewhat less relevant.

V. CONCLUSION

Branch and bound algorithm is able to find the best transportation route based on the heuristic it is assigned with, thanks to its optimality. However, since transportation issue is dynamic and contains a lot of consideration beside a quantitative variable, the solution generated by this algorithm is not suitable to be directly implemented without further consolidation.

Despite its weakness, this method of finding shortest route may potentially be applicable on some cases. The ease of changing the heuristic used and the adaptability of the algorithm will definitely shorten the time required to mull over the best route and designing a better transportation system overall.

ACKNOWLEDGMENT

The Author would like to express his deep gratitude and appreciation to Drs. Nur Ulfa Maulidevi and Dr. Ir. Rinaldi Munir, M.T. as the lecturer of IF2211 Strategi Algoritma for giving this paper task to the students of IF2211, as well as the knowledge and support given until the finishing of this paper. A special appreciation is given to Dwiva Anbiya Taruna and Olivia Putri Purnama, students from TPSDA (Water Resource Engineering) ITB, who has supported the author by providing information about the transportation condition in Jatinangor ITB. He would also express his appreciation to his parents who have given the material and moral support for this project, as well as all friends in IF2211 who have given moral supports needed to finish this project.

REFERENCES

[1] Munir, Rinaldi. *Matematika Diskrit*. Bandung: Informatika Bandung, 2005.

[2] Liebermann. Hillier. Introduction to Operational Researches. Stanford University, 2001

[3] http://stackoverflow.com/

[4] http://cs.brynmawr.edu/Courses/cs380/spring2013/section02/assignments/assignment3.html

[5] http://www.anakui.com/

[6] http://rumah-metri.blogspot.com/

[7] http://jatinangor.itb.ac.id/

[8] https://www.seas.gwu.edu/~ayoussef/cs212/branchandbound.html

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 07 Mei 2016

Nikolas Wangsaputra / 13514048