UAV Waypoint Strategy for COVID-19 Medicine Delivery based on Cheapest Link and Hamilton Circuit Algorithm

Leonard Matheus Wastupranata
School of Electrical Engineering & Informatics
Bandung Institute of Technology
leo.matt.547@gmail.com

***Abstract*— The COVID-19 pandemic is an unavoidable disaster. Many COVID-19 sufferers must self-isolate at their respective residences. However, difficult access has hampered the distribution of medicines from nearby hospitals. This paper aims to find the shortest route for UAVs so that they can deliver drugs in the shortest possible time and the minimum distance. The method used is the application of the Hamilton Circuit with the Cheapest Link Algorithm approach so that the distance between houses can be reached without being neglected at all. Flying construction was carried out using Software in The Loop (SITL) and ArduPilot Mission Planner. The results obtained are that routes created using the Cheapest Link Algorithm have an average efficiency of 66.86% better than other Hamilton circuits formed on the same graph.**

***Keywords*— UAV, Cheapest Link, Graph, Hamilton Circuit**

I. INTRODUCTION

The year of 2020 will be a unique and unforgettable year in world history. The COVID-19 pandemic has infected tens of millions of world citizens and caused many fatalities. Various logistical needs and medicines must be distributed to COVID-19 sufferers as quickly as possible. Self-isolation must be carried out so that the chain of distribution of COVID-19 can be quickly broken.

The problem that arises is the large number of patients who carry out independent isolation due to the capacity of the hospital ward which is full or problems with limited access. Drug delivery media are needed for COVID-19 patients who are self-isolating at their respective homes. The media used is UAV (Unmanned Aerial Vehicle) which can be called drones.

The method of determining drone flight routes has a variety of alternative algorithms such as the Cheapest Link Algorithm, Dijkstra Algorithm, Prime Algorithm, and others. Suwatno[1] has conducted research on how to determine the fastest flight route by using Dijkstra's Algorithm as input for the route graph that will be created. Unfortunately, the existing methods are not optimal for managing the effective and fast delivery of drugs. In addition, this algorithm has not been specified to make visits to all points, so it has not been able to reach independent isolation houses for sufferers of COVID-19.

For this reason, this paper aims to apply the theory of finding the shortest route using the Cheapest Link Algorithm to adjust drug delivery routes to increase the efficiency of flight routes and avoid delays in treating COVID-19 patients. In addition, the application of graph theory is expected to have a major impact so that the distribution of medical logistics assistance is able to reach residents' homes that are difficult to reach and have limited access.

This paper is divided into five major sections, which are Introduction, followed by a Literature Study that presents the theoretical basis and formulas. Then the Method will be presented along with the Results & Discussion of an effective flight route algorithm. Finally, Conclusions and Future Works will be conducted for further research.

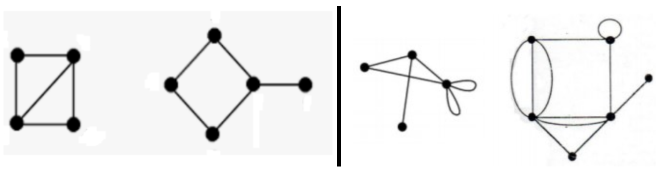
II. LITERATURE STUDY

A. Graph Theory

A.1. Graph Definition

Graphs are used to represent discrete objects and the relationships between them. Graph G = (V, E), which in this case, V is = the non-empty set of nodes $\{v_1,v_2,...,v_n\}$, while E is the set of sides connecting a pair of vertices $\{e_1,e_2,...,e_n\}$.

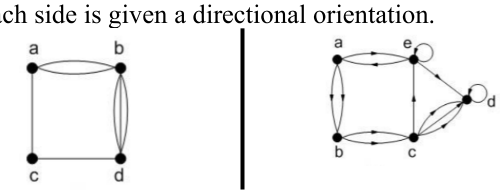
Based on the presence or absence of bracelets or double sides on a graph, graphs are classified into two types, namely Simple Graphs and non-simple Graphs. A Simple Graph is a graph that contains neither bracelets nor double sides, while a non-simple graph is a graph that has double sides or bracelets.



a) b)

Figure 1. Illustration of Simple Graph (a) and Directed Graph (b) [2]

Based on the directional orientation on the side of a graph, graphs are classified into two types, namely undirected graphs and directed graphs. An undirected graph is a graph whose sides have no directional orientation, while a directed graph is a graph whose each side is given a directional orientation.



a) b)

Figure 2. Undirected Graph (a) and Directed graph (b) [2]

A.2. Graph Terminology

There is some graph terminology underlying this paper. First, two knots are said to be neighbors when they are directly connected. In addition, $e(v_j,v_k)$ it is said that e is side by side with the node v_j or e is side by side with the node v_k.

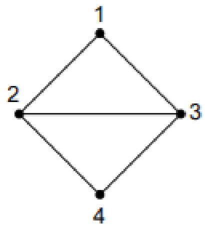


Figure 3. Examples of Neighborly and Side by Side Terminology [2]

In **Figure 3**, consider the node 1. Node 1 is neighboring with nodes 2 and 3 but is not neighboring with node 4. In addition, the sides (2,3) are side by side with node 2 and node 3, but the sides (1,2) are not side by side with node 4.

The next terminology describes the existence of a node that has no degree or is not side by side with other nodes. It is in the form of a Graph with a secluded node and an empty graph. A graph with a remote node means that the graph has a node that is not side by side with other nodes. Furthermore, a Blank Graph is a graph whose set of sides is an empty set (N_n).

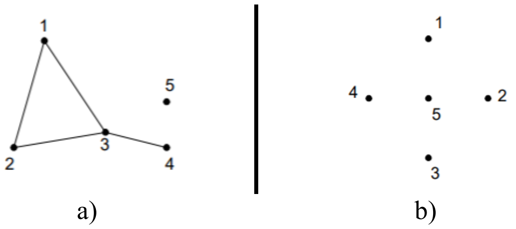


Figure 4. Illustration Remote node on node 5 (a) and Blank Graph (b) [2]

Other terminology that is important to discuss is regarding Trajectories and circuits. The trajectory whose length is n from the initial node v 0 to the destination node v n in the graph G is an alternating row of vertices and sides that are in the form of $v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n$ such that $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2), \ldots, e_n = (v_{n-1}, v_n)$ are the sides of the graph G. In **Figure 4**, trajectory 1,2,3,4 is the side-aligned trajectory (1,2),(2,3),and (3,4). The length of the track is worth 3 because there are 3 side slices.

Paths that start and end at the same node are called circuits or cycles. In **Figure 4**, the 1,2,3,1 track forms a circuit and has a Length of 3. The last Graph terminology that is also fairly important is the weighted graph. A weighted graph is a graph to which each side is assigned a price (weight). [3]

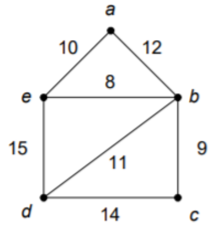


Figure 5. Weighted Graph Illustration [2]

B. Hamilton's Circuit Theory

B.1. Hamilton Circuit Definition

Hamilton's trajectory is one that goes through each node in the graph exactly once. The Hamilton circuit is a circuit that goes through each node in the graph exactly once, except for the origin node (as well as the final node) which is passed twice. Graphs that have Hamilton circuits are called Hamilton graphs, while graphs that only have Hamilton tracks are called semi-Hamilton graphs.

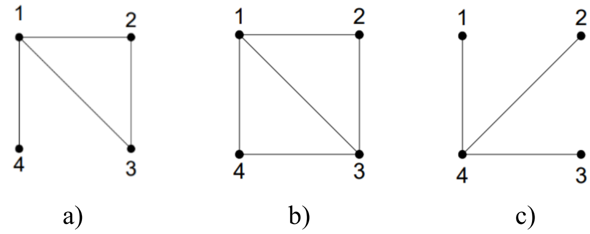


Figure 6. Hamilton Circuit Image Example [4]

Shown in **Figure 6**, for figure (a), the Graph has a Hamilton trajectory on a 3-2-1-4 path. While in the figure of section (b), the graph has a Hamilton circuit seen on the 1-2-3-4-1 line. In figure (c), however, the graph will never form Hamilton's track not even the Hamilton circuit.

B.2. Hamilton Circuit Theorem

The sufficient requirement for a simple graph G with n (≥ 3) vertices to be a Hamilton graph is if the degree of each node is at least n/2 (i.e., $d(v) \geq \frac{n}{2}$ for each node v in G). In **Figure 6**, it appears that on graph (a), $n = 4$, but vertices 4 have $d(v) = 1$, so it says it is not Hamilton's graph because it only has Hamilton's trajectory. On graph (b), n = 4, each node has $d(v) \geq 2$, so it can be said to be a Hamilton graph because it contains a Hamilton circuit in it.

With the above proof, it can be said that every complete graph is Hamilton's graph. In addition, the other Theorem is in the complete graph G with n vertices (n ≥ 3), there are $\frac{(n-1)!}{2}$ pieces of Hamilton circuits. In addition, in the complete graph G with n knots (n ≥ 3 and n is odd), there are $\frac{(n-1)}{2}$ pieces of Hamilton circuits that are mutually detached (no sides intersect). If n is even and n 4, then in G there are $\frac{(n-2)}{2}$ hamilton circuits that are loose from each other.

C. Travelling Salesperson Problem (TSP)

C.1. Travelling Salesman Problem (TSP) Definition

Travelling Salesman Problem (TSP) is a common problem in combinatorial optimization where a salesman must visit several N cities, where each city is only visited once. This salesman had to choose a route so that the total distance he traveled was minimum[5].

The rules identify that the problem is that the journey begins and ends in the same city as the city of origin. Furthermore, the whole city must be visited without a single city being missed. Finally, Sales must not return to their hometown before the entire city is visited. [6]

C.2. Travelling Salesman Problem (TSP) Math Models

Mathematically, the TSP formulation can be expressed as a graph G=(V,A) with V = {1,2,3, . . . , n} represents the set of graph points indicating the location of the point to be visited and A = {(i, j)| i, j ∈ V, i ≠ j} are sets of sides expressing the link between points. Point 1 represents the place of origin. Suppose $x_{i,j}$ is the mileage from city i to city j and is defined as a changer:

$$x_{i,j} = \begin{cases} 1, if\ side\ (i,j) \in A\ is\ traversed\ by\ route \\ 0, if\ side\ (i,j) \in A\ is\ nottraversed\ by\ route \end{cases}$$

The mathematical formulation of the TSP is as follows:

$$\text{Minimize } z = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{i,j}$$

This equation is a function aimed at minimizing the total mileage. With the following constraints:

$$\sum_{i=1}^{n} x_{i,j} = 1, i = 1,2, \dots, n$$

$$\sum_{j=1}^{n} x_{i,j} = 1, j = 1,2, \dots, n$$

The above constraints aim to show that the UAV came and left each point exactly once. While

$$\sum_{i \in Q} \sum_{j \in Q} x_{i,j} \geq 1, \forall Q \subset V, Q \neq \phi$$

is a constraint to ensure that there is no sub route, where $Q = \{1,2,3, \dots, n\}$. And the following last constraint is used to guarantee that x i,j is a binary changer i.e. $x_{i,j} \in \{0.1\}$, where $i, j = 1, \dots, n$.[7].

D. Cheapest Link Algorithm Theory

D.1. Definition of the Cheapest Link Algorithm

In Hamilton's circuit, of course, the delivery weight is attempted to be the shortest. Of course, the Brute Force Algorithm can be used to solve this problem. However, in his search, there are $\frac{(n-1)!}{2}$ Hamilton circuits that are loose at n even so this can lead to ineffective workloads. For this reason, it is necessary to implement an algorithm that has good performance so that it does not linger in determining the route.

There is an efficient algorithm called the Cheapest Link Algorithm. The idea of this Algorithm is to structure the tour by connecting separate lines (i.e., sides) of the tour based on the weight of that side. This algorithm does not care about the separately connected nodes at first. However, in the end, if the drafting is done well, the shortest Hamilton circuit will be formed. [8]

D.2. How the Cheapest Link Algorithm Works

According to Gregory J. Pottie[9]there are three steps in determining the route using the Cheapest Link Algorithm, namely:

1. Choose the side with the smallest weight, Mark the side.
2. Select the side with the next smallest weight, provided that:
   a. The side does not form a new circuit.
   b. The Side result after being marked on a node is not a third node. Or in other words, a node must not have more than two marked sides.
3. Repeat Step 2 until a Hamilton circuit is formed (All nodes have been visited and marked).

III. Method

A. Preparation of Route Planner (Mission Planner)

Figure 7. Sample conditions of self-isolation houses that have limited access.

To create a flight route for a drug delivery *drone* , the drug delivery area needs to be defined. HOME is defined as starting from Hasan Sadikin Hospital Bandung (Pins 1 & 14). The location of Hasan Sadikin Bandung is at a Latitude of -6.896413 degrees with a Longitude figure of 107.59 degrees. Flight simulation will be done using Software in the Loop (SITL) with ArduPilot Library and Mission Planner version 1.3.74.

The ride must start the mission from HOME and return to land on HOME when the mission is over. The green pin is where the ride must place the shipment in the form of medicines. Drones must also be able to deliver goods to all predetermined green points quickly and efficiently. All designated points must be visited in any way. Of the conditions that must be met, the graph must be formed first, then determine the waypoint to be used using calculations.

B. Weighted Graph Formation

Figure 8. The condition of the sample house after the graph is completed.

The existing weight is calculated using the GPS Map Tools feature on the ArduPilot. The size in this paper has been adjusted to the scale of the existing map. The final graph formed is a complete graph with consistent degrees at each vertex, which amounts to 12 degrees. A complete graph is the main choice because a complete graph is one of the conditions for the formation of the Hamilton circuit.

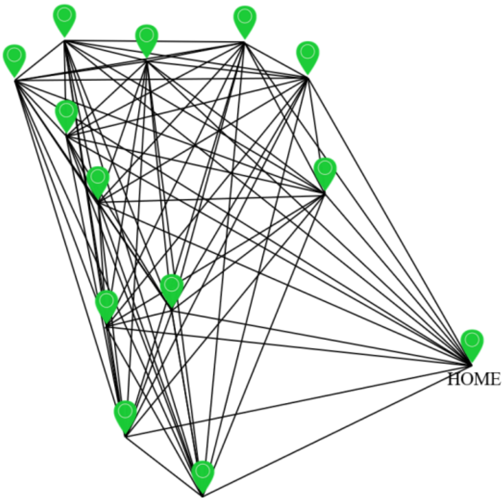


Figure 9. Final weighted graph

As a side note, the 14th node in this paper (for landing nodes) has the same place as HOME, so the node is adequately represented as HOME. The following is presented a table of distances from waypoints to each other in meters.

Table 1. List of Distances between waypoints (WP) representing weighted graphs in units of meters

Side (Node to Node)	Distance (meters)	Side (Node to Node)	Distance (meters)	Side (Node to Node)	Distance (meters)
HOME – 2	217,00	3 – 7	215,09	6 – 9	115,89
HOME – 3	247,43	3 – 8	259,10	6 – 10	103,71
HOME – 4	258,50	3 – 9	279,92	6 – 11	152,82
HOME – 5	215,81	3 – 10	263,55	6 – 12	171,04
HOME – 6	289,18	3 – 11	285,35	6 – 13	158,90
HOME – 7	331,66	3 – 12	284,79	7 – 8	52,40
HOME – 8	382,85	3 – 13	218,32	7 – 9	66,41
HOME – 9	369,99	4 – 5	48,80	7 – 10	73,95
HOME – 10	317,89	4 – 6	84,32	7 – 11	142,12
HOME – 11	278,98	4 – 7	139,54	7 – 12	174,01
HOME – 12	235,17	4 – 8	185,84	7 – 13	188,41
HOME – 13	169,00	4 – 9	205,91	8 – 9	45,80
2 – 3	70,6	4 – 10	192,01	8 – 10	97,67
2 – 4	135,02	4 – 11	221,13	8 – 11	166,86
2 – 5	136,49	4 – 12	228,89	8 – 12	207,51
2 – 6	226,21	4 – 13	180,37	8 – 13	236,34
2 – 7	279,49	5 – 6	93,30	9 – 10	61,00
2 – 8	323,14	5 – 7	145,60	9 – 11	126,89
2 – 9	337,54	5 – 8	197,38	9 – 12	173,09
2 – 10	311,93	5 – 9	205,87	9 – 13	213,72
2 – 11	321,40	5 – 10	175,33	10 – 11	69,00
2 – 12	309,43	5 – 11	192,18	10 – 12	110,78
2 – 13	231,63	5 – 12	190,16	10 – 13	160,56
3 – 4	77,30	5 – 13	215,56	11 – 12	49,80
3 – 5	93,00	6 – 7	51,60	11 – 13	117,74
3 – 6	166,19	6 – 8	102,69	12 – 13	83,70

C. Table Filling with Cheapest Link Algorithm

The reason why the Cheapest Link algorithm was chosen is because in its application, the specified point is mandatory to visit and should not ignore a single point. The new flight route will be created using the Cheapest Link algorithm according to the shortest iteration of the path. The iteration table filling will be shown below.

Table 2. Route iteration table with Cheapest Link Algorithm

2	217											
3	247	71										
4	259	135	77									
5	216	136	93	49								
6	289	226	166	84	93							
7	332	279	215	140	146	52						
8	383	323	259	186	197	103	52					
9	370	338	280	206	206	116	66	46				
10	318	312	264	192	175	104	74	98	61			
11	279	321	285	221	192	153	142	167	127	69		
12	235	309	285	229	190	171	174	208	173	111	50	
13	169	232	218	180	216	159	188	236	214	161	118	84
WP	1/14	2	3	4	5	6	7	8	9	10	11	12

From Table 2, each of the maximum nodes has only 2 correspondence sides. This is indicated by the yellow coloring of the nodes marked as the final route. In another way, table arrangement can be done by making a maximum node only have two degrees. That waypoint, the waypoints can be arranged in such a way that they are formed as in **Figure 11**.

Figure 10. Weighted graphs after drafting.

D. Simulation Environment

The program will run on Lubuntu Operating System version 18.04 as most applications will run on the Linux environment. For Visualization, a Gazebo application is used so that the UAV's flying behavior can be seen. The UAV will fly following the distance on the GPS so that the visualization of the waypoint passed can be seen properly.

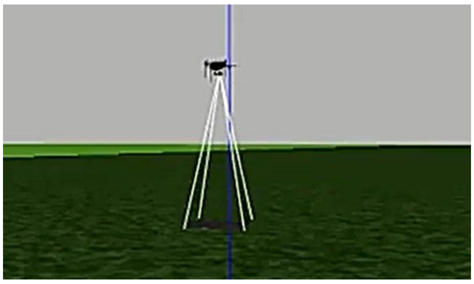


Figure 11. Drone View via Gazebo application in *real time*

IV. RESULTS & DISCUSSION

A. Results

Two sets of experiments have been carried out to deliver medicines to the homes of residents whose access is difficult to reach as seen on the map. In the first experiment, UAV flight routes were used that did not follow the rules of the Cheapest Link and only concerned with delivery having arrived at all points of destination. Meanwhile, in the second experiment, the UAV route was compiled using the Cheapest Link Algorithm which will be described in the next section.

Figure 12. Weighted graphs without using the cheapest link algorithm (Experiment 1)

After carrying out a flying mission and successfully returning to the starting point, namely Hasan Sadikin Hospital Bandung, the results were obtained, namely that the UAV had flown for 9 minutes 27 seconds measured from the time the UAV took *off* to *landing*. The distance traveled is 1,678 meters (assuming only horizontal height is calculated). This route consumes 13.76 V of existing battery power. Of course, the power required exceeds the battery capacity which is only 12 V.

Figure 13. Display during experiments via *Software in The Loop* (Experiment 2)

After carrying out a flying mission and successfully returning to the starting point, namely Hasan Sadikin Hospital Bandung, the results were obtained, namely that the UAV had flown for 5 minutes 54 seconds measured from the time the UAV *took off* to *landing*. The distance traveled is 1,090 meters (assuming only horizontal height is calculated). This route consumes 10,072 V of existing battery power. That is, there is still 1.9 V left on the battery.

B. Discussion

From the two experiments that have been carried out, it can be seen that the difference is quite significant. First of all, it is seen from the aspect of time. The route on the second try saved 3 minutes and 33 seconds from the first try and had a time efficiency of 62.43% against the first try. This is certainly very much needed considering the large number of patients who need medicines and are an urgent need.

Furthermore, from the aspect of distance, the route in the second experiment had a distance difference of 588 meters shorter than the first experiment. This proves that the weight of each side is very important to pay attention to. The efficiency of the second experiment from the point of view of distance gave a result of 64.96% against the first try.

Finally, from the aspect of battery power. A battery with a capacity of 12 V greatly affects the flight resistance of the UAV. As it turned out, it was found that the second experiment had a better difference, amounting to 3.688 V from the first try. The efficiency value obtained in the second experiment, which was 73.19% more efficient than the first experiment.

The battery power aspect is considered important because considering the battery capacity in UAVs generally has a capacity of 12 V only. If the required power is more than this value, the battery power may be exhausted at the time of delivery of medicines. A very fatal effect is that the UAV may fall because it runs out of energy to fly.

V. CONCLUSION

In determining flight routes, especially in delivering medicines for COVID-19 sufferers who are self-isolating in their respective homes, a mature strategy is needed. The strategy of determining UAV flight routes can be done by applying the principles of Hamilton's circuit theory so that all homes that need a supply of medicines can be well reached even if access is limited.

In addition, optimization is also needed in the delivery of medicines so that the fastest route is obtained on the condition that all houses that are self-isolating can be reached in a fast time and the shortest possible distance. The average efficiency obtained, especially in terms of time, distance, and power, is 66.86% better than all possible routes.

VI. FUTURE WORKS

The next suggestion for researchers is that the determination of the fastest route is not only measured based on time, distance, and power, but can be added to the influence of the load weight, rotor efficiency and optimal high stability to avoid obstacles. It can be further developed regarding efficient delivery innovations so that UAVs can automatically detect delivery automatically according to the priority scale of each patient's needs.

BIBLIOGRAPHY

[1] M. I. Suwatno, "Aplikasi Pencarian Rute Penerbangan Terpendek Dengan Menggunakan Algoritma Dijkstra pada PT Dwidaya Indoexchange," 2015.

[2] R. Munir, "Graf (Bag.1)." http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf (accessed Dec. 04, 2020).

[3] R. Munir, *Matematika Diskrit edisi ketiga*. Bandung, 2009.

[4] R. Munir, "Graf (Bag.3)." http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian3.pdf (accessed Dec. 07, 2020).

[5] B. Santosa, *Konsep dasar optimasi - Pengantar metaheuristik implementasi dengan Matlab*. 2017.

[6] H. Fachrudin, "Optimasi Penentuan Rute Perjalanan Sales pada UD. Aster," *Repos. Univ. Islam MAJAPAHIT*, 2019, [Online]. Available: http://repository.unim.ac.id/id/eprint/169

[7] S. Wahyuningsih, D. Satyananda, and D. Hasanah, "Kajian Karakteristik Solusi Varian Traveling Salesman Problem (TSP) dan Aplikasinya," *Pros. Semin. Nas. Mat. dan Pendidik. Mat.*, no. 978, pp. 490–498, 2015, [Online]. Available: http://fmipa.um.ac.id/index.php/2016/08/24/karya-ilmiah-dosen-fmipa-an-sapti-wahyuningsih-pada-prosiding-semnastika-2015-di-unesa/

[8] P. Tannenbaum, *Excursions in Modern Mathematics, 7th Edition*, 7th ed. Fresno: Pearson, 2010.

[9] G. J. Pottie and W. J. Kaiser, *Principles of Embedded Networked Systems Design*. Cambridge: Cambridge University Press, 2005.