Proceedings of Science and Mathematics
Faculty of Science, Universiti Teknologi Malaysia
Vol. 4, 2021, page 124-134

Optimal Paths in Chinese Postman Problems Using Graph Theory

[1]Fatin Faridah Samrin and Fong Wan Heng

[1,2]Department of Mathematical Sciences
Faculty of Science, Universiti Teknologi Malaysia,
81310 Johor Bahru, Johor, Malaysia.

e-mail: [1]fatinfaridahs@gmail.com, [2]fwh@utm.my

Abstract Graph theory is rapidly moving into the mainstream of mathematics mainly due to its applications in various fields including the arc routing. Theoretically, the concept of routing is basically used to determine the optimal path within a graph or a network. In this research, the Chinese Postman Problem (CPP), a famous arc routing problem is studied. The Chinese Postman Algorithm is proposed to solve the CPP on both undirected and directed graphs. The application of the algorithm in solving the arc routing problem is shown where two examples are presented to illustrate them.

Keywords Graph theory, arc routing, Chinese Postman Problem, Chinese Postman Algorithm.

1 Introduction

Applications of graph theory can be found in other mathematical branches, as well as other fields like engineering, operations research, and sciences, etc. A graph is a set of vertices (also called nodes) interconnected by edges (also called arcs). Nodes and arcs can be used in a wide range of applications. They may reflect physical networks like electric circuits, roads, or organic molecules [1]. Graph theory could solve the arc routing problem at an intersection like that of Euler's solution on the Königsberg bridge problem.

The increasing number of streets these days has affected the arc routing problems such as the routing of waste collection, newspaper distribution, fuel deliveries, etc. [2]. This problem leads to an increase in the time taken in delivery jobs, network maintenance, road gritting, etc. [3]. Although the problem of arc routing is practically important and has a huge range of real-life applications, it is very challenging especially when the actual road map has large number of streets and customers [4].

2 Literature Review

Arc routing problems are also known as vehicle routing problems in which the tasks to be carried out are placed on some edges or arcs of a network [5]. The purpose of the arc routing problem is to determine a least-cost route that services a subset of arcs in a graph, with or without constraints [2].

The Königsberg bridge problem was the first known theorem on the arc routing [4] where the vertices represent the landmasses while the bridges represent the edges. Euler observed that when a vertex is visited during the process of tracing a graph, there must be one edge that enters the vertex and there must also be another edge that leaves the vertex [6]. Thus, the order of the vertex must be even. Another well-known problem is the Chinese Postman Problem (CPP) that will be used in this research to solve the arc routing problem. In this problem, the edges and nodes represent streets and intersections. In general, there are three main arc routing problems:

- the Chinese Postman Problem (CPP),
- the Rural Postman Problem (RPP),
- the Capacitated Arc Routing Problem (CARP).

Graph theory is very important in solving the CPP. The concept of graph theory is used in CPP where the edges of the graph represent the streets (one-way or two-way) and the vertices represent the intersections of the streets. In CPP, graph theory is used to find the shortest distance or path between the vertices. In graph theory, the shortest path is determined by finding a path between vertices in a graph such that the sum of the weights of its edges is minimised.

A walk in CPP that starts at a vertex, traces each edge exactly once and ends at the starting vertex, is called an Euler trail [8]. If it ends at some other vertices, it is called an open Euler trail [9]. An open Euler trail is possible if and only if there are exactly two vertices of odd degree while an Euler trail is possible if and only if every vertex is of even degree [7].

3 Methodology

The Chinese Postman Problem (CPP) is studied in this research and an algorithm is proposed to solve the problem. In this section, two algorithms are discussed which are: the undirected Chinese Postman Algorithm to solve CPP on an undirected graph and the directed Chinese Postman Algorithm to solve CPP on a directed graph. Before that, the finding of the all-pairs shortest path using the Floyd-Warshall Algorithm is presented first in Section 3.1 since it will be used in the Chinese Postman Algorithm to find the odd degree vertices pairing with minimum weight.

3.1 All-Pairs Shortest Path

In this section, the Floyd-Warshall Algorithm is discussed. This algorithm will help in completing the Chinese Postman Algorithm. Floyd-Warshall Algorithm is one type of all-pairs shortest paths algorithm to find the shortest route for all pairs of nodes that exist on a graph [10]. As a result, this algorithm generates a matrix which represents the minimum distance from any node to all other nodes in the graph [11]. The values obtained from the Floyd-Warshall Algorithm are used to find the odd degree vertices pairing with minimum weight (or the length of the edges). The Floyd-Warshall Algorithm is shown in Figure 3.1.

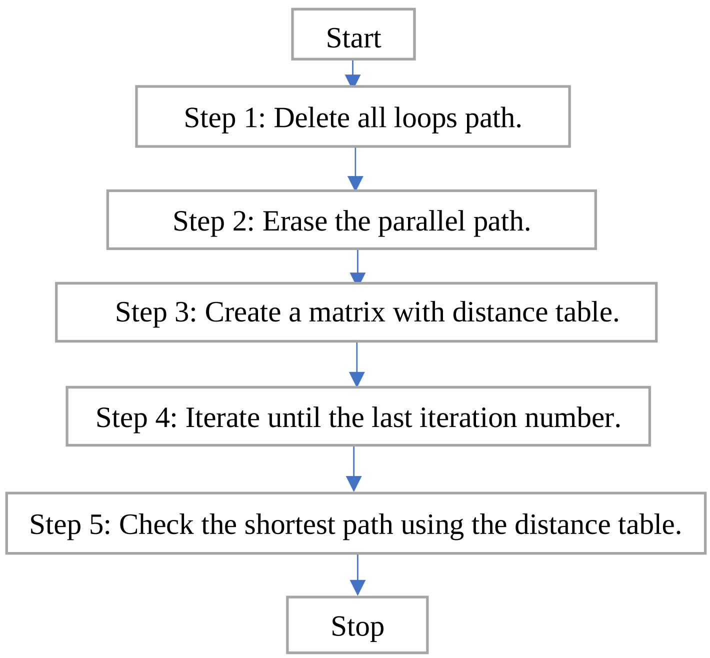

Figure 3.1 Floyd-Warshall Algorithm

3.2 The Chinese Postman Problem on an Undirected Graph

The first known result concerning the undirected CPP is due to Euler [2] who showed that a connected undirected graph $G(N, A)$ has a circuit that traverses every edge exactly once if and only if G contains exactly zero nodes of odd degrees i.e., if every node is an even degree. An Eulerian graph is a path in a graph, which visits each edge exactly once. Similarly, an Euler tour is the path in such a graph. With an Euler graph, it is easy to find the Euler tour. If the graph has an Euler tour, thus an optimal postman route exists. The algorithm to solve the CPP on an undirected graph is described in four steps:

Step 1: All odd degree vertices in G are identified. If there are *m* of it, where *m* is an even number, proceed to Step 5.

Step 2: The shortest path between all vertices is computed. A minimum length pairwise matching of the *m* odd degree vertices is obtained.

Step 3: For each of the pairs of odd degree vertices in the minimum length pairwise matching found in Step 2, construct G'.

Step 4: G' is added to the original G, obtain G^*. The graph G^* obtained contains no nodes of odd degree.

Step 5: An Euler tour on G^* need to be found. This Euler tour is an optimal solution to the Chinese Postman Problem on the original graph *G.* The length of the optimal tour is equal to the total length of the edges in *G* plus the total length of the edges in the minimum length matching.

3.3 The Chinese Postman Problem on a Directed Graph

Concerning directed graphs, there is an extremely important condition for CPP to work, unlike for undirected graphs. The graph has to be strongly connected: in other words, there must be a directed path connecting every node [12]. As with undirected graphs, the first step to solve a CPP problem on directed graphs is to verify whether the graph has an Euler tour, which can be done by using manual approach.

A strongly connected $G(N, A)$ directed graph has an Euler tour if and only if in all the nodes, the number of edges leading into that node and the number of edges leading away from it are the same, where in the literature it is referred to as a balanced graph. To trace whether the graph is balanced or not, we need to find the indegrees and outdegrees. If the graph is imbalance, then balance the graph by adding additional path considered only on the imbalanced vertex. Thus, an Eulerian graph is formed since all the vertices are even. If the graph has an Euler tour, thus an optimal postman route is exists. The algorithm to solve the CPP on directed graph is described in four steps:

Step 1: Compute the indegree, $d^-(i)$ and outdegree, $d^+(i)$ of each vertex. Then, calculate the difference of indegrees and outdegrees to find the imbalanced nodes.

Step 2: Find the additional paths from the imbalanced nodes.

Step 3: Add the additional path to the graph *G* such that the indegrees and outdegrees match all nodes after inserting the new paths. New symmetric graph G^* is obtained.

Step 4: Find Euler tour in G^*. This Euler tour is an optimal solution to the Chinese Postman Problem on the original graph G.

4 Application of Chinese Postman Algorithm

In this section, two examples are discussed to illustrate the Chinese Postman Problem for both undirected and directed graphs.

4.1 An Example of Chinese Postman Problem on an Undirected Graph

The following network in Figure 4.1 shows the distances of roads in kilometers, km connecting seven towns in a city X. A policeman, based at town A in city X, had to travel along each of the roads at least once before returning to base A before his lunch break. In order to complete his travel in a minimum time, thus an optimal path and minimum total distance that the policeman travel must be obtained.

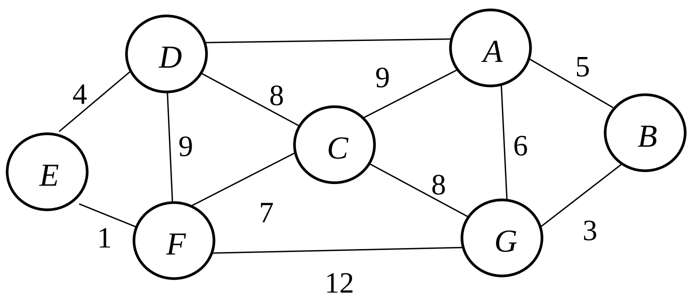

Figure 4.1 Weighted graph G_1 of city X

Figure 4.1 shows a weighted graph G_1 of city X. The edge of the graph represents the street length of city X while the vertices represent the towns in city X. To solve the policeman's problem, an optimal postman route will be obtained by using the undirected Chinese Postman Algorithm. Let G_1 be a connected graph with zero nodes of odd degree. Then, the order of the vertices of G_1 are listed in Table 4.1.

Table 4.1 The order of each vertex in graph G_1 of city X

Vertex	Order
A	4
B	2
C	4
D	4
E	2
F	4
G	4

From Table 4.1, the connected graph G_1 is said to be Eulerian since it has no odd degree vertices. A connected graph G is Eulerian if and only if the degree of each vertex of G is even [9], thus G_1 is an Eulerian graph [9]. To solve the problem, an Euler tour need to be obtained so that an optimal postman route can be found. Hence, by manual approach and to traverse all the edges once, one possible Euler tour is obtained:

$$A - G - C - D - A - B - G - F - D - E - F - C - A\,.$$

The minimum total distance of the optimal postman route in G_1 is equal to the total length of the graph which is 85. The solution is as shown below.

Let a weighted graph (G_1, w), where $G_1 = (V, E)$, $V_{G_1} = \{A, B, C, D, E, F\}$ and $E_{G_1} = \{(A,B), (B,G), (G,C), (C,D), (D,F), (F,G), (G,A), (A,C), (C,F), (F,E), (E,D), (D,A)\}$. Then, find the total edges in G_1 referring to Figure 4. Note that the total length of the graph is the total weight of all the edges of the graph denoted as $w(E_{G_1})$. Thus, $w(E_{G_1})$ is obtained:

$$\begin{aligned}
E_{G_1} &= \{(A,B), (B,G), (G,C), (C,D), (D,F), (F,G), (G,A), (A,C),\\
&\qquad (C,F), (F,E), (E,D), (D,A)\},\\
w(E_{G_1}) &= w(A,B) + w(B,G) + w(G,C) + w(C,D) + w(D,F) + w(F,G) +\\
&\qquad w(G,A) + w(A,C) + w(C,F) + w(F,E) + w(E,D) + w(D,A),\\
&= 5 + 3 + 8 + 8 + 9 + 12 + 6 + 9 + 7 + 2 + 1 + 4 + 8 + 9\\
&= 85.
\end{aligned}$$

Since the total length of graph G_1 is 85, thus the total distance of the optimal postman route in G_1 is $85\mathrm{km}$. The policeman can complete his travel without repeating any streets and finishing his travel before lunch break with a total distance of $85\mathrm{km}$ by following the optimal postman route:

$$A - G - C - D - A - B - G - F - D - E - F - C - A.$$

4.2 An Example of Chinese Postman Problem on a Directed Graph

An ambulance driver is based at *a* and has to respond to an emergency at d and returns back to a immediately in a network G_3 of city Z. However, the streets he used are one-way streets. In order to arrive as early as he can, an optimal postman route could help him. Thus, we need to find the optimal postman route he needs to take on the network in Figure 4.6 so that he can arrive at a early.

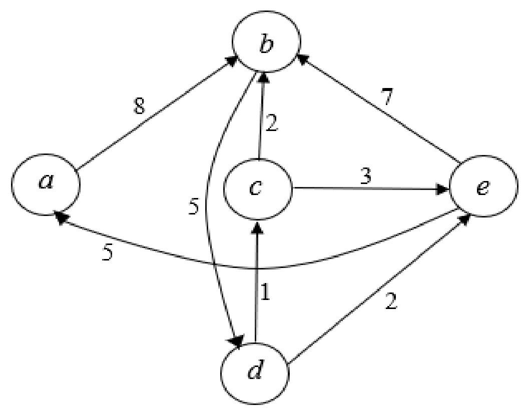

Figure 4.2 Weighted graph G_2 of city Y

Figure 4.2 shows a weighted graph G_2 of city Y. The vertices of the graph represent the towns and edge of the graph represents the roads of the city. To solve the ambulance driver's problem, an optimal postman route will be obtained. The directed Chinese Postman Algorithm is used to find the optimal Chinese postman route in graph G_2.

First, to solve the problem, the graph must be strongly connected. In this case, graph G_2 is strongly connected since any node in G_2 can be reached from any other node. There is also a round tour that uses all the edges. There are no negatives cycles since all the edges are positive-weight edges. Thus, the problem can be solved on G_2. We find the imbalanced vertices by computing the indegree and outdegree of each vertex as shown in Table 4.2.

Table 4.2 The difference of indegrees and outdegrees

Vertex	Indegree, $\boldsymbol{d^-(i)}$	Outdegree, $\boldsymbol{d^+(i)}$	Net Supply, $\boldsymbol{D(i)}$
a	1	1	0
b	3	1	-2
c	1	2	1
d	1	2	1
e	2	2	0

From Table 4.2, since there exists $d^-(i) \neq d^+(i)$ of i, therefore G_2 is not symmetric. The imbalanced vertices are b, c and d. An Euler tour would exist in the graph if and only if all the vertices are balanced. A set of paths between the imbalanced vertices need to be found such that G_2 will completely be balanced after adding the additional paths. The additional paths must be minimal, thus the all-pairs shortest distance between all the vertices are obtained in Table 4.3.

Table 4.3 Shortest distance matrix of graph G_2

	a	b	c	d	e
a	0	8	14	14	15
b	14	0	6	5	7
c	8	2	0	7	3
d	9	3	1	0	2
e	5	7	13	12	0

To solve the shortest path problem, by referring to Table 4.3, a new graph G_2^* is constructed containing three additional paths: two paths (b, d) and one path (d, c) as shown in Figure 4.3.

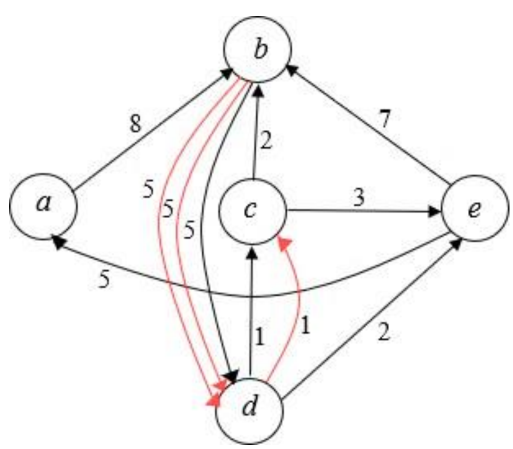

Figure 4.3 The Eulerian graph G_2^* of city Y

After inserting the additional paths, the in-out difference for all imbalanced graphs will be reduced to 0. By referring to Figure 4.3, there exists an Euler tour in G_2^* with total distance of the optimal postman route of 44km. The solution is as shown below.

Let a weighted graph be (G_2, w), where $, G_2 = (V, E)$, $V_{G_2} = \{a, b, c, d, e\}$ and $E_{G_2} =$

$\{(a,b),(b,d),(d,c),(c,b),(b,d),(d,c),(c,e),(e,b),(b,d),(d,e),(e,a)\}$. Then, find the total length between all the edges in G_2 based on Figure 6. Note that the total length of a graph is the total weight of all its edges denoted as $w(E_{G_2})$:

$$\begin{aligned}
E_{G_2} &= \{(a,b),(b,d),(d,c),(c,b),(b,d),(d,c),(c,e),(e,b)\},\\
w(E_{G_2}) &= w(a,b)+w(b,d)+w(d,c)+w(c,b)+w(b,d)+w(d,c)+w(c,e)+\\
&\quad c(e,b),\\
&= 8+5+1+2+5+1+3+7\\
&= 33.
\end{aligned}$$

Next, find the additional path in $G_2^* = (V,E)$. Let E'_{G_2} be the added path in G_2^*. Referring to Figure 4.3, the additional path is:

$$\begin{aligned}
E'_{G_2} &= \{\,(b,d),(b,d),(d,c)\,\},\\
w(E'_{G_2}) &= w(b,d)+w(b,d)+w(d,c)\\
&= 5+5+1\\
&= 11.\\
w(E_{G_2})+w(E'_{G_2}) &\Rightarrow 33+11=44.
\end{aligned}$$

In conclusion, the ambulance driver can respond to the emergency immediately with a total distance of 44km. By manual approach of finding the Euler tour in a graph, an optimal Chinese postman route corresponding to the length is obtained:

$$a - b - d - c - b - d - c - e - b - d - e - a.$$

5 Result and Discussion

The results for undirected Chinese Postman Algorithm turn out to be similar as presented in the literature [2] where from the literature, for an Eulerian graph, the optimal postman route should be the total length of the graph itself. This can be seen in Section 4.1. The weighted graph G_1 of city X has no odd degree vertices, thus it is Eulerian. Thus, in this network, the optimal path is the total length of the graph itself which is 85. In other case where the undirected graph is a non-Eulerian graph, an artificial edge with the minimum length should be added to make the graph Eulerian, then the optimal postman route can be obtained.

Next, the results obtained for the directed Chinese Postman Algorithm is also similar as presented in the literature where from the literature in order for a graph to have a solution, the graph must be strongly connected. As discussed in Section 4.2, a weighted graph G_2 of city Y is strongly connected but has imbalanced vertices, therefore G_2 is not symmetric. An Euler tour would exist in the graph if and only if all the vertices are balanced. Thus, additional paths are added in G_2 to balance it. Three additional paths are added in G_2, hence G_2 is balanced and it becomes Eulerian since all the orders of the vertices are even. There exists an Euler tour that traverses all the edges in G_2 where it is the optimal postman route with a total distance of 44km. In other case where the directed graph is already balanced, the solution is similar to the undirected CPP.

6 Conclusion

In conclusion, the Chinese Postman Algorithm proposed has successfully been applied in the arc routing with the help of Floyd-Warshall Algorithm in finding the all-pairs shortest path between the vertices. The optimal path in the illustrated problems have been obtained both in an undirected network and a directed network. The algorithm used are the undirected Chinese Postman Algorithm to solve CPP on undirected graph and the directed Chinese Postman Algorithm to solve CPP on a directed graph. The algorithm is capable to solve all the arc routing problems. The results obtained are similar to the literature [2] where from the literature, for an Eulerian graph, the optimal postman route should be the total length of the graph itself. For the non-Eulerian graph, an artificial edge with the minimum length should be added to make the graph Eulerian, then the optimal postman route is obtained [3]. The CPP can further be explored using any other methods such as using softwares like Matlab, TORA, Maple and so on. It is a dynamic problem and many more types of problem may be extended from this work.

The CPP can further be explored using any other methods such as using softwares like Matlab, TORA, Maple and so on. It is a dynamic problem and many more types of problem may be extended from this work.

7 References

[1] Gross, J. L and Yellen, J. *Graph Theory and its Application*. Boca Raton, Florida: CRC Press. 2006.

[2] Eiselt, H.A., Gendreau, M. and Laporte, G. Arc routing problems, part I: the Chinese postman problem. *Operations Research.* 1995. 43(2): 231-242.

[3] Assad, A. A., and Golden, B. L. Chapter 5 Arc routing methods and applications. In: Barnhart, C. and Laporte, G. *Handbooks in Operations Research and Management Science*: *Network Routing*. Amsterdam: Elsevier. 375-483; 1995

[4] Dror, M. (Ed.), *Arc routing. Theory, solutions, and applications*. Boston: Kluwer Academic Publishers. 2000.

[5] Aráoz, J., Fernández, E., and Zoltan, C. Privatized rural postman problems. *Computers & Operations Research*. 2006. 33(12): 3432–3449.

[6] Trudeau, R.J. *Introduction to Graph Theory*. 2nd ed. Dover Publications. 1993.

[7] Grotschel, M., & Yuan, Y. X. Euler, Mei-ko Kwan, Königsberg, and a Chinese postman. *Documenta Math.* 2012. Extra Volume ISMP: 43-50.

[8] West, D. B. *Introduction to Graph Theory*. 2nd ed. Upper Saddle River, N.J: Prentice-Hall. 1996

[9] Risald, A., Mirino, E. Best routes selection using Dijkstra and Floyd-Warshall algorithm. *2017 11th International Conference on Information & Communication Technology and System (ICTS)*. Surabaya. 2017.155-158.

[10] A.Singh, P. Mishra, and H. Performance Analysis of Floyd-Warshall Algorithm vs Rectangular Algorithm. *International Journal of Computer Applications (0975–8887)*. 2014. 107(16). 57-62.

[11] Wang, H. and Wen, Y. Time constrained Chinese postman problems. *Computers and Mathematics with Applications*. 2002. 44(3): 375–387.