

Generalization of Dijkstra's Algorithm for Extraction of Shortest Paths in Directed Multigraphs

Siddhartha Sankar Biswas, Bashir Alam and M.N. Doja

Department of Computer Engineering,
Faculty of Engineering and Technology, Jamia Millia Islamia University, New Delhi-110025, India

Received 2013-02-01, Revised 2013-04-18; Accepted 2013-04-29

ABSTRACT

The classical Dijkstra's algorithm to find the shortest path in graphs is not applicable to multigraphs. In this study the authors generalize the classical Dijkstra's algorithm to make it applicable to directed multigraphs. The modified algorithm is called by Generalized Dijkstra's algorithm or GD Algorithm (GDA in short). The GDA outputs the shortest paths and the corresponding min cost. It is claimed that GDA may play a major role in many application areas of computer science, communication, transportation systems, in particular in those networks which cannot be modeled into graphs but into multigraphs.

Keywords: Min-Weight Multiset, Multigraphs, Shortest Path Estimate, Relaxation, GDA

1. INTRODUCTION

Graph theory has wide applications in several branches of Engineering, Science, Social Science, Medical Science, to list a few only out of many. Graph is also an important non-linear data structure in Computer Science. Multigraph (Balakrishnan, 1997; Biswas *et al.*, 2012) is a generalized concept of graph where multiple edges (or arcs) may exist between vertices. Many real life situations of communication network, transportation network, cannot be modeled into graphs, but can be well modeled into multigraphs because of the scope of dealing with multiple edges (or arcs) connecting a pair of nodes. A huge and rich volume of literature is available in the area of 'Graph Theory', but unfortunately the 'Theory of Multigraphs' has not so far developed upto that extent to meet the present requirements to deal with real life network problems. Since the notion of multigraphs (Balakrishnan, 1997; Biswas *et al.*, 2012) is an extension of the notion of graphs, we cannot take it granted that all the rich theories and properties of graphs will be true in case of multigraphs too, unless studied rigorously in the

context of multigraphs. There is a genuine need to make algebraic characterization of multigraphs (Biswas *et al.*, 2012), to define various fundamental operations on multigraphs and then to study the various properties of multigraphs. Though Dijkstra's algorithm (Cormen *et al.*, 2009) works well for the kind of network model which are graphs, but there is also serious need to find out how Dijkstra's algorithm can be modified or generalized so that it can be implemented to find out the shortest path in multigraphs also. The data structure Multigraph is a generalized notion of graph. A multigraph or pseudograph is like a graph but it is permitted to have multiple edges (also called "parallel edges") that have the same end nodes. Thus two vertices may be connected by more than one edge. Some authors also allow multigraphs to have loops, i.e., an edge that connects a vertex to itself, while others call these pseudographs reserving the term multigraph for the case with no loops. Throughout in our work here, we have worked only with multigraphs, not pseudograph.

Obviously, a classical graph is a special case of multigraph where between a pair of vertices there are no

Corresponding Author: Siddhartha Sankar Biswas, Department of Computer Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia University, New Delhi-110025, India

multi-edges, only single edge (or no edge). The following diagram (**Fig. 1**) shows a multigraph consisting of four cities A, B, C and D in a country, where the edges denote bus routes and alternative bus-routes from one city to another, with the distances mentioned in miles against each edge.

Multigraphs have great implementations in many real life situations (Ivanco, 2010; Giasemidis *et al.*, 2012; Meszka and Skupien, 2012; Knisley *et al.*, 2012; Martschat *et al.*, 2012). For example, two neighboring routers in a network topology might share more than multiple direct connections between them (instead of just one), so as to reduce the bandwidth as compared if a single connection is used. If we can find a method to implement Dijkstra's algorithm in this network, we can easily deduce the shortest path from a source host to all the routers on that network, where many of them might be have multiple connections between each other and based on this result and as well as the data traffic volume on each inter-connections, we can find an optimized path between various routers and viz-a-viz, between the two end hosts. In this study we have modified the Dijkstra's algorithm to make it applicable in multigraphs.

1.1. Classical Dijkstra's Algorithm

The Classical Dijkstra's algorithm (Cormen *et al.*, 2009) is not applicable to multigraphs but to graphs to find the shortest path from a source node to a destination node, as well as the corresponding cost. Many real life applications, such as Link State routing protocol, are based on the concepts of Dijkstra's algorithm. Most of these applications were based on the fact that the concerned network is a simple directed graph, where there is only one arc between any two nodes in a single direction, i.e., in a network or a directed graph, between any two of its nodes X and Y, there can be maximum two arcs, XY and YX.

1.2. Shortest Path Estimate of a Vertex in a Directed Graph

Consider a weighted directed graph $G = (V, E)$. Shortest path estimate $d[v]$ of any vertex v , as shown in the diagram (**Fig. 2**), where vertex v is one of the neighboring vertices of the currently traversed vertex u , is the distance between the vertex v and vertex u , added with the shortest distance between the starting vertex s and vertex u , where $s, u, v \in V[G]$:

$$\therefore d[v] = (\text{shortest distance between } s \text{ and } u) + (\text{weight of arc between } v \text{ and } u)$$

1.3. Relaxation of an Arc in Dijkstra's Algorithm

For the relaxation process of an arc to happen, we must first initialize the graph along with its starting vertex and shortest path estimate for each vertices of the graph G .

INITIALIZE-SINGLE-SOURCE (G, s)

1. FOR each vertex $v \in V[G]$
2. $d[v] = \infty$
3. $v.\pi = \text{NIL}$
4. $d[s] = 0$

Now on the basis of this initialization process, Dijkstra's algorithm proceeds further and the process of relaxation of each arc begins as shown in the diagram (**Fig. 3**). The sub-algorithm RELAX, plays the vital role to update $d[v]$ i.e., the shortest distance value between the starting vertex s and the vertex v (which is neighbor of the current traversed vertex u , $\forall u, v \in V[G]$)

The RELAX algorithm runs as shown below:

1. IF $d[v] > d[u] + w(u, v)$
2. THEN $d[v] \leftarrow d[u] + w(u, v)$
3. $v.\pi \leftarrow u$

where, $w(u, v)$ is the weight of the arc from vertex u and vertex v and $v.\pi$ denotes the parent node of a vertex v , $\forall v, v \in V[G]$.

Dijkstra's algorithm solves the single-source shortest-path on a weighted directed graph $G = (V, E)$ for the case in which all edge weights are non-negative. Dijkstra's algorithm maintains a set S of vertices whose final shortest path weights from the source s has already been determined. The algorithm repeatedly selects the vertex $u \in V-S$ with the minimum shortest-path estimate, adds u to s and relaxes all edges leaving u . The algorithm is as follows:

DIJKSTRA (G, w, s)

- 1 INITIALIZE-SINGLE-SOURCE (G, s)
- 2 $S \leftarrow \emptyset$
- 3 $Q \leftarrow V[G]$
- 4 WHILE $Q \neq \emptyset$
- 5 DO $u \leftarrow \text{EXTRACT-MIN}(Q)$
- 6 $S \leftarrow S \cup \{u\}$
- 7 FOR each vertex $v \in \text{Adj}[u]$
- 8 DO RELAX(u, v, w)

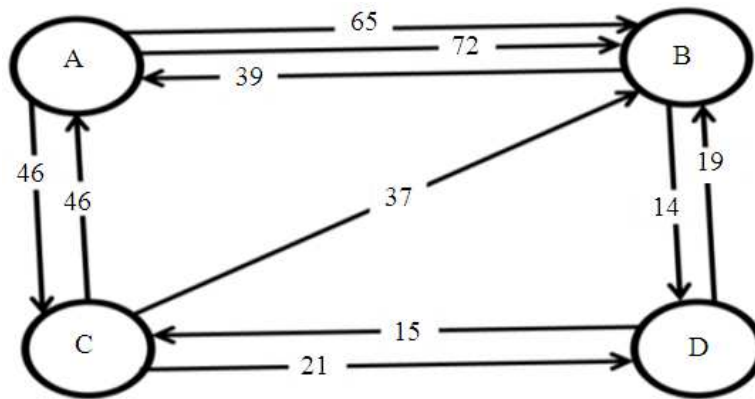


Fig. 1. A multigraph G

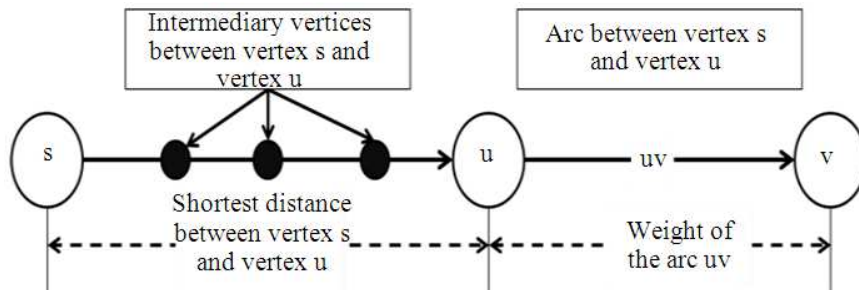


Fig. 2. Diagram showing $d[s, u]$ in G

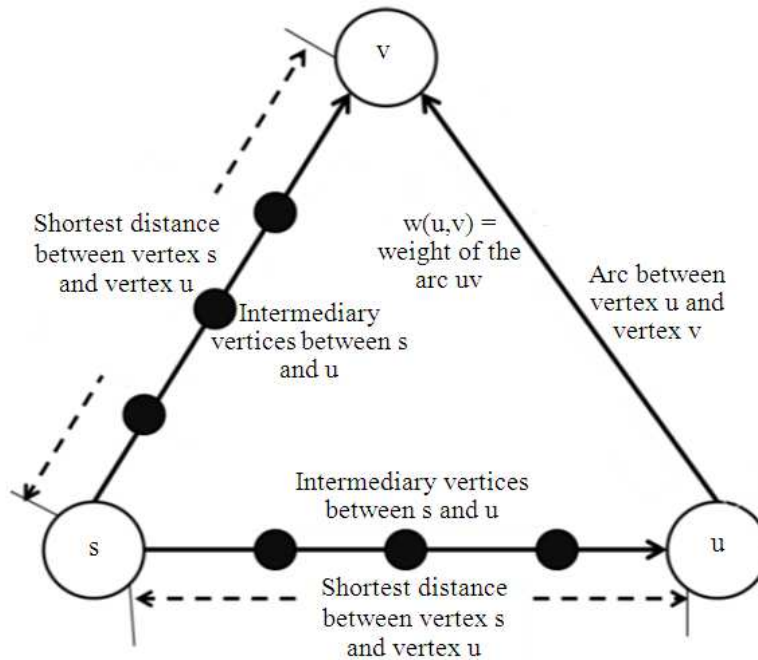


Fig. 3. Diagram showing how RELAX algorithm works

1.4. Generalization of Dijkstra’s Algorithm for Multigraphs

The Classical Dijkstra’s algorithm (Cormen *et al.*, 2009) is not applicable to multigraphs as multigraph is a generalization of graph. The main objective of the work in this study is to modify the classical Dijkstra’s algorithm, retaining the same philosophy of Dijkstra for graph, so as to make it applicable in directed multigraphs. Let us name the modified algorithm as ‘Generalized Dijkstra’s Algorithm’ or ‘GD Algorithm’ (or GDA) in short. For this, first of all we need to define the terms: Weight multiset, Min-Weight multiset, Shortest path estimate ($d[v]$) of a vertex, relaxing an arc, in the context of the multigraphs.

1.5. Weight Multiset and Min-Weight Multiset of a Directed Multigraph

Suppose that there are n number of arcs from the vertex u to the vertex v in a directed multigraph G , where n is a non-negative integer. Let W_{uv} is the set whose elements are the arcs between vertex u and vertex v , keyed and sorted in non-descending order by the value of their respective weights.

$$\therefore W_{uv} = \{(uv_1, w_{1uv}), (uv_2, w_{2uv}), (uv_3, w_{3uv}), \dots, (uv_n, w_{nuv})\}$$

where, uv_i is the arc- i from vertex u to vertex v and w_{iuv} is weight of that arc, for $i = 1, 2, 3, \dots, n$. If two or more number of weights is equal then they may appear at random at the corresponding place of non-descending array of weights with no loss of generality in our discussion.

Without any confusion, let us denote the multiset $\{w_{1uv}, w_{2uv}, w_{3uv}, \dots, w_{nuv}\}$ also by the name W_{uv} . Let w_{uv} be the min value of the multiset $W_{uv} = \{w_{1uv}, w_{2uv}, w_{3uv}, \dots, w_{nuv}\}$. Clearly, $w_{uv} = w_{1uv}$, as the multiset W_{uv} is already sorted. The collection of all w_{uv} in a directed multigraph forms a multiset which is denoted by W and is called by ‘Min Weight Multiset’ of the Multigraph. The collection of all w_{uv} in a directed multigraph forms a multiset which is denoted by w and is called by ‘Min-Weight Multiset’ of the Multigraph.

1.6. Shortest Path Estimate ($d[v]$) of a Vertex in a Directed Multigraph

Dijkstra’s algorithm works fine if there is a single arc between any two vertices. If we consider a multigraph (as shown below by an example of multigraph G_1 in Fig. 4), where there is multiple arcs between vertex u and its neighbor vertex v , the RELAX sub-algorithm would fail to run as there is no single value of weight for arc between vertex u and vertex v , rather there are multiple value of weights as there are multiple arcs between vertex u and vertex v .

In other words, to modify Dijkstra’s algorithm for application in directed multigraph, we must upgrade the RELAX sub-algorithm so as to enable it to choose the minimum value out of all the weights for defined for each of the arcs between vertex u and vertex v . Using the value of w_{uv} from the min-weight multiset w of a directed multigraph, we can now find the shortest path estimate i.e., $d[v]$ of any vertex v , in a directed multigraph.

(Shortest path estimate of vertex v) = (Shortest path estimate of vertex u) + (weight of the smallest arc between vertex u and vertex v) or, $d[v] = d[u] + w_{uv}$.

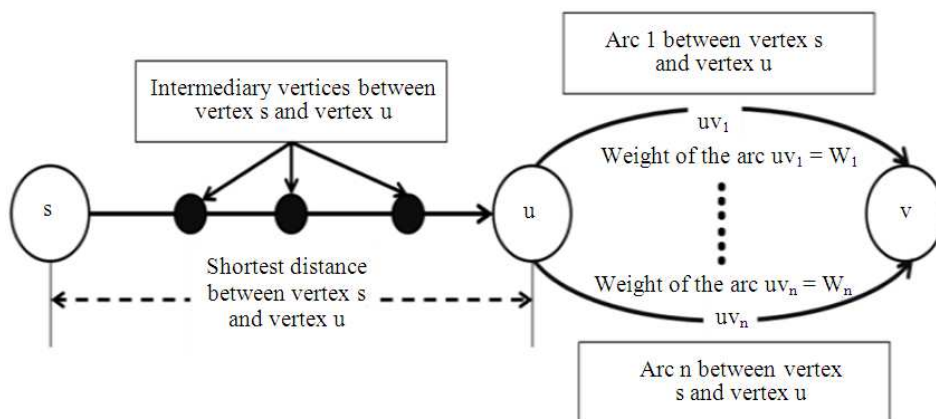


Fig. 4. Graph G_1

1.7. The Process of Relaxing an Arc in a Directed Multigraph

With the revised notion of the multiset w of multigraphs, let us now upgrade the RELAX algorithm below, which can be used in case of directed multigraph. Call this modified algorithm by 'RELAX.MG' algorithm.

RELAX.MG (u, v, W)

1. IF $d[v] > d[u] + w_{uv}$
2. THEN $d[v] \leftarrow d[u] + w_{uv}$
3. $v.\pi \leftarrow u$

Now we design the Generalized Dijkstra's Algorithm or GD Algorithm.

1.8. GD Algorithm (GDA) for Directed Multigraphs

Dijkstra's algorithm for finding single source shortest path in case of a directed graph has over the years found many real life applications, particularly in network theory, optimization, transportation. One of the reasons for large acceptance of Dijkstra's algorithm is its simplicity. But one shortcoming of this algorithm, in its present form is that, it is limited to finding single source shortest path in case of simple graph and cannot be implemented in case of multigraphs. Here we generalize the Dijkstra's algorithm so that it can be used to find single source shortest path in case of multigraphs also, we name this by 'Generalized Dijkstra's Algorithm' or 'GD Algorithm' by the title GDA:

GDA (G, W, s)

- 1 INITIALIZE-SINGLE-SOURCE(G, s)
- 2 $S \leftarrow \emptyset$

- 3 $Q \leftarrow V[G]$
- 4 WHILE $Q \neq \emptyset$
- 5 DO $u \leftarrow \text{EXTRACT-MIN}(Q)$
- 6 $S \leftarrow S \cup \{u\}$
- 7 FOR each vertex $v \in \text{Adj}[u]$
- 8 DO RELAX.MG (u, v, W)

Multigraphs are a generalization of graphs. It cannot be taken into granted automatically that all theories or tools applicable to graphs can also be applicable to multigraphs. The classical Dijkstra's algorithm returns the shortest path (if exists) from a source vertex to a destination vertex in a graph and also the corresponding cost. The GDA returns the shortest paths (if exist) from a source vertex to a destination vertex in a multigraph and also the corresponding cost.

Example 3.1

Consider a weighted and directed Multigraph G_2 , (as shown in Fig. 5), in which we want to solve the single-source shortest-paths problem, taking the source vertex as the vertex A.

Clearly The GDA algorithm yields the following result:

- $S = \{A, C, B, D\}$, i.e., the shortest path from the source vertex A is:

$$A \rightarrow C \rightarrow B \rightarrow D$$

- And the d -values i.e., shortest distance values of each vertex from the starting vertex A is:

$$d[A] = 0, d[C] = 3, d[B] = 7, d[D] = 9$$

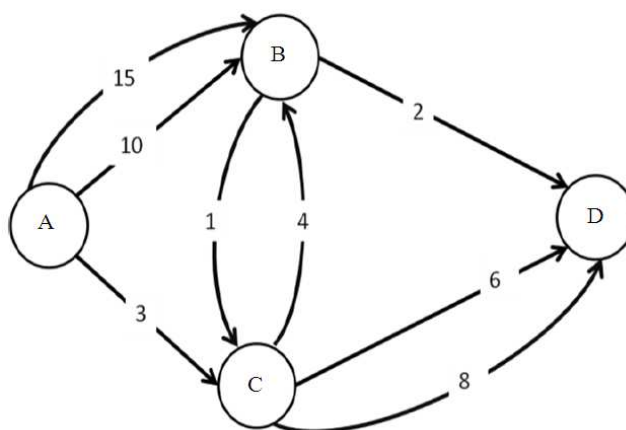


Fig. 5. Multigraph G_2

2. CONCLUSION

Multigraph is a generalization of graph. There are many real life problems of network, transportation, circuit systems, which cannot be modeled into graphs but into multigraphs. only. The classical Dijkstra's algorithm (to find the shortest path in graphs) is not applicable to multigraphs. In this study we have done slight adjustment in the classical Dijkstra's algorithm to make it applicable to multigraphs to find the shortest path from a source vertex to a destination vertex. The modified algorithm is called as Generalized Dijkstra's algorithm or GD Algorithm (GDA in short). The GDA results are shown using a hypothetical example of multigraph.

3. REFERENCES

- Balakrishnan, V.K., 1997. Schaum's Outline of Graph Theory: Including Hundreds of Solved Problems. 1st Edn., McGraw Hill Professional, New York, ISBN-10: 0070054894, pp: 293.
- Biswas, S.S., B. Alam and M.N. Doja, 2012. A theoretical characterization of the data structure 'multigraph'. J. Contemporary Applied Math.
- Cormen, T. H., C.E. Leiserson, R.L. Rivest and C. Stein, 2009. Introduction to Algorithms. 3rd Edn., The MIT Press, ISBN-10: 0262033844, pp: 1312.
- Giasemidis, G., J.F. Wheeler and S. Zohren, 2012. Multigraph models for causal quantum gravity and scale dependent spectral dimension. J. Phys. A. DOI: 10.1088/1751-8113/45/35/355001
- Ivanco, J., 2010. Decompositions of multigraphs into parts with the same size. Discuss. Math. Graph Theory, 30: 335-347.
- Knisley, D., J. Knisley, C. Ross and A. Rockney, 2012. Classifying multigraph models of secondary RNA structure using graph-theoretic descriptors. ISRN Bioinform., 12: 157135-157135. DOI: 10.5402/2012/157135
- Martschat, S., J. Cai, S. Broscheit, E. Mujdricza-Maydt and M. Strube, 2012. A multigraph model for coreference resolution. Proceedings of the 12 Joint Conference on EMNLP and CoNLL-Shared Task, (CST' 12), Association for Computational Linguistics Stroudsburg, PA, USA., pp: 100-106.
- Meszka, M. and Z. Skupien, 2012. Decompositions of a complete multidigraph into almost arbitrary paths. Discuss. Math. Graph Theory, 32: 357-372.