International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 4 (2017), pp. 631-640
© Research India Publications
http://www.ripublication.com

Fuzzy Real Time Dijkstra's Algorithm

Siddhartha Sankar Biswas

Department of Computer Science & Engineering, Faculty of Engineering & Technology, Jamia Hamdard University, Hamdard Nagar, New Delhi-62, India.

Abstract

For a public road transportation system or a MANET or an Adhoc Network, or any present day networks of communication systems, it carries a lot of uncertainties regarding traffic jam, logging due to sudden flood or due to some kind of urgent maintenance work (in case of public road network), and also regarding unpredictable damage or attack from external or internal sources, etc. Consequently, few or all of the existing links/arcs of a given network (graph) may not be always in their excellent original condition, rather in a weaker or much weaker condition at a real instant of time concerned with some user of the network. Sometimes even few links all of a sudden become disabled or blocked temporarily and waiting for maintenance/repair from the authority. This is a kind of unavoidable circumstances, but causes finally delay in communication or transportation. The classical Dijkstra's algorithm is applicable if we assume that all the links of the concerned graph are available at their best (ideal) condition, but the reality is that it is not the case always. Consequently, if the classical Dijkstra's algorithm is used to calculate the shortest path, it may be mathematically correct but not at reality, because the computed shortest path may become costlier in terms of time and/or in terms of other overhead costs; whereas some other path may be the more efficient or more optimal which is remaining hidden. In [2] the authors solved this SPP problem by introducing a new term called by 'condition factor (CF)' in the network graph by developing "Real Time Dijkstra's Algorithm" or RTD Algorithm (RTDA). In this paper the author considers the same problem where the condition factor (CF) is not always crisp but fuzzy, and develops the corresponding version of Dijkstra's Algorithm called by "Fuzzy Real Time Dijkstra's Algorithm" or FRTD Algorithm (FRTDA). It is claimed that FRTDA may play a significant role in many real time situations in a communication or transportation network.

Keywords: Condition factor (CF); Fuzzy Condition Factor (FCF), Link status; Effective cost (EC), RTDA, FRTDA.

MSC Code (2000) : 05C85

1. INTRODUCTION

This work is sequel to the work [2] in which the authors developed a new type of solution but in the philosophy of the classical Dijkstra's algorithm [6], for a real time SPP problem in graph theory, in particular for the transportation or communication network.

The author in [2] mentioned that though the classical Dijkstra's algorithm [6] performs well for the network model which are graphs, but there is also a genuine need to check-up the physical or logical condition of its links at the real instant of communication or transportation. For example [2, 8, 9], suppose that in a public road network one existing road AB from one city A to another city B is of distance 60 km which usually covered by a Super-Delux bus in one hour time or nearly so if it is in its original good condition; but at this time the road AB is not in its original condition due to water-logging caused by flood and rainfall. The final reality is that the Super-Delux bus now managed to cover this distance AB in two hours time! This type of real situation is very common in reality and can not be ignored while transportation/communication be in progress in the concerned application domain network. Sometimes it may also happen that there is no alternative road/link, but due to exigency the communication/transportation work can not be halted. The classical Dijkstra's algorithm [6] can provide a mathematical solution to such type of graphs, but in reality this solution is of no significance to the communication systems or to the transportation systems at that real instant of communication period because the mathematically so computed shortest path is not the optimal path for that period of time. For an analogous example, consider a MANET or an Adhoc Network in which there exist a link from node u to its neighbor node v, but because of some unfortunate reasons this link is not in the ideal condition (may be partially or temporarily damaged) at real instant of transportation time. Therefore although this particular link is well available officially for transmission of packets by the node u to its neighbor node v, but will cause the communication delay because of its so weak condition at this period of time. In case it is a fully damaged condition at this moment, then there will be even no scope for communication at all for some period of time!. This is a very useful information to the communication system if available to the sender node in advance. To solve such type of problems, the author in [2] introduced a new kind of mathematical parameter corresponding to each link (edge) called by **'Condition Factor (CF)' (or 'link status')**. They[2] made a slight adjustment of the classical Dijkstra's algorithm so that it can be applied successfully to solve the above type of real time SPP, by developing "Real Time Dijkstra's Algorithm" or RTD algorithm (RTDA). In [2] the authors considered only crisp values of 'Condition Factor (CF)'. Our observation is that all the CF can not be always crisp, few or all may be sometimes fuzzy numbers. If so, then the RTDA can not be applied to the SPP. In this paper we consider crisp as well as fuzzy CF and make the necessary modification of the existing RTDA algorithm (of Dijkstra's type) to introduce "Fuzzy Real Time Dijkstra's Algorithm" or FRTD algorithm (FRTDA). We do not reproduce in our work here any preliminaries of Graph Theory and of the famous Dijkstra's algorithm [6], which are available in any good book of Graph Theory, Algorithm (viz. [1], [3-5], [7]).

2. FUZZY REAL TIME DIJKSTRA'S ALGORITHM (FRTD ALGORITHM)

In our daily networks of transportation or communication systems, be it a public road transportation system, or air lines communication or packet transfer in a MANET (or in an Adhoc Network), in reality there are a lot of uncertainties because of the reason that one link may be good for communication process at this moment of time, but may get damaged after few hours because of several possible reasons which are usually unpredictable. Consequently, the existing links/edges of a given network (graph) may not be always in their original condition, but in a weaker or bad condition at a real instant of time of communication, or even few links sometimes could be disabled or blocked temporarily and waiting for maintenance/repair from the authority. If the Classical Dijkstra's algorithm [6] is applied to such type of graphs then it can produce mathematically optimal results which may not be truly optimal results, as few of the links are not in their ideal original conditions. In [2] the authors developed a solution for SPP to such real times scenario in application domain networks called by "Real Time Dijkstra's Algorithm" (RTD Algorithm). In our work here we generalize the mathematical parameter CF of [2] by introducing fuzzy CF, as all the condition factors may not be always crisp because of imprecise nature.

2.1 'Fuzzy Condition Factor'(FCF) of a Link

Consider a graph G, one node u and its neighbor node v. For the link uv we define the **fuzzy condition factor (FCF)** or **link status** which is a real time fuzzy parameter defined as below, and convert the FCF values into CF values as explained below:-

(i) The ideal (i.e. best or original) value of 'fuzzy condition factor' for each link uv is not fuzzy but the real value 1, if it is available at its best or original condition without any damage or attack internally or externally at this moment of time being fluently available for communication. We write FCF(uv) = CF(uv) = 1.

(ii) The worst value of 'fuzzy condition factor' for each link uv is also not fuzzy but the real value 0, if it is not available, i.e. at a condition of fully damaged or blocked at this moment of time. We write FCF(uv) = CF(uv) = 0.

(iii) Otherwise, the 'fuzzy condition factor' for each link uv is a fuzzy number $\tilde{f}$ where f is in between 0 and 1, which means that the link is available but not in its best condition. It is partially damaged or in a traffic-jam or at a similar other real time adverse circumstances (could be a temporary problem) which may cause the communication to be at slow pace.

(iv) For a decision parameter α pre-choosen by the concerned decision maker, consider the closed interval [p,r] which is the α-cut of the fuzzy number $\tilde{f}$.

Define CF(uv) by : CF(UV) = (p+r)/2. Consequently, the FCF gets defuzzified.

Thus, whatever be the CF or FCF, we have correspondingly $0 \leq CF(uv) \leq 1$ for each of the existing links like uv of G at the real time under consideration during the period of communication or transportation.

It is presumed that all the real time information will automatically get updated at every node of the graph at every q quantum of time (a quantum amount q is to be pre-fixed by the authority depending upon the properties of the concerned network, depending upon the kind of communication/transportation it is doing).

Let us present an example below (see Figure 1) to explain the model, with hypothetical data and information. Consider a public transport network used by the petroleum company "All India Petroleum" which transports its materials across the cities in India. Consider a real instant of transportation process. Suppose that at this instant of time, its link-road CD is the only link at its original physical condition, link-road AB is not available due to rainfall and water-logging , link-road BC is not available for 10 hours next due to PWD work for maintenance, link-road AC is available but with intense traffic jam, link-road CB is available in almost original condition, link-road BD is available but with little amount of water logging on it (which has caused the traffic slow for the plying busses, trucks, vehicles, etc.). The corresponding FCFs for all its links estimated and/or computed by the IT Manager of the "All India Petroleum" transportation system are shown in the graph G (figure-1).

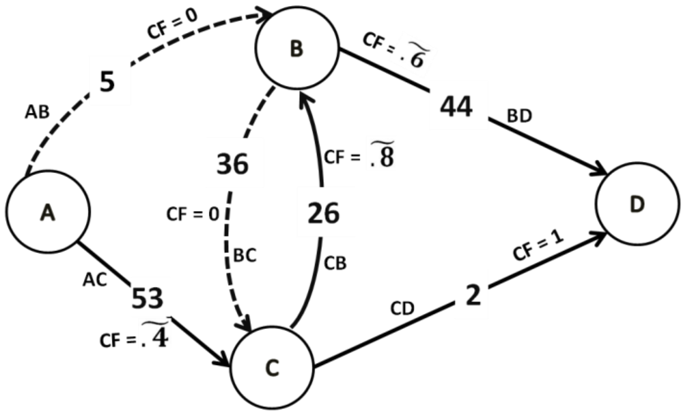

Figure 1: A public transport network G with FCF of its links at some period of time.

After conversion of all FCFs into CFs, the above public transport network G may be called the graph G1 with CF of its links at that period of time is as shown in Figure-2 below.

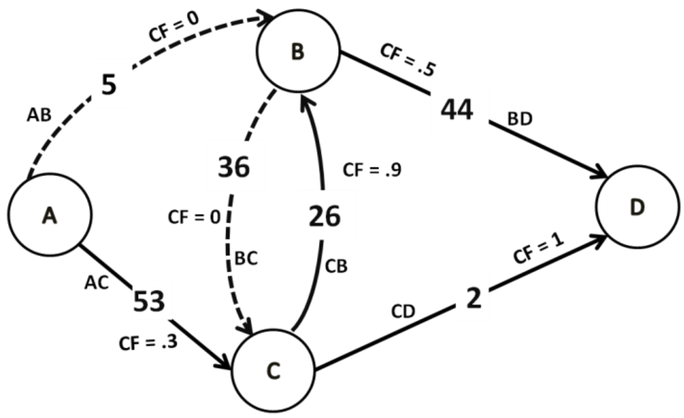

Figure 2: The graph G1 with FCF converted into CF for all of its links at the same instant

It may be noted that the graph G and the graph G1 are basically same, but the only difference is that in the graph G the links have FCF and the graph G1 is the converted version of the graph G where the FCFs are converted into CFs.

2.2 Effective Cost (EC) of a link/edge

We will consider here the graph G1 with FCF converted into CF for all of its links. Consider a node u and its neighbor node v of G1. Suppose that the cost or weight of this link is C_{uv}. If due to some unavoidable circumstances this link is not available at its original or ideal condition, then a fraction of its ideal condition will happen to be available to the system for communication at this period of time. Consequently, if this link is choosen for communication of a packet from node u to node v in the graph G1, the effective cost of this link will not be in reality equal to C_{uv} but a little higher side, depending upon the physical or logical condition of the link uv at that real instant of time. The effective cost (EC) of the link uv at that real instant of time will be defined by

$$EC(uv) = C_{uv} / CF(uv), \quad \text{where } CF(uv) \neq 0.$$

If $CF(uv) = 0$, then we say that $EC(uv) = \infty$, i.e. the link uv is unavailable at this period of time.

The actual costs or weights of the links are fixed in a graph, but the effective costs are a dynamic data which vary with time.

Consider the transport network of Figure-2 above. If we apply the classical Dijkstra's algorithm in this graph, we will certainly get a mathematically optimal result which in reality will not be effective and significant to the communication system for an effectively optimal action at this real instant of time. Incorporating the CFs of its links, the effective costs (revised temporarily for that real instant of time period) are estimated and the real time snapshot of the graph is presented below in Figure-3 in a new graph G2. Here each cost C_{uv} is replaced by its corresponding effective cost EC(u,v).

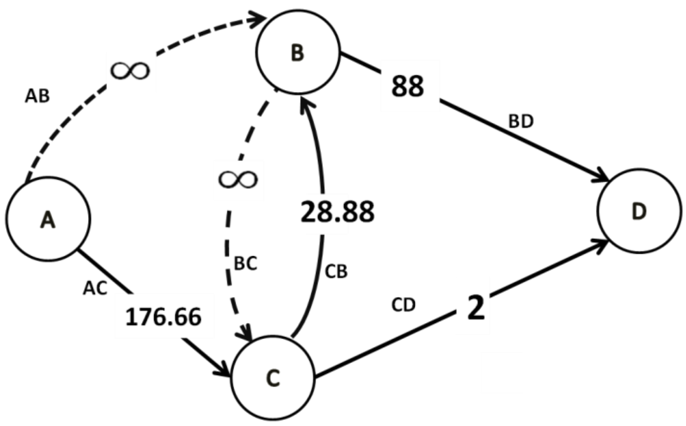

Figure 3: The real time scenario of the public transport network by graph G2 at some instant of time.

In the next section we consider this graph G2. However, if there is no confusion let us use the original title G for the sake of presentation (instead of writing G2).

2.3 "Fuzzy Real Time Shortest path estimate" of a vertex in a directed graph

Let us consider a weighted directed graph G = (V, E). The fuzzy real time shortest path estimate d[v] of any vertex (or node) v of G, where the vertex v is one of the neighboring vertices of the currently traversed vertex u , is the effective real time cost E(u,v) between the vertex v and the vertex u but added with the fuzzy real time shortest distance between the starting vertex s and the vertex u, where s, u, v ∈ V[G].

∴ d [v] = (fuzzy real time shortest distance between s and u) + EC(u,v)

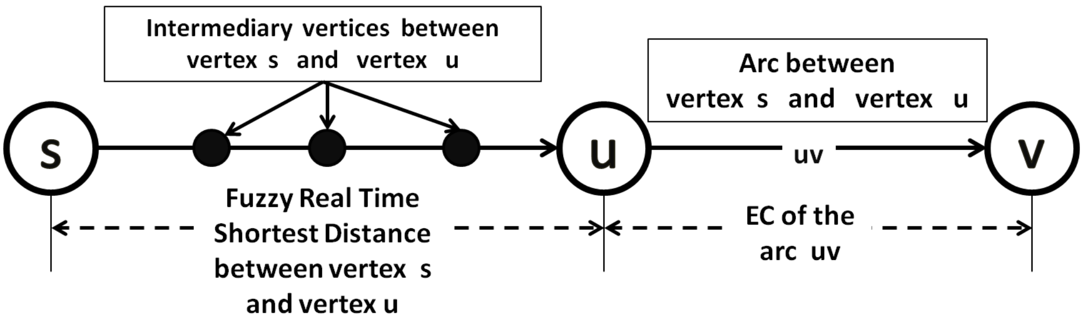

Figure 4: Diagram showing d[v] in G.

2.4 "Fuzzy Real Time Relaxation" of a link/Arc

For the relaxation process of a link or arc to happen at some real instant of time, we must first initialize the graph G 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 ∈ V[G]
2. d[v] = ∞
3. v.π = NIL
4. d[s] = 0

Once this initialization process be executed, our proposed "Fuzzy Real Time Dijkstra's Algorithm" then proceeds further and the process of relaxation of each arc begins at the real time only. The sub-algorithm "FUZZY REAL-TIME-RELAX" updates d[v] i.e. the fuzzy real time shortest distance value between the starting vertex s and the vertex v *(which is neighbor of the current traversed vertex u, ∀ u, v ∈ V[G]).* Suppose that W is the multiset of weights (costs) and C is the multiset of the corresponding CFs.

The "FUZZY REAL-TIME-RELAX" algorithm can be pseudo-coded as shown below :

FUZZY REAL-TIME-RELAX (u, v, W, C)

1. EC(u,v) ← C_{uv}/CF(u,v)
2. IF d[v] > d[u] + EC(u,v)
3. THEN d[v] ← d[u] + EC(u,v)
4. v.π ← u

where, EC(u,v) is the effective cost of the arc between vertex u and vertex v, and v.π denotes the parent node of the vertex v, ∀ u, v ∈ V[G].

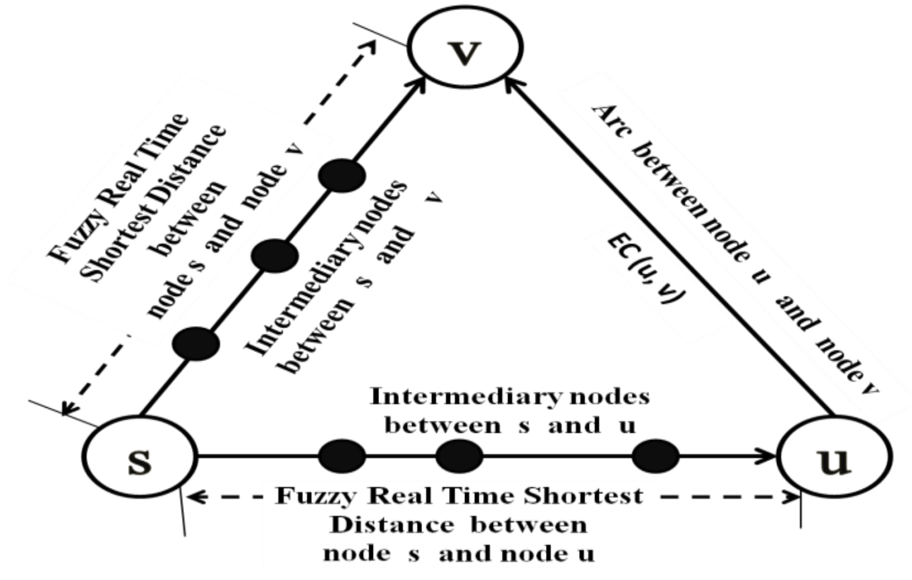

Figure 5: Diagram showing how the 'FUZZY REAL-TIME-RELAX' algorithm works.

2.5 FRTD Algorithm

The Fuzzy Real Time Dijkstra's algorithm (FRTDA) solves the single-source shortest-path at real instant of time on a weighted directed graph G = (V, E) for the case in which all costs are non-negative. The FRTDA maintains a set S of vertices whose final shortest path weights from the source s has already been determined. This algorithm repeatedly selects the vertex u ∈ V – S with the minimum value of fuzzy real time shortest- path estimate, adds u to s, and relaxes all edges leaving u at the real instant of time for application.

The FRTDA algorithm is as follows:

FRTDA (G, W, C, s)

```
INITIALIZE-SINGLE-SOURCE (G, s)
S ← Ø
Q ← V[G]
WHILE  Q ≠ Ø
        DO  u ← EXTRACT-MIN (Q)
            S ← S ∪ {u}
            FOR  each vertex v ∈ Adj[u]
                 DO  FUZZY REAL-TIME-RELAX (u, v, W, C)
```

3. CONCLUSION

In a classical transportation or communication graph, at any given instant of time, it is presumed that a link is always considered to be available for communication or transportation (i.e. the link-status or CF of every link is always 1). Hence, the classical Dijkstra's algorithm is applicable to solve SPP for such type of ordinary transportation or communication graphs. But in real life situation, in a network graph the link-status CF of a link, after converting the FCF of this link, can be any amount in a closed interval [0,1], depending upon its present physical condition or capability for communication at that real instant of time of communication or transportation. Under such a real time circumstances, the classical Dijkstra's algorithm provides a mathematical solution which is not of that much significance during the real time period of communication or transportation via that network, in case the communication is planned to be completed in minimum amount of time efficiently. The author in [2] proposed "Real Time Dijkstra's Algorithm" (RTDA) to solve this problem but with CF only, without considering FCF. We introduce the notion of FCF here because of the reason that condition factor of a link may not be always a precise quantity but an ill-defined quantity. Consequently, in this paper we make a slight modification of the "Real Time Dijkstra's Algorithm" (RTDA) by developing "Fuzzy Real Time Dijkstra's Algorithm" (FRTDA) which can provide to the communication system the effective shortest path with respect to any condition of the links of the network, and thus providing a real time optimal solution. Surely FRTDA can play a major role in the existing and future communication systems considering its potential to deal with real time topology of the platform networks with respect to any ill condition of its links or edges.

REFERENCES

[1] Balakrishnan, V. K., Graph Theory, McGraw Hill; 2005.

[2] Biswas, Siddhartha Sankar., Alam, Bashir. and Doja, M.N., A Slight Adjustment of Dijkstra's Algorithm for Solving SPP in a Network in Real Time Environment, Third International Conference on Computational Intelligence and Information Technology – CIIT 2013, Held in Mumbai, India., Published at International Conference on ComNet CIIT & ITC 2013 Proceedings (Elsevier Publication), pp: 256-259 ISBN: 978-81-910691-6-3.

[3] Bollobas, Bela., Modern Graph Theory, Springer; 2002.

[4] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford. Introduction to Algorithms, MIT Press. 2009.

[5] Diestel, Reinhard., Graph Theory, Springer 2000.

[6] Dijkstra, E. W., A note on two problems in connexion with graphs, Numerische Mathematik1: 269–271 (1959). DOI:10.1007/BF01386390.

[7] Harary, Frank., Graph Theory, Addison Wesley Publishing Company, 1995.

[8] Jenson P, Barnes J., Network Flow Programming, John Wiley and Sons, NY 1980.

[9] Marzolf, F., Tr.panier, M., Langevin, A., "Road network monitoring: algorithms and a case study," Comput Oper Res, Vol.33(12), pp.3494-3507, December 2006.