International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Unitary Partition Algorithm in Shortest Path Problems

P. Geetha, A.Gnanam, S. Suseela

Abstract*: *In Existing system, a network consists of N(V,E) to find a shortest path to minimize the total cost from source to destination. Researchers have been proposed many algorithms for finding the shortest path like two familiar algorithms namely Prim's algorithm and Kruskal's algorithm. In this paper we provide a new approach to solve the shortest path problems using unity partitions algorithm in C programme. We may apply this unitary partitions algorithm to find the shortest path in physical networks and social networks like road networks, railway networks, airline traffic networks, electrical networks and organizational charts, friendship networks, Interaction networks

Keywords* : *Partitions, Complete Partitions, Unitary partitions algorithm.

I. INTRODUCTION

The hypothesis of parcels has a fascinating history. Certain unique issues in parcels [1], [2], [3] absolutely go back to the medieval times. Euler in reality established the frameworks of the hypothesis of allotments. A considerable lot of the other incredible mathematicians – Cayley, Gauss, Hardy, Jacobi, Lagrange, Legendre, Littlewood, Rademacher, Ramanujan, Schur and Sylvester have added to the improvement of number hypothesis. The hypothesis of segments is one of not very many parts of science that can be valued by any individual who is blessed with minimal in excess of an enthusiastic enthusiasm for the subject. In the hypothesis of numbers or combinatorial issues from all sources any place discrete items are checked or orders its applications are found. Portrayal of a positive whole number as an entirety at least two squares is additionally a segment, where each part is a square or a square number. There are numerous sorts of segments [4], [5] relying upon the parts spoke to. The parcel of a positive number n can be characterized as : A limited non – diminishing succession with the end goal that and for all The

Manuscript published on 30 September 2019
* Correspondence Author
P. Geetha* Assistant Professor, Department of Mathematics, Periyar Maniammai Institute of Science & Technology, Periyar Nagar, Vallam, Thanjavur, maxgeetha@gmail.com
A.Gnanam Assistant Professor, Department of Mathematics, Government Arts College, Thuvakkudi malai, Trichy, gnaanam@yahoo.com
S. Suseela Assistant Professor, Department of Compute Science and Engineering, Periyar Maniammai Institute of Science & Technology, Periyar Nagar, Vallam, Thanjavur, suseelas@pmu.edu

are known as the pieces of the segment and is known as the length of the segment.

Number hypothesis, an intriguing part of science that manages whole numbers and their properties assumes a significant job in discrete arithmetic. Diagrams [6], [7], [8] are discrete structures comprising of vertices and edges that associate these vertices. A way is a chain where the terminal vertex of each circular segment is characterized to the underlying vertex of the following bend. The issue of finding the briefest (way of least length) from vertex 1 to some other vertex in a system is known as a Shortest Path Problem. A briefest way between two vertices in a weighted diagram is a way of least weight. In an unweighted chart, a briefest way [9], [10] implies one with minimal number of edges. On the off chance that the subgraph T of an associated chart G is a tree containing all the vertices of G, at that point T is known as a crossing tree of G. In the event that G is an associated weighted diagram, the crossing tree of G with the littlest complete weight is known as the base traversing tree of G. In Prim's calculation edges of least weight that are episode on a vertex as of now in the spreading over tree and not framing a circuit are chosen.

We currently give a calculation to locate the briefest way of a diagram G by unitary parcels calculation in C [11], [12], [13], [14], [15] and [16] for finding most brief way. Definition: A complete partition of an integer n is a partition $\mu = (\mu_1, \mu_2, ..., \mu_k)$ of n, with $\mu_1 = 1$, such that each integer r, $1 \le r \le n$, can be represented as a sum of elements of $\mu_1, \mu_2, ..., \mu_k$. In other words, each r can be expressed as $\sum_{i=1}^{k} \beta_j \mu_j$, where each β_j is either 0 or 1. In a complete partition the unitary partition is μ_1^{l} where l is the length of the partition and $l = n$.

Algorithm 1 (Unitary Partitions for network weight >1) :
INPUT : Number of vertices and edges
OUTPUT : Shortest path
Float path = tl, l, minicost = 0, min 1, mini weight
Define n
Get edges of cost (i, j)
Find partition p(n)
$p = l$
$Cost(i, j) / l \; \forall \, i, j$
Sort $Cost(i, j) / l \; \forall \, i, j$
Find sp $\forall \, i, j$
Add sp

10. $Mc = Tc \times l$

End

Algorithm2 (Unitary Partitions for network weight >=1) :

Start

INPUT : Number of vertices and edges

OUTPUT : Shortest path

Float path = tl, l, minicost = 0, min 1, mini weight

Define n

Get edges of $Cost(i, j) > 1$

Find partition p(n)

$p = l$

$Cost(i, j) > 1 / l \; \forall \; i, j$

Sort $Cost(i, j) > 1 / l \; \forall \; i, j$

Find sp $\forall \; i, j$

Add sp

$Mc = Tc \times l + 1$

End

II. Working Procedure for Algorithm:

Stage 1 : - For a given system the edges are orchestrated in the request for expanding loads.

Stage 2 :- Choose an edge with least weight to be an edge of the system.

Stage 3 :- Divide the loads of each edge by unitary segment length

Stage 4 :- Edges with least proportions that don't frame a circuit with the effectively chosen edges are progressively included.

Stage 5 :- The system is halted after edges have been chosen in the system

Stage 6 :- To locate the most brief way with the base load in the system

Case (I) : If for all edges just duplicate by .

Case (ii) : If for any edge discover the entirety of proportions of the loads of the system duplicate by and include 1.

Stage 7 :- The adjusted of number is taken as weight of the most limited way in the system.Result and Discussion:

Fig.1 to Fig. 8 demonstrates that the system N1(N,E), it comprises of four hubs, N2 comprises of four hubs, N3 comprises of five hubs, N4 comprises of five hubs, N5 comprises of six hubs, N6 comprises of six hubs, N7 comprises of seven hubs and N8 comprises of eight hubs which are associated with one another in the system. Our point is to locate the briefest parth from the accompanying systems by utilizing calculation 1. Similarly we continue the calculation 2 with marginally unique system.

Example 1: Consider the following set of cities:

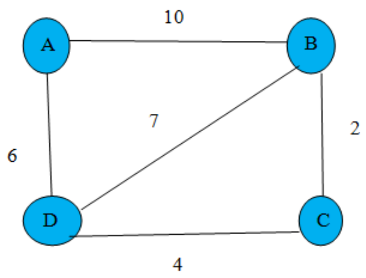

Fig.1

Nodes	A	B	C	D
A	0	10	0	6
B	10	0	2	7
C	0	2	0	4
D	6	7	4	0

Table.1

Example 2: Consider the following set of cities:

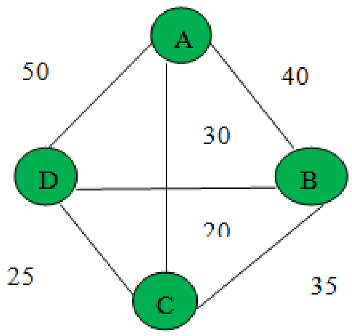

Fig.2

Nodes	A	B	C	D
A	**0**	**40**	**30**	**50**
B	**40**	**0**	**35**	**20**
C	**30**	**35**	**0**	**25**
D	**50**	**20**	**25**	**0**

Table.2

Retrieval Number: C6874098319/2019©BEIESP
DOI:10.35940/ijrte.C6874.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Example 3: Consider the following set of cities:

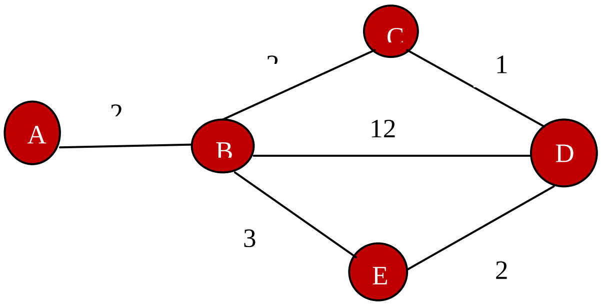

Fig.

Nodes	A	B	C	D	E
A	0	20	0	0	0
B	20	0	25	12	30
C	0	25	0	15	0
D	0	12	15	0	20
E	0	30	0	20	0

Table 3

Example 4: Consider the following set of cities:

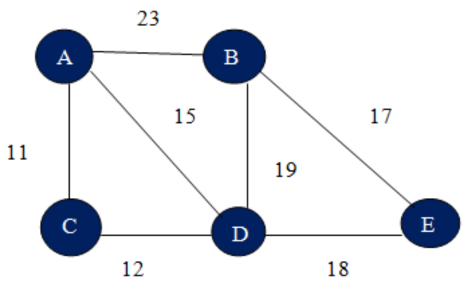

Fig.4

Nodes	A	B	C	D	E
A	0	23	11	15	0
B	23	0	0	19	17
C	11	0	0	12	0
D	15	19	12	0	18
E	0	17	0	18	0

Table 4

Example 5: Consider the following set of cities:

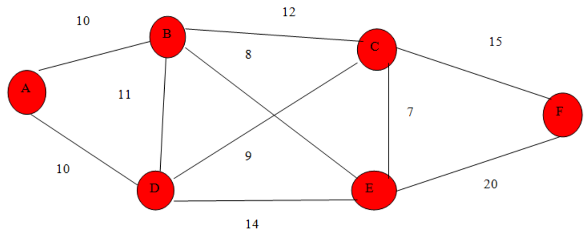

Fig.5

Nodes	A	B	C	D	E	F
A	0	10	0	10	0	0
B	10	0	12	11	8	0
C	0	12	0	9	7	15
D	10	11	9	0	14	0
E	0	8	7	14	0	20
F	0	0	15	0	20	0

Table 5

Example 6: Consider the following set of cities:

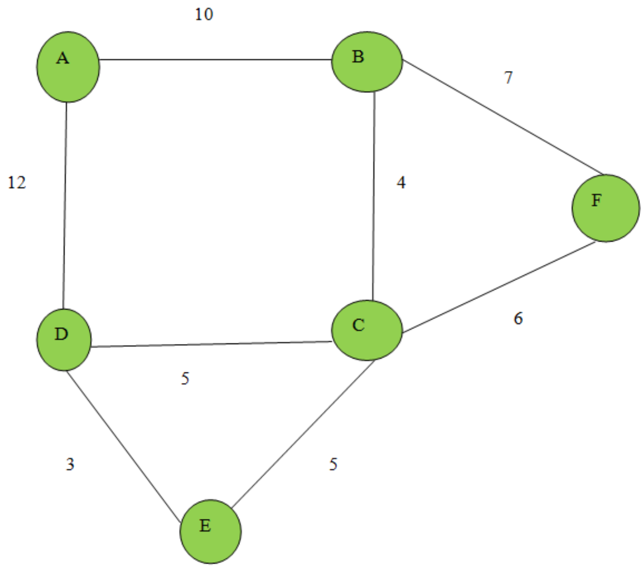

Fig.6

Retrieval Number: C6874098319/2019©BEIESP
DOI:10.35940/ijrte.C6874.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Nodes	A	B	C	D	E	F
A	0	10	0	12	0	0
B	10	0	4	0	0	7
C	0	4	0	5	5	6
D	12	0	5	0	3	0
E	0	0	5	3	0	0
F	0	7	6	0	0	0

Table 6

Example 8:

Nodes	A	B	C	D	E	F	G
A	0	23	0	0	0	25	12
B	23	0	15	0	0	0	8
C	0	15	0	10	0	0	9
D	0	0	10	0	17	0	16
E	0	0	0	17	0	29	25
F	25	0	0	0	29	0	40
G	12	8	9	16	25	40	0

Table 8

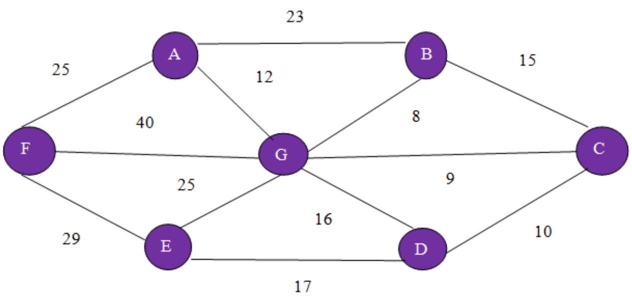

Fig.8

Example 8: Consider the following set of cities:

Nodes	A	B	C	D	E	F	G	H
A	0	15	10	20	0	0	0	0
B	15	0	11	0	23	0	0	0
C	10	11	0	15	0	12	0	0
D	20	0	15	0	0	0	14	0
E	0	23	0	0	0	19	0	17
F	0	0	12	0	19	0	20	18
G	0	0	0	14	0	20	0	16
H	0	0	0	0	17	18	16	0

Table 8

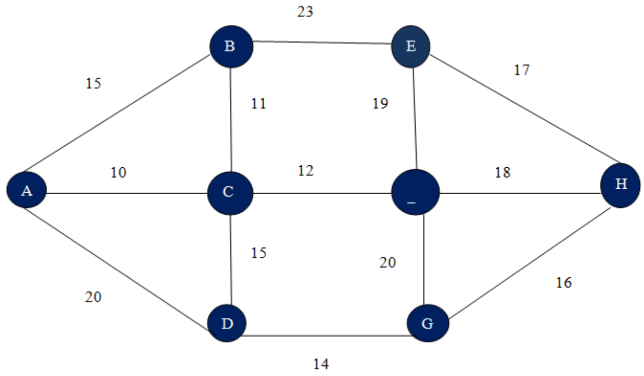

Fig.8

Analysis of the shortest path

The analysis of the shortest path has been analysed and which is represented in the table 9.
It shows that the shortest path from the source to the destination from the various networks of the parameters such as number of nodes, shortest path, minimum weight, partition algorithm weight, cost etc.,

III. Analysis of the Shortest Path

The examination of the most limited way has been broke down and which is spoken to in the table 9.

Figure	Number of nodes	Shortest Path	Minimum weight	By unitary partition algorithm weight	Cost
1.	4	BC, CD, AB	2.000000	6.000000	12.000000
2.	4	BD,CD,AC	20.000000	3.750000	75.000000
3.	5	AD, CD, AB, DE	12.000000	5.583333	67.000000
4.	5	AC,CD,BE,DE	11.000000	5.272727	58.000000
5.	6	CE,BE,CD,AB,CF	7.000000	7.000000	49.000000
6.	6	DE,BC,CD,CF,AB	3.000000	9.333333	28.000000
7.	7	BG,CG,CD,AG,DE,AF	8.000000	10.125000	81.000000
8.	8	AC,BC,CF,DG,CD,GH,EH	10.000000	9.500000	95.000000

Table 9

It demonstrates that the most brief way from the source to the goal from the different systems of the parameters, for example, number of hubs, most limited way, least weight, segment calculation weight, cost and so forth.,

IV. Conclusion

This proposed calculation give an answer for gap the base load by utilizing unitary parcels length and the count is made simpler to locate the most limited way in C program. The benefits of this segment calculation are effectively dividing the given least weight from the system and we can locate the most brief way in a powerful manner. However, to get the base load in the most limited way we receive a marginally extraordinary strategy. This calculation might be reached out to some other segments calculation in C programming which can be spoken to as a system applications like street systems, railroad systems, carrier traffic systems, electrical systems and hierarchical diagrams, fellowship systems, connection systems.

REFERENCES

1. G. E. Andrews, The Theory of Partitions, Encyclopedia of Mathematics and Its Applications. Vol.2. Reading, Mass.: Addison-Wesley, 1976.
2. George E. Andrews, Number Theory, W. B. Saunders Company, London
3. Ivan Niven, Herbert S. Zuckerman, Hugh L. Montgomery, An Introduction to the theory of Numbers, 5th Edition, John Wiley & Sons, Inc. New York.
4. S. K. Park, Complete Partitions, Fibonacci Quart., to appear.
5. Hokyu Lee and Seung Kyung Park, The Double Complete Partitions of Integers, Commun. Korean Math. Soc. 17(2002), No.3, pp. 431 – 437.
6. T.Veerarajan, Discrete Mathematics with Graph Theory and Combinatorics, 2008.
7. Douglas B. West, Introduction to Graph Theory, Second Edition, 1995.
8. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science, 2014.
9. F. Harary, Graph Theory, Addison-Wesley, Reading, MA, 1969.
10. R. Balakrishnan, K. Ranganathan, A text book of Graph Theory, 2012.
11. Yashavant Kanetkar, Let Us C, 15th Edition, BPB Publications.
12. Yashavant P. Kanetkar, Let Us C Solutions, 15th Edition, BPB Publications, 2006.
13. Yashavant P. Kanetkar, Data Structure Through C, 2nd Edition, BPB Publications.
14. Deepali Srivastava and S. K. Srivastava, C in Depth, BPB Publications.
15. David Griffiths, Dawn Griffiths, Head First C, O'Reilly publisher.
16. Brian W. Kernighan and Dennis Ritchie, Programming with C, Prentice Hall Publisher.

AUTHORS PROFILE

P. Geetha Assistant Professor, Department of Mathematics, Periyar Maniammai Institute of Science & Technology, Periyar Nagar, Vallam, Thanjavur**,** maxgeetha@gmail.com

A.Gnanam Assistant Professor, Department of Mathematics, Government Arts College, Thuvakkudi malai, Trichy, gnaanam@yahoo.com

S. Suseela Assistant Professor, Department of Compute Science and Engineering, Periyar Maniammai Institute of Science & Technology, Periyar Nagar, Vallam, Thanjavur**,** suseelas@pmu.edu