

,

Thesis (Dissertation) Research Area College Name MSc. Major Field  
Prof. Dr. Mushtak A.K. Shiker Al-Jenabi  
Babel, Iraq Jun 2022

# Some Applications of Network Optimization

SAFA MOHAMMED HADI AL-KHAFAJI

November 3, 2022

I am pleased to extend my sincere thanks to my loving mother and dear father who helped me and watched my upbringing and education since my childhood ,I am also pleased to extend my thanks to my beloved and my life partner who provided me with psychological and scientific support, I am also pleased to extend my sincere thanks and appreciation to Dr and Professor MushtaK Abdul Ghani Al-Janabi, who had the main credit for preparing this thesis, he is the virtuous educator before he was the successful professor, We ask God to help us to return part of the gift to all of them.

At the outset, I thank God Almighty, Who gave me strength and made it easy for me in my academic career, I also extend my thanks and gratitude to my dear supervisor, spiritual father, and first supporter (Prof. Dr. Mushtak A.K. Shiker Al-Jenabi) who provided me with scientific guidance and advice throughout the study period, as well as thanking and gratitude to my parents and my husband for their support and encouragement for me throughout the period of my studies.

## LIST OF SYMBOLS & ACRONYMS

No	The Symbols	Full name
1.	$t_o$	Optimistic time Estimate
2.	$t_m$	Most likely time Estimate
3.	$t_p$	Pessimistic time Estimate
4.	$t_e$	Time estimate
5.	MST	Minimum Spanning Tree
6.	V/S	All nodes without (S)
7.	A	Set of an activity
8.	E	Edge
9.	V	Vertices
10.	S	Set of Nodes Selected
11.	$SA_1$	Suggested method to Find the minimum spanning tree and its relationship to the minimum cut
12.	$SA_2$	Improvement Time estimate
13.	$SA_3$	Improvement maximum flow
14.	$SA_4$	Improvement minimum cut

Table 1:

## Abstract

We propose an improvement in estimating the time to be closer to reality, and the results will be achieved without any negative impact on the project, where we can give less weight to the estimate of the likely time, 3 instead of 4 to estimate the potential time of the distributors. As for the pessimistic and optimistic time estimates, we give them a weight of 1 for each of them, where the results are intermediate between the normal time estimate and the optimistic time estimate, we will take a map of one of the random cities and study on it the problem of maximum flow as well as the problem of minimum cutoff if we pass one of the paths from the starting point to the target point, we will encounter paths whose edges have different capacities. We look at the edge that has the highest amplitude. Then we make the rest of the edges of that path the same capacity in order to accommodate the largest possible amount of flow. Thus, improve the maximum flow but in the case of improving minimum cut. We cut the edge with the largest weight to ensure that the process is more efficient. We suggest a new way to find the minimum spanning tree and that is to connect it with the topic of minimum cut and the states to find the weight of each path. Then, we gradually take the paths as needed from the largest to the smallest and cut the edge with the largest weight continue this process. until the conditions of the spanning tree are met. in the end we get the same results as the old methods or slightly less than them.

# CHAPTER 1

---

## INTRODUCTION AND CONCEPT

### 1.1 Introduction

The network is a graph that connects activities to points complete any project, there must be an estimated time, and there are many estimates of time called the network term for all that constitutes a certain set of points where linking their own links and is also a series of relationships and activities and events associated with each other as exemplified by a series of actions which we will mention some of them in general. For example the manufacturer of a product after complement their production process manufactured goods to the stores or shops consuming business to be the executive director of the company, determine the path shortest distance to reduce the cost of transport the common examples of networks is upon receipt of company for a project must these complete projects in their estimated time or less than the agreed time. It can't be exceeded that time in order to maintain the company and its reputation in the commercial market. The spanning tree of the graph is a sub tree that must contain all nodes in the graph, provided. There is no edge rotation and the number of

edges is  $v-1$  where  $v$  is the vertices. A connected graph may contain many spanning trees, and a graph can contain more than one spanning tree due to the general properties of the spanning tree. A spanning tree has no rotation at its edges such that all the spanning trees we get from the original graph have the same nodes and the same number of edges, removing one edge of the spanning tree will separate the graph and adding one new edge to the spanning tree will rotate so that the bottom line of the tree is the lowest measure of weights or costs, it is possible to find the minimum spanning tree by extracting all possible spanning trees and calculating the weight of each tree so that the minimum spanning tree represents the least weight of them if the network is of a simple type, if it is somewhat complicated. We will discuss several algorithms for example, kruskal algorithm and Prim algorithm are which important to find the minimum spanning tree. One of the very important problems in linear programming problems are the problems of maximum flow and minimum cutoff, as well as the problem of minimum cost flow is the basic network flow problem, where determining max flow at min cost from the source to the target. The shortest path problem is the specializations of this problem, in this paper we discuss the concept of maximum flow and its algorithm. As well as the minimum cut and its algorithm, and then we suggest improving for these two algorithms. The authors presented many papers in varied fields of science such as operation research, and optimization [2] [5]. Usually we want to deliver certain types of goods, electric energy, water flow, or send messages through telecommunication networks or internet towers from one point to another. Thus, we try to make these flows as much as possible efficiency at a precise level of time, effort and cost, where the roots of these problems go back to the great scientists of the last century in 1845, especially the work of Gustav Kirchhoff, Also the idea of improving the

network with the development of linear programming has its origins in the forties and since then the noticeable development began on a large scale for computers that are able to perform large arithmetic operations and other operations [1]. Evans, J. (2017). He studied (optimization algorithms for problems to formulated on graphs and networks). This edition provides many new applications and algorithms while maintaining the classic foundations on which contemporary algorithm. Glover, F., Klingman, D., Phillips, N. V. (1992). He studied the Network models in optimization and their applications in practice. Ahuja, R. K., Magnanti, T. L., Orlin, J. B., Reddy, M. R. (1995). In book (Applications of network optimization) discussed several applications of the following network optimization problems: shortest paths, maximum flows, minimum cost flows, assignment and matching's, minimum spanning trees, convex cost flows, generalized flows, multi commodity flows, the traveling salesman problem, and network design. The book describes the following forty two applications, drawn from the fields of operations research, computer science, the physical sciences, medicine, engineering, and applied mathematics. Some basic notation and definitions from graph theory as well as a mathematical programming formulation of the minimum cost flow problem, which is the core network flow problem that lies at the heart of network optimization is introduced.

## 1.2 The Thesis Objectives

- 1- The goal of this thesis is to obtain satisfactory results in the least time or the least cost without having any negative impact on production.
- 2- When we done through the use of proposed methods and new ideas to be the same or slightly less than the results of the previous methods.

## 1.3 The Structure of the Thesis

The first chapter includes basic concepts and important acquaintances, including definitions of the topic of the network and definitions of the topic of optimization.

The second chapter contains the basic idea of the minimum spanning tree and how to find it, We used kruskal algorithm and prime algorithm to find the minimum spanning tree, then we proposed a new method related to the minimum cutoff to find the minimum spanning tree and we called it  $SA_1$ .

The third chapter includes definitions and steps of the maximum flow and minimum cut-off and the relationship between them, we have proposed an improvement for the maximum we called it  $SA_3$ , and minimum cut off we called it  $SA_4$  and the results between them are equal that is, the relationship between them has been preserved.

The fourth chapter includes how to find the project network and its importance, as well as the estimated time for any project through estimating the optimistic time and estimating the pessimistic time and estimating the normal time and the relationship between them through a mathematical law, A simple amendment has been proposed in that law so that the estimated time ranges between estimating optimistic time and estimating regular time, which is closer to estimating regular time because it is close to reality, and we called it  $SA_2$ .

## 1.4 Definitions of Some Network Optimization

Definition 1.5.1:(Network)

A Network is a continuum of events and activities that have a starting point and a goal point and this represents the end of a service or it means reaching the desired goal for example, when we want to travel we have to book a ticket first, then go to the airport, then the plane takes us to the desired goal and this sequence cannot be changed, the activities must follow the specified sequence otherwise we will not get a chain [29] [29] [8] [15].

Definition 1.5.2:( Activity)

An Activity is the performance of a work and it is represented in the form of an arrow in the network diagram, usually the project consists of several activities, and each activity takes a specific time [31] [28].

Definition 1.5.3:(Event)

An Event is the beginning or end of an activity, and events in networks are called nodes for example, the start of the clock hand movement is one event and the clock hand stop moving is another event and the events are represented in the network diagram in the form of circles [6] [28].

Definition 1.5.4:(Predecessor event)

A Predecessor event that immediately precedes an event of another event is called a Predecessor event [28] [15].

Definition 1.5.5:(Successor event)

A Successor event is immediately following the event of another event is called a Successor event [17].

Definition 1.5.6:(Dummy activity)

A Dummy activity is an activity that maintains the logical sequence of nodes

and edges and does not take any time and its main benefit is the introduction of a dummy activity that provides a direct connection to the network [2] [5].

Definition 1.5.7:(Construction)

A Construction the project network are the main components of this network are events and activities, an event occurs at a specific time, while activity occurs from time to time, usually in a network, there is a beginning event and an end event, and other events are between the beginning and end events [6] [15].

Definition 1.5.8:(Critical path and critical activities)

A Critical path and critical activities are The path that carries the most time is called the critical path, and the activities in this path are called critical activities, as all paths start from the beginning of the event and stop at the end of the event [15] [5].

Definition 1.5.9:( Earliest start time of an activity)

An Earliest start time of an activity Is the soonest possible time to start this activity [5] [4].

Definition 1.5.10:(Earliest finish time of an activity)

An Earliest finish time of an activity Is the earliest possible time to do this activity [5] [15].

Definition 1-5.11:(latest finish time)

A latest finish time of an activity Is the last possible time to complete the activity. [5] [15]

Definition 1.5.12:(latest start time)

A latest start time of an activity Is the last time the activity starts [5] [15].

Definition 1.5.13:(Total float of an activity)

A Total float of an activity is the time that does not delay the entire project,

but only the activity [25].

Definition 1.5.14:(Optimization)

An Optimization is the process in which the best result is obtained by reducing time, effort and cost under certain conditions and through repeating work sessions, which gives the maximum and minimum job [13].

Definition 1.5.15:(Objective function)

An Objective function is the goal to be reached as it is formulated as a linear programming issue in the form of achieving the greatest profit or achieving the lowest cost or greater economy in terms of time and effort, as it is expressed in a function of profit in the case of maximizing and loss in the case of minimization, and this depends on the nature of the problem to be studied and solve it [9] [2].

Definition 1.5.16:(Constraints)

A Constraints are equations or inequalities in which the resources are specific and measurable, they represent linear relationships that link the decision variables. variables [5].

Definition 1.5.17:(Decision variable)

A Decision variable are variables or unknowns whose values need to be determined to reach the desired goal [5] [2].

Definition 1.5.18:(Slack variable)

A Slack variable are the variables that are used specifically in linear programming, where they are added to the inequality constraint to convert it to equality and are added to the type of constraints that are in the form of less than or equal to, and this type of variable takes positive or zero values and does not take negative values [5] [21].

Definition 1.5.19:(Surplus variable)

A Surplus variable are the variables that are used specifically in linear programming, where they are added to the inequality constraint to turn it into an equality and are added to the type of constraints that are greater than or equal to, and this type of variables take negative values [5] [21].

Definition 1.5.20:(Artificial variable)

An Artificial variable is one of the types of variables that is entered in a linear program model to get the possible basic solution and can be added to the equality constraints and the largest or smallest inequality constraints, and it is usually added in the first step of the solution to get the acceptable solution [2] [21] [5].

Definition 1.5.21:(Basic variable and non basis variable)

A Basic variable and non basis variable are the variables that fulfill the non-negative condition of the linear programming problem and that they take any value other than zero and that the solution formed by it is called the basic potential solution and it is usually used in the simplex method in linear programming problems where the surplus variables or stagnation variables represent the basic variables of recurrence first, the rest of the variables are known as non-basic variables [2] [21].

Definition 1.5.22:(Optimal solution)

An Optimal solution Is the best possible solution among a certain number of solutions so that we can reach the target point with the least losses in terms of time, effort and money, this solution is the best measurably than all other solutions, which meets all the constraints of a problem improvement, suppose we have a subway system that connects the stations in the least amount from A to D, and we want to move from station A to station D, and there are only three paths in which we can move to reach the goal, which is station D, if we reach the target point in the least time and effort, then that path will be

the optimal solution [2] [5] [1].

Definition 1.5.23:(Feasible solution)

A Feasible solution are the variables that achieve all the limitations of the optimization problem, as most optimization algorithms work by trying to determine any solution first and then finding any other solution that is better than the first solution in terms of reducing time, effort and cost, which works to improve the value of the objective function [2] [21] [7].

Definition 1.5.24:(Multiple Solutions)

A Multiple Solutions are the solutions that exist in multiple ways through the form in which the objective function line and one of the constraint lines have the same slope and these solutions work directly to enlarge or reduce the objective function according to the graphical method for example, through the simplex method we can know whether the system has more than one solution, after completing the steps of the usual simplex method that is after removing the negative values in the operands of the objective function, but we observe if all the idle operands are replaced by the original variables, there is no more than one solution but if it remains one of the negative variables, then this solution must be completed and this variable replaced, this indicates that there is more than one solution [2] [5].

Definition 1.5.25:(Graph)

A Graph that links a group of points or objects together with sides or arrows. These arrows may have directions or be without direction only sides. This theory studies the properties of networks.

Definition 1.5.26 :(Connected graph)

A Connected graph is a graph that contains at least one path between each pair of vertices so that we can move from any vertex to another because it is connected.

Definition 1.5.27 :(Sub graph)

A Sub graph is a sub-graph of the original graph, provided that all its edges and vertices are present in the original graph and represent part of it.

Definition 1.5.28 :(Indirectd graph)

An indirectd graph is a graph whose edges are not directed, that is, its edges do not contain any direction.

Definition 1.5.29:(Complex graph)

A complex graph is contains many edges and vertices, and its edges may be parallel or self-contained.

Definition 1.5.30 :(Simple graph)

A Simple graph is a graph in which the number of edges and vertices are few and there are no self-loops and no parallel edges.

Definition 1.5.31 :(Non periodic)

A non periodic is a graph whose edges have no period.

Definition 1.5.32 :(Edge)

An Edge is the activity that connects any two vertices within the network and is denoted by an arrow within the network diagram.

Definition 1.5.33 :(Vertices)

A Vertices is a set of events that are connected together by edges to form a network and are symbolized by a circle within the network diagram.

## CHAPTER 2

# THE MINIMUM SPANNING TREE AND RELATIONSHIP WITH MINIMUM CUT

### 2.1 Minimum Spanning Tree

It is a special type of tree that reduces the weights, lengths, or capacities of the edges of that tree. It is considered the least measure of the weights or costs of the spanning tree, the graph must be connected in order to find the minimum spanning tree for it. If the graph is offline, we can calculate the minimum the spanning tree for each of its connected components is collectively known as the Minimum Spanning Tree, the weights of the edges may be different or they may be the same (if they are different, the minimum spanning tree is not unique because the cost of the possible edges that reach the target point may be similar). The spanning tree must contain all the vertices of the original graph and the edges of the spanning tree must also be non-periodic, an example of this is the cable company that saves money because one of the conditions of this company is that the cable path is single and without going back, It in turn runs a line for

multiple revival operations by reducing the amount of cables [24] [20] [18].

## 2.2 The Basic Principles Of The Spanning Tree

- . Adding an edge connecting two vertices in a tree creates another path (i.e. a single cycle), an example of this

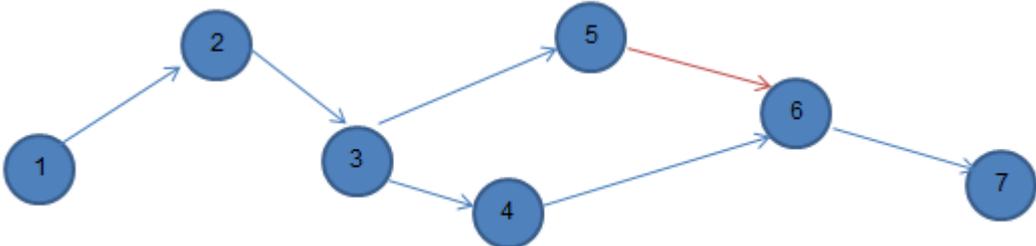


Figure 2.1: The adding from basic principles for spanning tree

**path 1:** 1-2-3-4-6-7

**path 2:** 1-2-3-5-6-7

- . Cutting an edge in a graph is splitting the vertices into two separate groups and the intersecting edge is the edge that connects a vertex of one group with a vertex of another group, an example of this.

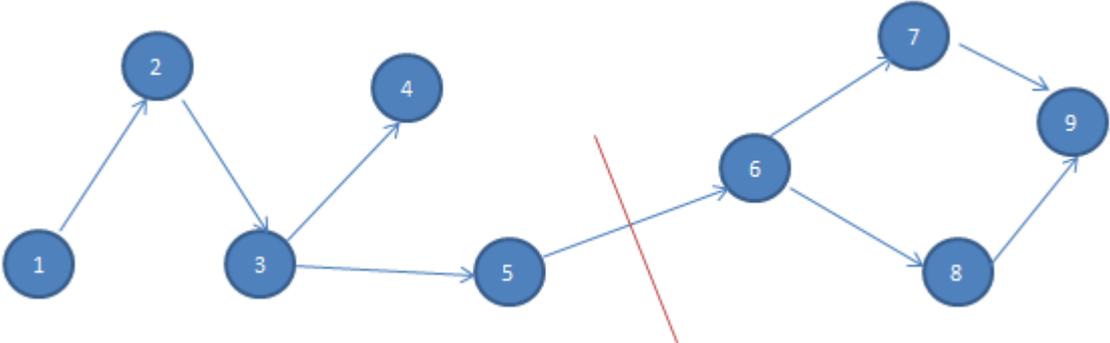


Figure 2.2: The cutting from basic principles for spanning tree

## 2.3 Find The Minimum Spanning Tree

The sub-graphs are too many, so we find it difficult to find a minimum spanning tree that can reach millions or billions of sub-trees, Larger graphs have more nodes and many other possibilities of the sub-graph as well as lengths usually the weights are different so that one edge can be given a weight of 8 and a length of 5 meters and another edge of the same length can be given 6 weights to find the minimum spanning tree and here certain algorithms are used to find the minimum spanning tree, If the graphs are of simple type , we will find all possible spanning trees in the given graph and then find the weight of each tree and then take the lowest weight which is the minimum spanning tree and denote it with the symbol (MST), for example [10] [18].

**Example 2.3.1.** From this example we will extract all possible spanning trees.

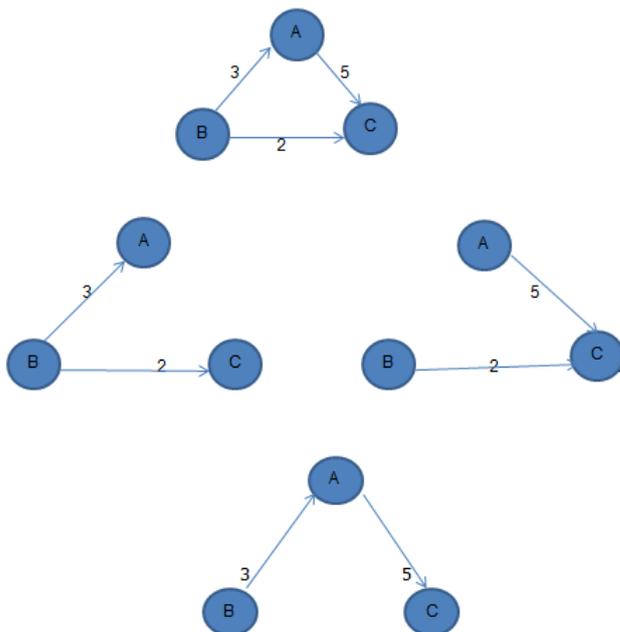


Figure 2.3: Example to find the minimum spanning tree for simple graph

If the graph is complex and contains a lot of nodes and edges, we need to use Kruskal's algorithm and Prim's algorithm to get the minimum spanning tree from the indirect graph, which are simple and intuitive strategies used to solve some optimization problems. Algorithms do the optimum test at every step because they try to find the way to the overall point to solve the entire problem and succeed in solving some problems because in many cases they do not guarantee the optimal solution.

## 2.4 Kruskal's algorithm

It is an algorithm that relies mainly on edges and is one of the well-known algorithms for finding the minimum spanning tree where the spanning tree is built by adding the edges with the least weight and gradually to the growing spanning tree with the possibility of no closed cycle within the tree as well as calling all the vertices [20] [22].

### 2.4.1 Kruskal's algorithm steps

1. We delete all the edges, leaving only the vertices
2. We keep all edges in order from least to heaviest
3. We list the edges in order to the graph from least weight to heaviest, ensuring that no cycle occurs inside the tree until we meet the stopping condition, which represents the number of vertices-1,  $(V - 1)$  [22].

**Example 2.4.1.** This example to find the MST by kruskal algorithm.

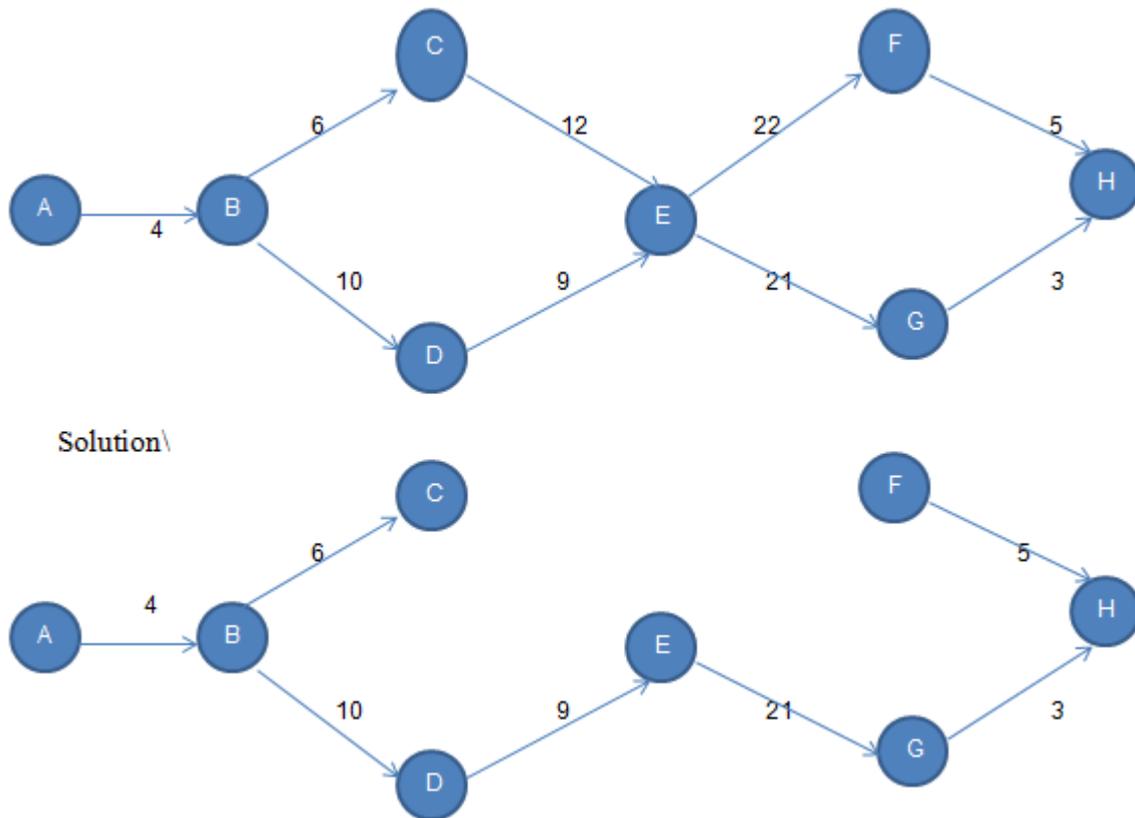


Figure 2.4: An example of Kruskal's algorithm

GH =3, AB =4, FH =5, BC =6, DE =9, BD =10,

CE =12 ×

EG=21

EF =22 ×

Vertex =8, Edge = Vertex -1 ,E=8-1=7 ,the cost of this spanning tree is 58

## 2.5 Prim Algorithm

It is an algorithm that generally depends on nodes and is one of the well-known algorithms for finding the minimum spanning tree, Where we choose any node randomly, In the spanning tree, it is considered a partial tree of the original drawing [14].

### 2.5.1 Prim Algorithm Steps

1. We randomly choose one of the nodes from the graph
2. We distinguish the edge with the least weight that is connected to the first knot chosen in the first step because this knot can contain more than one edge
3. We search for all the edges that connect the tree in the above step to the new nodes of these edges that were found and we specify the minimum edge to add to the tree and also it is indicated to repeat this step until we reach the minimum weight of the span until the spanning tree is found and until the condition of stopping is fulfilled and the number of heads -1 [26].

**Example 2.5.1.** This example to find the MST by prim algorithm.

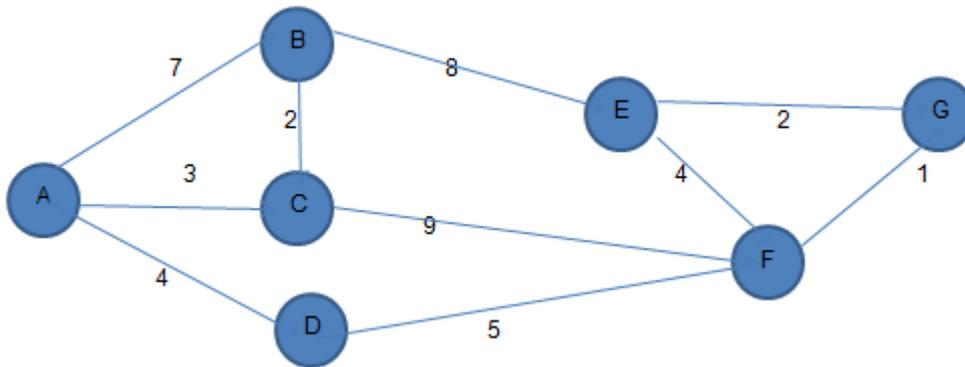


Figure 2.5: An example of Prim's algorithm

**Step 1 :**  $S=\{A\}$

$V/S=\{B, C, D, E, F, G\}$

$A=\{\}$

Lighted edge  $=\{A, C\}$



Figure 2.6: An example of prim's algorithm step 1

**Step 2:**  $S=\{A, C\}$

$V/S=\{B, D, E, F, G\}$

$A=\{A, C\}$

Lighted edge  $=\{C, B\}$

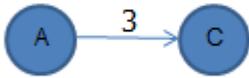


Figure 2.7: An example of prim's algorithm step 2

**Step 3:**  $S = \{A, C, B\}$

$V/S = \{D, E, F, G\}$

$A = \{A, C\}, \{C, B\}$

Lighted edge =  $\{B, E\}$

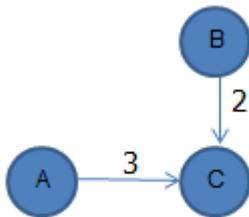


Figure 2.8: An example of prim's algorithm step 3

**Step 4 :**  $S = \{A, C, B, E\}$

$V/S = \{D, F, G\}$

$A = \{A, C\}, \{C, B\}, \{B, E\}$

Lighted edge =  $\{E, G\}$

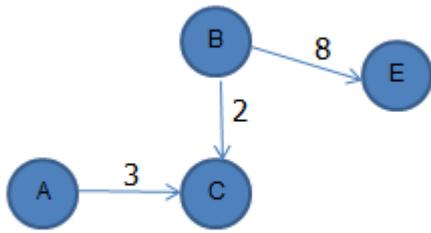


Figure 2.9: An example of prim's algorithm step 4

**Step 5 :**  $S = \{A, C, B, E, G\}$

$V/S = \{D, F\}$

$A = \{A, C\}, \{C, B\}, \{B, E\}, \{E, G\}$

Lighted edge =  $\{G, F\}$

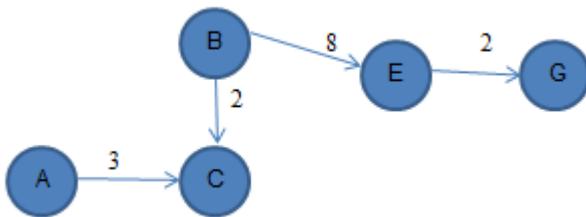


Figure 2.10: An example of prim's algorithm step 5

**Step 6 :**  $S = \{A, C, B, E, G, F\}$

$V/S = \{D\}$

$A = \{A, C\}, \{C, B\}, \{B, E\}, \{E, G\}, \{G, F\}$

Lighted edge =  $\{F, D\}$

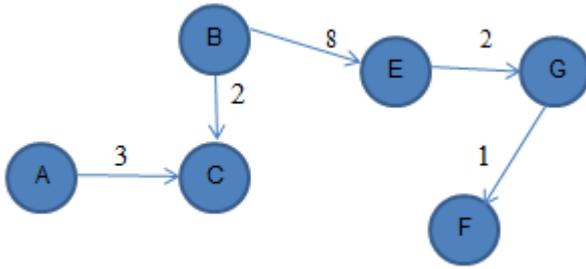


Figure 2.11: An example of prim's algorithm step 6

**Step 7 :**  $S=\{A, C, B, E, G, F, D\}$

$V/S=\{\}$

$A=\{A, C\}, \{C, B\}, \{B, E\}, \{E, G\}, \{G, F\}, \{F, D\}$

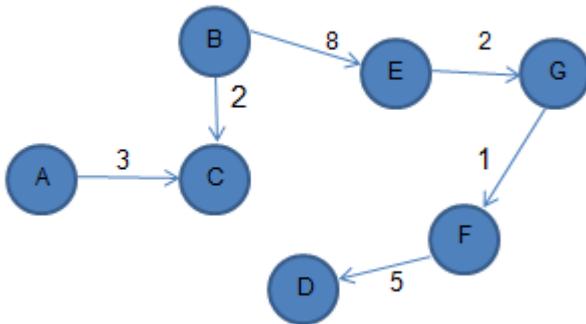


Figure 2.12: An example of prim's algorithm step 7

Cost =21

## 2.6 Suggested method: SA1

We supposed a different method from the previous algorithms, that is was obtained through kruskal algorithm or prim algorithm or similar to it, but it is better that the total cost of that tree be the same or less than the cost obtained from the previous algorithms a few of them, Thus this method is

considered a proposed method and a new idea in which we link the issue of the spanning tree with the subject of the minimum cut so that they are in the form of paths where it requires us to extract all the paths and know their weights in order to choose the path that carries the largest weight and cut off the edge with the largest weight according to the idea of the minimum cut, We continue to use these paths as needed from the largest to the smallest until we meet the conditions of the spanning tree, And we will touch on the presence of the symbol \* which indicates the path with the greatest weight from which the edge with the greatest weight was cut in this method ,We called this method  $SA_1$ .

### 2.6.1 Steps of Suggested Method $SA_1$

1. Extract all paths from the original graph.
2. We calculate the weight of each path.
3. We take the path that carries the largest weight, and we gradually take the paths as needed, in descending order from largest to smallest.
4. We cut off the edge with more weight than that path.
5. We continue with this process until the conditions of the spanning tree are met so that its edges do not form any rotation in addition to that we must reach all the vertices in the original graph and we must bear in mind that the number of edges in a spanning tree is one degree less than the number of vertices in that tree.

**Example** 2.6.1. This example to find the MST by suggested method  $SA_1$ .

Solution

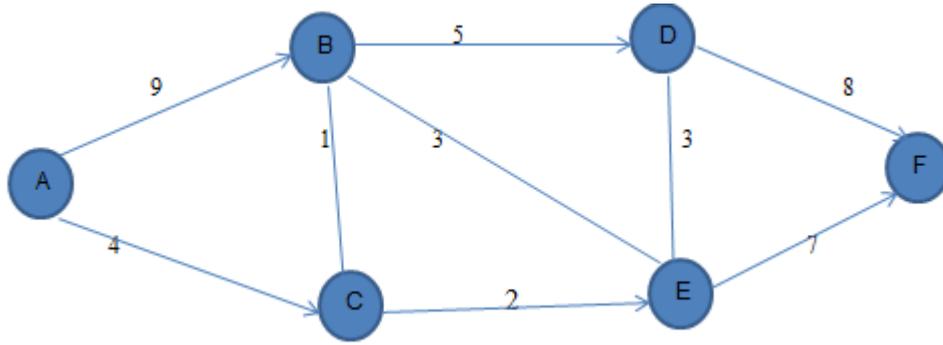


Figure 2.13: An example 2.6.1 of  $SA_1$

**path 1:** A-B-D-F ,the weight of this path is 22\*

**path 2:** A-B-E-D-F ,the weight of this path is 23\*

**path 3:** A-B-E-F ,the weight of this path is 19

**path 4:** A-B-D-E-F ,the weight of this path is 24\*

**path 5:** A-B-C-E-F ,the weight of this path is 19

**path 6:** A-B-C-E-D-F ,the weight of this path is 23\*

**path 7:** A-C-E-F ,the weight of this path is 13

**path 8:** A-C-B-D-F ,the weight of this path is 18

**path 9:** A-C-B-E-D-F ,the weight of this path is 19

**path 10:** A-C-B-E-F,the weight of this path is 15

**path 11:** A-C-E-D-F,the weight of this path is 17

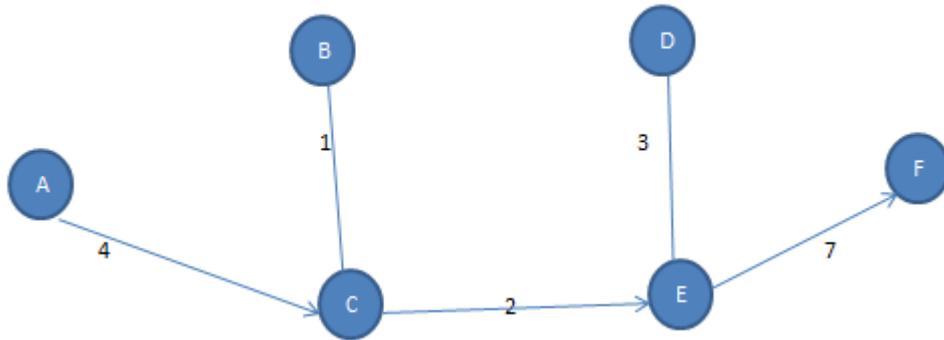


Figure 2.14: Solution an example 2.6.1 of  $SA_1$

Cost =17

**Example 2.6.2.** This example to find the MST by suggested method  $SA_1$ .

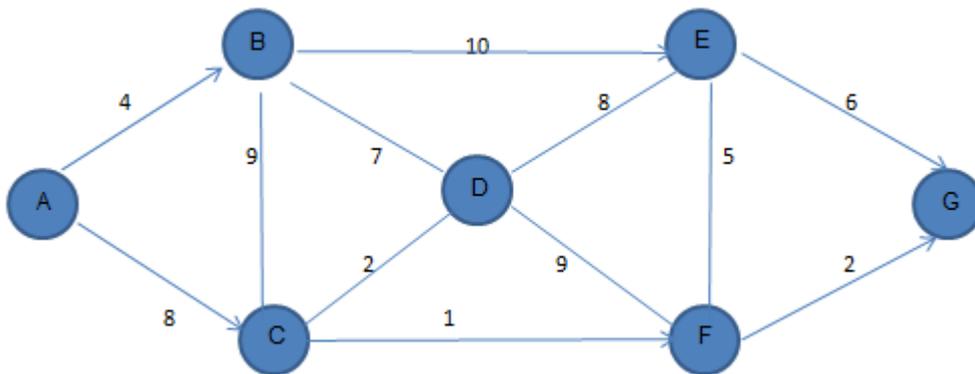


Figure 2.15: An example 2.6.2 of  $SA_1$

Solution

**path 1:** A-B-E-G ,the weight of this path is 20

**path 2:** A-B-D-E-G ,the weight of this path is 25

**path 3:** A-B-D-F-G ,the weight of this path is 22

**path 4:** A-B-D-F-E-G ,the weight of this path is 31

**path 5:** A-B-D-E-F-G ,the weight of this path is 26

**path 6:** A-B-C-F-G ,the weight of this path is 16

**path 7:** B-C-D-F-G ,the weight of this path is 26

**path 8:** A-B-C-D-F-E-G ,the weight of this path is 35\*

**path 9:** A-B-C-D-E-F-G ,the weight of this path is 30

**path 10:** A-B-C-D-E-G ,the weight of this path is 29

**path 11:** A-C-F-G ,the weight of this path is 11

**path 12:** A-C-D-E-F-G ,the weight of this path is 21

**path 13:** A-C-D-E-G ,the weight of this path is 24

**path 14:** A-C-D-F-E-G ,the weight of this path is 30

**path 15:** A-C-D-F-G ,the weight of this path is 21

**path 16:** A-C-B-D-E-F-G ,the weight of this path is 39\*

**path 17:** A-C-B-D-E-G ,the weight of this path is 38\*

**path 18:** A-C-B-D-F-E-G ,the weight of this path is 44\*

**path 19:** A-C-B-D-F-G ,the weight of this path is 35\*

**path 20:** A-C-B-E-F-G ,the weight of this path is 34\*

**path 21:** A-C-B-E-G ,the weight of this path is 33

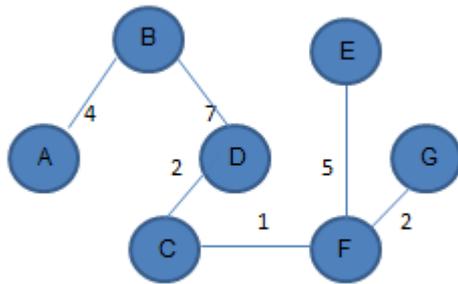


Figure 2.16: Solution an example 2.6.2 of  $SA_1$

Cost =21

**Example** 2.6.3. This example to find the MST by suggested method  $SA_1$ ( same example 2.4.1).

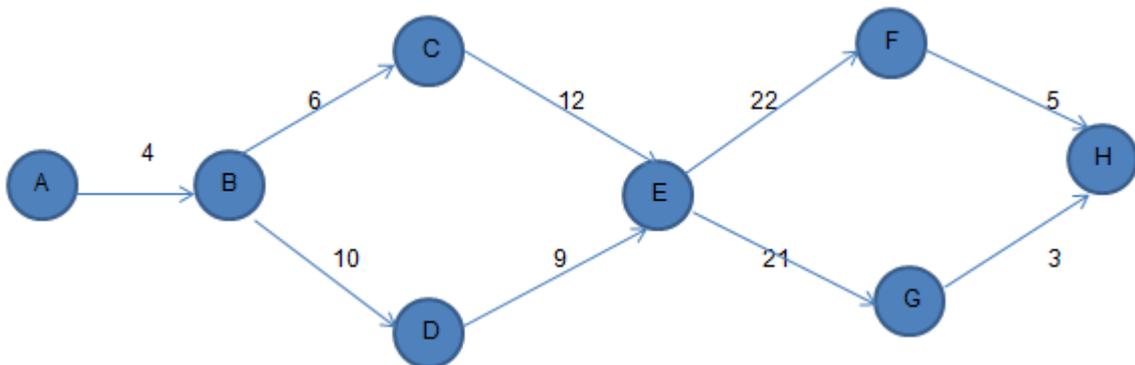


Figure 2.17: An example 2.6.3 of  $SA_1$

Solution

**path 1:** A-B-C-E-F-H ,the weight of this path is 49\*

**path 2:** A-B-C-E-G-H ,the weight of this path is 46

**path 3:** A-B-D-E-F-H ,the weight of this path is 50\*

**path 4:** A-B-D-E-G-H ,the weight of this path is 47

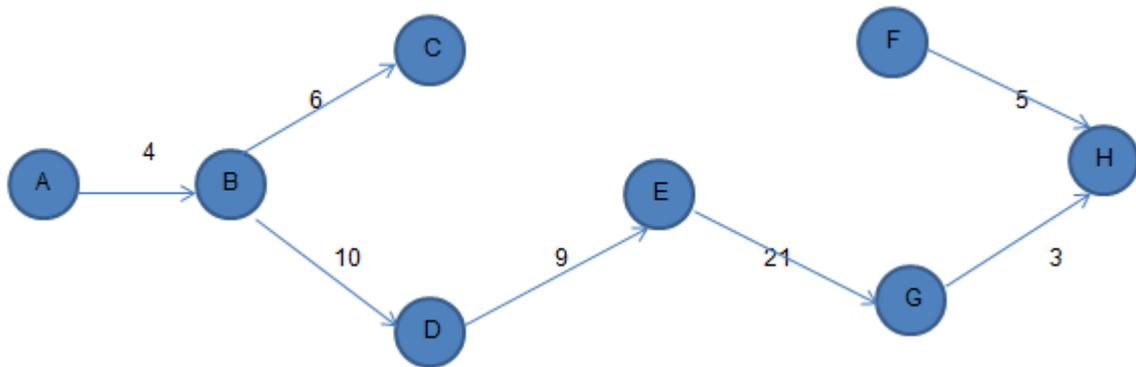


Figure 2.18: Solution an example 2.6.3 of  $SA_1$

Cost =58

**Example 2.6.4.** This example to find the MST by suggested method  $SA_1$ .

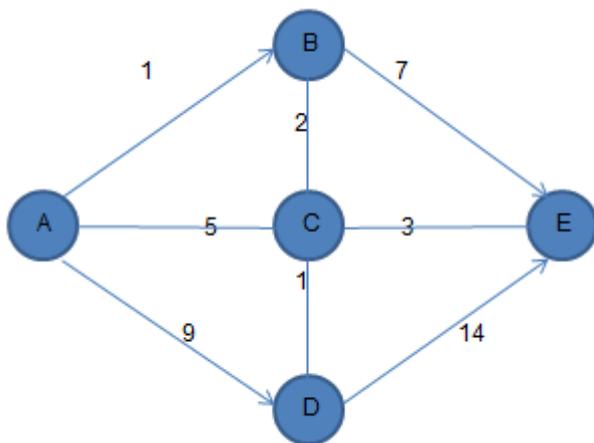


Figure 2.19: An example 2.6.4 of  $SA_1$

Solution

**path 1:** A-B-E ,the weight of this path is 8

**path 2:** A-B-C-E ,the weight of this path is 6

**path 3:** A-C-E ,the weight of this path is 8

**path 4:** A-D-C-E ,the weight of this path is 13

**path 5:** A-D-E ,the weight of this path is 23\*

**path 6:** A-B-C-D-E,the weight of this path is 18\*

**path 7:** A-C-B-E,the weight of this path is 14

**path 8:** A-C-D-E,the weight of this path is 20\*

**path 9:** A-D-C-B-E,the weight of this path is 19\*

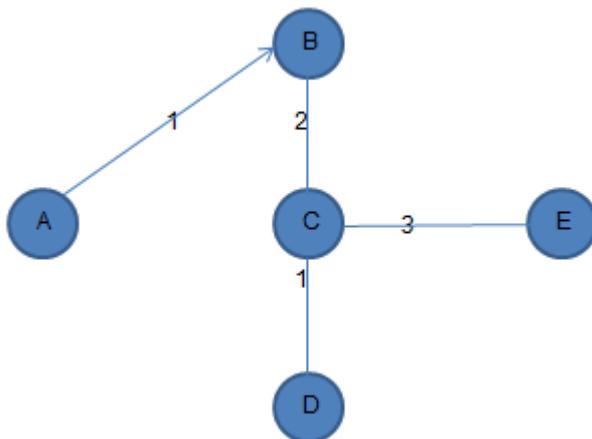


Figure 2.20: Solution an example 2.6.4 of  $SA_1$

Cost =7

**Example** 2.6.5. This example to find the MST by suggested method  $SA_1$ .

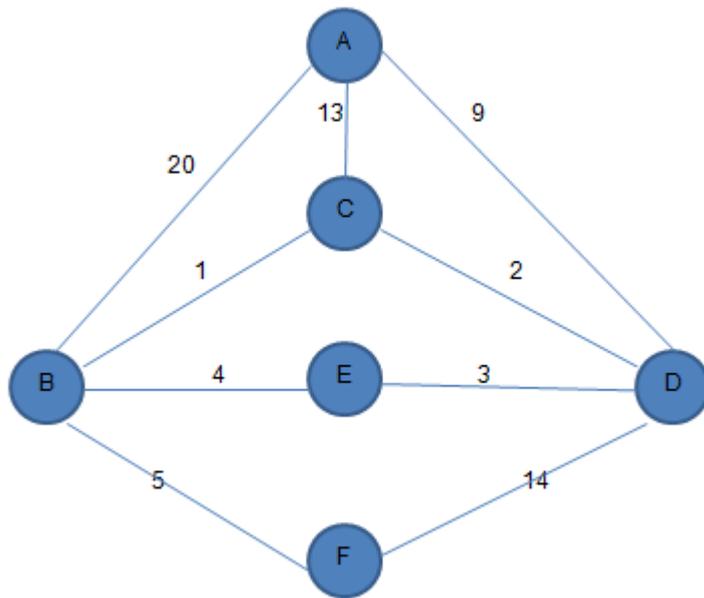


Figure 2.21: An example 2.6.5 of  $SA_1$

Solution

**path 1:** B-A-D ,the weight of this path is 29\*

**path 2:** B-A-C-D ,the weight of this path is 35\*

**path 3:** B-C-D ,the weight of this path is 3

**path 4:** B-E-D ,the weight of this path is 7\*

**path 5:** B-F-D ,the weight of this path is 19\*

**path 6:** B-C-A-D,the weight of this path is 23

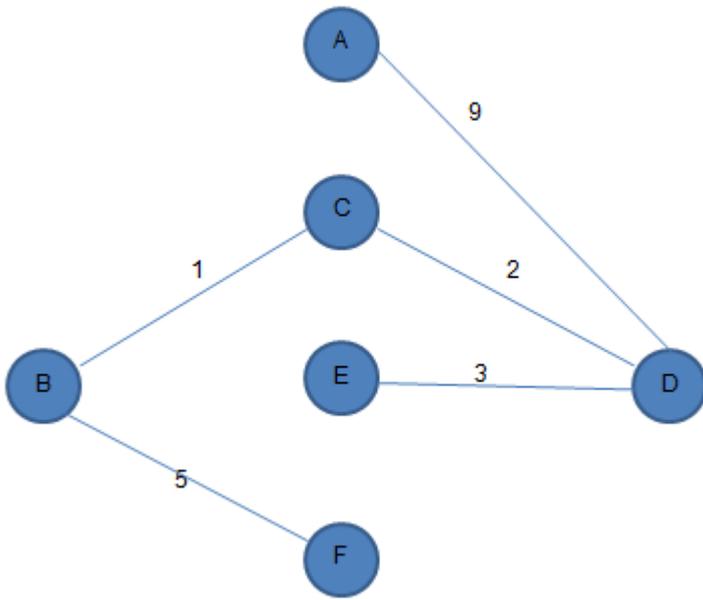


Figure 2.22: Solution an example 2.6.5 of  $SA_1$

Cost =20

**Example** 2.6.6. This example to find the MST by suggested method  $SA_1$ .

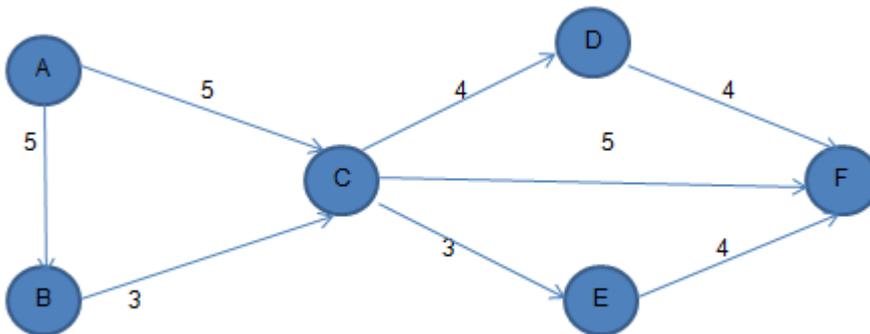


Figure 2.23: An example 2.6.6 of  $SA_1$

Solution

**path 1:** A-C-D-F ,the weight of this path is 13

**path 2:** A-C-F ,the weight of this path is 10

**path 3:** A-C-E-F ,the weight of this path is 12

**path 4:** A-B-C-D-F ,the weight of this path is 16\*

**path 5:** A-B-C-F ,the weight of this path is 13\*

**path 6:** A-B-C-E-F ,the weight of this path is 15\*

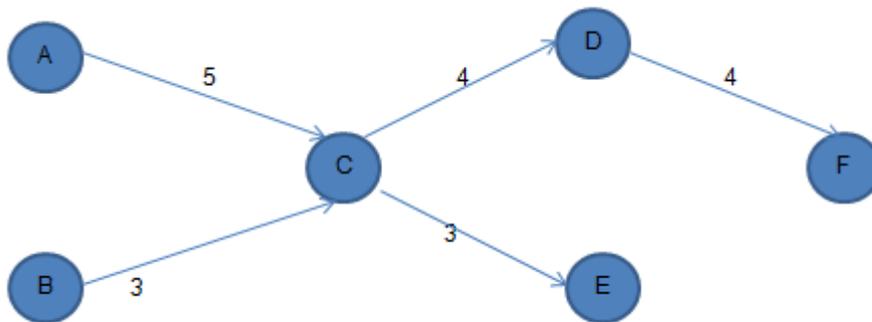


Figure 2.24: Solution an example 2.6.6 of  $SA_1$

Cost =19

**Example** 2.6.7. This example to find the MST by suggested method  $SA_1$ ( same figure 2.5.).

Solution

**path 1:** A-B-E-G ,the weight of this path is 17

**path 2:** A-B-C-F -G ,the weight of this path is 19\*

**path 3:** A-B-C-F-E-G ,the weight of this path is 24\*

**path 4:** A-B-E-F-G ,the weight of this path is 20\*

**path 5:** A-C-B-E-F-G ,the weight of this path is 18\*

**path 6:** A-C-B-E-G ,the weight of this path is 15

**path 7:** A-C-F-E-G ,the weight of this path is 18

**path 8:** A-C-F-G ,the weight of this path is 13

**path 9:** A-D-F-G ,the weight of this path is 10

**path 10:** A-D-F-E-G ,the weight of this path is 15

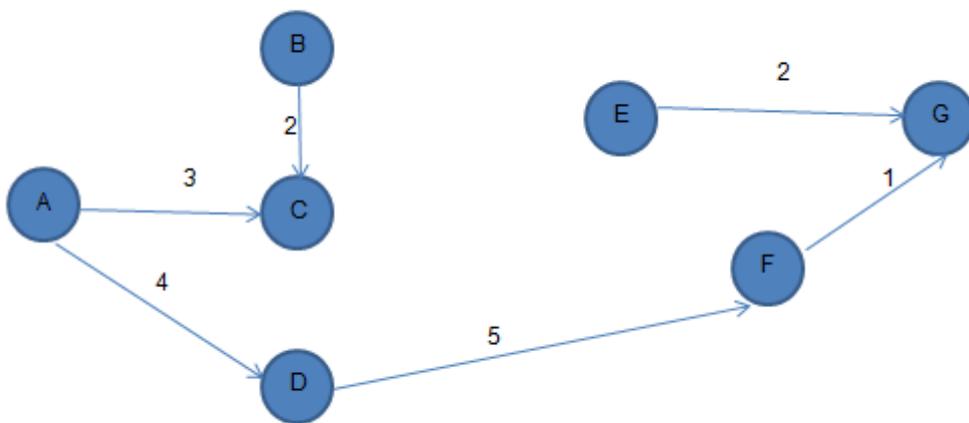


Figure 2.25: Solution an example 2.6.7 of  $SA_1$

Cost =17

**Example** 2.6.8. This example to find the MST by suggested method  $SA_1$ .

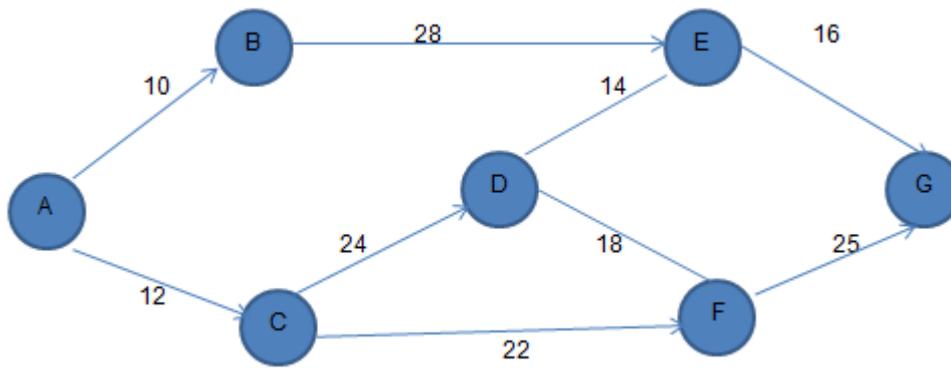


Figure 2.26: An example 2.6.8 of  $SA_1$

Solution

**path 1:** A-B-E-G ,the weight of this path is 54

**path 2:** A-B-E-D-F-G,the weight of this path is 95\*

**path 3:** A-C-D-F-G,the weight of this path is 79\*

**path 4:** A-C-F-G ,the weight of this path is 59

**path 5:** A-C-D-E-G ,the weight of this path is 66\*

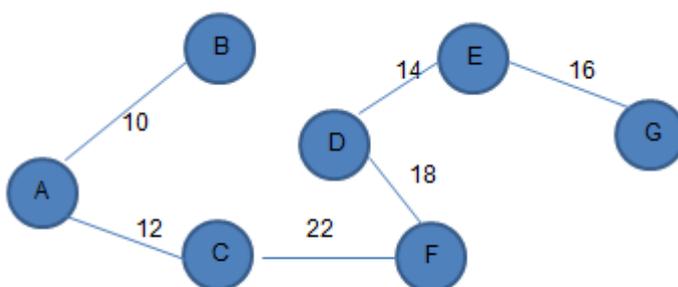


Figure 2.27: solution An example 2.6.8 of  $SA_1$

Cost =92

## 2.7 Results

A comparison between the results of the previous methods (Kruskal's algorithm, Prime's algorithm) and the Suggested Method

Examples	Results of Kruskal's algorithm	Results of Prime's algorithm	Results of Suggested Method $SA_1$
Example 2.6.1	17	19	17
Example 2.6.2	21	22	21
Example 2.6.3	58	60	58
Example 2.6.4	7	18	7
Example 2.6.5	20	21	20
Example 2.6.6	19	19	19
Example 2.6.7	17	21	17
Example 2.6.8	92	92	92

Table 2.1: The results of the previous examples and suggested method

The results of the new suggested method are equal or slightly less than the results of the previous methods, so the cost has become less.

## CHAPTER 3

# IMPROVEMENT MAXIMUM FLOW AND MINIMUM CUT

### 3.1 Maximum Flow Problem

. In order to explain this problem in a simplified way, we will express the point in the drawing as a city and about the edges as the road that accommodates a certain number of cars that want to pass from the starting point to the target point so that we can take the largest number possible cars that can pass on one road [4] [16].

#### 3.1.1 Steps of Maximum Flow Problem

1. The weight on edge represents maximum of cars to pass on this road.
2. Send maximum number of cars from first edge to second edge.
3. Find set of paths to send cars through them from start to end.
4. The flow will be from the source point to the target point.

For example, to find the maximum flow for the following graph [4] [16]

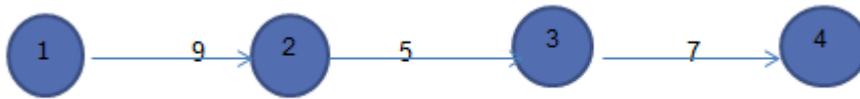


Figure 3.1: Example of max flow

**We can send on  $1-2 = 9$  cars**

**We can send on  $2-3 = 5$  cars**

**We can send on  $3-4 = 7$  cars**

But edge 2-3 is bottleneck as if we sent from  $1-2 = 9$  cars we can't pass more than 5 so max flow of path is the minimum value on it.

## 3.2 Minimum Cut Problem

We will explain this problem by giving an example from the real life and linking it to the topic directly, for example, in a war, there is a network of roads and the enemy delivers weapons through these roads to other areas in order to address this problem needs to destroy these roads and it's better to destroy the road that has the least capacity for cars to pass because we are looking for the lowest cost, cutting one of these roads prevents weapons from reaching to other areas, we can destroy all the roads or more of them, but it requires more cost, so we resort to less destruction to number of ways until the enemy is unable to deliver weapons to other ares [30] [12].

### 3.2.1 Steps of Minimum Cut Problem

1. we are in a war we would like to destroy the road to prevent city (A) to send weapons to city (B)
2. Edges costs is cost of destroying a road.
3. The minimum edges to achieve the task are min cut edges. for example, to find the minimum cut( same figure 3.1) [3] [5]

We can destroy all edges but this is very costive, as well as we have just 1 path make sense to destroy 1 edge, this edge is the minimum edge  $e = 5$ . (There is a theory that relates maximum flow to minimum cut-off, which is) maximum flow - minimum cut-off

The maximum flow through any network from a given source to a given sink is exactly equal to the minimum sum of a cut .

### 3.3 Improvement Maximum Flow

We have previously mentioned the maximum flow and explained that the points in the drawing represent the cities and the edges represent the streets leading to the cities, we suggest that improve the maximum flow by choosing the road that accommodates greater number of flow and making the rest of the roads with the same capacity for that larger road so that it can pass the maximum number flow in each path and the same amplitude, we call this  $SA_3$ , as in the following examples.

**Example** 3.3.1. This example to Find the maximum flow by  $SA_3$  (same figure 3.1).

**We can send on  $(1-2) = 9$  cars**

We can send on  $(2-3) = 5$  cars

We can send on  $(3-4) = 7$  cars

But edge  $(1-2)$  is the largest value of the path, we must make all the paths the same value for the largest flow:  $(2-3) = 5$  cars,  $5+4 = 9$  cars,  $(4-3) = 7$  cars,  $7+2 = 9$  cars, the maximum flow is 9.

**Example 3.3.2.** This example to Find the maximum flow by  $SA_3$  .

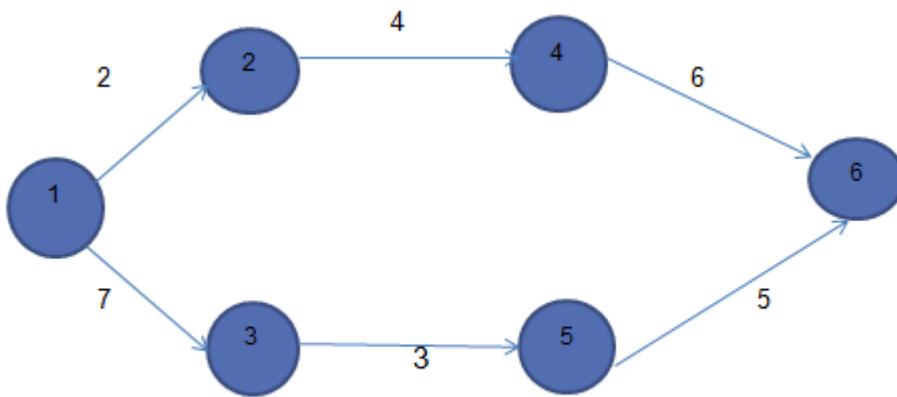


Figure 3.2: Example 3.3.2 of  $SA_3$

**Path 1:** 1- 2- 4- 6 ,the edge  $(4- 6) = 6$  is the largest value, we must make all the path the same value for the largest flow:  $(1- 2) = 2+4 = 6$  ,  $(2- 4) = 4+2 = 6$

**Path 2:** 1- 3- 5- 6 ,the edge  $(1- 3) = 7$  is the largest value, we must make all the path the same value for the largest flow  $(3- 5) = 3+4 = 7$  ,  $(5- 6) = 5+2 = 7$  , the maximum flow is  $7+6 = 13$

**Example 3.3.3.** This example to Find the maximum flow by  $SA_3$  .

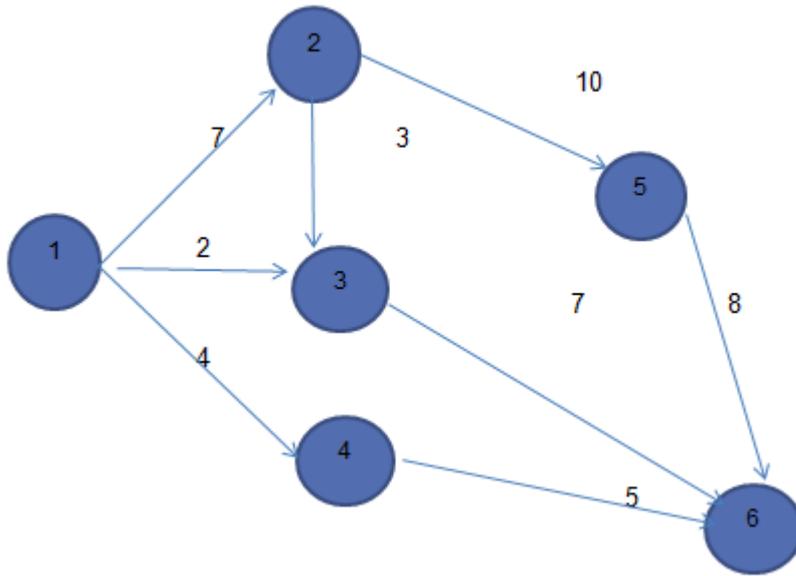


Figure 3.3: Example 3.3.3 of  $SA_3$

**Path 1:** 1- 2- 5- 6 ,the edge (2- 5) is the largest value, we must make all the edges of the path same value for the largest flow.  $(1- 2) = 7+3 = 10$  ,  $(5- 6) = 8+2 = 10$  ,

**Path 2:** 1- 2- 3- 6 ,the edges (1- 2) and (3- 6) are the largest values of this path, we must make edge (2- 3) same value for largest flow:  $(2- 3) = 3+4 = 7$ .

**Path 3:** 1- 3- 6 ,the edge (3- 6) is the largest value of the path, we must make edge (1- 3) same value for largest flow:  $(1- 3) = 2+5 = 7$ .

**Path 4:** 1- 4- 6 ,the edge (4- 6) is the largest value of the path:  $(1- 4) = 4+1= 5$  the maximum flow is  $10 +7+7+5 = 29$ .

**Example 3.3.4.** This example to Find the maximum flow by  $SA_3$  .

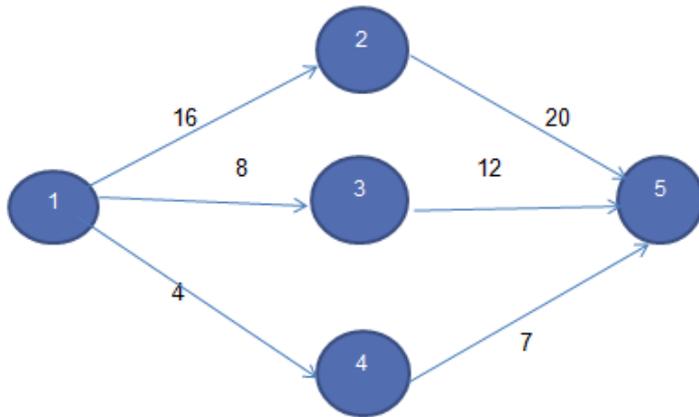


Figure 3.4: Example 3.3.4 of  $SA_3$

**Path 1:** 1- 2- 5 ,the edge (2- 5) is the largest value of the path then the edge (1- 2) =  $16+4 = 20$

**Path 2:** 1- 3- 5 ,the edge (3- 5) is the largest value of the path then the edge (1- 3) =  $8+4 = 12$

**Path 3:** 1- 4- 5 ,the edge (4- 5) is the largest value of the path then the edge (1- 4) =  $4+3 = 7$  the maximum flow is  $20+12+7= 39$ .

**Example 3.3.5.** This example to Find the maximum flow by  $SA_3$  .

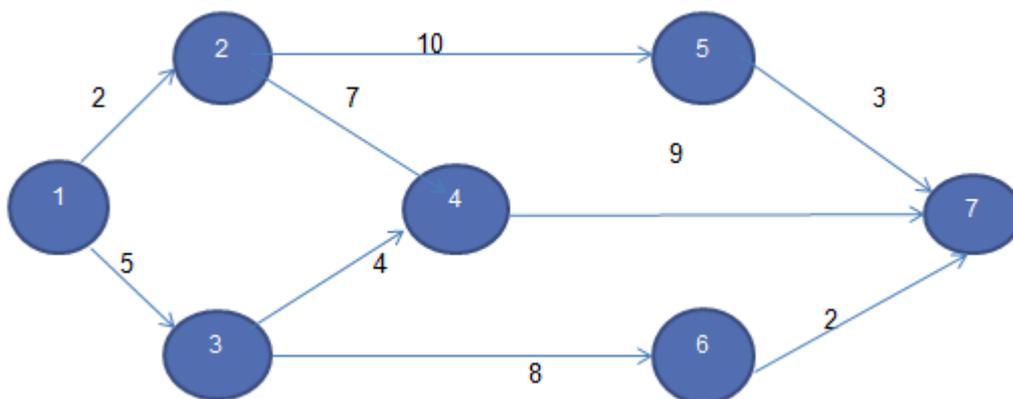


Figure 3.5: Example 3.3.5 of  $SA_3$

**Path 1:** 1- 2- 5- 7 ,the edge (2- 5) is the largest value of the path then the edge (1- 2) =  $2+8 = 10$  and (5- 7) =  $3+7 = 10$ .

**Path 2:** 1- 2- 4- 7 ,the edge (4- 7) is the largest value of the path then the edge (1- 2) =  $2+7= 9$  and (2- 4) = $7+2 = 9$ .

**Path 3:** 1- 3- 4- 7 ,the edge (4- 7) is the largest value of the path then the edge (1- 3) = $5+4 = 9$  and (3- 4) =  $4+5 = 9$ .

**Path 4:** 1- 3- 6- 7 ,the edge (3- 6) is the largest value of the path then the edge (1- 3) =  $5+3 = 8$  and (6- 7) =  $2+6 = 8$  the maximum flow is  $10+9+9+8 = 36$ .

### 3.4 Improvement Minimum Cut

We have previously mentioned the problem of minimum cutting and this problem has been clarified through an example from real life which is the example of war and how to deliver equipment and weapons from one area to another through a network of roads and cutting one of these roads leads to the failure of the set plan by the enemy, and we can suggest an improvement for this problem, which is that instead of cutting the edge with the least capacity we cut the edge that, accommodates the most capacity in order to ensure more safety and to prevent the enemy from reaching its goal provided that the cost is increased to small extent, we call this  $SA_4$ , for examples:

**Example 3.4.1.** This example to Find the minimum cut of the following graph by  $SA_4$  , (same figure 3.1).

**We can send on (1- 2) = 9 cars**

**We can send on (2- 3) = 5 cars**

We can send on  $(3- 4) = 7$  cars

.

We must cross the edge with the largest capacity in order to increase the chance of the enemy not reaching the other side. we could destroy the edge (1- 2) with largest capacity, min cut = 9 = max flow of example 3.3.1.

*Example 3.4.2.* This example to Find the minimum cut by  $SA_4$  ,(same figure 3.2).

.

**Path1:** 1- 2- 4- 6 ,we must cross the edge with the largest capacity in order to increase the chance of the enemy not reaching the other side. we can destroy the edge (4- 6) with largest capacity in this path.

**Path 2:** 1- 3- 5- 6 ,we could destroy the edge (1- 3) with largest capacity in this path. min cut is  $6+7= 13 = \text{max flow}$  of example 3.3.2.

.

*Example 3.4.3.* This example to Find the minimum cut by  $SA_4$  , (same figure 3.3).

.

**Path1:** 1- 2- 5- 6 ,we must cross the edge with the largest capacity in order to increase the chance of the enemy not reaching the other side. we could destroy the edge (2- 5) with largest capacity in this path.

**Path 2:** 1- 2- 3- 6 ,we could destroy the edge (1- 2) or (3- 6) with largest capacity in this path.

**Path 3:** 1- 3- 6 ,we could destroy the edge (3- 6) with largest capacity in this path.

**Path 4:** 1- 4- 6 ,we could destroy the edge (4- 6) with largest capacity in this path. min cut is  $10+7+7+5 = 29 = \text{max flow}$  of example 3.3.3.

*Example* 3.4.4. This example to Find the minimum cut by  $SA_4$  , (same figure 3.4).

**Path1:** 1- 2- 5 ,we could destroy the edge (2- 5) with largest capacity in this path.

**Path 2:** 1- 3- 5 ,we could destroy the edge (3- 5) with largest capacity in this path.

**Path 3:** 1- 4- 5 ,we could destroy the edge (4- 5) with largest capacity in this path. min cut is  $20+12+7 = 39 = \text{max flow}$  of example 3.3.4.

*Example* 3.4.5. This example to Find the minimum cut by  $SA_4$  , (same figure 3.5).

**Path1:** 1- 2- 5- 7 ,we could destroy the edge (2- 5) with largest capacity.

**Path 2:** 1- 2- 4- 7 ,we could destroy the edge (4- 7) with largest capacity.

**Path 3:** 1- 3- 4- 7 ,we could destroy the edge (4- 7) with largest capacity.

**Path 4:** 1- 3- 6- 7 ,we could destroy the edge (3- 6) with largest capacity.  
the minimum cut is  $10+9+9+8 = 36 = \text{max flow}$  of example 3.3.5.

### 3.5 Results

We will notice in the table below the results of the maximum flow and minimum cut-off are equal.

Examples	max flow	min cut	Improvement max flow SA <sub>3</sub>	Improvement min cut SA <sub>4</sub>
Example 3.3.1	5	5	9	9 (Example 3.4.1)
Example 3.3.2	7	7	13	13 (Example 3.4.2)
Example 3.3.3	16	16	29	29 (Example 3.4.3)
Example 3.3.4	18	18	39	39 (Example 3.4.4)
Example 3.3.5	10	10	36	36 (Example 3.4.5)

Table 3.1: The results of the previous examples

We note that the results of optimizing the maximum flow and improving the minimum cut-off are equal, in order to achieve the theory that links them together.

## CHAPTER 4

# IMPROVEMENT TIME ESTIMATE OF NETWORK OPTIMIZATION

### 4.1 Project Network

Is a diagram linking the activities and duration allocated to complete this project any that displays the order in which must be completed by the activities of the project and to identify the network plan for the project and the duty of the project manager and any member of the team because it helps them to visualize business and activities that must be completed during the duration of the project ,so that the chronology of the project reflects and is a key step towards building the agenda of any project viable is to regulate the activities where represents the tail of the arrow the beginning of the activity while ultimately edged him represent the end of the activity and point in order to sequentially activities are scheduled to use charts stocks in the planning stage for the management of any project where they are represented by shares to be that project deadlines agreed upon or less than the original duration of the appointment [5] [27] [15] [11].

# 4.2 The Importance of the project network

The main importance of creating a schedule and an arrow chart for a particular project is not only to organize the tasks and know when they must be implemented, but also to know which of the tasks must be completed and which of them can be overcome while maintaining the achievement of the goals and objectives of the project [11] [27] [23].

*Example* 4.2.1. Construct the network diagram for project with the following activities.

Activity	Immediate predecessor activity
A	-
B	-
C	A
D	B
E	A
F	C,D
G	E
H	E
I	F,G
J	H,I

Table 4.1: Example 4.2.1 of project network

**Solution** The two activities A,B do not have a previous event, that is, they are the first in the network.

An activity A is an event that precedes the activity C

An activity B is an event that precedes the activity D

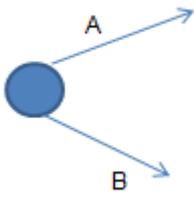


Figure 4.1: Solution of example 4.2.1 for project network step 1

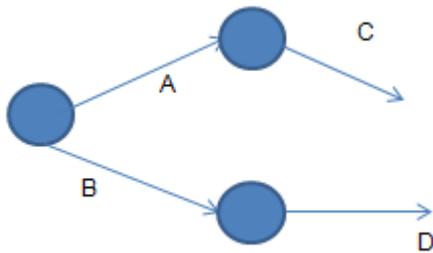


Figure 4.2: solution of Example 4.2.1 for project network step 2

An activity A is an event that precedes the activity E

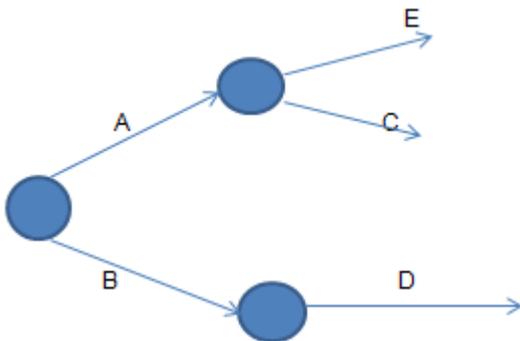


Figure 4.3: solution of Example 4.2.1 for project network step 3

The two activities C,D are preceding the activity F

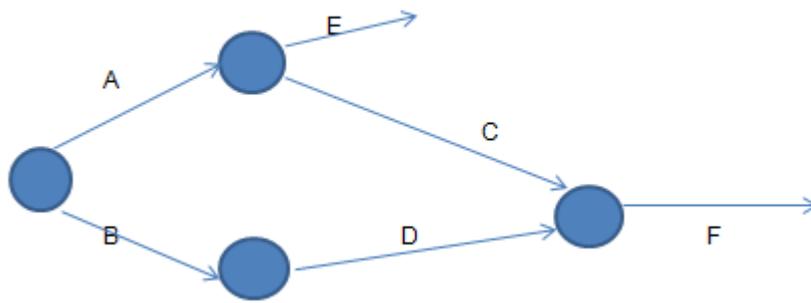


Figure 4.4: solution of Example 4.2.1 for project network step 4

An activity E is an event that precedes the activity H,G

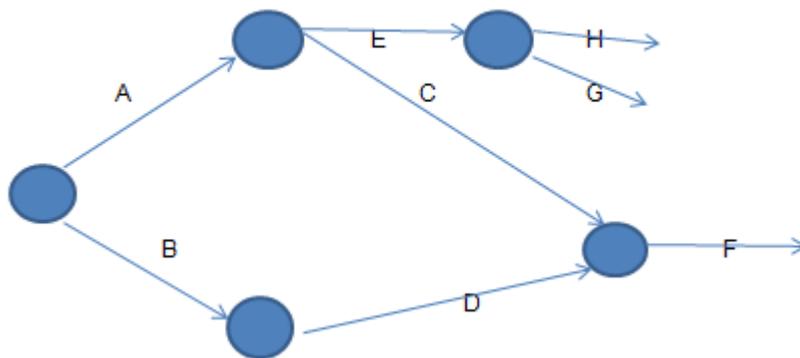


Figure 4.5: solution of Example 4.2.1 for project network step 5

The two activities F,G are preceding the activity I

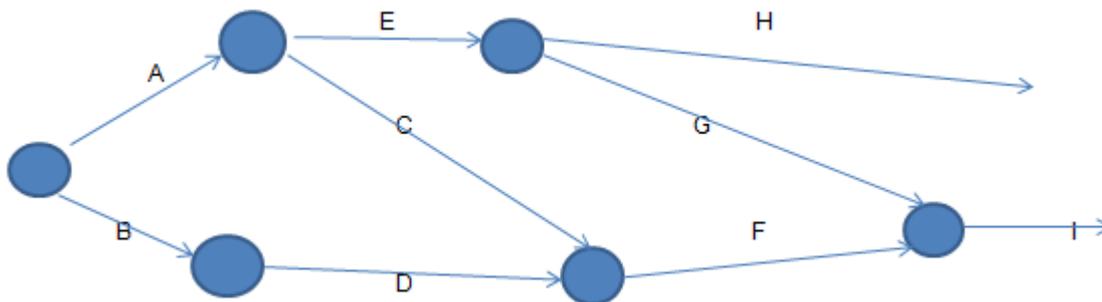


Figure 4.6: solution of Example 4.2.1 for project network step 6

The two activities I,H are preceding the activity J

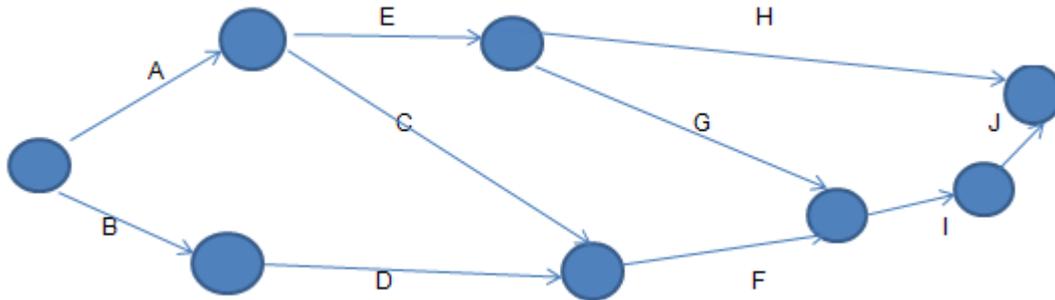


Figure 4.7: solution of Example 4.2.1 for project network step 7

**Example 4.2.2.** Develop a network diagram for the project specified below.

Activity	Immediate predecessor activity
A	-
B	A
C,D	B
E	C
F	D
G	E,F

Table 4.2: Example 4.2.2 of project network

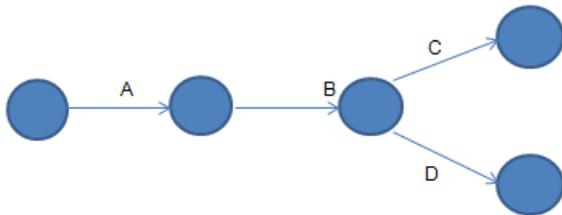
**Solution** Activity A does not have any past event, it is the first activity.



An activity A is an event that precedes the activity B

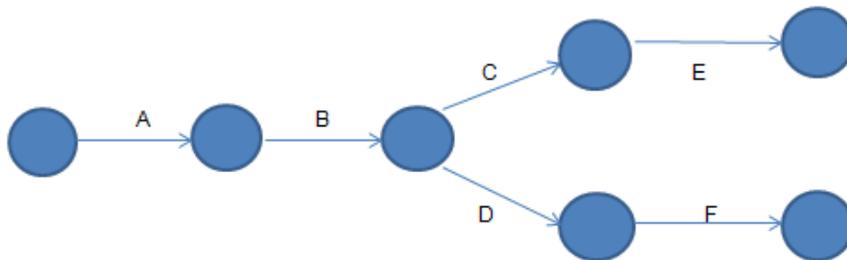


An activity B is an event that precedes the two activities C,D



An activity C is an event that precedes the activity E

An activity D is an event that precedes the activity F



the two activities F,E are previous events of the activity G

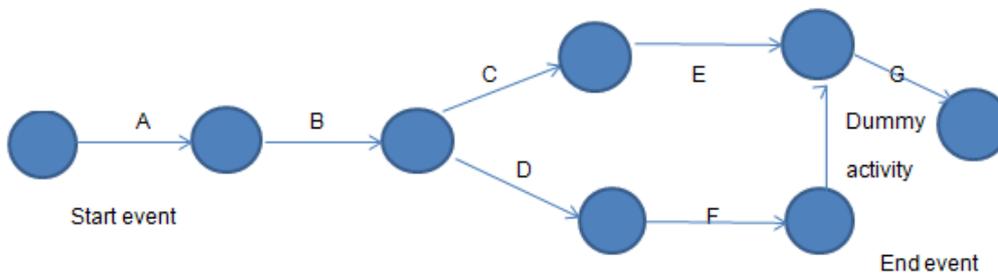


Figure 4.8: Solution of example 4.2.2 for project network step 1,2,3

Figure 4.9: Solution of example 4.2.2 for project network step 4,5

**Example 4.2.3.** Develop a network diagram for the project specified below.

activity	Immediate predecessor activity
A	-
B	A
C	A
D	A
E	B
F	C
G	D,E,F

Table 4.3: Example 4.2.3 of project network

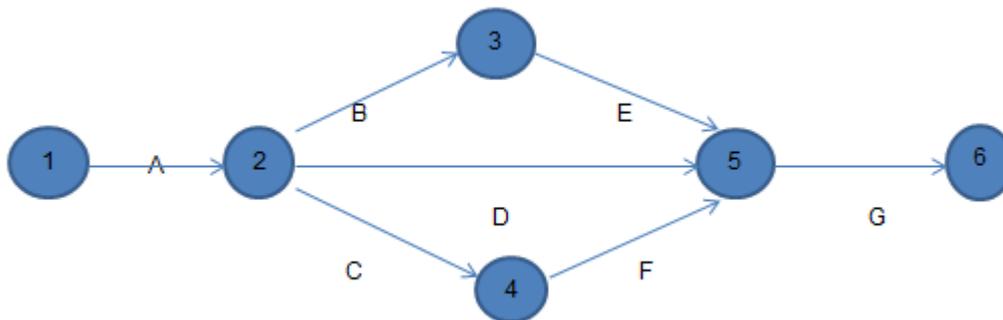


Figure 4.10: Solution example 4.2.3 of project network

### 4.3 Time Estimate

Basic preparations to start work on any project should be an estimate of the time of termination of this project and it is by the administrators of recipients of the project, where one of them a short time of optimism is assumed on the assumption that there will be no problems during the work and then another pessimistic time suggests assuming that there is a problems unexpected during the period of work and thus will consume a lot

of time and in fact, what utter the mind is the estimated time between optimistic and pessimistic, and is the so-called time-usual is likely and obviously that estimate the time optimistic less than the estimated time normal and is in turn, less than the estimated time pessimistic and former researchers have an estimate of the most likely time estimated for any project where they assumed in practice it is likely not withstand appreciation of the pessimistic or optimistic so they decided to give more weight to estimate the time likely of 4 to estimate the likely time for distributors while time estimates pessimistic and optimistic weight of 1 each [19] [5] [6] [11].

therefore, we note that there are three types of time estimation:

1. Pessimistic time estimate
2. Most likely time estimate
3. Optimistic time estimate

The mathematical law is  $t_e = (t_o + 4t_m + t_p) / 6$

Such that

$t_e$  : Time Estimate

$t_o$  : Optimistic Time Estimate

$t_m$  : Most likely time Estimate

$t_p$  : Pessimistic time Estimate

**Example 4.3.1.** To complete the next project we must find the time required and then find the critical path and critical activities.

Activity	Optimistic time estimate $t_o$	Most likely time estimate $t_m$	Pessimistic time estimate $t_p$
1-2	2	5	8
2-3	8	11	20
3-4	2	4	6
2-4	4	7	10
2-5	4	8	12
4-6	6	10	14
5-6	3	5	7
3-6	4	9	14

Solution\From the three time estimate  $t_p, t_m$  and  $t_o$  calculate  $t_e$  for each activity we obtain the following table.

Table 4.4: Example 4.3.1 of time estimate

Activity	Optimistic time estimate $t_o$	4*Most likely time estimate $t_m$	Pessimistic time Estimate $t_p$	time estimate $t_e=(t_o + 4t_m + t_p)/6$
1-2	2	5(4)	8	5
2-3	8	11(4)	20	12
3-4	2	4(4)	6	4
2-4	4	7(4)	10	7
2-5	4	8(4)	12	8
4-6	6	10(4)	14	10
5-6	3	5(4)	7	5
3-6	4	9(4)	14	9

Table 4.5: solution Example 4.3.1 of time estimate

Using the single time estimate of the activities we get the following network diagram for the project.

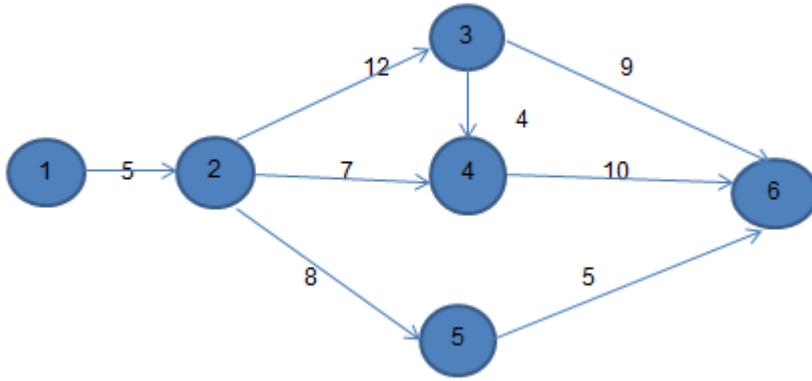


Figure 4.11: Solution example 4.3.1 of time estimate

Consider the path starting at the starting point and stopping at the end point.

**path 1:** path 1:1-2-3-6.

time of the path  $5+12+9=26$

**path 2:** path 2: 1-2-3-4-6.

time of the path  $5+12+4+10=31$

**path 3:** path 3: 1-2-4-6.

time of the path  $5+7+10=22$

**path 4:** path 4:1-2-5-6.

time of the path  $5+8+5=18$

Compare the times for the four path maximum of  $\{26, 31, 22, 18\}=31$ , we see that path two has the maximum time there for it is the critical path

and the critical activities are 1-2, 2-3, 3-4, 4-6 the rest of the edges are non-critical activities , such that project time 31 weeks.

**Example 4.3.2.** A project consists of seven activities with the following time estimate to find the probability that the project will be completed in 30 weeks or less.

Activity	Predecessor activity	Optimistic time estimate( $t_o$ day)	Most likely time estimate( $t_m$ day)	Pessimistic time estimate( $t_p$ day)
A	-	2	5	8
B	A	2	3	4
C	A	6	8	10
D	A	2	4	6
E	B	2	6	10
F	C	6	7	8
G	D,E,F	6	8	10

Table 4.6: Example 4.3.2 of time estimate

From the three time estimate

$[t_o]$  Optimistic Time Estimate,

$[t_m]$  Most likely time estimate,

$[t_p]$  Pessimistic time estimate,

calculatet<sub>e</sub>: each activity we obtain the following table.

Activity	Optimistic time estimate $t_o$	4*Most likely time estimate $t_m$	Pessimistic time Estimate $t_p$	Time estimate $t_e=(t_o + 4t_m + t_p)/6$
A	2	20	8	5
B	2	12	4	3
C	6	32	10	8
D	2	16	6	4
E	2	24	10	6
F	6	28	8	7
G	6	32	10	8

Table 4.7: Solution example 4.3.2 of time estimate

with the single time estimate of the activities the following network diagram is constructed for project.

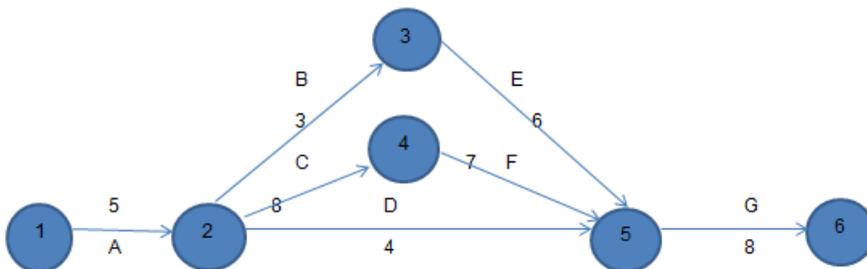


Figure 4.12: Solution example 4.3.2 of time estimate

Consider the path starting at the starting point and stopping at the end point.

**Path 1:** A-B-E-G.

time of the path  $5+3+6+8=22$ .

**Path 2:** A-C-F-G.

time of the path  $5+8+7+8=28$ .

**Path 3:** A-D-G .

time of the path  $5+4+8=17$ .

Compare the times for the three paths maximum of  $\{22, 28, 17\}=28$  we see that path two has the maximum time there for it is the critical path and the critical activities are A, C, F, G the rest of the edges are non-critical activities , such that project time 28 weeks.

**Example 4.3.3.** To complete the next project we must find the time required and then find the critical path and critical activities.

Activity	Optimistic time estimate( $t_o$ day)	Most likely time estimate( $t_m$ day)	Pessimistic time estimate( $t_p$ day)
1-2	3	6	9
1-6	2	5	8
2-3	6	12	18
2-4	4	5	6
3-5	8	11	14
4-5	3	7	11
6-7	3	9	15
5-8	2	4	6
7-8	8	16	18

Solution\From the three time estimate  $t_p, t_m$  and  $t_o$  calculate  $t_e$  for each activity we obtain the following table.

Table 4.8: Example 4.3.3 of time estimate

Activity	Optimistic time estimate $t_o$	4*Most likely time estimate $t_m$	Pessimistic time Estimate $t_p$	Time estimate $t_e=(t_o + 4t_m + t_p)/6$
1-2	3	24	9	6
1-6	2	20	8	5
2-3	6	48	18	12
2-4	4	20	6	5
3-5	8	44	14	11
4-5	3	28	11	7
6-7	3	36	15	9
5-8	2	16	6	4
7-8	8	64	18	15

Table 4.9: solution Example 4.3.3 of time estimate

With the single time estimate of the activities the following network diagram is constructed for project.

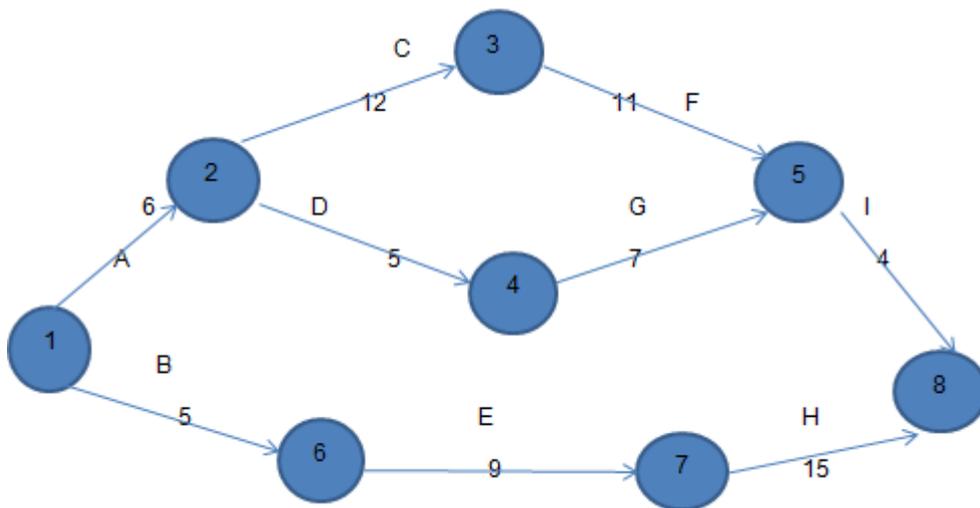


Figure 4.13: Solution example 4.3.3 of time estimate

Consider the path starting at the starting point and stopping at the end point.

Path 1: A-C-F-I

time of the path  $6+12+11+4=33$ .

**Path 2:** A-D-G-I

time of the path  $6+5+7+4=22$ .

**Path 3:** B-E-H,

time of the path  $5+9+15=29$ .

Compare the times for the three path maximum of  $\{33, 22, 29\}=33$

we see that path 1 has the maximum time there for it is the critical path and the critical activities are A, C, F, I .

The rest of the edges are non-critical activities , such that project time 33 weeks.

## 4.4 Improvement Time Estimate

The time has come to propose improved to estimate the most time likely for any project where we can give less weight to estimate the time weighted of 3 instead of 4 to estimate the likely time for distributors while estimates of time pessimistic and optimistic weight of 1 each because we see that giving weight 3 instead of 4 gives reasonable results and closer to reality and the appropriate degree of reliability, as well as to reduce the time the project is completed fully with achieve the desired goals and achieve results less than the estimated time the normal time of a few percent and more than estimate the time optimistic a few percent that's mean it takes the medium term between them and this leads to reduce the time and cost without causing

a negative impact on the project in general, and this leads not to schedule exceeded the project and that exceeded the time for any reason, whether an event was possible or potential problems do not take a large socket of time, but rather more than a little thing and due to the application of existing controls within the scheme of the project to coincide with the time of each paragraph as it is the law would be as follows called this improvement  $SA_2$ .

The mathematical law is:  $t_e = [(t_o + 3t_m + t_p)/5] - 1$

We will demonstrate this by applying the idea of optimization to the previous examples.

**Example 4.4.1.** To complete the next project by  $SA_2$  we must find the time required and then find the critical path and critical activities (same table 4.4).

**Solution** From the three time estimate  $t_p, t_m$  and  $t_o$  calculate  $t_e$  for each activity we obtain the following table.

Activity	Optimistic time estimate $t_o$	3*Most likely time estimate $t_m$	Pessimistic time Estimate $t_p$	Time estimate $t_e = [(t_o + 3t_m + t_p)/5] - 1$
1-2	2	5(3)	8	4
2-3	8	11(3)	20	11.2=11
3-4	2	4(3)	6	3
2-4	4	7(3)	10	6
2-5	4	8(3)	12	7
4-6	6	10(3)	14	9
5-6	3	5(3)	7	4
3-6	4	9(3)	14	8

Table 4.10: Solution of example 4.4.1 for improvement time estimate

Using the single time estimate of the activities we get the following network diagram for the project.

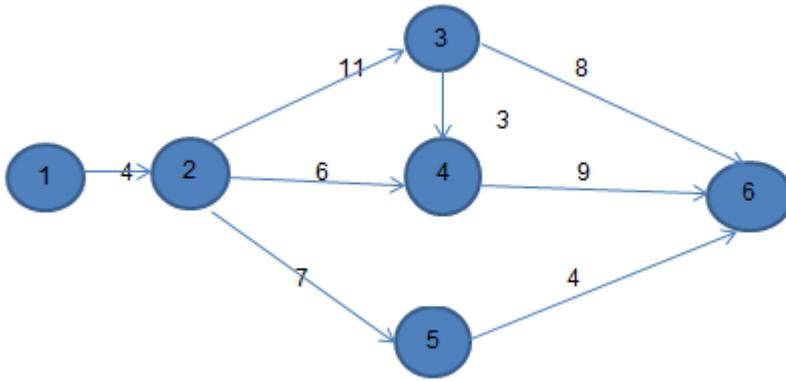


Figure 4.14: Solution of example 4.4.1 for improvement time estimate

Consider the path starting at the starting point and stopping at the end point.

**Path 1:** 1-2-3-6.

time of the path  $4+11+8=23$ .

**Path 2:** 1-2-3-4-6.

time of the path  $4+11+3+9=27$ .

**Path 3:** 1-2-4-6.

time of the path  $4+6+9=19$ .

**Path 4:** 1-2-5-6.

time of the path  $4+7+4=15$ .

Compare the times for the four path maximum of  $\{27, 19, 23, 15\}=27$   
 we see that path two has the maximum time there for it is the critical

path and the critical activities are 1-2, 2-3, 3-4, 4-6 the rest of the edges are non-critical activities , such that project time 27 weeks.

**Example 4.4.2.** A project consists of seven activities with the time estimate to find the probability that the project by  $SA_2$  will be completed in 30 weeks or less (same table 4.6) .

**Solution** From the three time estimate  $t_p, t_m$  and  $t_o$  calculate  $t_e$  for each activity we obtain the following table.

Activity	Optimistic time estimate $t_o$	3*Most likely time estimate $t_m$	Pessimistic time Estimate $t_p$	Time estimate $t_e = [(t_o + 3t_m + t_p)/5] - 1$
A	2	15	8	4
B	2	9	4	2
C	6	24	10	7
D	2	12	6	3
E	2	18	10	5
F	6	21	8	6
G	6	24	10	7

Table 4.11: Solution of example 4.4.2 for improvement time estimate

With the single time estimate of the activities the following network diagram is constructed for project.

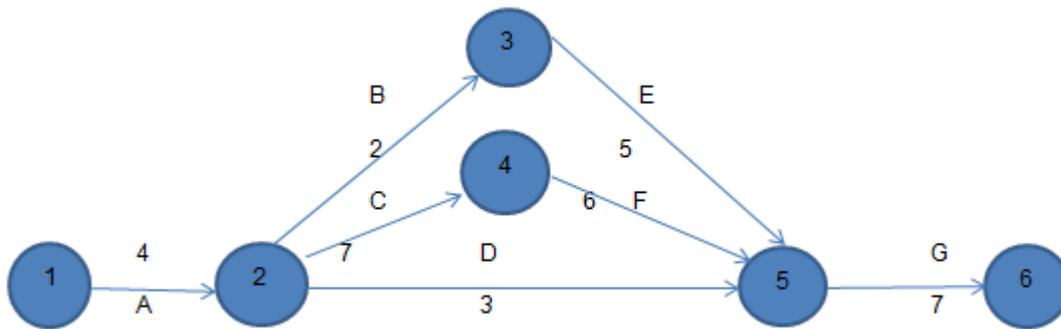


Figure 4.15: Solution of example 4.4.2 for improvement time estimate

Consider the path starting at the starting point and stopping at the end point.

Path 1: A-B-E-G.

time of the path  $4+2+5+7=18$ .

**Path 2:** A-C-F-G.

time of the path  $4+7+6+7=24$ .

**Path 3:** A-D-G.

time of the path  $4+3+7=14$ .

Compare the times for the three path maximum of  $\{14, 24, 18\}=24$  we see that path two has the maximum time there for it is the critical path and the critical activities are A, C, F, G the rest of the edges are non-critical activities, such that project time 24 weeks.

**Example 4.4.3.** To complete the next project by  $SA_2$  we must find the time required and then find the critical path and critical activities (same table 4.8) .

Solution\From the three time estimate  $t_p, t_m$  and  $t_o$  calculate  $t_e$  for each activity we obtain the following table.

Activity	Optimistic time estimate $t_o$	3*Most likely time estimate $t_m$	Pessimistic time Estimate $t_p$	Time estimate $t_e = [(t_o + 3t_m + t_p)/5] - 1$
1-2	3	18	9	5
1-6	2	15	8	4
2-3	6	36	18	11
2-4	4	15	6	4
3-5	8	33	14	10
4-5	3	21	11	6
6-7	3	27	15	8
5-8	2	12	6	3
7-8	8	48	18	13.8=14

Table 4.12: Solution of example 4.4.3 for improvement time estimate

With the single time estimate of the activities the following network diagram is constructed for project

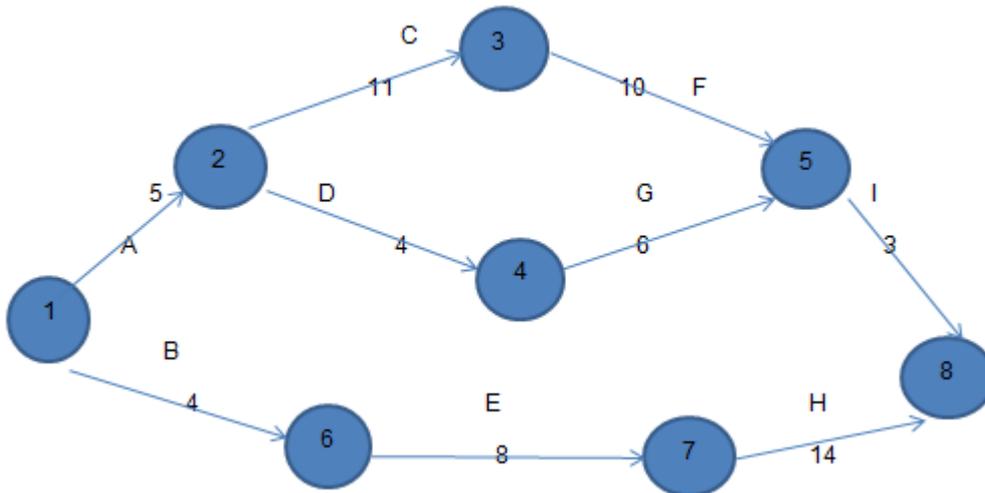


Figure 4.16: Solution of example 4.4.3 for improvement time estimate

Consider the path starting at the starting point and stopping at the end point.

**Path 1:** Path 1: A-C-F-I.

time of the path  $5+11+10+3=29$ .

**Path 2:** A-D-G-I.

time of the path  $5+4+6+3=18$ .

**Path 3:** B-E-H.

time of the path  $4+8+14=26$ .

Compare the times for the three path maximum of  $\{29, 18, 26\}=29$ .

We see that path one has the maximum time there for its the critical path and the critical activities are A, C, F, I the rest of the edges are non-critical activities, such that project time 29 weeks.

## 4.5 Results

Comparison of example results for the estimated time before optimization and the estimated time after optimization.

Examples	Results of time estimate	results of improvement time estimate
Example 4.3.1	31	27 (Example 4.4.1)
Example 4.3.2	28	24 (Example 4.4.2)
Example 4.3.3	33	29 (Example 4.4.3)

Table 4.13: The results of the previous examples and improvement time estimate

We note that the results of the estimated time after the improvement are less than the results of the estimated time before the improvement, and this indicates a reduction in the total time of the project

## CHAPTER 5

# CONCLUSIONS AND FUTURE WORKS

### 5.1 Conclusions

In the topics of our thesis, we deal with many ways and steps to reach the optimal solution, through the application of those methods and the proposed ideas to reduce the cost or time. In order to get the same results from the previous methods or slightly less than them. We suggested improving the time estimate to be closer to reality, and the results were achieved without any negative impact on the project, as we were able to give less weight to the potential time estimate, 3 instead of 4 to the distributors' likely time estimate, as for the pessimistic and optimistic time estimates, we give them a weight of 1 for each of them, as the results were intermediate between the normal time estimate and the optimistic estimate of time, and we took a map of one of the random cities and studied the maximum flow problem as well as the minimum cutoff problem. If we pass one of the paths from the starting point to the target point, We will face the paths with edges of different amplitudes, look at the edge with the highest amplitude, and then make the rest of the edges of that path the same amplitude to accommodate

as much flow as possible and thus improve the maximum flow, but in the case of optimizing the minimum cut, we cut the edge with the greatest weight to ensure that The process is more efficient and we proposed a new way to find the minimum spanning tree which is to relate it to the subject of the minimum cutoff and we find the weight of each path and then we take the paths gradually as needed from largest to smallest and cut off the edge with the most weight and in the end we got the results with the least cost, time and effort.

## 5.2 Future Work

1. We proposing new ideas and various more general methods and applying them directly to the networks to obtain results similar to or slightly less than the previous methods.
2. We recommend reducing the time and cost as much as possible without having any negative impact on production efficiency.
3. We recommend not to change the basic conditions of the topic in the event that new ideas or different methods are proposed, but rather it is supposed to be relied upon in setting the basics of the proposed methods.

## REFERENCES

- [1] AHUJA, R. K., AND ORLIN, J. B. Inverse optimization. *Operations Research* 49, 5 (2001), 771–783.
- [2] ARUMUGAM, S., BRANDSTÄDT, A., NISHIZEKI, T., AND THULASIRAMAN, K. *Handbook of graph theory, combinatorial optimization, and algorithms*, vol. 34. CRC Press, 2016.
- [3] AWERBUCH, B., AND LEIGHTON, T. Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing* (1994), pp. 487–496.
- [4] BANG-JENSEN, J., AND GUTIN, G. Z. *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.
- [5] COOK, W., LOVÁSZ, L., SEYMOUR, P. D., ET AL. *Combinatorial optimization: papers from the DIMACS Special Year*, vol. 20. American Mathematical Soc., 1995.
- [6] DOĞAN, B., AND ÖLMEZ, T. A new metaheuristic for numerical function optimization: Vortex search algorithm. *Information Sciences*

293 (2015), 125–145.

- [7] EASTTOM, C., AND ADDA, M. Application of the spectra of graphs in network forensics. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)* (2021), IEEE, pp. 0846–0852.
- [8] FENG, Y., DEB, S., WANG, G.-G., AND ALAVI, A. H. Monarch butterfly optimization: a comprehensive review. *Expert Systems with Applications* 168 (2021), 114418.
- [9] GAN, G., MA, C., AND WU, J. *Data clustering: theory, algorithms, and applications*. SIAM, 2020.
- [10] HOLM, J., DE LICHTENBERG, K., AND THORUP, M. Polylogarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Journal of the ACM (JACM)* 48, 4 (2001), 723–760.
- [11] JUNGNICHEL, D., AND JUNGNICHEL, D. *Graphs, networks and algorithms*, vol. 3. Springer, 2005.
- [12] KARGER, D. R. Minimum cuts in near-linear time. *Journal of the ACM (JACM)* 47, 1 (2000), 46–76.
- [13] KAUFMANN, D., KLAMPFL, L., KLÜCK, F., ZIMMERMANN, M., AND TAO, J. Critical and challenging scenario generation based on automatic action behavior sequence optimization: 2021 iee autonomous driving ai test challenge group 108. In *2021 IEEE International Conference on Artificial Intelligence Testing (AITest)* (2021), IEEE, pp. 118–127.

- [14] KAY, B., DATE, P., AND SCHUMAN, C. Neuromorphic graph algorithms: Extracting longest shortest paths and minimum spanning trees. In *Proceedings of the Neuro-inspired Computational Elements Workshop* (2020), pp. 1–6.
- [15] KIM, B.-H., AND YE, J. C. Understanding graph isomorphism network for rs-fmri functional connectivity analysis. *Frontiers in neuroscience* (2020), 630.
- [16] LI, H., ZHANG, T., ZHANG, Y., WANG, K., AND LI, J. A maximum flow algorithm based on storage time aggregated graph for delay-tolerant networks. *Ad Hoc Networks* 59 (2017), 63–70.
- [17] LIU, Q., WEI, W., YUAN, H., ZHAN, Z.-H., AND LI, Y. Topology selection for particle swarm optimization. *Information Sciences* 363 (2016), 154–173.
- [18] MAMUN, A.-A., AND RAJASEKARAN, S. An efficient minimum spanning tree algorithm. In *2016 IEEE Symposium on Computers and Communication (ISCC)* (2016), IEEE, pp. 1047–1052.
- [19] MANCHANDA, S., MICHEL, S., DRAKULIC, D., AND ANDREOLI, J.-M. On the generalization of neural combinatorial optimization heuristics. *arXiv preprint arXiv:2206.00787* (2022).
- [20] OSIPOV, V., SANDERS, P., AND SINGLER, J. The filter-kruskal minimum spanning tree algorithm. In *2009 Proceedings of the Eleventh Workshop on Algorithm Engineering and Experiments (ALENEX)* (2009), SIAM, pp. 52–61.
- [21] RAO, L., LIU, X., XIE, L., AND LIU, W. Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-

- market environment. In *2010 Proceedings IEEE INFOCOM* (2010), IEEE, pp. 1–9.
- [22] RUTHMAIR, M., AND RAIDL, G. R. A kruskal-based heuristic for the rooted delay-constrained minimum spanning tree problem. In *International Conference on Computer Aided Systems Theory* (2009), Springer, pp. 713–720.
- [23] SCHAUB, M. T., ZHU, Y., SEBY, J.-B., RODDENBERRY, T. M., AND SEGARRA, S. Signal processing on higher-order networks: Livin’ on the edge... and beyond. *Signal Processing* 187 (2021), 108149.
- [24] SEN, Z., HEPING, Z., AND JINSONG, T. Dendriiform branch cut algorithm based on minimum spanning tree for phase unwrapping. *Procedia Engineering* 29 (2012), 1154–1159.
- [25] SENGUPTA, S., BASAK, S., AND PETERS, R. A. Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives. *Machine Learning and Knowledge Extraction* 1, 1 (2018), 157–191.
- [26] SOON, J. J., AND LOW, K.-S. Photovoltaic model identification using particle swarm optimization with inverse barrier constraint. *IEEE Transactions on Power Electronics* 27, 9 (2012), 3975–3983.
- [27] SPORNS, O. Graph theory methods: applications in brain networks. *Dialogues in clinical neuroscience* (2022).
- [28] TANWEER, M. R., SURESH, S., AND SUNDARARAJAN, N. Self regulating particle swarm optimization algorithm. *Information Sciences* 294 (2015), 182–202.

- [29] WANG, L., YANG, B., AND CHEN, Y. Improving particle swarm optimization using multi-layer searching strategy. *Information Sciences* 274 (2014), 70–94.
- [30] WEISSTEIN, E. W. *CRC concise encyclopedia of mathematics*. Chapman and Hall/CRC, 2002.
- [31] ZHANG, Q.-B., WANG, P., AND CHEN, Z.-H. An improved particle filter for mobile robot localization based on particle swarm optimization. *Expert Systems with Applications* 135 (2019), 181–193.