STABLE MATCHING PROBLEM AND ITS APPROACH TO DETERMINE VITAL LINK IN 2-WAY NETWORK

Enrollment No. - 122208
Name of Student - Ekta Gupta
Name of supervisor - Dr. Nitin

May – 2014

Submitted in partial fulfillment of the Degree of
Master of Technology

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND INFORMATION TECHNOLOGY
JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TABLE OF CONTENTS

CERTIFICATE

This is to certify that the work titled "STABLE MATCHING PROBLEM AND ITS APPROACH TO DETERMINE CRITICAL LINK" submitted by "EKTA GUPTA" in partial fulfillment for the award of degree of Master of Technology (M.Tech.) of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor ……

Name of Supervisor Dr. Nitin

Designation Associate Professor

Date .....

ACKNOWLEDGEMENT

I am grateful to Nitin for his valuable comments and his efficient use of time for painstakingly checking the correctness of proof of complexity in the preliminary versions of the thesis. We would like to acknowledge Jaypee University of Information Technology for its continual support both financially and technically to help improve the presentation of this paper. Finally, I would like to thank the anonymous referees for their constructive comments that greatly helped in improving the thesis work

Signature of the student

Name of Student Ekta Gupta

Date ……..

SUMMARY

We have studied the stable marriage problem and have given a detailed description of its variations and its real world instances. We have shown the implications and what changes those real world instances have and what modifications were made by other researchers to handle them. Afterwards, we proposed algorithm which describe the transition of a male pessimal matching set to optimal when it is a men- oriented approach by deleting a pair from a matching set considering score based approach. Resulting the best case scenario and thereafter applying the parallel approach to reduce the complexity. Through this result we can consider hard problem of parallel problem solvable in polynomial time. Further, the proposed approach is applied in the application of two sided network which consist of connectivity between clients and server in order to determine vital link to provide the shortest path length.

The significance of the SMP is clearly proven by all its variations and application in real life. Along with that every algorithm that manages to improve the certain aspects of the solutions and not just stability should be beneficial.

Modified approach to Gale Shapely algorithm manages to give an alternative point of view on the situation by providing with the solution that can raise the network reliability by holding stability.

_____	_____
Signature of Student	Signature of Supervisor
Name: Ekta Gupta	Name: Dr Nitin
Date:	Date:

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Stable Matching Problem

The Stable matching problem was first introduced by David Gale and Lloyd Shapely in 1962 in a paper entitled "College Admission and the Stability of Marriage" [1]. Gale and Shapely analysed the matching at abstract and general level where they used marriage problem as an illustrative example. Idea behind the marriage problem was "How matching should be done between set of men and set of women with their respective preference list?" The main challenge behind the simple idea was that matching should be done in such a way that no pair should break up and form new pair which would make them better off. For such idea, the solution was "Gale Shapely – deferred acceptance algorithm" consisting of set of simple rule and solvable in polynomial time.

Further the stable marriage theory was extended to whole research areas with the result arising from the field of computer science, game theory, mathematics and economics. Such results founded the solution to real time problem such as Hospital/Resident problem where assigns new doctors to hospital, 3-way kidney problem, assigning human organ for transplant to recipient. The generalized variant of Hospital/Resident problem was used as practical application in matching scheme such as National Resident Matching Program [2], the Canadian Resident Matching Service [3] and the Scottish Pre-registration house officer Allocations schema [4].A recent research [5] has emphasized on each and every aspect of Stable Matching Problem (SMP), such as incomplete lists, blocking pairs, ties etc. and has enlisted the research till now done to eradicate it. Another variant of stable matching problem is known as Stable Roommate Problem. Stable Roommate problem is non-bipartite variant of stable matching problem. In this problem there is no pair of elements from different set need to broke the set into male and female subset, also any'2n' member in the set can prefer '2n-1'menber in the same set. There is some instance for which stable roommate problem could not find any optimal matching and was stated as hard

problem but an algorithm by Irving provided the solution in polynomial time stating the existence and non- existence of matching for the instance. Further many researcher provided solution for finding almost stable matching to roommate problem. Latest work has been approved as "Almost stable matching in Roommate problem with bounded preference list"[6].

There were other models which gave new dimension to stable matching definition. Man-Exchange Model [7]: it stated the stability property that no two men exchange their partner and its existence is NP- Complete. Many-to-many stable matching[8][9]: here both set is provided with defined quota which was initially applied to hospital/ resident problem and found algorithm for minimum egalitarian matching[10]. One sided preference list[11]: in this one set has the preference over the other set.

Decade later, the work was extended by Mayr and A. Subramaian in the field of network stability. They showed that network stability problem is NP- Complete Problem. However, later it was proved that when network is multistage interconnection Network (MIN) then stability problem becomes equivalent to Stable matching problem. They showed the situation in which MINs becomes unstable and how can stability be achieved in the unstable network.

Further, In 2012, Lloyd Shapley and Alvin Roth got Nobel Prize in economic science for extending abstract theory developed in 1960 over empirical work in 1980, to on-going effort on finding further analytical developments as well as practical design in market institute.

1.2 Problem Statement

David Gale and Lloyd Shapley introduced the problem of stable matching in 1962 in a paper entitled "College Admissions and the Stability of Marriage. The basic algorithm states people to be heterosexual with n-men and n-women providing their preference lists for the opposite gender and hence the matching being 1:1. In some instances it came out to be a fact that man-oriented approach led to a man-pessimal matching where they did not get their best possible partner contrary to what Gale-Shapely proposed which leads to a worst case scenario. Though it has already been experimentally proved that the chances of having a worst case scenario for stable matching is extremely low, but occurrence of it prevents the parallel algorithm for stable matching to run, which could have reduced the running time dramatically. Therefore describes

the transition of a male-pessimal matching set to optimal when it is a man-oriented approach by deleting a pair from matching set considering the score based approach. Resulting the best case scenario and thereafter applying the parallel approach so to reduce time complexity. Through this result we can consider hard problem of parallel problem solvable in polynomial time. Further this problem has been carried on the two sided network where we consider on one side as set of client or user and on other side the set of server. Applying stable matching solution to two sided network targets the stability of network reliability by providing hop length stability and network connectivity. We use the stable matching approach on the network to find the vital link to be protected so that user can access the server within small hop count even if non – vital links fails such that the problem can solved in polynomial time.

1.3 Objective

The objective behind the thesis is to study, find and describe the problem instances of stable matching problem and its variants in real life problem. Also by studying the native algorithm by gale and Shapely to create a solution based on score based approach that focus on finding solution that may not be stable. For Example we consider problem on content distribution network in order to provide stability of network reliability and network connectivity.

1.4 Thesis Formation

The thesis has been divided into five chapters.

The chapter1 focuses on the problem statement of the thesis and the importance of the stable matching.

The chapter2 focus on the background details which has been divided into two sub- section. The former sub-section describes in detail about classical Gale Shapley approach and how simple idea was extended to its various variant. Also describe about algorithm structure and how authors modified the algorithm to improve the time complexity. The later sub-section describes the literature study of the previous work done on the network and to find the shortest path in two sided network during occurrence of failure.

Chapter 3 describes the approach proposed to eradicate the worst case scenario of the stable matching approach with its comparative result

Chapter 4 describe the stable matching approach applicable on the content delivery network with its proof of correctness and experiment result

Chapter 5 focus on the tools used and algorithm data structure and discusses on the various parameter used to provide input to the algorithm.

Chapter 2

Background Overview

Section I: Stable Matching Approach

2.1. Definitive Approach to Stable Matching [1]:

The stable matching problem consist of set of n men and n women, each person mentions their respective preference list by ordering n people from opposite side in strict manner.

Objective: To find the stable matching such that no pair leave their current partner.

Stability: If a man 'm' and woman 'w' is assigned with other partner, but 'm' prefers 'w' to his current partner and 'w' prefer 'm' to her current partner then they have bonus to leave their current partner and switch to another partner. Such pair (m, w) is said to be unstable pair.

Stable Matching: A matching is said to be perfect if there is one –to-one correspondence between men and women i.e. each men is assigned at most one women and each women is assigned at most one man. Therefore matching is stable if there is no unstable pair and has a perfect matching.

Example: Given the below dataset table, there are three men and three women with their respective preference list from most priority to less priority.

	1st	2nd	3rd
X	A	B	**C**
Y	**B**	A	C
Z	**A**	B	C
	1st	2nd	3rd
A	Y	X	**Z**
B	X	**Y**	Z

C	**X**	Y	Z

ii) Women Preference List

i) Men Preference List

Table 1: Men and Women Data Set

The question arises is assignment X-C, Y-B, Z-A stable?

By observing Table 1, we came to know that pair (X, B) will hook up, leading to blocking pair. Hence pair (X, C) (Y, B) (Z, A) is unstable pair since (X, B) is a blocking pair.

This problem was solved by Gale Shapley by intuitive based method known as deferred acceptance algorithm which always finds the stable matching and runs in O(n2) time. Hence, it is a positive proof that any instance admits at least one stable matching.

Table 2: GSA Algorithm[1]
Input: Data set Instances of Stable Matching *Output:* Stable Matching Pair, M
1. Initialize each person to be free. 2. while (some man is free and hasn't proposed to every woman) 3. Choose such a man m 4. w = first woman on m's list to whom m has not yet proposed 5. if (w is free)

```
6.          assign m and w to be engaged
7.        else if (w prefers m to her fiancé m')
8.          assign m and w to be engaged, and m' to be free
9.        End if
10.   End while
```

Theorem: Algorithm finishes the execution after at most n^2 iterations.

Proof: From above algorithm two points were observed, they are as follows:

Ob 1: Men propose to women in decreasing order of preference.

Ob 2: Once a woman is matched, she never becomes unmatched; she only "trades up."

Therefore, each time through the while loop a man proposes to a new woman. There are only n^2 possible proposals.

2.1.1 Men Optimality and Female Pessimality:

The Gale Shapley in best case produces matching that is men optimal and women pessimal. Men optimal means that there is always best possible partner for each men and Women pessimal means that women get worst partner. This is followed by following lemma:

Lemma 1.1 (a): During the execution of GS algorithm if women w reject men m then (w, m) cannot be stable pair or valid partner.

Proof: Say there is an unstable pair (Z, A). Let the partner of Z be B and partner of A be Y. Then two cases arise:

1. *Z has not offered to A:*

According to ob1, men always propose in decreasing order. Therefore B is on higher list of Z than A. thereby contradict the instability of (Z, A).

2. *Z has offered to A:*

Assuming A rejects Z and now paired with X. therefore now matching M is (X, A). Since proposal is in increasing order, Y is in highest priority list. Thereby, contracting the instability of pair (Z, A).

Theorem 2: If paring is Men optimal then it is also Female pessimal.

Proof: Given the following dataset shown in Table 3

Men Preference List

	1st	**2nd**
X	A	B
Y	B	A

Women Preference List

	1st	**2nd**
A	X	Y
B	Y	X

Table 3: Entities Preference Profile

Say X is rejected by optimal pair A, and now paired with B. But when Y proposes to B, being higher on B list, Pair (Y, B) gets accepted and lead to instability of (X, A). Hence contradict the stability of (X, A). According to induction to female list, if matching list is (Y, A) and (X, B) then it lead to blocking pair (X, A) and (Y, B). Thereby contract the men optimality.

2.2 Extension variant to Preference list:

In Stable matching problem, each person preference list includes all member of the opponent set. Represented by: $w_1 >_m w_2$ (Men 'm' prefers w1 to w2) and if in a matching M, Man '*m*' and

Woman '*w*' are matched together, indicated by *M(m) = w* and *M(w) = m*. But in large scale application scheme it becomes impossible to include all members of the opponent team. Therefore, there are two relaxing extensions that can be included in preference list i.e. Ties and Incomplete List.

2.2.1 Incomplete Preference Lists (SMI)

In this variant one or more than one person list may be incomplete. If person P's list include a person 'R' then it is said 'R' is acceptable to 'P'.

Stable Matching with Incomplete List (SMI): SMI is defined as a matching M of a disjoint set of pair (P, R) such that 'P' and 'R' are acceptable to each other.

Hence Blocking Pair is defined under three conditions:

a) $M(m) \neq w$ but w and m are acceptable to each other;

b) $w >_m M(m)$ or m is single in M.

c) $m >_w M(w)$ or w is single in W.

Property 1: SMI partition entities into two set. One set containing entities with stable pairs and another set containing entities that are Single.

Further, it was showed that SMI can find stable matching by using Gale- Shapley algorithm with minimal modification [12].

2.2.2 Preference Lists with Ties (SMT)

In this variant one person can include two or more person with same priority, denoted by $w_1 =_m w_2$. SMT stability is categorized into three notations. They are as follow:

- *Super-Stability*: Blocking pair is defined as a pair(m, w) such that

$$M(m) \neq w, \; w \geq_m M(m),$$

and $m \geq_w M(w)$.

- *Strong-Stability:* Blocking Pair is defined as a pair (m, w) such that

$$M(m) \neq w,\ w >_m M(m),$$

and $m \geq_w M(w)$.

- *Weak-Stability*: Blocking Pair is defined as a pair (m, w) such that

$$M(m) \neq w;\ \ w >_m M(m)$$

and $m >_w M(w)$

Super stable matching is most stable among three notations. Also it has been proved that weakly stable matching is solvable in polynomial time [13] but for some instances super-stable and strong-stable is unsolvable. However, there is an algorithm to decide whether super-stable/strong-stable exist or not and by recent research is solvable in polynomial time i.e. $O(n^2)$ [14].

2.2.3 Incomplete preference list with ties (SMTI)

SMTI is generalization to combination of SMT and SMI approach.

SMTI is defined as Matching:

- *Weakly Stable:* if there is no unpaired couple (m; w) and finds each other acceptable or each of them strictly prefers the other to his/her partner in M.
- *Strongly Stable:* if there is no unpaired couple (m; w) such that
- Either ‘m’ is unpaired in M and finds ‘w’ acceptable, or ‘m’ strictly prefers ‘w’ to his/her partner in M.
- Or ‘w’ is unpaired in M and finds ‘m’ acceptable, or ‘m’ strictly prefers ‘w’ to his/her partner in M, or ‘w’ is indifferent between ‘m’ and his/her partner in M.

- *Super Stable:* if there is no unpaired couple (m, w), each of whom is either unpaired in M and finds the other acceptable, or strictly prefers the other to his/her partner in M, or is indifferent between them.

Men preference profile		
	1st	**2nd**
M_1	W_1	
M_2	W_1	W_2

Women Priority profile	
	1st
W_1	(M_1, M_2)
W_2	M_2

Table 4: Instance of SMTI

From Table 4, there are two weakly stable matching of distinct size, that is $\{(M_1, W_1), (M_2, W_2)\}$ and $\{(M_2, W_1)\}$.

For strong and super stability, the same results have been shown as in SMT. The execution time of Strong Stable matching O(*a*) and Super Stability matching is O(*na*)[14] where 'a' is the length of the preference list which is equivalent to $2n^2$ if preference list is complete. For Weak Stable, problem is identifiable for many instances and exit in O(*a*) time. But due to different size problem of finding MAX SMTI is NP hard problem [15][16].

2.2.4 CHEATING BY PLAYERS

If any player cheats by specifying a deceitful preference list to get his/her desired partner. If one man lies and others are true about their preferences then man cannot cheat, though the woman can cheat arranging the men in her preference list in a deceitful manner. Cheating is desirable on

the women side as it's always a man oriented approach when man proposes. An example of such an instance is shown in the figure below. The way woman can cheat to get her man of dreams is now explained. At first she remains blank declaring any of the men un-acceptable for her. Then after knowing the order of proposal of men she can specify which man she wants to be paired up with and can accordingly specify her preference list stating all men as unacceptable after the one she wants to tie knot with. This makes her get that man in order to have the whole marriage stable. But in some cases, woman is not allowed to keep the men unacceptable.

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

Men's Preference List

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

Women's True Preference Profile

	1st	2nd	3rd
A	Y	Z	X
B	X	Y	Z
C	X	Y	Z

***A* Lies**

Table 5: Cheating by women

A natural extension suggests woman' w' to:

- Accept a proposal, and then reject all future proposals.
- From the list of men who proposed to *w* but were rejected, find her most preferred partner; repeat the Gale-Shapley algorithm until the stage when this man proposes to her.
- Reverse the earlier decision and accept the proposal from this most preferred partner, and continue the Gale-Shapley algorithm by rejecting all future proposals.
- Repeat (b) and (c) until the woman cannot find a better partner from all other proposals.

DISADVANTAGE: The above strategy does not always yield the best stable partner a woman can achieve. The reason is that this greedy improvement technique does not allow for the possibility of rejecting the current best partner, in the hope that this rejection will trigger. To

solve this issue, the authors of [17] described one more approach, but it was not even free from anomalies.

2.3 REAL LIFE APPLICATION

2.3.1 Stable Roommate Problem (SRM)

Stable roommate is the generalized view of non-bipartite graph of stable matching problem. Here there is no disjoint sets of entities i.e. does not require to categorize entity into two set (in general, set of men and set of women). It is different from stable marriage problem as any entity can prefer any other entity in the same set. Further it was also proved that stable matching for the instance of stable roommate may not exit.

Example: Give the data set with their preference list as shown in table 6

No. of entities	**1st**	**2nd**	**3rd**
P_1	P_2	P_3	P_4
P_2	P_3	P_1	P_4
P_3	P_1	P_2	P_4
P_4	P_2	P_1	P_3

Table 6: Instance I_1 of SRM with preference list

In above give instance if any of the pair say {(p1, p4); (p2, p4); (p3, p4)} is involved in matching set M then M will be blocked by following pair {(p1, p3); (p2, p1); (p3, p2)}. Knuth theoretical proved that problem of finding the existence of stable matching is a NP-Complete problem. Later, Irving [18] provided the polynomial time algorithm that find the existence of stable or report if does not exist. Irving algorithm consists of two distinct phase:

Phase1: This phase is similar to Gale-Shapley algorithm where each participant proposes to other participants based on their preference list or continues to propose next person if current proposal is rejected. A participant rejects the proposal if he already hold proposal or respectively receive proposal from one he prefer the most. An important observation was notice in phase1 that one participant might be rejected by all of the others person thereby proving that no stable matching exist or end with proposal hold by each person.

On the termination of phase1 stable roommate foreshorten preference list is generated known as Stable Table or Phase1 table.

Participants	**PREFERENCE LIST**				
P_1	P_3	P_4	P_2	P_5	P_6
P_2	P_6	P_5	P_4	P_1	P_3
P_3	P_2	P_4	P_5	P_1	P_6
P_4	P_5	P_2	P_3	P_6	P_1
P_5	P_3	P_1	P_2	P_4	P_6
P_6	P_5	P_1	P_3	P_4	P_2

Table 7: Instance I_2 Stable Roommate involving 6 participants P1 to P6

After phase 1 execution Table7 ends with reduced preference list as shown in Table 8 where 1st preference shows the sequence of proposal and deletion shows rejections

Participants	**PREFERENCE LIST**				
P_1	P_4	P_2	P_5	P_6	
P_2	P_6	P_5	P_4	P_1	P_3
P_3	P_2	P_4	P_5		
P_4	P_5	P_2	P_3	P_6	P_1

P_5	P_3	P_2	P_4
P_6	P_1	P_4	P_2

Table 8: Stable Table for Instance I2

Table 9: Stable Roommates Phase1 Algorithm

Input: Stable Roommate Instance

Output: Stable Table

```
1. Assign each person to be free;
2. While some free person Pi has a non-empty list do
3.      Pj:= first person on Pi's list;
4.      if some person Pk is semi-assigned to Pj then
5.              Assign Pk to be free;
6.              Assign Pi to be semi-assigned to Pj ;
7.              for each successor Pl of Pi on Pj's list do
8.                      delete the pair (Pl, Pj);
9.              End for
10.     End if
11. End while
```

Phase 2: In this phase the preference list is reduced by eliminating sequence of rotations. This phase continues until all lists have one person indicating stable matching have been found or participant list becomes empty indicating no stable matching exist for such instance.

Table 10: Stable Roommate Phase II Algorithm

Input: Stable Table Instance

Output: Stable Matching M

```
1.  T:= Stable Table;
2.      While(T list is non-empty) do
3.          Identify rotation R in T;
4.          Eliminate R from T;
5.          If(T list becomes empty)
6.              Return Null; // no stable matching exit
7.          Else If(reduced List become of size 1)
8.              Return Stable –Pair (x,y); // found the stable pair
9.          End if
10.     End while
```

Using stable table of I2, rotation R1 is identified i.e. (P_1, P_4), (P_3, P_2) as P_2 is P_1 second favourite and P_4 is second favourite of P_3. Thus Remove R_1. Next, the rotation $R_2 = \{(P_1, P_2), (P_2, P_6), (P_4, P_5)\}$ is identified. After eliminating stable match found is P_1 and P_6. At end of phase 2 stable matching found is $\{(P_1, P_6), (P_2, P_4), (P3, P5)\}$.

Irving proved that stable roommate algorithm terminate in $O(n^2)$ and also stated that given SM instance we can construct SR instance such that stable matching is one–to–one correspondence.

Stable Roommate Problem with Ties and Incomplete List: SR generalization includes Ties and Incomplete List as was consider in SM problem. SR blocking pair is also defined in terms of three notations i.e. Weakly Stable, Strong Stable, Super Stable as discussed in earlier section of

Stable marriage problem. As already discussed there is no guarantee of finding stable matching in SR without ties. Hence breaking ties in SR does not guarantee that stable matching will be reported. Ronn [19] proved that the problem of resolving whether SRT instance admits a weakly stable is NP-complete. Further Irving and Manlove [20][21] showed the decision on weak stability in SRT and SRTI and also proved weak stable matching instances of SRT may have distinct sizes.

Scott [22] presented $O(a^2)$ algorithm for finding strong stable matching for SRTI instances where a is the length of the preference list. Algorithm also processes in two- phases and is somewhat complex than super stable matching.

2.3.2 Hospital/Resident problem

In USA during early 1940, competition for student graduating from the medical school forced hospital to offer internship even several years before getting graduating. Matches were made before the student could complete their qualification and even before they knew which branch of medicine they would like to prefer. When an offer was rejected, it was too late to provide offer to other candidate. Therefore it resulted in producing unstable match because sufficient offer cannot be made on time in order to benefit from mutual trades. To provide solution to this problem, The National Resident Matching Program [NRMP] was introduced in 1950. In 1984, the problem was studied by Alvin Roth and found that it closely matches to Gale-Shapely algorithm [23]. Further, Roth[24] described the centralized matching scheme that organizes doctor recruitment in the US by referring the NRMP.

Hospital / Resident problem is most well known application in stable matching also known as college admission problem. It is a process that matches the medical student and hospital having opening for interns. The matching is based on one to many mapping because hospital has limited position and only top students graded by the hospital gets admission. Later, new centralized matching scheme have been implemented for high schools in many other countries like Boston[25], Spain[26], Turkey[27], etc. but there is lack of information about the details of these schemes.

2.3.3 Kidney Exchange

In U.S. more than 70,000 people are awaiting to obtain a kidney transplant. But there are few problems such as deficit of donors due to lesser donors that is on a count of 10,000 a year but among then surviving donors are 7000 a year. In 2005, forty thousand patients died waiting for the appropriate donor. Further the problem is not simple supply and demand. There is other problem also such as donor kidney requires being compatible with the patient. Therefore patient having a living donor but cannot use the kidney because of issue. Also two or more patients can be involves in trading donor kidneys by participating in a kidney exchange. Hence matching theory help to understand this problem and make use of a limited consortium of donors.

To deal with such problem many researcher proposed an algorithm. In 1986 F.T. Rapaport, the medical doctor proposed the algorithm *live-donor paired kidney exchange.* The algorithm considered input as incompatible patients- donor pair such as donor in each pair contributes a kidney to the other pair compatible patient [28].

Fig1: A three- way kidney exchange. P_i denotes the pateint and D_i indicates the donor in each pair of the exchange

In 90's, Korea and Netherlands started to form the database to form such trades and reported that the algorithm constitute up to more than 10% of live donor transplants [29]. On the success many other countries started conducting live-donor kidney exchange procedure such that it has been estimated that 2,000 additional transplant were conducted successfully per year [30]. But main problem in maintaining the kidney exchange was deficit of proper procedure to clear the trade in an efficient and motivational- compatible manner. To solve such problem, [31] Roth et al proposed the first framework that was based on the core mechanism that was designed for house allocation problem with existing renters named as *Gale's top trading cycle algorithm [32]*.

2.3.4 3-WAY KIDNEY EXCHANGE PROBLEM [33]

3 Way Kidney Exchange Problem is an application of the Circular Stable Matching. Circular Kidney Exchange problem has been first introduced by Knuth with his question "It is not always the case that we have 2 pools of datasets, so can there be any way that Stable Matching "In Circular Stable matching instead of 2 pools of data we have 3/ more pools of datasets. Taking an example of a 3 pool dataset say we have men preferring women, women preferring dogs and dogs preferring men, where each entity from each dataset prefers another entity from other dataset and this way entity dataset-preferred dataset form a circular representation. The goal is to organize them into family units (man, woman, dog) so that no three of them have incentive to leave their assigned family members to join in a new family. The family units are in the form of a triplet. Supposing that we have family units as {($m1$,$w1$,$d1$), ($m2$,$w2$,$d2$), ($m3$,$w3$,$d3$)} in the matching set M, if $m1$ prefers $w2$ to $w1$, $w2$ prefers $d3$ to $d2$, and $d3$ prefers $m1$ to $m3$, then ($m1$, $w2$, $d3$) becomes a blocking triple. But it may also be the case that $w2$ prefers $d1$ to $d2$. Then ($m1$, $w2$, $d1$) can also be conceived as a (weaker) blocking triple, since only $m1$ and $w2$ are really preferring each other in such a triple, while $d1$ is indifferent. Hence, the concept of blocking pairs is more complex in circular stable matching than in basic stable matching.

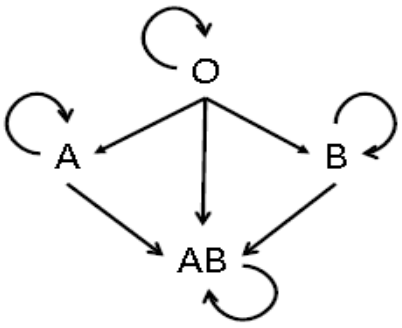


Figure 2: Blood Compatibility

This concept has been extended to be applied in the field of medical science for Kidney Exchanges. Currently, 118, 617 patients are waiting for a prospective donor for vital organ transplantation, and among these about 93, 000 are waiting for Kidney Transplantation. In most of the cases it is the family member who is interested to donate a kidney. But, unfortunately in many cases, they fail to match the tissue type or blood group type of the patient. Such cases are named as Tissue Incompatibility and Blood Group type Incompatibility respectively. Such a (patient donor) pair is referred to as Incompatible Pair. Figure 3 represents the blood

compatibility issues. To this Circular Stable Matching comes forward with solution of 3-Way Kidney Exchange. The Figure 4 below shows an example of how 3-Way Kidney Exchange works. Here, we have 3 donors and three recipients. Recipient 1 (wife) having the blood group O seeks a kidney here. Both husband and husband's sister try giving the kidney, but both fail due to blood group incompatibility and person with Blood Group O can only receive from another person with the same blood group, thought he/she can give to anybody. Recipient 2 (wife's brother) having blood group A receive from the husband and Recipient 3 (Wife's brother's friend) having the universal receptor type blood group can receive from anybody (Donor 2).

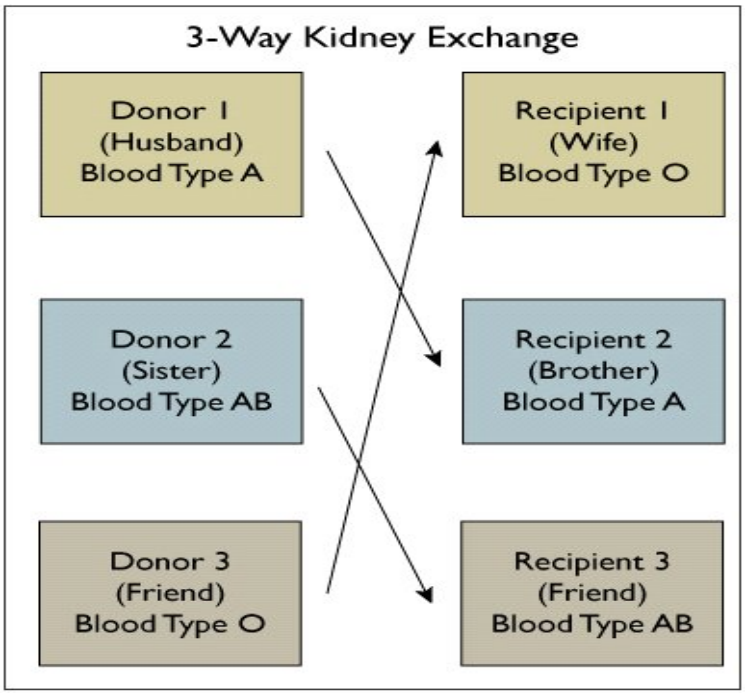


Figure 3: 3-Way kidney exchange problem

Stable matching concept is applied in 3-WAY KIDNEY TRANSPLANT, as matching is composed of oriented triples. Here, we write such a triplet as ($k1$,$k2$,$k3$) to denote that $k2$,$k3$,$k1$ are the successors of $k1$,$k2$,$k3$, respectively. Here, each triplet (k1, k2, k3) represents a couple which is often a married couple in case on stable matching, but here it consists of a person needing a new kidney (patient) and a potential kidney donor (donor). If $k2$ follows $k1$ in a triplet, then the donor from the (patient, donor) pair $k2$ will be donating a kidney to the recipient of $k1$. Thus, it is $k1$'s preference (degree of compatibility) that needs to be considered.

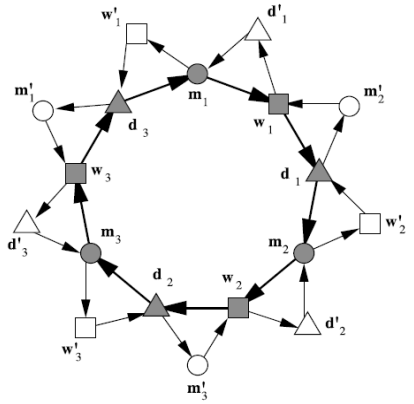
At this point, a point to note here is that an oriented couple matching set ($k1$,$k2$,$k3$) can be a blocking triplet itself ($k1$,$k3$,$k2$), if $k1$ prefers $k3$ to $k2$, $k3$ prefers $k2$ to $k1$, and $k2$ prefers $k1$ to $k3$. Such phenomenon has been an open stable matching problem for the researchers in the stable matching literature till now. For tissue incompatibilities, the serum from a patient and lymphocytes from a donor must be physically mixed to affirmatively determine compatibility issues before a transplant can take place. Such a test is known as the Human Leukocyte Antigen (HLA) Test. Each such test requires a non-trivial amount of blood; therefore it is not feasible to exhaustively determine all compatibilities in an exchange with hundreds of participants. But, here a problem lies in coordinating the subsets of the test domain as the incompatibility pairs led from blood type incompatibilities and tissue incompatibilities may be located at multiple hospitals across India. For this till now there is centralized national testing centre. Therefore, instead of real data compatibility tests, virtual compatibility tests based on computer programming by running the Circular Stable Matching algorithm is done, which is then later verified by real compatibility tests before transplantation.

Further, 3-Way Kidney Exchange has been extended to the case where there are multiple donors for a single patient interested to donate. For example, if we have a child (patient) then both the parents may be interested to donate. In this case we have a problem similar to ties in SMP and it is solved in the same way too by arbitrarily selecting one. When represented in the form of graphs, a back arc always implies an embedded pair-wise exchange.

2.3.5 3-SIDED CYCLIC NETWORKS

Knuth proposed the concept of 3-way stable matching in the year 1976 in [34] with three types of agents men-women-dogs, popularly known as three gender SMP. Matching-set is formed of disjoint families, in the form of triples. Such a triplet is known to be stable, if no blocking family exists i.e. preferred to by all members in current families in a match. In 1988, Alkan [35] gave an instance where no stable matching exists in 3DSM, which was further filtered by Ng and

Hirschberg in 1991 and it was shown that the problem of deciding the existence of solution for 3DSM is NP Complete[36]. In 1994, an alternative proof was given by Subramanium[37]. In 2002, Manlove proposed a solution to find SMP of maximum cardinality for an instance of SMTI, and proved that in some cases it is NP Hard. Later, in 2004 Boros et al [38] proved that for *n* equals to 3 stable matching always exists, which was further extended for n equals to 4 by Eriksson et al in 2006 [39]. Huang in 2007 [40] showed that the problem remains NP complete for strong stability even if the preference list is consistent. A preference list is inconsistent if, for example, man m ranks (w1, d1) higher than (w2, d1), but he also ranks (w2, d2) higher than (w1, d2), so he does not consistently prefer woman w1 to woman w2. Structural and #P completeness results for string stable matchings. Later in 2009, Peter brio and Eric McDermid [41] showed that given a stability criterion, strong stability (there exists no weakly blocking family), the problem is NP complete even for complete lists. Taking incomplete lists into consideration without occurrence of ties, Fig. 5 shows that no stable matching exists where n equals 6.



$m_1 : w_1, w_1'$ $w_1 : d_1, d_1'$ $d_1 : m_2, m_2'$
$m_2 : w_2, w_2'$ $w_2 : d_2, d_2'$ $d_2 : m_3, m_3'$
$m_3 : w_3, w_3'$ $w_3 : d_3, d_3'$ $d_3 : m_1, m_1'$
$m_1' : w_3$ $w_1' : d_3$ $d_1' : m_1$
$m_2' : w_1$ $w_2' : d_1$ $d_2' : m_2$
$m_3' : w_2$ $w_3' : d_2$ $d_3' : m_3$

Fig 4: Non-existence of 3DSM stable matching for n=6

The main questions that remain unsolved are

- whether there exists an instance of cyclic 3DSM that admits no stable matching, and
- whether there is a polynomial time algorithm to find such a matching or report that none exists, given an instance of cyclic 3DSM.

Table 11: Success of Stable Mechanism

Market	**Stable?**	**Still Used?**
NRMP	Yes	Yes
US Medical Specialties (about 30)	Yes	yes*
For UK Residency Match		
Edinburgh	Yes	Yes
Cardiff	Yes	Yes
Birmingham	No	No
Newcastle	No	No
Sheffield	No	No
Cambridge	No	Yes
For Other Market		

London hospital Canadian lawyers	No Yes	Yes yes*
Pharmacists	Yes	Yes
Reform rabbis Clinical psychologists	Yes Yes	Yes Yes

2.4 STABLE MATCHING PROBLEM REPRESENTATIVE

This section gives brief introduction to some classes of problem and algorithm used to solve them. Those problems are as follow:

a) *Graph Theory and Stable Matching:*

A bipartite graph is an extension of the stable matching problem that models situations when objects are assigned to other objects. The goal is to find a matching of maximum size. The algorithm inductively builds up the matching, and is representative of a class of problems called network flow problems.

An instance of bipartite graph in stable matching is represented by G =(M U W, E) where M and W represent the graph node and E represent the possible matching, also adjacency list are linearly ordered with ties. The possible condition of stable matching in terms of bipartite graph are stated as below

- If edges (m, w) and (m, w_i) are tied then 'm' is indifferent between w and w_i.
- If edge (m, w) proceeds (m, w_i) then m prefers w to w_i.
- Stable matching problem==complete, if |men| = |women| and G is complete bipartite graph with $a = (b/2)^2$ here a = number of node and b = number of edges.

- Matching M= Set of edges no two of which share a end-point.

If (m, w) Є M we say that *a* and *b* are partners. An edge e = (m, w) Є E\M is a blocking pair if m is unmatched/ m strictly prefers w to his or her currently matched partner in M and w is unmatched/ w strictly prefers m to his or her currently matched partner or is indifferent between them i.e. if m prefers to match with w, w would not object.

- Blocking pairs can the solved locally by divorcing the current partners and marrying each other, but it forms a cycle if the divorced partners marry each other (though the chances are rare).

In Bipartite Graphs An instance of stable marriage is considered as bipartite graph, *G=(XUW,E)*, where *G* denotes the graph with *X* , *W* vertices and *E* edges. Adjacency lists are linearly ordered with ties being allowed. Considering *n* as no. of nodes (men/women) and *m* as no. of edges an edge can be denoted as (a, b). In case *(a, b)* and *(a, b')* are tied, we say that *a* is indifferent between *b* and *b'*. If edge (a, b) preceeds (a, b'), then we can say that a prefers b to b'. Stable matching problem is said to be complete, if number of men equals to number of women and hence, G is complete bipartite graph with $m = (n/2)^2$. In a graph, a matching □ is a set of edges where no two of which share an end-point. If (a, b) Є □ we say that a and b are partners. An edge E = (a, b) Є E\M is a blocking pair if a is unmatched/ a strictly prefers b to his or her currently matched partner in M and b is unmatched/ b strictly prefers a to his or her currently matched partner or is indifferent between them i.e. if a prefers to match with b, b would not object. Blocking pairs can the solved locally by divorcing the current partners and marrying each other, but it forms a cycle if the divorced partners marry each other (though the chances are rare). Iriving in 1994 showed that the time complexity, for computing strongly stable matching in complete instances is $O(n^4)$. In 1999, Manlove showed it for incomplete bipartite graphs to be $O(m^2)$. Later in 2003, an improved Iriving algorithm was proposed taking O(mn) time.

m_i	Pref_list(m_i)	w_i	m_1 w_1 m_2 w_2

				Pref_list(w_i)	
m_1	w_1	w_2	w_1	m_2	m_1
m_2	w_1, w_2		w_2	m_2	m_1

Table 12: Problem Instance for SMP in graphs

Here both women prefer to m_2 over m_1. m_1 prefers w_1 over w_2 and m_2 is indifferent between women, w_1 and w_2. The matching $\{(m_1, w_1), (m_2, w_2)\}$ is not strongly stable since, w_1 prefers to m_2 over m_1 and m_2 is indifferent between w_1 and w_2.

b) *Competitive facility Location:*

This is a game-playing problem where players alternate choosing valued locations, wanting the maximum possible value, but they are limited because the set of all chosen locations must be an independent set. The problem is whether, given a target value, if one player can reach the target, regardless of what other player(s) do. This problem is in the class of PSPACE-complete problems, which are difficult to solve and prove.

c) *Multi Interconnection Network and stable matching:*

It has been proved that when network is multistage interconnection Network (MIN) then stability problem becomes equivalent to Stable matching problem. The objective is to overcome the problem of network instability. This problem has been mentioned by A. subrmanian and nitin considering different network i.e. regular, irregular, hybrid with ties, colouring scheme.

d) *Interval Scheduling:*

In this problem resources are given say lecture room but participant can use it for a period of time. The objective is to use resource such that usage does not overlap at a given period of time.

CHAPTER 3

PRESERVING BASIC PROPERTY OF STABLE MATCHING

Till previous section we are cleared with basic approach and various work done by various authors in order to improve the approaches to stable matching problem. But In some instances it came out to be a fact that man-oriented approach led to a man-pessimal matching where they did not get their best possible partner contrary to what Gale-Shapely proposed which leads to a worst case scenario. Though it has already been experimentally proved that the chances of having a worst case scenario for stable matching is extremely low, but occurrence of it prevents the parallel algorithm for stable matching to run, which could have reduced the running time dramatically. Therefore, this section targets the above issue and solves it by deleting least happy pair from the already formed matching set using the score-based approach.

• Basic Notation

Basically, the stable matching problem considers two sets M and W each of size n. M is a matrix of n-men along with the respective preference list for women. Similarly, W is a matrix of n-women along with the respective preference list for men. Here, we are considering the preference lists to be complete and strictly ordered. A complete list is such that a man needs to specify the ranks for all of his partners that are participating in the game, and a strict ordering of lists puts a bound that man needs to be clear about his thoughts for the preferences of his partners and therefore he cannot assign the same rank to more than one partner. In any instance of the matchmaking problem we uniquely match each man in set M with its woman (partner) in the set W for a man-oriented approach and vice versa if it is a woman-oriented approach, which means that GSA is partial. Either it favours men leading to a man optimal and woman pessimal solution or the other way round. To achieve global optimality with respect to both the sides we have egalitarian approach with a time complexity, $O(n^4)$.

Given a problem-instance, I_i a matching □ is a pairing of man M_i to woman W_j. If (M_1, W_3) belongs to □, then we can say that M_1 and W_3 is a couple. A complete one to one matching of each man in set M to each woman in set W uniquely is known as a marriage. If a man and a woman in different couple in the matching set □ prefers to each other to their present partner then we say we have a blocking pair and the marriage is not stable. Therefore, a stable matching is a marriage with no blocking pairs. Rank refers to the priority (position) in a person's preference list of his or her partner. We will denote the rank of woman W_j for man M_i in his preference list as $RM_i(W_j)$. In this paper we are considering score as the sum of all the ranks in □, denoted by S□(I).

In the basic Gale Shapely algorithm it has been clearly specified that, in a man-oriented stable matching, it should always be man-optimal. But in some instances, we have men not assigned their highest possible ranked partners leading to a man-pessimal matching. For such instances merit of the Gale Shapley algorithm cannot be used efficiently. Therefore, has tried to eradicate the worst case scenario in $O(n^3)$ time, by making minimal changes in the preference list of the contestants. In paper we are trying to reduce this time complexity to some extent by achieving a lower bound of (n^3) by considering an alternative score based approach.

Here we illustrate the scenario with the help of an example. Suppose we have set of 4 men and 4 women. Each preference list is ordered in increasing order from left to right as shown in Table 19. Lower the priority higher is the preference.

Man_i	**Pref. List (PL_{Mi})**	**$Woman_j$**	**Pref. List (PL_{Wj})**
$M_{1:}$	W_1, W_2, W_3, W_4	$W_{1:}$	M_2, M_1, M_3, M_4
$M_{2:}$	W_3, W_1, W_2, W_4	$W_{2:}$	M_1, M_2, M_3, M_4
$M_{3:}$	W_2, W_4, W_3, W_1	$W_{3:}$	M_4, M_3, M_1, M_2

Table 19: Instance I_0: Set of 4 men and women along with their respective preference lists			
$M_{4:}$	W_2, W_3, W_4, W_1	$W_{4:}$	M_3, M_2, M_1, M_4

If we follow the general GSA we will end up having:

□ = {(M_1, W_2), (M_2, W_1), (M_3, W_4), (M_4, W_3)}

As we can see in the men-table (Table19) we have all the men getting paired up with their second preferences, and the women get the men from their first preferences (Table 20).

Assignment of women for men in the men-table	
Man_i	**Pref. List (PL_{Mi})**
$M_{1:}$	W_1, **W_2**, W_3, W_4
$M_{2:}$	W_3, **W_1**, W_2, W_4
$M_{3:}$	W_2, **W_4**, W_3, W_1

$M_{4:}$	W_2, **W_3**, W_4, W_1

Assignment of women for men in the women-table	
$Woman_j$	**Pref. List (PL_{Wj})**
$W_{1:}$	**M_2**, M_1, M_3, M_4
$W_{2:}$	**M_1**, M_2, M_3, M_4
$W_{3:}$	**M_4**, M_3, M_1, M_2
$W_{4:}$	**M_3**, M_2, M_1, M_4

Table 20: Worst case of Instance I_0

Gale Shapely Algorithm says that in case of a man-oriented approach a man always gets its best possible partner and a woman its worst possible partner i.e. it should be man-optimal and woman-pessimal. But the result we got in this case is a man-pessimal and woman-optimal solution. As the result shows, we have women happier than men when men initiate the proposal, contrary to what GSA says.

- # MODIFIED GALE SHAPELY ALGORITHM (MOD_{GSA})

The above result we got is the worst case scenario where the proposing party who is expected to be happier than the non-proposing one is sad rather. It has been proved that if there are 16 men and 16 women then the probability that the worst case occurs is 10^{-45}, which is very low [64]. Parallel algorithm is based upon divide and conquer principle having a time complexity of n^2-2n+ [log n] as stated in [64], which is better than the time complexity of basic Gale-Shapely

Algorithm. But, parallel algorithms do not work for the worst case. We can avoid such worst case scenario to some extent by following the Modified GSA proposed in this paper.

In this algorithm we are taking as input a matrix of M for men and W for women, and their preference lists ordered according to their priority. We apply GSA on the basic problem instance Io and we denote the matching set found, by □0. For say, we have a set of 4 men and 4 women then the matching set formed will have 4 pairs with each man paired with his respective partner, we then denote the matching set □$_0$=[p1, p2, p3, p4] where pi denotes a pair i. Therefore, we can say that the matching set is a matrix with n-pairs for n being the size of the problem. Though the problem size is considered n × n, but for simplicity we will consider it n throughout the paper.

The for-loop in the Modified Gale-Shapely Algorithm runs for each pair which is given by the problem size only i.e. n. For each pair we delete the pair first. Then we apply GSA to the new matrix set of (n-1) men and (n-1) women, leading to a matching set □$_i$. Finally we calculate the score of □$_i$, as S□$_i$(I_i). We continue to do so for the entire pairs pi in the original GSA matching set, and select to delete the pair with the minimum score, S_{min}. Therefore, the GSA matching set retained now has the minimum score. In the matching set □, the score is found as the sum total of all the ranks of the partners in the preference list of the proposing party.

Table 21: MODIFIED GSA(MODGSA) ALGORITHM
Input: The problem instance with the men matrix M and the women matrix W along with their preference lists. **Output:** A man-optimal matching set, □i for a man-oriented approach. **Precondition:** The problem instance should produce the worst case scenario.
1. Calculate the score for the problem instance I_0

2. For each pair in matching set $\Box_0$ do

3. Delete the pair pi

4. Form the matching set $\Box_i$

5. Calculate the score, $S_{\Box}(I_i)$

6. End for

7. Delete the pair pi for which the score, $S_{\Box}$ is minimum

8. Output the matching set $\Box_i$ for which the pair has been deleted.

Time Complexity: The time complexity T(n)GSA of the above algorithm is given by O(n3).

Proof of Complexity or Correctness: The Gale Shapely algorithm takes $O(n^2)$ for a problem size n. We have the for-loop run for each pair. For a problem size n we always end up having n pairs. Therefore, we get the complexity to be calculated as:

Time Complexity, $T(n)GSA = (n-1)^2 \times n$

$$=O(n^3)$$

We can improve this time complexity by following parallel GSA (MODP-GSA) instead of basic GSA at line 4. At line 1 we will follow the basic GSA. Here we have made possible for the parallel algorithm to execute successfully with high probability in case of a worst case scenario by deleting or ignoring one couple from the matching set $\Box_0$.

We can improve this time complexity by following parallel GSA ($MOD_{P\text{-}GSA}$) instead of basic GSA at line 4. At line 1 we will follow the basic GSA. Here we have made possible for the parallel algorithm to execute successfully with high probability in case of a worst case scenario by deleting or ignoring one couple from the matching set $\Box_0$.

- # IMPLEMENTATION DETAIL:

This section describes the implementation details of the algorithms both for MODGSA and MODP-GSA. Then we have done a comparative analysis of these algorithms taking number of steps as the parameter. At a later stage we represented our results graphically for both space complexity (number of steps) and time complexity (T(n)). Finally, we analyse the performance metric enhancement for MODP-GSA w.r.t. MODGSA.

MOD_{GSA}Algorithm Explanation:

Here we consider the Table 18 and based upon it we will describe our algorithm for MODGSA in a stepwise manner. We will consider the Modified Parallel GSA denoted by MODP-GSA in the next section of implementation.

For problem instance in Table1 the following steps describe the flow of the algorithm.

Step 1: Applying GSA we get the matching

$\square_0 = \{(M_1, W_2), (M_2, W_1), (M_3, W_4), (M_4, W_3)\}$

The score, $S\square(I_o)= 2+2+2+2=8$.

The score we will calculate at a later stage should come less than this as we are trying to maximize happiness for men. This constraint would verify the correctness of our algorithm as less is the score more is the happiness.

Step 2: For deletion, we need to consider each pair in M. As for a problem size n we always end up having n-pairs, this for loop will run for n-times.

Step 3: We proceed first by considering $\square_0[0]$ i.e. (M_1, W_2). Now we are left with the matrix:

Man$_i$	**Pref. List (PL_{Mi})**	**Woman$_j$**	**Pref. List (PL_{Wj})**
$M_{2:}$	W_3, W_1, W_4	$W_{1:}$	M_2, M_3, M_4
$M_{3:}$	W_4, W_3, W_1	$W_{3:}$	M_4, M_3, M_2
Table 22: Reduced Table $M_{4:}$	W_3, W_4, W_1	$W_{4:}$	M_3, M_2, M_4

Step 4: Applying GSA, □1 = {(M_2, W_1), (M_3, W_4), (M_4, W_3)}

Step 5: For the above reduced problem instance I1, score is given by,

$$S□(I_1) = 5$$

Step 6: Similarly doing it for all other pairs in □, we have

Delete (M2, W1): □$_1$={(M_1, W_2), (M_3, W_4), (M_4, W_3)}

$$S□(I_2) = 6$$

Delete (M3, W4): □$_1$={(M_1, W_1), (M_2, W_3), (M_4, W_2)}

$$S□(I_3) = 3$$

Delete (M4, W3): □$_1$ = {(M_1, W_1), (M_2, W_2), (M_3, W_4)}

$$S□(I_4) = 4$$

Step 7: Choosing the pair with minimal score we delete (M3, W4), and we are left with the matrix:

Man$_i$	**Pref. List (PL_{Mi})**	**Woman$_j$**	**Pref. List (PL_{Wj})**

$M_{1:}$	**W_1**, W_2, W_3	$W_{1:}$	M_2, **M_1**, M_4
$M_{2:}$	**W_3**, W_1, W_2	$W_{2:}$	M_1, M_2, **M_4**
$M_{4:}$	**W_2**, W_3, W_1	$W_{3:}$	M_4, M_1, **M_2**

Table 23: Result Table

Table23 clearly shows that now the men get their first preferences and women either their second or third, resulting into a man-optimal and therefore woman pessimal solution.

As we can see here, our problem size has been reduced to (n-1), but as we go on increasing the value of n, this hardly matters, if by doing so we get an overall happiness and preserve the basic property of Gale-Shapely Algorithm by reducing the occurrence of a worst case.

MODP_GSA Algorithm Explanation:

Parallel GSA follows divide and conquer principle to solve the matchmaking problem in a parallel way taking (n^2-2n+log(2n)) steps, where n is the size of the main problem. As the name indicates, this has got two phases i.e. the division phase and the conquering (merging) phase. The division of the problem into sub-problems led to a tree like structure and problems at the same tree level are solved in a parallel fashion to produce a partial matching set which is then merged to form a higher level match. The conflict where a single man is matched twice at the same level with two women is solved by consulting the women's preference list. This whole process continues until we get the final result.

As cited in [64], parallel GSA does not work is a worst case scenario, therefore the input to MODP-GSA i.e. the matching set □0 is calculated following the GSABASIC, which takes at most n^2 number of steps in a worst case. Inside the algorithm where we obtain the matching set □i we will use the parallel GSA. Even here there is a chance that the worst case scenario may occur. But it has already been stated that the chances of the occurrence of worst case are very rare and the rarity increases even more as we are searching for a worst case within the worst case.

For problem instance in Table19 the following steps describe the flow of the parallel algorithm rest is same as MODGSA algorithm

Division Phase: During this phase, problem is divided into individual sub-problem containing each sub-problem contains single men. Thus each man can propose his choice as shown in below Fig 15.

(M4, W2)

(M3, W2)

(M2, W3)

(M1, W1)

Fig 15: Division Phase

ALGORITHM A

(An algorithm which produces a male optimal stable solution)

Input: A male ranking matrix and a female ranking matrix.

Output: A male optimal stable solution.

Step 1: Divide the problem into two sub-problems, by halving the male ranking matrix. Call these two sub-problems P1 and P2.

Step2: Recursively apply this algorithm to find male optimal stable solutions for P1 and P2- Call these two solutions S1 and S2.

Step3: Apply Algorithm B which is a merging algorithm to combine S1 and S2 into S.

In this algorithm, a merging procedure is used. Let us first introduce some definitions. In a solution, suppose Mi_1.... Mi_k , $k >= 2$, propose to the same woman Wi. Without loss of

generality, we shall assume that so far as Wi is concerned, the ranking of Mi_k is the highest, i.e. it will accept Mi_k and will reject m1…..Mi_{k-1}. We shall say that {Mi_1, ... Mi_{k-1} } is the set of rejected men of Wi. In a solution S, let Wi denote the set of women who are proposed to by more than one man. Let R_s be the union of the sets of rejected men of members in W_s. Then R_s is called the set of rejected men associated with S.

Merging Phase:

(M4, W2)

(M3, W2)

(M2, W3)

(M1, W1)

(M1, W1) (M2, W3)

(M3, W2) (M4, W2)

Fig 16: First Merging Phase

In fig 16, it is noticeable that participant M3 and M4 has same priority for W2. Therefore conflict is resolved by choosing more preferable person from women list. The resultant is shown in Fig17. In Similar fashion, merging phase iterate until stable match is found.

(M4, W2)

(M3, W2)

(M2, W3)

(M1, W1)

(M1, W1) (M2, W3)

(M3, W2) (M4, W3)

(M2, W1)(M3, W4)(M4, W3) (M2, W1)

Fig 17: Resultant to Parallel Approach

Fig 17 shows the final result after applying parallel approach to GSA.

ALGORITHM B

(A merging algorithm which produces a male optimal stable solution out of two male optimal stable sub-solutions)

Input: Two male optimal stable solutions S1 and S2 and their associated ranking matrices.

Output: A male optimal stable solution which combines S_1 and S_2.

Step 1: Let S be the union of S_1 and S_2.

Step 2: If no two men propose to the same woman in S, then accept S as the solution and return. Otherwise, go to Step 3.

Step 3: For each man M_i in R_s, and for the set of rejected men associated with S, replace (M_i, W_j) in S by (M_i, W_k) where W_k is the next best choice of M_i.

Step 4: Go to step 2.

When the research was done based upon the above algorithms, the following results were concluded:

- For a worst case execution of McVitie-Wilson's algorithm the following two statements are true:

  STATEMENT 1: Let M_k finally propose to this N^{th} choice W_i. Then the Nth column of the male ranking matrix consists only of one woman W_i and the $(N-1)^{th}$ column consists of all the other N-1 women.

STATEMENT 2: Let M_j propose to his $(N - 1)^{th}$ choice W_a. Then the first choice of W_a must be M_j.

- If $N^2 - N + 1$ proposals are needed in McVitie-Witson's algorithm to obtain the male optimal stable solution, then the probability that this worst case occurs is of the order $CN^{-2N+7/2}e^{-N}$ with $2.5 < C < 3.5$.

- The number of steps needed to obtain the male optimal stable solution by using our parallel algorithm is at mostN^2-2N+floor ($\log_2 N$).

- In the worst case of our parallel algorithm the first column of the male ranking matrix consists of exactly N-1 women.

- The probability that the worst case occurs in parallel algorithm is less than $CN^{-2N+5}e^{-2N}$ with $30<C<70$, for $N>2$.

CONCLUSION: Parallel algorithms are slower than the sequential ones and they get even slower as the number of processors increase if no special processors are used [66]. Communication is expensive also.

As parallel GSA does not work is a worst case scenario, therefore the input to MODP-GSA i.e. the matching set □0 is calculated following the GSABASIC, which takes at most n^2 number of steps in a worst case. Inside the algorithm where we obtain the matching set □i we will use the parallel GSA. Even here there is a chance that the worst case scenario may occur. But it has already been stated that the chances of the occurrence of worst case are very rare and the rarity increases even more as we are searching for a worst case within the worst case.

Time Complexity: The time complexity, $T(n)_{P\text{-}GSA}$ of the Modified Parallel GSA is given by $O(n^3)$.

Proof of Complexity or Correctness: The Parallel Gale Shapely algorithm takes (n^2-2n+log2n) number of steps for a problem size n. As for-loop runs after we delete a pair, the problem size reduces to (n-1). This for-loop runs for each pair. For a problem size n we always end up having n pairs. Therefore, we get the complexity to be calculated as:

$$\text{Time Complexity, } T(n)_{P\text{-}GSA} = ((n-1)^2-2(n-1)+\log2(n-1)) \times n$$

$= O(n^3)$

- **Comparison (MOD_{GSA} 'vs' $MOD_{P\text{-}GSA}$)**

The time complexity of both the algorithms MODGSA and MODP-GSA is given by $O(n^3)$ when calculated. Taking the worst case scenario as input and the number of steps required as the parameter here we have done a theoretical comparative analysis. Finally we have deduced the performance enhancement for MODP-GSA in comparison to MODGSA and we concluded that as the value of n increases the performance enhancement metric goes on giving better results.

NUMBER OF PEOPLE IN EACH SET (n)	**NUMBER OF STEPS**			**PERFORMANCE ENHANCEMENT IN $MOD_{P\text{-}GSA}$ W.R.T. MOD_{GSA}**
	GSA	**MOD_{GSA}**	**$MOD_{P\text{-}GSA}$**	
3	9	12	6	LOW
4	16	36	20	LOW
5	25	80	55	LOW
6	36	150	108	INTERMEDIATE
7	49	252	189	INTERMEDIATE
8	64	392	304	INTERMEDIAE
9	81	576	468	INTERMEDIATE
10	100	810	670	INTERMEDIATE

11	121	1100	924	HIGH
12	144	1452	1236	HIGH
13	169	1872	1612	HIGH
14	196	2366	2058	HIGH
15	225	2940	3184	HIGH
16	256	3600	3184	HIGH

Table 24: Comparative Analysis

Representing the data from Table 24 in a graphical form we obtain Fig 18 showing the comparison of performance for each algorithm. Here also we can see that the difference between the peak points for MODGSA and MODP-GSA keeps on increasing as the value of n increases.

Figure 18: Comparison graph considering 'No. of Steps' as a parameter

We know that the number of steps required for an algorithm to run is directly proportional the time it will take to run on any machine. When we run the algorithm for various problem instances we obtained the graph given in Figure 19. The graph shows the variation in time complexities for MOD_{GSA} and $MOD_{P\text{-}GSA}$ and the edge $MOD_{P\text{-}GSA}$ obtains over MOD_{GSA} for larger values of n.

Figure 20: Comparison of Time Complexities of MOD_{GSA} AND $MOD_{P\text{-}GSA}$

3.6 Conclusion

Explores the worst case scenario of Gale Shapely Algorithm (GSA) and improves it by deleting one pair to achieve greater happiness. For this we have proposed an algorithm, MODGSA, based on GSABASIC which takes $O(n^3)$ time. Further, we have been trying to decrease this time complexity, by following the parallel GSA (GSAPARALLEL), denoted by MODP-GSA. We have taken a number of steps to dry - run the algorithm as our parameter and represented our results both in tabular and graphical form. However, on comparing we deduced the result that MODP-GSA gives better performance than MODGSA for higher values of n and hence achieving greater stability.

Chapter 4

STABLE MATCH APPROACH TO DETERMINE VITAL LINK TO PRESERVE SHORTEST PATH LENGTH

4.1 Introduction

Content Delivery Network is a large distributed system consisting of multiple servers deployed in much geographical location that delivers the content to end user with high performance and high availability. But the increase load of content delivery server and network degrades the quality of service. Also 70% of unintentional failures are Single link failure which creates potential failure along the delivery chains. To get over the problem some mirror server are placed in the network where a request is transferred to one of the mirror servers such that connectivity between user and server is within small hop count even during the failure. This paper targets the stability of network reliability by providing hop length stability and network connectivity. We use the stable matching approach on the network to find the vital link to be protected so that user can access the server within small hop count even if non – vital links fails. Moreover, we prove that the problem can solved in polynomial time conditioned when hop count is determined to be one.

In last seven years Internet access has established web as a mass market media channel. Where the Internet has become essential part of office life and plays a key role for home users. The number of people with web access continues to rise resulting into more Internet resources to use Internet application well-situated. Hence it is crucial to ensure comfortable and easy access for all users to server by Content Distribution Services/Network (CDS/CDN). For this reason mirror server are situated in the network to serve the same content shown in fig21. A request is transferred from user to one of the mirror server to proportionate the load and reduce the delay time. In construct many content distribution provider use mirror servers to get advantage over less delay time and high reliability against failure.

The content delivery network must serve connectivity between user and server within small hop count even in the case of link failure. But under normal condition, we face the risk of failure on many links due to break down of fibres, failure of transmission devices, etc [1]. Therefore, network must be robust enough to hold the connectivity between nodes and preserve the stable services in consequences of failure. To fulfil these requirements the simplest way to provide better connectivity between nodes during failure is to deploy many large capacity links but it leads to high operational and investment cost. Another way is to provide users with high reliable services at a lost cost by designing optimum network topology while minimizing the total cost from given traffic demand matrix [2]. But the traffic demand matrix is hard to estimate accurately and as the traffic demand continues to change, the network optimality degrades.

Further resilient schemes have been provided to protect user traffic from network disturbances which use combination of restoration and protection scheme providing the benefit of sufficient spare capacity and avoiding sharing network facilities so to give no impact on Internet services.

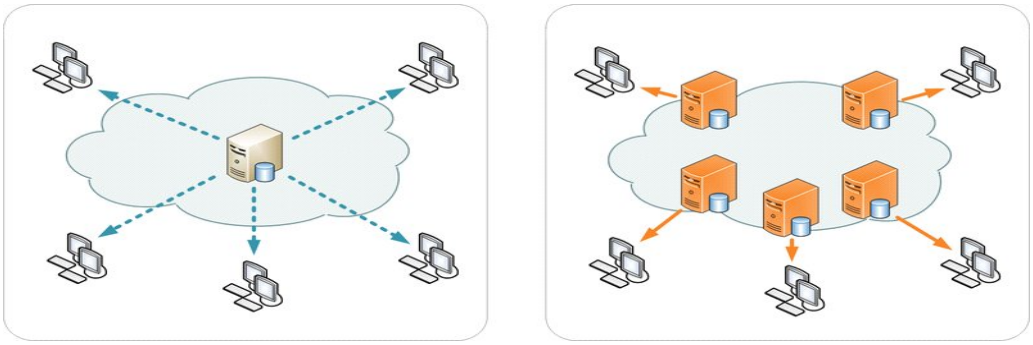


Fig 20: a) Traditional Scheme of Distribution b) CDN scheme of distribution

Restoration scheme is available at IP layer which actively looks for backup paths and distance between nodes in the network. The protection scheme is used at transport layer which reserves dedicated backup path and distance between nodes in the network. Therefore if recovery is provided at lower layer then the link between IP routers is considered not to be failed because failure of the link is quickly recovered and is not detected over the IP layer. Such link are said to be protected link. Practically it is not possible to protect every link as it takes much of network resources which lead to high investment and operational cost for network service provider.

This paper focus on above issues where only vital links are protected whose failure significantly degrades the performance of the network. The Stable Matching approach [3] is applied on the

network in order to reduce the cost by finding vital link to be small. Here network is considered to be set of users and set of servers. Where each client and server is described by their respective profile that aggregates information like online- availability, bandwidth capacity, etc. Based on the profile, two Matrix is build one for client set and other for server set. In Client Matrix, each client provides the preference list for the server and vice versa. Based on the preference list the stable matching approach is applied to find the smallest number of link to be protected such that accessibility from users to servers is within small hop count. Furthermore, the problem can be solved in polynomial time conditioned when hop count is one.

The remainder of the paper is organized as follows. Section 2 addresses the related work done so far. Section 3 deals with basic concept, notation and parameter that have been considered in this paper. Section 4 and 5 describes the proposed algorithm and implementation details with its result. Finally Section 6 concludes the paper and discuss on future work.

4.2 NETWORK DESIGN USING STABLE MATCHING

In general stable matching problem consists of two sets. Here according to our problem, we consider two set as a set of users say U of size n and set of servers say M of size m. The set is defined as a matrix (say U) of n user along with their preference list for suitable servers. Similarly m-set of servers is defined as a matrix M of m servers along with their preference list for suitable client. The preference list is build on the basis of profile. Profile is tables that is maintained by each client and server and provide aggregate information that follows the network related criteria such as on-line availability, bandwidth capacity, etc shown in fig22.

- Network related criteria:

The criteria for client and server may be different. The most precise and generic are discussed below:

- *User profile U_i :*

  User maintains a profile containing information such as distances between users i and server j, maximum amount of bit rate successful transferred, etc as mentioned below:

  • Bandwidth ($B_{i,j}$): It is defined as maximum bit rate of successful message delivery over communication channel.

  • Reliability ($R_{i,j}$): Reliability is the probability that the link between two user i and server j is working without failure

  • Performance ($P_{i,j}$): The performance is turnaround time from request being sent to when the response is received.

  • Proximity ($D_{i,j}$) : The proximity is the measure of distance between the geographical distance between user i and server j.

For proximity measurement we use great circle distance method. In Euclidean space the distance is measured between two points that is the length of straight line between them. But on sphere there is no straight line. Therefore, great circle distance measure the shortest distance between two points on surface of the sphere. The requirements are geographical latitude of two points given by $\square_i$, $\square_j$ and longitude of two points given by β_i, β_j. Their absolute distance is given by $\square$, β, then the central angle between them is represented as:

$$\square = arccos(Sin\ \square_i Sin\ \square_j + Cos\ \beta_i\ Cos\beta_j\ Cos\ \beta\ ) \qquad (1)$$

Hence, the distance d between two points is the arc length for radius r and $\square$ given in eq2.

$$D_{i,j} = r.\ \square \qquad (2)$$

- Server Profile (S_j):

Each server maintains its profile in form of table where each column maintains information such as:

• Frequency of Number of resources accessed by the user.

• Availability: Availability is expressed as percentage of uptime in a week. Availability is equated as given in equation 3.

$$Aj = (\text{total time} - \text{down time}) / \text{total time} \qquad (3)$$

For our practical concern we consider the preference list to be strictly ordered and complete. Preference list is said to be complete if user specify the ranks for all his partners (servers). On the other hand the preference list is strictly order when user is clear about priority of the partner and hence cannot assign same rank to more than one partner.

After creating matrix with complete preference list the stable matching approach is applied in the network design where user initiate the request to respective server considering its preference list with respect to the matrix given shown in fig 21. The server responds to request if it is idle and available. And if the server is busy responding to another user and cannot connect with the initiating user then user initiates the request considering the next higher preference server. This process keeps on repeating until user is linked with most suitable server. This suitable link is said to be stable link between user i and server j.

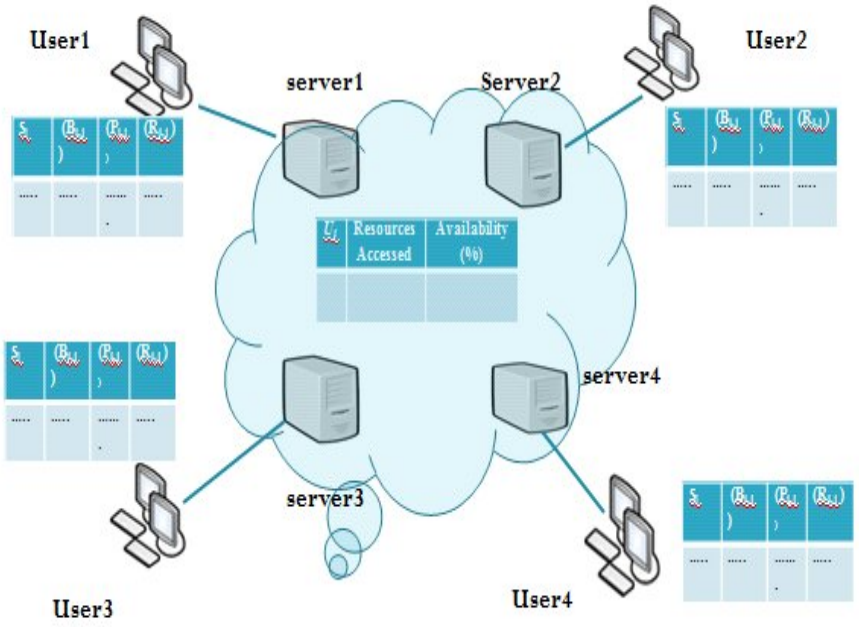


Fig 21: Content Distribution Scheme over the network with each user and server updated with respective profile table.

4.3 Network link Stable matching

As mentioned earlier stable matching approach is adopted to determine the vital link so that stable connectivity is maintain between user and server with small hop count. This section explains two proposed algorithm in order to target the objective of the paper.

4.3.1. Network Stability Algorithm:

This algorithm targets to determine the vital links between the user and server. The size of user may differ when compare with the size of server because in the network the total number of server is less than the total number of user. But the network is designed in such a way that some or all server are able to deliver the requested service to the entire user. The parameter to determine whether single server can process the multiple users is through Server load. Therefore, single servers can be paired with multiple users. Hence the whole complexity of the algorithm depends on size of participating user.

Table 25 : Network Stability Algorithm NSA(n,m)

Input: Matrix *U* and Matrix *S* build through profile of each user and server.

Output: Stable Link Between User and Server.

- Initialize all user u_i □ *U* and s_i □ S are free
- While(u_i is free)
- {
- *s* = highest preference server to whom request still not send **;**
- if(server *s* is available and capable to input more load)
- {
- Stable_Link = (u_i, s) link;
- $Server\_Profile_i$ [resource_accessed] is incremented by the amount of accessed resource;
- }
- else some link (u_j, *s*) already exist
- {
- if(*s* prefer u_i to u_J)
- Stable_link = (u_i, *s*) link;
- else
- Stable_link= (u_j, s) link;
- }
- Select the vital link from stable link if there is direct link from user to server else ignore the link.
- }

Time Complexity: The time complexity of NSA(*m,n*) is given by O(n)

Proof of correctness: The NSA(m,n) takes the problem size of *n* user and *m* server. The problem runs unless each user is linked with suitable server. In general number of participating server is very less than number of participating user, hence the algorithm ends with less than or equal to *n-1* stable link.

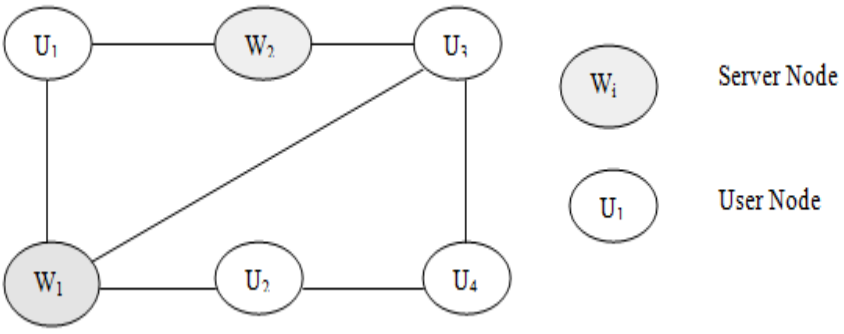


Fig 22: Example of vital Link determination

Table 26: i) Matrix of User *U* ii) Matrix of Server *S*

Matrix S

S_j / # user request	***Server j preferences from most to list***			
W_1 / 2	***U_1***	***U_4***	***U_2***	***U_3***
W_2 / 2	***U_1***	***U_4***	***U_3***	***U_2***

Matrix S

S_j / # user request	***Server j preferences from most to list***			
W_1 / 2	***U_1***	***U_4***	***U_2***	***U_3***
W_2 / 2	***U_1***	***U_4***	***U_3***	***U_2***

Matrix U

U_i	User *i* preferences from most to list	
U_1	W_2	W_2
U_2	W_1	W_2
U_3	W_2	W_1
U_4	W_1	W_2

Table26 is assumed on the basis of proximity parameter considering network shown in Fig 22. Also w_i / 2 indicate the at a time number of request can be initiated by server W_i. Applying, the above algorithm on the Fig 22, we get stable link as $\{(u_1, w_2), (u_2, w_1), (u_3, w_2), (u_4, u_1)\}$ but out of them vital link are select as $\{(u_1, w_2), (u_4, w_1), (u_2, w_1)\}$ shown in fig 23.

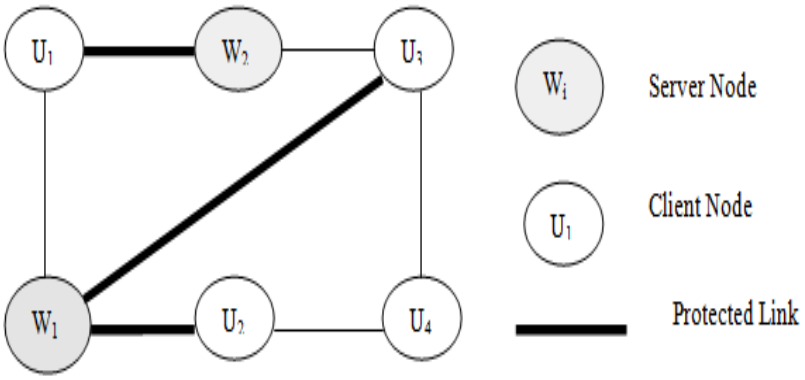


Fig 23: Network Link Stability Result

As we can notice if any of the link except the vital link fails simultaneously still user can connect with server within hop count of less than or equal to two.

4.3.2 Vital link stability from failure algorithm (VLSF):

This algorithm determines the vital link within small hop count even when failure is from one of the stable link. Considering fig 24 and assuming vital link $\{u_1, w_2\}$ fails. By applying above

algorithm we can obtain alternate vital link which comes out to be ({w_1, u_4}, {w_1, u_2}). In the algorithm score is determined by the summation of rank.

Table 27: Vital Link stability from failure (VLSF) algorithm

Input: Set of users and servers represented by Matrix *M* and *U*
Output: Vital link to be protected from failure
Pre-condition: Suspect of single point of failure

- **For** each link in the matching set of stable_link
- Delete the link *(u_i,s_j);*
- Apply the Network Stability Algorithm for *n-1* user and *m-1* server, NSA(*n-1*,*m-1*);
- Form the stable_link set;
- Calculate the score S_m determining minimum hop path length;
- **End for**
- Output the stable _link set stating vital link to be protected in case of failure with minimum hop path length.

Time Complexity: Time complexity of VLSF algorithm is $O(n^2)$

Proof of correctness: For "n-1" link it calls NSA() which itself takes O(n) time. Hence the VLSF algorithm complexity comes to be $O(n^2)$.

But if link {u2, w1} fails then path length from u_2 to any server is more than two. Hence lead to NP- complete problem if vital link fails.

Theorem 1: Vital Link Stability from failure is a NP – Complete

Theorem 2: Network stability Algorithm is solvable in a linear time.

4.3.3 Proof of correctness of Network Stability algorithm:

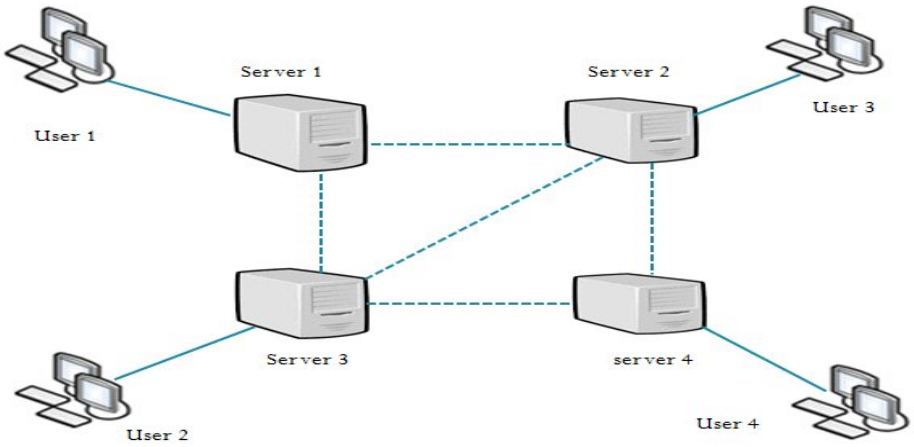


Fig 24: Network shows the connection between User and server

From Fig 24 we determine that user are only connected to single server hence there is only one path for user to get connected with server. If these links get failed then there is no other way for user to get connection with other server also. Therefore, to proof the correctness of network stability algorithm we randomly generate the Matrix of user U without considering any profile. Hence any preference list can be considered we consider table 4.i. But for server matrix S preference shown in table 4.ii is based on the user access because if we consider user1 there is always one link that is connected to server1 to provide access. Even if this link fails then user1 has no path way to get connected with any other of them**.**

Table 28: User Matrix and Server matrix

Server S_j	**Server j preference from most to least**			
S_1	**U_1**	**U_2**	**U_3**	**U_4**
S_2	**U_2**	**U_1**	**U_3**	**U_4**
S_3	**U_3**	**U_2**	**U_1**	**U_4**
S_4	**U_4**	**U_1**	**U_2**	**U_3**

User U_i	**Server S_j**	**Server j preference from most to least**			
	S_1	**U_1**	**U_2**	**U_3**	**U_4**
	S_2	**U_2**	**U_1**	**U_3**	**U_4**

	S_3	**U_3**	**U_2**		**U_1**	**U_4**
	S_4	**U_4**	**U_1**		**U_2**	**U_3**

	User i preference list from most to least			
U_1	S_1	S_2	S_3	S_4
U_2	S_1	S_3	S_2	S_2
U_3	S_4	S_3	S_1	S_2
U_4	S_4	S_1	S_2	S_3

Applying Network link stability algorithm, we get vital link as $\{(u_1, s_1), (u_2, s_2), (u_3, s_3), (u_4, s_4)\}$. Even using the vital link stability failure (VLSF) algorithm if any of them is failed still the vital link are resulted in the either set of $\{(u_1, s_1), (u_2, s_2), (u_3, s_3), (u_4, s_4)\}$.

Hence proves the correctness of network stability algorithm to determine the vital link and the shortest path of connection between client and server during failures.

4.5 Experimental Details

In this section we discuss simulation result to the specified approach. The experiment was setup covering geographical area within the campus shown in fig25. The campus connectivity supports the IEEE 802.11 standard supporting the Ethernet connection.

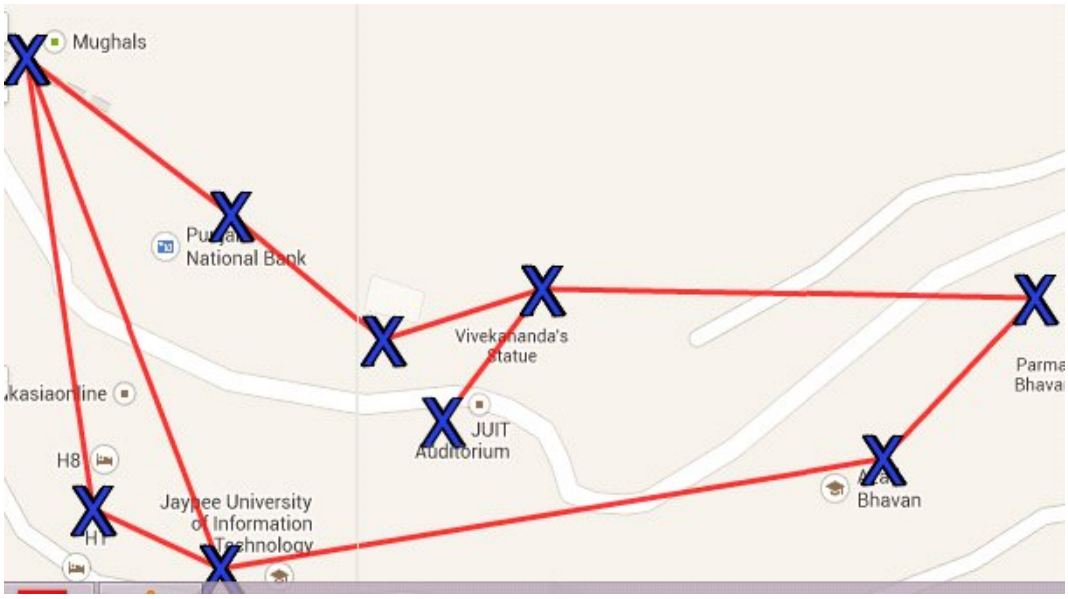


Fig 25: Geographical View of Client and Server Connectivity

The setup consist of six clients and three server connected through the Ethernet across different area in the campus shown as Fig 26. The connection between by the client and server is represented by the edge number. For testing each client sends the multimedia TCP packet through the dedicated transmission line and each server must be able to respond the request. For this purpose we use following tools

- Jperf 2.0.2 : Java Performance tool
- Distance Monitoring tool

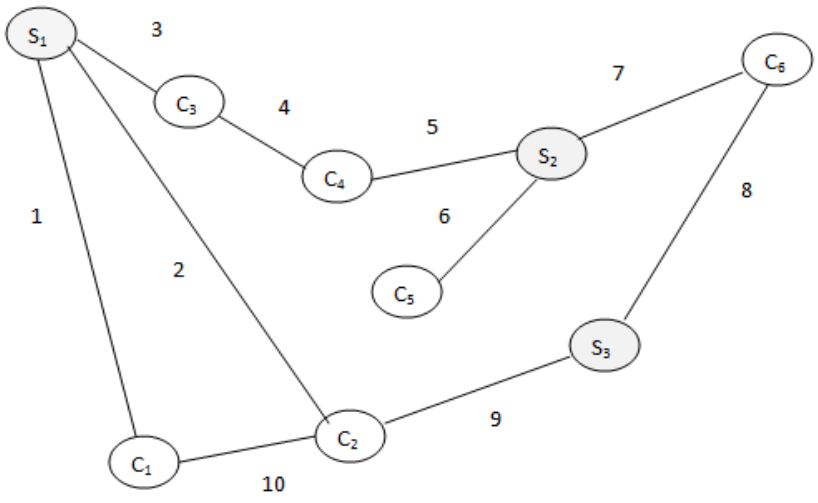


Fig26: Graphical representation to client server connectivity

Jperf version 2.0.2 is used for measuring network performance and bandwidth, distance monitoring tools is used for calculating the proximity between the links such as great circle distance measurement and the software JDK 1.7 is used for algorithm development. Fig 27 shows the use of Jperf tool for measuring the bandwidth at server side when different client requests and for measuring throughput at client side that is the total time taken from request to respond to client.

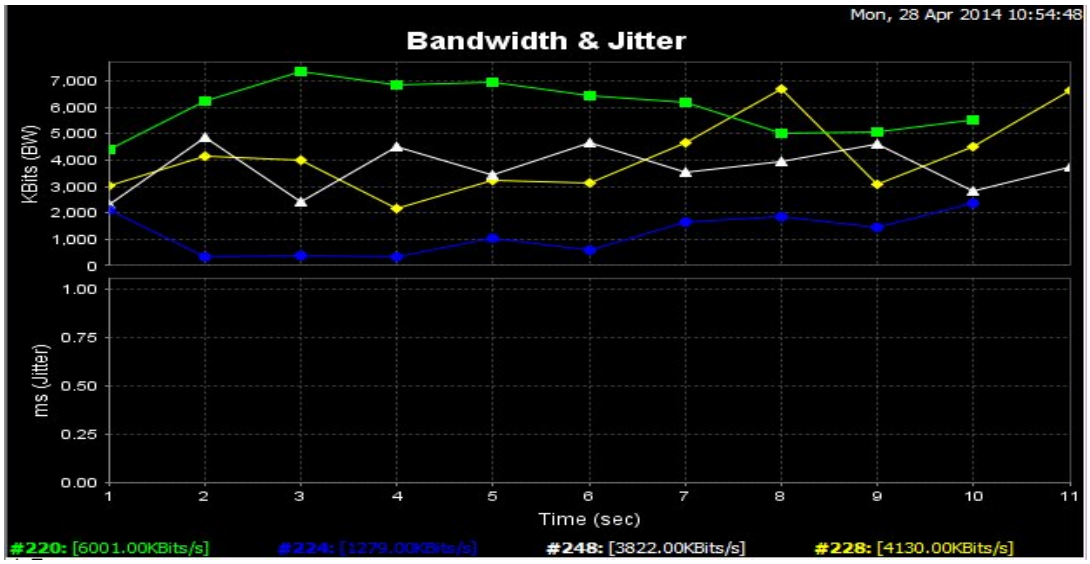


Fig 27: Bandwith Representation by Server 1 for 5 Clients in range

Circle #	Point #	Point/Center latitute	Point/Center longitute	Distance/ Radius	Azimuth	Total distance / Circumference
1	0	31 00 59 N	077 04 11E			
1	1	31 01 01 N	077 04 10E	70	-25	70
2	0	31 00 56 N	077 04 11E			
2	1	31 01 01 N	077 04 10E	173	-9	173

3	0	31 00 57 N	077 04 12E			
3	1	31 00 56 N	077 04 11E	64	-135	64
4	0	31 00 57 N	077 04 12E			
4	1	31 01 01 N	077 04 10E	144	-29	144
5	0	31 00 59 N	077 04 15E			
5	1	31 00 57 N	077 04 12E	77	-127	77
6	0	31 01 00 N	077 04 11E			
6	1	31 01 01 N	077 04 10E	56	-58	56
7	0	31 01 00 N	077 04 13E			
7	1	31 01 00 N	077 04 11E	44	-59	44
8	0	31 01 00 N	077 04 14E			
8	1	31 01 00 N	077 04 13E	39	-103	39
9	0	31 01 00 N	077 04 16E			
9	1	31 01 00 N	077 04 14E	51	-112	51
10	0	31 01 01 N	077 04 16E			
10	1	31 00 59 N	077 04 15E	67	-145	67
11	0	31 00 59 N	077 04 13E			
11	1	31 01 00 N	077 04 14E	46	36	46
11	2	31 01 00 N	077 04 14E	0.42	0	46

Table 29: Great Circle Distance Measurement

Table29 shows the longitude and latitude of different nodes represented by edges in Fig27 where the total distance is measured in meters using great circle distance method. Based on the proximity and bandwidth results the preference list are generated and feed into the algorithm which detected the vital link of the network as shown in Fig 28

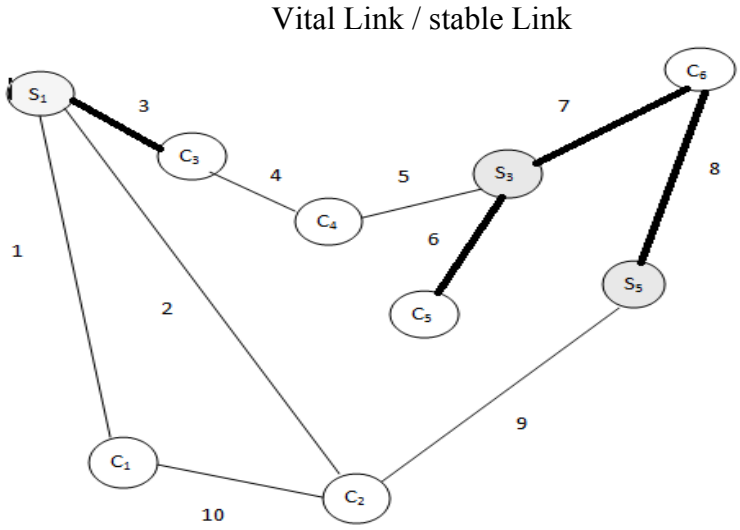


Fig 28: Graphical representation of detected vital link

4.6 Conclusion

In this paper, we formulated the vital link detection method and proved the correctness of stable matching approach in determining the vital link and shortest path between client and server within small hop count. Also we proposed an algorithm supporting the situation if one of the vital links fails during failure. Furthermore, the conditions are made clear where the problem is solvable in polynomial time. In general, a polynomial time algorithm is presented to solve the problem when the number of simultaneous failed link is restricted to one. This algorithm is useful when 70% of unintentional failures are single link failures. Also we deployed the approach within campus showing the vital link to be protected.

For future work we will focus on the design of approximation algorithm considering ties and indifference from the concept of stable matching. Ties implies when two users at a time wants to access the single resource of the server resulting into race condition and synchronization problem which can result into network performance detour. Indifference implies when two or more links fails simultaneously. Also will focus on the other conditions that prove the problem can be solvable in polynomial time.

CHAPTER 5
IMPLEMENTATION MODULES

Initialization Function:

```
public LinkStability(int n)
{       super(n);
        this.n = n;
        ClientPref = new int[n][];
        ServerPref = new int[n][];
        for(int i=0; i<n; i++)
        {       ClientPref[i] = new int[n];
                System.out.println("Enter the preference for Client" + i);
                createPref(ClientPref[i]);
                ServerPref[i] = new int[n];
                System.out.println("Enter the preference for Server" + i);
                createPref(ServerPref[i]);
        }
}

private void createPref(int[] v)
{       Scanner sc = new Scanner(System.in);
        for(int i=0;i<v.length;i++)
        { mlist:{
                v[i]= sc.nextInt();
                if(v[i] > n)
                { System.out.println("perference list should be less than equal ton" + n +
"Enter Again");
```

```
                i--;
                break mlist;
                    }}
            }
}
```

Modified_GSA Function:

```
public int[] stable()
{       // refers the woman i currently engaged to men i.e current[i]
        int[] current = new int[n];
        final int NOT_ENGAGED = -1;
        for(int i=0; i<current.length; i++)
                current[i] = NOT_ENGAGED;  //currentl no proposal
        // List of Client not connected to Server
        LinkedList<Integer> freeClient = new LinkedList<Integer>();
        for(int i=0;i<current.length; i++)
                freeClient.add(i);
        next = new int[n]; //men i proposal to women position to next[i] , initialized to 0.
        while(!freeClient.isEmpty())
        { int m = freeClient.remove();
          int w = ClientPref[m][next[m]];
           next[m]++;
        if(current[w] == NOT_ENGAGED)
        {       current[w]=m; }
        else
        {       int m1= current[w];
                if(prefers(w, m, m1))
                {       current[w] = m;
```

```
                    freeClient.add(m1);
                }
                else
                    freeClient.add(m);
        }}}
        return current;
}
```

Preference Function:

```
private boolean prefers(int w, int x, int y)
{ for (int i = 0; i < n; i++)
  {     int pref = ServerPref[w][i];
        if (pref == x)
            return true;
         if (pref == y)
            return false;
    }
       // For debugging
    System.out.println("Error in womanPref list " + w);
     return false;
  }
```

Score Function:

```
protected int calScore()
{       for(int i=0;i< n;i++)
        {       score = score + next[i];
        }
```

```
        return(score);
}
```

OptimalLink Fuction :

```
protected void printStablePair(int[] m)
{       System.out.println("optimal couples (Client + Server): ");
        for (int i = 0; i < m.length; i++)
         System.out.println(m[i] + " + " + i);
   }
```

GreatCircle_Distance_Measurement Function:

Given the latitude and longitude (in degrees) of two points compute the great circle distance (in nautical miles) between them. The following formula assumes that sin, cos, and arcos are computed in degrees, so need to convert back and forth between radians.

$$d = 60 * \text{acos}(\sin(L1)*\sin(L2) + \cos(L1)*\cos(L2)*\cos(G1 - G2))$$

```
public class GreatCircle
{    double x1 = Math.toRadians(Double.parseDouble(args[0]));
      double y1 = Math.toRadians(Double.parseDouble(args[1]));
      double x2 = Math.toRadians(Double.parseDouble(args[2]));
      double y2 = Math.toRadians(Double.parseDouble(args[3]));
      // great circle distance in radians using law of cosine
      double angle1 = Math.acos(Math.sin(x1) * Math.sin(x2
                        + Math.cos(x1) * Math.cos(x2) * Math.cos(y1 - y2));
    // convert back to degrees
      angle1 = Math.toDegrees(angle1);
      // each degree on a great circle of Earth is 60 nautical miles
      double distance1 = 60 * angle1;
```

```
        System.out.println(distance1 + " nautical miles");
    }
```