International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 4, April - 2014

An Efficient Algorithm for Three Dimensional Cyclic Stable Matching

Nikita Panchal
StudentComputer Science Department, BIT (MESRA) Noida CampusNoida [U.P.], India

Seema Sharma
Assistant ProfessorComputer Science Department, BIT (MESRA) Noida CampusNoida [U.P.], India

***Abstract*: Classic cyclic three dimensional stable matching problem (3DSM) models many real world scenarios. One such well known application is the "3-way kidney transplant". It has been proven that to find whether there exists a stable matching for a given instance of 3DSM is NP-complete. In recent years this cyclic 3DSM problem has been redefined as "Three sided matching with size and cyclic preferences (TMSC)" problem which perfectly models three sided network setups and the matching problem in such setups. An algorithm was proposed for restricted version of this problem in context of three sided networks.**

This paper proposes an efficient algorithm for reduced version of TMSC problem in the context of three sided networks, using partially distributed and graph theoretic approach. This paper also shows that if applied to the restricted model of the problem, proposed algorithm is efficient.

Keywords: Cyclic 3 dimensional stable matching, Three sided networking, Tripartite graphs, Stable marriage problem.

I. INTRODUCTION

Stable marriage problem (SMP) defined by Gale and Shapley [1] in 1962 is one of the most successful simulations of some real world problems. Motivation behind defining SMP was United States of America's "National Residents Matching Program (NRMP)" which is a student hospital matching system. Following is the brief introduction of this system which can be generalised to any 2 dimensional matching problem. Each candidate ranks hospitals according to its preferences and vice-versa. The goal is to find a "stable" one-to-one assignment of candidates to the hospitals. Where "Stable" means there is no candidate and hospital which prefers each other to their current assignments. Two dimensional SMPexists pervasively in various areas like job market where a consultant can maximise the gains by materializing maximum employer-employee deals, or in internet auctions where objects are matched withbuyers and where prices are decisive. Lloyd S. Shapley and Alvin E. Roth were awarded 2012's Nobel Prize in economic sciences for "the theory of stable allocations and the practice of market design".

Some computer networking issues have also been addressed using solutions defined for 2D SMP like mobility tolerable route selection algorithm for wireless networks, scheduling for input and output queued switch etc. however, in some economic and networking scenarios, three sets of agents are involved and this is the genesis of three dimensional stable matching (3DSM). In 2012, "Three sided matching with size and cyclic preference (TMSC)" problem was modelled by Lin Cui and Weijia Jia. And they proposed an algorithm for reduced and restricted version of the problem; specifically for three sided networks. Any system where information from sources, flows to users through servers can be regarded as three sided network. Example of such networks are military surveillance systems, assisted-living and residential monitoring networks, environmental sciences study systems, traffic monitoringsystem etc...

Some of the above mentioned three sided networks exhibit special cyclic relationship among its agents. A user would want the data from its most preferred data source only (based upon its requirements). There would be no concern about through which server data is being served. Similarly, a data source would have different priority for different servers (depending upon bandwidth and connection quality). And the servers would have preferences for users while receiving the requests for information. The goal is to generate a matching set of triples (data source, server, user) such that there exists no triple which is not member of the matching set but all three partners of that triple prefer each other to their current partners in the matching.

In this paper, reduced version of TMSC problem has been taken as basis for further investigation and following contributions are made:

1) An Algorithm and supporting data structure for reduced version of TMSC problem is designed.
2) Proposed Algorithm is also applied on restricted model of TMSC problem and results were evaluated for stability criteria against greedy approach algorithms.

The remainder of this paper is organized as follows:

Section 2 provides brief about the related work in two and three dimensional matching. In section 3, the TMSC problem is revisited and explained thoroughly for its stability

criteria. Section 4 describes the solution approach and explains the data structure and algorithm. In Section 5, results are compared and analyzed. Section 6 shows the future scope of work. And Section 7 concludes the paper.

II. RELATED WORK

In 1962, While devising an algorithm for matching applicants to college places, David Gale and Lloyd Shapley defined Stable Marriage problem in their paper "College Admissions and the Stability of Marriage"[1]. The problem is: Given N men and N women, where each person has ranked all members of the opposite sex in order of preference, marry the men and women together such that there are no two people of opposite sex who would both rather have each other than their current partners. If there are no such people, all the marriages are "stable". Gale and Shapely gave $O(n^2)$ algorithm for this problem. The idea is to iterate through all free men while there is any free man available. Every free man goes to all women in his preference list according to the order. For every woman he goes to, he checks if the woman is free, if yes, they both become engaged. If the woman is not free, then the woman chooses either says no to him or dumps her current engagement according to her preference list. So an engagement done once can be broken if the woman gets better option.

Since then, many versions of SMP has been defined and studied. Stable marriage problem can be extended to the SMI (Stable marriage with incomplete lists), SMT (Stable marriage with ties) and SMTI (Stable marriage with ties and incomplete lists) [2]. There are three stability notions for SMT problem: *weakly stable, strongly stable and super stable.*

Almost all the algorithms for SMP and its variants are central algorithms.Ismel Brito and Pedro Meseguer studied the distributed version of SMP[3] and modeled it as distributed constraint satisfaction problem (DisCSP) and gave an distributed algorithm for SMP.Esteban Arcaute and Sergei Vassilvitskii studied stable matching in the context of job market and social networks[4].

Knuth extended the SMP by introducing three dimensional stable matching problem (3DSM).In 3DSM, there are three sets of agents and a stable matching is a set of triples where there exists no blocking triple that is preffered by all members to their current partners in the triple[5]. Ng and Hirschberg proved that the problem of deciding whether a stable matching exists in NP-complete [6]. Thay introduced cyclic 3DSM as a restriction on 3DSM. In cyclic 3DSM, agent A has preferences over agent B and agent B has preferences over agent C and agent C has preferences over agent A. It has been proved that it is NP-complete to decide whether there exists a stable matching for an instance of cyclic 3DSMI (3DSM with incomplete lists)[7]. There has been researches on circular stable matching and 3 way kidney tranplant problem, which proved the np compllteness for different stability notions.

In 2013, Lin Cui and Weijia Jia modeled "three sided matching with cyclic and size preferences (TMSC)" problem[9]. TMSC differes from classic 3DSMI problem as each agent can be assigend to more than one triple. This liberty makes TMSC perfect simulation of matching scenario in three sided sensor networks. Lin Cui and Weijia Jia proved that TMSC can be generalized to 3DSMI and thus it is NP-complete to determine the existance of stable matching for an instance of TMSC problem. They defined a resticted model R-TMSC for reduced version of TMSC problem and proposed an algorithm for the same.

III. PROBLEM DESCRIPTION

Three sided networks are best example of three dimensional stable matching with cyclic preferences[9]. In this paper also, same example has been taken to explain the problem model.

3.1 Network Setup

A three sided network is consists of three types of agents : (Figure 1 depicts one such network)
Data sources : These are the devices to capture the required information. The devices could be cameras or environment sensing equipments etc. Depending upon system type. Role of these devides is to get the information and send it to servers for further distribution.

Servers : Servers will distribute the information to the users based on the requests made by users. In some cases, servers also perform additional tasks of optimization and storage etc..

Users : Users are basically user end equipments or softwares which need information captured by data sources, to perform the assigned tasks in any system.

3.2 Cyclic Relationship

In three sided networks, the agents exhibit preferences for the other agents in a cyclic manner. Following is the explanation:

User → Data source
Users are only concerned about the information which is needed, it doesn't matter through which server that information is coming. Thus Users have preference lists only for data sources.

Data source → Server
While receiving requests from servers, there can not be any priority but a DS can have choices for servers, to which it would send its data so as to minimize the time delay or loss of information.i.e. based on the connection quality between data source and server, the data sources have preference lists for servers.

Server→ User
Similar to data source and server links, the available bandwidth between server and user and the required bandwidth for user's requested data will decide it for servers that which user it can server first. Thus the srvers have prefrence lists for users.

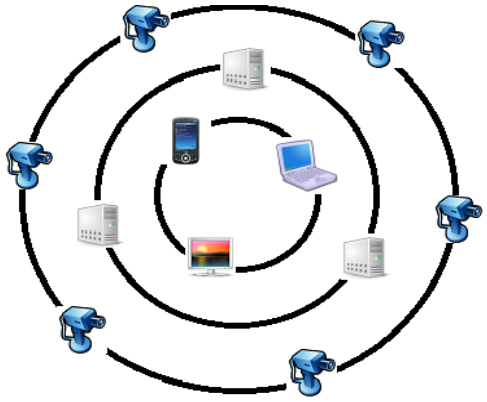

Figure 1: Three sided networks as a layered setup

3.3 Stability Criteria

Suppose D is a set of data sources where
$D = \{d_1,d_2,.....,d_{|D|}\}$ and |D| is the total number of data sources.
Similarly S is the set of servers where
$S = \{S_1,S_2,.....S_{|S|}\}$ and |S| is the total number od servers.
U is the set of users where
$U = \{u_1, u_2,.....,u_{|U|}\}$ and |U| is the total number of users.
Let T be the set of all possible triples such that
$T = \{(u_i,d_j,s_k) \mid 1 \le i \le |U|, 1 \le j \le |D|, 1 \le k \le |S|\}$
And M be the output matching set which is a subset of T i.e.
$M \subset T$.
Now any n^{th} triple in matching set M would be as follows :
$(u_{xi}, d_{yj}, s_{zk})$where for the nth triple
$n = x = y = z$.
Now if there exists any triple $t \in T$ but $t \notin M$
Such that $x \neq y \neq z$ and
$u_{xi} \mapsto d_{yi} \mapsto s_{zk}$ where $x \mapsto y$ means x prefers y over its current partner in matching, then that triple is a blocking triple.
Thus a stable matching M does not contains any blocking triple.
For example, a matching set M is as follows:
M = {(u1,s2,d1),
(u2,s1,**d3**),
(u3,**s2**,d2),
(**u4**,s3,d4)}
If u4 ⟼ d3 ⟼ s2 ⟼u4, then (u4,s2,d3) is a blocking pair.

3.4 Constraints

TMSC has been proven to be NP-hard[9]. General TMSC problem covers many scenarios where a user might request different data sources at the same time or where each data source can only accept some user requests. Also a server has a maximum allowed simultaneous streams or sessions. Thus a pair (u_i,s_j) in any triple is acceptalbe only if the bandwidth between the user and the server is more than the minimum required bandwidth for the service. Same constraint exists for the data source and server pair.
The algorithm presented in next section is for the reduced version of TMSC problem that is when each user only requests one data source and each data source can only be assigned to one server.
Thus in the assumed network:

- Each server has capacity to serve more than one user and which in total servers all the users.
- No. of data sources may or may not be equal to no. of users. I.e. one data source can be assigned to more than one users.

Which is a genuine scenario of any such networking setup.

IV. ALGORITHM

4.1 Graph Theoritic approach

This kind of network is a tripartite cyclic digraph. A tripartite graph is a graph in which vertices can be divided in three seperate non empty disjoint sets and vertices in the same set are not adjacent. Edges's weights are priorities. And there are |T| cycles in total of length 3(As three vertices u,d and s are involved in each cycle.).

Therefore, a matching corresponds to a disjoint packing of directed 3-cycles in this graph. Figure 2 shows an example of three sided network shown as tripartite cyclic digraph.

The algorithm starts with finding the minimum weight cycle and add the triple to the matching set M. In the next iteration, the next minimum weight cycle would be chosen which do not break the earlier made triples.

While adding triples to the matching set M, a trailing list is attached to every agent/vertix which contains the agents/vertices which were at higher priority in the agent's/vertix's priority list.

This trailing list is used to find if there is any blocking triple being formed in the matching set M.
Based on this approach, there are many ways to write this algorithm but to keep the time complexity at minimum a special matrix like data structure is suggested which isdistributed among any one set of agent (either users or servers).

4.2Data Structure

Graphs are generally represented as an adjacency matrix. Here the complete network graph will be represented as a two dimensional matrix. But each column which is itself a one dimensional array will be distributed among the users/servers.

Thus for the complete matrix:

Number of columns = number of users i.e. |U|

Number of rows = number of data sources i.e. |D|

Speciality of this data structure is that the third dimension of the network which are servers, would be represented in each cell. Each cell is itself a two dimensional array. Thus for each cell:

Number of columns = 4

Number of rows = number of servers i.e. |S|

The contents of the matrix are the preferences. Following example explains the reason behind fixed number of columns in each cell:

Any single row with in a cell would be like {a,b,c,d}. The Users and the data source can be identified by the column number and row number respectively in the outer matrix.

Let's take user as ui and data source as dj. The row number within the cell identifies the server. Let's take server as sk.

Now that particular row {a,b,c,d} where a, b, c and d are digits, shows the weights in the cycle ui ⟹ dj⟹ sk. I.e.

ui's preference for dj is 'a'.

dj's preference for sk is 'b'.

sk's preference for ui is 'c'.

And d = a+b+c. I.e. sum of the weights of that particular cycle.

Figure 3 is an instance of the network where |U| = 6, |D| = 6 and |S| = 2.

4.3 Algorithm

This algorithm is partially distributed and partially central. Each user maintains its own column of the matrix in form of an array list. And each user has a sorted queue (in ascending order) of the 4th column i.e. cycle sum where each element of the queue is a (index, sum) pair.

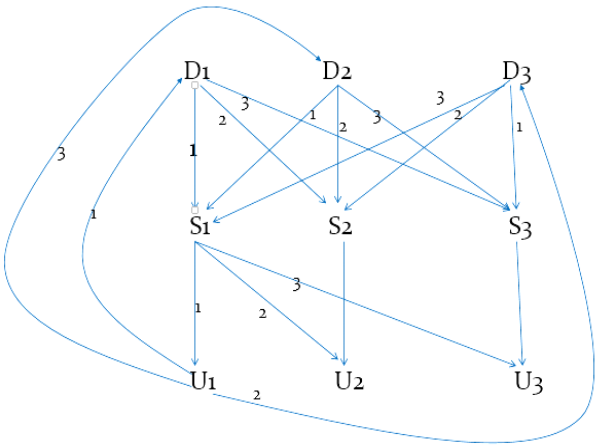

Figure 2: An instance of three sided network shown as tripartite cyclic digraph

Index in the queue identifies the corresponding data source and server pair.Server which runs the main algorithm keeps a hash map which stores the data sources and count of maximum number of users that data source can be assigned to. Each user shares the popped elements from its queue as and when requested by the main algorithm. Also, during its run the algorithm keeps on updating the user's list and queue.

Input:
1. List of n*s Quadruples (Identified by [DS, S] pair) for each user.

2. Sorted queue of cycle sum's where single element of queue is an index and sum pair [index, sum].

Output: Matching Set of (U, D, S) triples.

Algorithm:

Set Total=0;
1) Pop the elements from each user's queue and add to the tempList.

2) Find the min. sum in the tempList.

3) Look for the corresponding index in the checklist
1) If found
1) Pop the next element of corresponding queue

2) Add to the tempList

3) Continue to step 2
4) Add the min. sum triple (U, D, S) to the matching set
5) Set Total=Total+1

6) Set corresponding min. sum entry in tempList = 0.

7) If DS < DS_{max}
1) Continue;

2) Else if DS = DSmax

3) Set the corresponding DS rows in the list for each user.

4) Set index entry in the queue to null for each user.

8) Add the index to the checklist

9) If Total <= N

Repeat step2

Data source	Server	User1	User2	User3	User4	User5	User6
D1	S1	1 1 6 8	1 1 5 7	1 1 4 6	1 1 3 5	1 1 2 4	1 1 1 3
	S2	1 2 5 8	1 2 6 9	1 2 3 6	1 2 4 7	1 2 1 4	1 2 2 5
D2	S1	2 2 6 10	2 2 6 10	2 2 4 8	2 2 3 7	2 2 2 6	2 2 1 5
	S2	2 1 5 8	2 1 6 9	2 1 3 6	2 1 4 7	2 1 1 4	2 1 2 5
D3	S1	3 2 6 11	2 1 6 9	2 1 3 6	2 1 4 7	2 1 1 4	2 1 2 5
	S2	3 1 5 9	3 1 6 10	3 1 3 7	3 1 4 8	3 1 1 5	3 1 2 6
D4	S1	4 1 6 11	4 1 5 10	4 1 4 9	4 1 3 8	4 1 2 7	4 1 1 6
	S2	4 2 5 11	4 2 6 12	4 2 3 9	4 2 4 10	4 2 1 7	4 2 2 8
D5	S1	5 1 6 12	5 1 5 1'	5 1 4 10	5 1 3 9	5 1 2 8	5 1 1 7
	S2	5 2 5 12	5 2 6 13	5 2 3 10	5 2 4 11	5 2 1 8	5 2 2 9
D6	S1	6 2 6 14	6 2 5 13	6 2 4 12	6 2 3 11	6 2 2 10	6 2 1 9
	S2	6 1 5 12	6 1 6 13	6 1 3 10	6 1 4 11	6 1 1 8	6 1 2 9

Figure 3: an Instance of network where data structure represents the whole tripartite graph as a matrix of preferences/ weight on the edges

V. ANALYSIS AND EVALUATION

In this paper, two criteria were chosen for evaluation: time complexity of algorithms and Stability of the resulting matching set.

Time Complexity

As each user sorts its own list, this task is done distributed and in parallel. If we consider worst case complexity of sorting random numbers, it comes out to be O (n log n). Main algorithm would yield at least one matching triple in one iteration of the loop. So, the maximum running cycles for the main loop would be n. Thus the worst case complexity would be O (n^2logn).

If applied on R-TMSC problem where that is a master list of data sources and preference list of eachserver contains at least one tie, suggested algorithm's time complexity would be O (n log n). Following table compares the suggested algorithm's time complexity with Lin Cui and Weijia Jia's R-TMSC algorithm and two other greedy approach algorithms. Greedy-I: each user just tries any server arbitrarily; if the server is available, it will try to assign the best possible available data sources to the user, otherwise, the user will go to next server. Greedy-II: each user tries the best available data source; if the data source has not been assigned, it will check from the best server for the data source until an available one is found.

Table 1: Comparative analysis on basis of time complexity

	Partially distributed algorithm	R-TMSC algorithm	Greedy-I	Greedy - II
Average case	N log N	N^2	N^3	N^3
Worst case	N^2 log N	O ($\|S\| \Sigma_{dk \in D} \|P(D_k)\|$)	N^3	N^3

Stability

The stability of an output matching indicates whether there is still room for some triples to improve their situation by trilateral cooperation or in other words number of blocking pairs in the output matching shows the level of stability. Based on this criteria, Algorithm suggested in this thesis is compared with R-TMSC algorithm and two other greedy approach algorithms. Number of users were increased from 10 to 100 to test the percentage blocking pairs in the output matching.

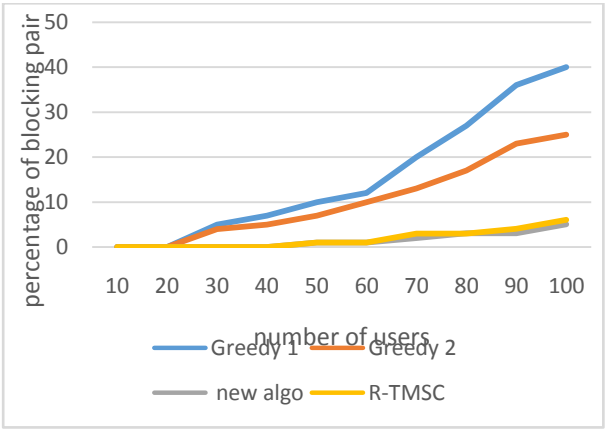

Figure 4: Performance on Stability of matching

Results show that the suggested partially distributed algorithm is more efficient and maximum percentage of blocking pairs is 5% in case of 100 users. Although there is no significant difference as compare to R-TMSC algorithm but with respect to greedy approach algorithms which were resulting in 20-30% blocking pairs, suggested algorithm certainly has an edge. Also this algorithm is egalitarian and gives the solution for unrestricted version of R-TMSC.

VI. FUTURE SCOPE

Finding approximate stability for general TMSC problem and considering local and distributed approach were two of the future scopes mentioned by Lin Cui and Weijia jia [16]. Both of them were that taken into account while devising the suggested algorithm in this paper. Some potential scenarios for future work are as follows:

(1) Case where a set of data sources is requested by users instead of just one data source, requires the remodelling of problem. In such a scenario, preference lists of users will consist of sets of data sources. The biggest challenge in finding stable matching in such case would be to find common set of data sources from user's preferences and server's remaining capacities.

(2) The situation when data forwarding is enabled among servers. This facility actually enables the algorithm to produce more stable triples. This can be evaluated for some changes in the algorithm.

(3) Possibilities for increasing efficiency of suggested algorithm can be explored.

(4) Various levels of stability can be defined for the reduced version of the TMSC problem.

(5) Algorithm suggested in this thesis work is evaluated for limited number of users in this paper. Further evaluation can be done for a large set of users, for decreasing number of data sources and for different server capacities.

VII. CONCLUSION

In this paper, a different version of three dimensional stable matching was considered where at least one set of agent can be assigned to more than one triple. This modification to the three dimensional matching problem allowed modelling it as a three sided network's matching problem. Under the context of a three sided sensor network which exhibits the same scenario of cyclic relationship among agents, an algorithm was devised for finding set of stable triples, using graph theoretic and distributed approach. The whole network of data sources, servers and users was perceived as a tripartite graph of three sets of agents respectively. And problem of creating stable triples was framed as packing the minimum sum triangles in the tripartite graphs with an additional logic to ensure that resulting matching set does not contain any blocking pair. The algorithm is egalitarian i.e. agent neutral. A distributed data structure was suggested to reduce the time complexity of the algorithm. Evaluation were done for two criteria. On the basis of stability criteria it was found that the proposed algorithm is efficient and finds a stable matching with 95% success rate. On the basis of time complexity, the suggested algorithm showed worst case time complexity to be O (n2 log n).

REFERENCES

1. D. Gale, L.S. Shapley, College admissions and the stability of marriage, The American Mathemati-cal Monthly 69 (1962)
2. D. Manlove, R. Irving, K. Iwama, S. Miyazaki, Y. Morita, Hard variants of stable marriage, Theo-retical Computer Science 276 (2002)
3. Ismel Brito and Pedro Meseguer, Distributed Stable Marriage, AAMAS-02 Workshop on Distrib-uted Constraint Reasoning (2002)
4. Esteban Arcaute and Sergei Vassilvitskii, Social Networks and Stable Matchings in the Job Market,Springer, Yahoo! Press Release, 2009. http://re-search.yahoo. com/news/2743 (2009)
5. A. Alkan, Nonexistence of stable threesome matchings, Mathematical Social Sciences 16 (2) (1988) 207–209.
6. C. Ng, D.S. Hirschberg, Three-dimensional sta-ble matching problems, SIAM Journal on Discrete Mathematics – SIAMDM 4 (2) (1991) 245–252.
7. K. Eriksson, J. Sjotrand, P. Strimling, Three-di-mensional stable matching with cyclic preferences, Mathematical Social Sciences 52(2006) 77–87.
8. C.-C.H. Huang, Circular stable matching and 3-way kidney transplant, Algorithmica 58 (1) (2010) 137–150.
9. Lin Cui , Weijia Jia, Cyclic stable matching for three-sided networking services, Computer Net-works Journal, Elsevier (2012)
10. Ravi Rastogi, , Nitin, Durg Singh Chauhan and Mahesh Chandra Govil, On Stability Problems of Omega and 3-Disjoint Paths Omega Multi-stage In-terconnection Networks , International Journal of Computer Science (2011)
11. P. Biro, E. McDermid, Three-sided stable matchings with cyclic preferences, Algorithmica 58 (2010) 5–18.

Appendix

Algorithm to find the blocking triple in the output matching set M.

Input:

Three array lists of users, data sources and servers, which are denoted as U, D and S respectively in the following algorithm. Each cell of array list has attached trail list which is the list of agents which were preferred more than the current partner in matching in descending order i.e. first priority first.

Output: Set of blocking triples

Algorithm:

For each user Ui in arraylist U

If trail list = Ø then Continue;

Else

For each data source Di in Ui's trail list

If trail list = Ø then Continue;

Else

For each server Si in Di's trail list

If trail list = Ø then Continue;

Else if Ui ∈ Si's trail list

Add set (Ui, Di, Si) to the set of blocking triples.