A Graph Based Algorithm For Solving The Compatible Pair Problem

KUSHAL CHAKRABORTY
Computer Science and Information System
Birla Institute Of Technology And Science, Hyderabad Campus
2022H1030089H
ARITRA KUMAR DUTTA
Computer Science and Information System
Birla Institute Of Technology And Science, Hyderabad Campus
2022H1030096H
MOHAMMAD AVESH HUSAIN
Computer Science and Information System
Birla Institute Of Technology And Science, Hyderabad Campus
2022H1030090H

***Abstract*—Compatible Pair Problem is an algorithmic mathematical problem aimed at choosing the right pairs within a group of items based on the preferences set for each item with respect to other items. The preferences are quantified in some mathematical parameters. It falls in the category of matching and assignment problem in which items of two disjoint sets are matched and groups are formed from them while in this problem pairing is done within a single set. The objective is to pair the items of a set in such a way that it maximizes the profit which in this case is the compatibility among different pairs of items. To tackle the mentioned problem graph-based algorithm is used. One of the applications of this problem is finding the right partner within a class of students based on the likes and dislikes information given by each student. Students are represented as the vertices of the graph, the likes contribute positive edge weight and the dislikes contribute negative edge weight, each vertices have two weights associated with them representing the count of likes and dislikes for the vertices from every other node. Using the information of likes and dislikes given by each student with respect to other students the main objective is to maximize the satisfaction level of students according to the preferences given by them.**

***Index Terms*—complete-graph, compatible pair, satisfaction constraint,**

I. Introduction

this extensive project aims on exploring the solution of the compatible pairing problem between a set of students to form groups which aims for maximizing the chances of pairing the willing students in the same group. These problems which we are exploring falls in the category of well-known matching and assignment problem which can be crucial in day-to-day life, for example finding compatibility between husband and wife, Scheduling n number of jobs to m number processors, mapping number of candidates to the vacancies, choosing tasks to allocated to machines in the production line and many more. In both the matching and assignment problem items are matched either within the same set or items of different sets are matched to form groups, the matching is done based on certain criteria and tries to optimize that criteria. In our example the items are students who needs to be matched with other favourable students within their same set which is their class. for the sake of simplicity we have assumed that the team size is restricted to two which is a common scenario in this type of settings. Also, we have assumed that each student will be allowed to choose fixed number of students to fill their like and dislike information. and we also have assumed that there are even number of students in the class so that everyone should be matched with their corresponding partner and no one is left without a partner. These assumptions are crucial assumptions to avoid adding any unnecessary complexity to the problem and focus on the core of the problem. In this paper a graph-based solution to the compatible problem is introduced which aims to pair each student with their most liked partner based on the preferences provided by them. In this paper the problem is defined as there are n number of students in a class, they are required to be grouped in the teams of two for some project work, in order to group them in teams it has to be made sure that they are compatible in working with each other. An algorithm is required to group them such that student is matched with their most preferred partner for the project work. A graph base algorithm in which students are represented as vertices(v) of the graph and their relation with each other is represented as the edges between them, Complete graph of n vertices are considered to map the relation of each student to every other student in the class. Further section of this paper is organised as follows,in section II related works has been discussed In section III representation of the the problem using graph is explained, further in section IV and example illustration is shown to better understand the problem representation steps, then in section V actual proposed algorithm has been discussed, In section VI a example is traced over the algorithm and in the subsection of section VI the complexity of the algorithm has been discussed,finally in Section VII Results has been shown

II. Literature Review

Since the matching and assignment problems are explore extensively in the field of computer science and mathematics, researches has come up with different solution of the problem, [1] models the problem of matching disjoint sets of men and women and tries to provide a mapping between them by using matrix based search algorithm in which according to the preference of each men and women best possible matches is formed. [2] proposed an general algorithm which also solve the problem of mapping and assignment between two disjoint sets in any n*n scenario Where n is count of the instances of objects, objects can be husband/wife, task/machine, schedulers/jobs etc. the problem is modelled around a satisfaction constraint which has to maximized or minimized according to the constraint. [3] in this they provided some extra criteria to be used along with Gayle shapely algorithm to increase the chances of pairing of compatible pairs and also introduces other variants of the problems which lie under the hood of matching and assignment problem of two disjoint sets. All the solution listed above takes in account only one parameter to produce matching from the disjoint set, our problem solves the problem taking in account multiple parameters and also these solutions are explored for matching and assignment problem of two disjoint sets, our problems differ in the sense that we have to make pairs within the set. .

III. Problem Representation

In this section representation of the above discussed problem in the form of complete un-directed graph is discussed. The problem is defined as complete un-directed graph G (V, E) where V are the vertices of the graph which represents individual students of the class, E represents the edges of the graph that represents relation between two students in the graph. To make the graph a complete graph, neutral edges are added to the graph after updating the relation provided, an edge weight win is associated with each edge $E_{i,j}$ which represent the quantification of the likes and dislikes of the students. There could be many schemes on the basis of which we can quantify the weights of the likes and dislike some of them could be

- assign +1 to likes, -1 to the dislikes and 0 to the neutral edges in the graph. this notation will give equal preference to the likes and dislikes
- assign +2 to likes, -1 to the dislikes and 0 to the neutral edges in the graph. this notation will give preference to the likes over dislikes
- assign +1 to likes, -2 to the dislikes and 0 to the neutral edges in the graph. this notation will give preference to the dislikes over likes
- assign +2 to likes, -1 to the dislikes and 1 to the neutral edges in the graph. this notation will give equal preference to the likes and dislikes and some preference to neutral edges

In order to further simplify the problem graph, all the directed edges between a particular set of nodes are merged and represented as one un-directed node between those two nodes. The resultant weight can be represented as the summation of every node $E_{i,j}$ and $E_{j,i}$. Further their will be weights associated with each node in the graph that will be represents as two sets L and D, where L_i will account for the number of nodes which likes node i and D_i will contain the number of nodes which dislike node i. we have also assumed that number of nodes in the graph will be even so that we can find pair for every node in the graph. So, the entire problem can be represented as a quintuple like G= (V, E, W, L, D) where V will represent students E will represent n(n-1)/2 edges in the graph which accounts for relation between students, W will represent the quantification of like and dislike for every edge E, L will represent the how many other students has liked pairing with him, D will represent how many other students has disliked pairing with him. This final graph which has represented as quintuple will be given as the input to the compatible pair algorithm.

IV. Illustration of problem Representation

In this section of the problem a problem has been modelled based on the modelling scheme considered in section III. In this illustration we have used following assumptions for assigning the weights to the edge weights

- For each like weight of +2, dislikes contribute -2, and a neutral edge contribute weight 1.

The likes and dislike information of each student has been provided as table 1.

TABLE I
Like Dislike Table

Student	Liked List	Dislike list
A	D,F,C	E
B	C, E, F	A
C	D, E, A	F
D	B, A, E	F
E	F, B, D	C
F	C, D, B	A

based on this information provided in the table 1, a new table will be deduced which will contain information of the node which does not come in neither like or dislike columns of table 1 which will indicate the neutral edges in the graph, more over it will also contain the L and D information for every node in the graph.

From the table 2 we can assign weights for nodes from the likes and disikes columns directly. To calculate the effective weight of edge between two nodes every like and dislike between them is considered. If there is no information about them in the table 1 then a neutral edge is added. Here an example for all the edges of student A is demonstrated, in similar concept weights for further edges can be calculated.

TABLE II
NODES WEIGHTS(L,D)

Student	Neutral	L(dislike count)	D(dislike count)
A	B	2	2
B	D	3	0
C	B	3	1
D	C	4	0
E	A	3	1
F	E	3	2

TABLE III
NODES WEIGHTS(L,D)

Edge from A	Edge to A	Weights	Resultant weight
A $\rightarrow B(neutral)$	B$\rightarrow A(dislikes)$	-2	-2
A $\rightarrow C(likes)$	C$\rightarrow A(likes)$	+2+2	4
A$\rightarrow D(likes)$	D$\rightarrow A(likes)$	+2+2	4
A$\rightarrow E(dislikes)$	E$\rightarrow A(neutral)$	-2	-2
A$\rightarrow F(likes)$	F$\rightarrow A(dislikes)$	+2-2	0

After calculating the edge weight for all n(n-1)/2 edges in the graph using the same notion as discussed above. An undirected complete graph can be drawn as shown in figure 1.

V. PROPOSED ALGORITHM

The algorithm takes as input the weighted undirected complete graph as input and produces a list of compatible sets of student S. algorithm starts as making a copy of G (V, E, W, L, D) as G' and feeding it to the algorithm. Algorithm performs n/2 iteration and carefully selects the sets and put it into the solution S. this algorithm runs until all the nodes in the graph is matched.

- A set V_1 is created by considering all the nodes I which have most dislike weight D_i.
- If the set V_1 contains more than one node then a new set V_2 is created which contains the V_1 node which contain least like L_i. If again all the node from V_1 is eligible then 1 node is selected randomly.
- All the edges incident to a node in V_2 and having the maximum weights are included in a set E_1.
- For all the Edges in E_1 the Edge With the node j which have maximum dislikes and minimum likes is selected and added to the solution set S, if both nodes have same like and dislike tie broken randomly.
- Node i and j are removed
- The sets of (L, D) for all the nodes are also updated

Algorithm 1 Min-Min Algorithm

```
Require: G'(E', V', W', L', D') = G(E, V, W, L, D)
Ensure: $S \subset E$ of n/2 disjoint edges with maximum total edge weight
 1: S = ∅
 2: for k = 1,2...,n/2 do
 3:     Find $V_1$, set of nodes $i \in V'$ with max $d_i$
 4:     Find $V_2$, set of nodes $i \in V_1$ with min $l_i$
 5:     Find $E_1 = \{(i,j) \in E', i \in V_2 or j \in V_2$ with max $w_{ij}\}$
 6:     Select $(i,j) \in E_1 | i \in V_2$ and $j$ satisfies max $d_j$ followed by min $l_j$
 7:     S = S ∪ selected $(i,j)$
 8:     V' = V'/$\{i,j\}$
 9:     E' = E' /$\{e \in E'$ incident to $i$ or $j$ $\}$
10:     Update L' and D' accordingly
11: end for
```

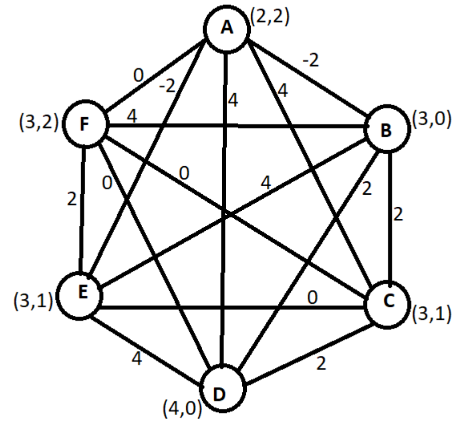


Fig. 1. Problem Graph

VI. ILLUSTRATION OF PROPOSED ALGORITHM

In this section we will take up the previous graph that we created in section 4 and trace the algorithm on it.
In the first iteration $V1 = \{F, A\}$ as both the nodes have maximum dislikes (2). Then algorithm creates V2 as $V2 = \{A\}$ as node A has min like (2) now edges are explored from A which has max weight$E1$= $\{(A, D), (A, C)\}$ both the edges have weight (4), from set E1 edge (A, C) is selected as it has less likes (3) and more dislikes (1) than the edge (A, D) which has like (4) and dislikes (0). S is updated with pair S= $\{(A, C)\}$. After the first iteration graph is reduced to as shown in fig 2.

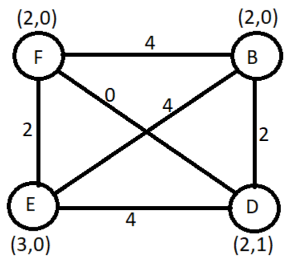


Fig. 2. Resultant Graph after First Iteration

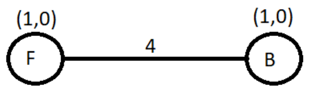


Fig. 3. Resultant Graph after Second Iteration

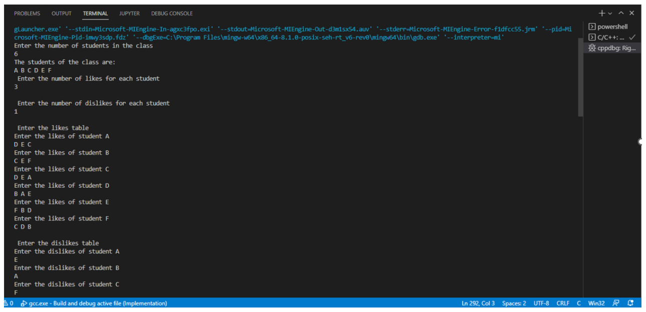
In the second iteration algorithm starts with $V1 = V2 = \{D\}$ as it has the max dislike (1), then it updates $E1 = \{(D, E)\}$ As its weight of edge (D, E) is 4, as we get only one pair in the E1 it will be the best match so the node (D, E) will be added to the Solution set $S = \{(A, C), (D, E)\}$ and (D, E) is removed from the graph. Graph after the second iteration becomes as shown in the figure 3.

In the third iteration, V1= F, B, V2=F, E1= (F, B), S= (A, C), (D, E), (F, B). As observed that it took n/2 iteration to pair all the nodes in the graph with solution produced as S= (A, C), (D, E), (F, B) which has weight 4+4+4= 12.

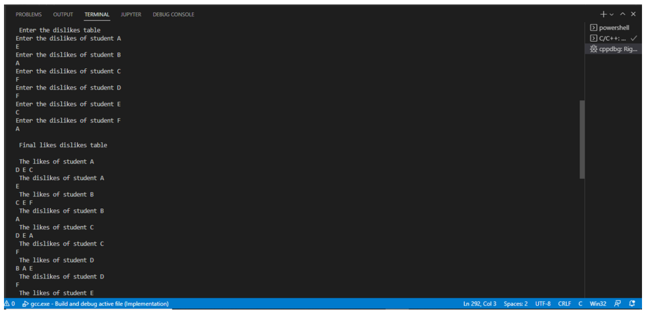
A. *Complexity of the algorithm*

Steps 3 and 4 takes n iterations for n vertices. Hence it takes O(n) time. Steps 6 and 7 in the worst case scenario takes n^2 iterations since the original graph is a complete graph and has $(n * (n - 1))/2$ edges. Hence it takes $O(n^2)$time. All the steps from 3 to 7 executes for $n/2$ times since there are$n/2$ pairs. Hence the overall time complexity is :$(n/2) * (O(n) + O(n^2)) = O(n^3)$.

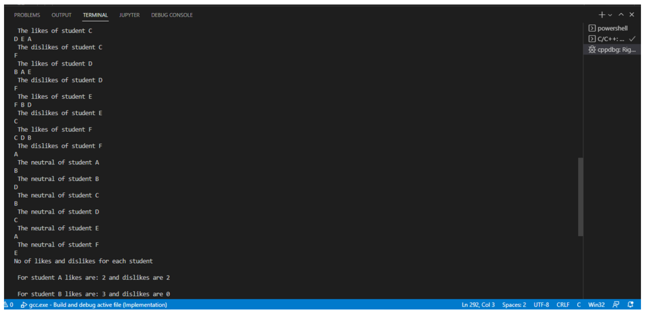
VII. Results



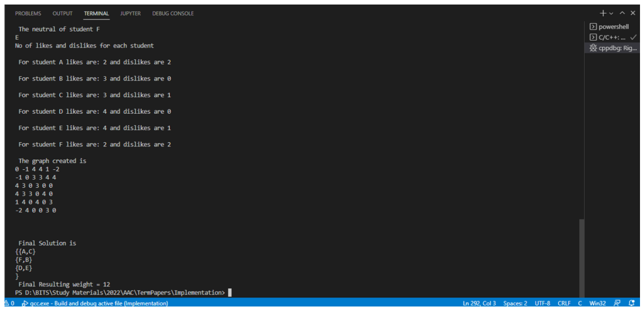
```
gLauncher.exe' '--stdin=Microsoft-MIEngine-In-agxc3fpo.exi' '--stdout=Microsoft-MIEngine-Out-d3m1sx54.auv' '--stderr=Microsoft-MIEngine-Error-f1dfcc55.jrm' '--pid=Mi
crosoft-MIEngine-Pid-imwy3sdp.fdz' '--dbgExe=C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin\gdb.exe' '--interpreter=mi'
Enter the number of students in the class
6
The students of the class are:
A B C D E F
 Enter the number of likes for each student
3

 Enter the number of dislikes for each student
1

 Enter the likes table
Enter the likes of student A
D E C
Enter the likes of student B
C E F
Enter the likes of student C
D E A
Enter the likes of student D
B A E
Enter the likes of student E
F B D
Enter the likes of student F
C D B

 Enter the dislikes table
Enter the dislikes of student A
E
Enter the dislikes of student B
A
Enter the dislikes of student C
F
```



```
 Enter the dislikes table
Enter the dislikes of student A
E
Enter the dislikes of student B
A
Enter the dislikes of student C
F
Enter the dislikes of student D
F
Enter the dislikes of student E
C
Enter the dislikes of student F
A

 Final likes dislikes table

 The likes of student A
D E C
 The dislikes of student A
E
 The likes of student B
C E F
 The dislikes of student B
A
 The likes of student C
D E A
 The dislikes of student C
F
 The likes of student D
B A E
 The dislikes of student D
F
 The likes of student E
```



```
 The likes of student C
D E A
 The dislikes of student C
F
 The likes of student D
B A E
 The dislikes of student D
F
 The likes of student E
F B D
 The dislikes of student E
C
 The likes of student F
C D B
 The dislikes of student F
A
 The neutral of student A
B
 The neutral of student B
D
 The neutral of student C
B
 The neutral of student D
C
 The neutral of student E
A
 The neutral of student F
E
No of likes and dislikes for each student

 For student A likes are: 2 and dislikes are 2

 For student B likes are: 3 and dislikes are 0
```



```
 The neutral of student F
E
No of likes and dislikes for each student

 For student A likes are: 2 and dislikes are 2

 For student B likes are: 3 and dislikes are 0

 For student C likes are: 3 and dislikes are 1

 For student D likes are: 4 and dislikes are 0

 For student E likes are: 4 and dislikes are 1

 For student F likes are: 2 and dislikes are 2

 The graph created is
0 -1 4 4 1 -2
-1 0 3 3 4 4
4 3 0 3 0 0
4 3 3 0 4 0
1 4 0 4 0 3
-2 4 0 0 3 0



 Final Solution is
{{A,C}
{F,B}
{D,E}
}
 Final Resulting weight = 12
PS D:\BITS\Study Materials\2022\AAC\TermPapers\Implementation>
```

References

[1] Teo, Chung Sethuraman, Jay Tan, Wee-Peng. (2000). Gale-Shapley Stable Marriage Problem Revisited: Strategic Issues and Applications. Management Science. 47. 10.1007/3-540-48777-832.

[2] H. W. Kuhn,"The Hungarian method for the assignment problem," Naval research logistics quarterly, vol. 2(1-2), pp. 83–97, 1955.

[3] K. Iwama and S. Miyazaki, "A Survey of the Stable Marriage Problem and Its Variants," International Conference on Informatics Education and Research for Knowledge-Circulating Society (icks 2008), 2008, pp. 131-136, doi: 10.1109/ICKS.2008.7.

[4] Y. Harrath, A. F. Salman and A. Alqaddoumi, "A Novel Graph-Based Algorithm for Solving the Right Partner Problem," 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2018, pp. 1-5, doi: 10.1109/3ICT.2018.8855794.