KIDNEY PAIRED DONATION OPTIMAL MATCHINGS IN BIPARTITE GRAPHS

Harish Udhay Kumar
hu33@scarletmail.rutgers.edu

Swathi Gopal
sd1322@scarletmail.rutgers.edu

Teja Paladgu
tp577@scarletmail.rutgers.edu

Yashraj Dhananjay Gangal
yg457@scarletmail.rutgers.edu

***Abstract*--** **Data structures and algorithms are essential tool sets for people from many walks of life. For patients with final-stage renal disease, kidney transplantation is the most effective therapeutic option. In this project, we deep-dive into Hungarian algorithm and showcase a practical application for kidney paired donation using the algorithm. Donor-recipient pairs (represented in a bipartite graph) are found such that the donors in one pair are compatible with the recipients in the other. To save as many lives as possible, a maximum matching on a graph is performed, with the vertices representing donor-recipient pairs and the non-negative weights of the edges representing the compatibility of the Donors with the patients.**

Keywords- Hungarian algorithm**, maximum match, Kidney Paired Donation**

1. INTRODUCTION

A. Kidney Paired Donation

The cases of kidney failure and the need for kidney transplantation is increasing nowadays. However, for a successful kidney transplantation, the donor and recipient pair must be compatible. This compatibility depends on various factors such as blood type, immunological characteristics, etc. Kidney Paired Donation also called KPD is a program that gives recipients other options for compatible transplantation. The vision statement of KPD is " Every kidney transplant candidate with an incompatible but willing and approved living donor receives a living donor kidney transplant".

Additionally, even after finding a compatible donor, there are more chances that the donor may back out at the last moment. In order to avoid those kinds of issues, maximum matching is done on donor-recipient pairs.

B. Optimal Matching in Bipartite Graph

In a Bipartite Graph, a matching is a group of edges chosen so that no two edges share an endpoint. A maximum match is a match that is the largest possible (maximum number of edges). If any edge is added to a maximum matching, it is no longer a matching.

Hungarian algorithm for Perfect Matching of a Bipartite Graph utilizes the fact that the optimality of a matching is a metric of its additive difference with other matchings. Thus, for a complete bipartite graph having equal nodes in both partitions, the same optimal perfect matching can be produced if all the edges incident on a node are reduced by a common subtrahend.

2. IMPLEMENTATION

I. Hungarian Algorithm

The Hungarian algorithm is a combinatorial optimization algorithm that solves the assignment problem in polynomial time, and which anticipated later primal-dual methods for optimization problems.

A. Implementation steps:

A bipartite graph represented as a matrix is given as input for the Hungarian algorithm. Where all columns belong to one subset and rows belong to another subset of the bipartite graph.

Every cell in the below-shown matrix represents the compatibility between the donor and patient. In a nonnegative $n \times n$ matrixe the element in the i-th row and j-th column represents the match of the j-th patient to the i-th donor. The goal is to find an assignment of the patient to the donor, such that each patient is assigned to one donor and each donor is

assigned to one patient, such that the total compatibility of the assignment is maximum.

Considering a matrix of donor-patient compatibility (percentage) between patient and donor.

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	82	83	69	92
Donor_2	77	37	49	92
Donor_3	11	69	5	86
Donor_4	8	9	98	23

Step 1: Subtract row minima

Subtracting the row minimum from each row. The smallest element in the first row is, for example, 69. The resulting matrix will be:

	Patient_1	Patient_2	Patient_3	Patient_4	
Donor_1	13	14	0	23	(-69)
Donor_2	40	0	12	55	(-37)
Donor_3	6	64	0	81	(-5)
Donor_4	0	1	90	15	(-8)

Step 2: Subtract column minima

Similarly, subtracting the column minimum from each column, giving the following matrix:

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	13	14	0	8
Donor_2	40	0	12	40
Donor_3	6	64	0	66
Donor_4	0	1	9	0
				(-15)

Step 3: Cover all zeros with a minimum number of lines

Determining the minimum number of lines (horizontal or vertical) that are required to cover all zeros in the matrix. All zeros can be covered using 3 lines:

	Patient_1	Patient_2	Patient_3	Patient_4	
Donor_1	13	14	0	8	
Donor_2	40	0	12	40	x
Donor_3	6	64	0	66	
Donor_4	0	1	9	0	x
			x		

Because the number of lines required (3) is lower than the size of the matrix (n=4), continuing with step 4.

Step 4: Create additional zeros

First, the smallest uncovered number is found. In this case, it is 6. Subtracting this number from all uncovered elements and adding it to all elements that are covered twice.

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	7	8	0	2
Donor_2	40	0	18	40
Donor_3	0	58	0	60
Donor_4	0	1	96	0

Now back to Step 3.

Step 5: Cover all zeros with a minimum number of lines

Again, determining the minimum number of lines required to cover all zeros in the matrix.

	Patient_1	Patient_2	Patient_3	Patient_4	
Donor_1	13	14	0	8	x
Donor_2	40	0	12	40	x

Donor_3	6	64	0	66	x
Donor_4	0	1	9	0	x

The number of lines required is 4. As the number of lines required (4) equals the size of the matrix (n=4), an optimal assignment exists among the zeros in the matrix. Therefore, the algorithm stops.

The Optimal Assignment

The following zeros cover an optimal assignment:

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	7	8	0	2
Donor_2	40	0	18	40
Donor_3	0	58	0	60
Donor_4	0	1	96	0

The below matrix corresponds to the following optimal assignment in the original cost ***matrix with minimum cost match*:**

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	82	83	69	92
Donor_2	77	37	49	92
Donor_3	11	69	5	86
Donor_4	8	9	98	23

In general, the Hungarian algorithm is used to find the minimum cost of the assignments. However, in this case, maximization of the assignment is found. The maximum is obtained by taking the complement of the given matrix and then minimizing the cost matrix. When all the elements of a matrix are subtracted with the maximum element of the matrix, the resulting matrix is called the complement.

The maximum match with the optimal assignment with ***maximum compatibility match* is:**

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	82	83	69	92
Donor_2	77	37	49	92
Donor_3	11	69	5	86
Donor_4	8	9	98	23

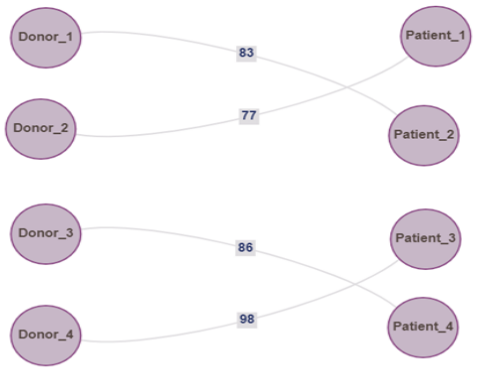
B. Output:

At each step the algorithm adds one edge to matching and this happens O(n) times.
It takes O(n) time to find the right vertex for the augmenting and O(n) time to flip the matching. Improving the labeling takes O(n) time to find a match and update the label accordingly. We might have to improve the labeling up to O(n) times if there is no augmenting path. This gives a total of $O(n^2)$ time.

In all there are O(n) iterations each taking O(n) work, leading to a total running time of the Hungarian algorithm for perfect bipartite matching to be $O(n^3)$.

The space required by the algorithm is in the order of $O(n^2)$.

Visualization of Hungarian algorithm*:*



This algorithm gives a maximum matching of weight 344 for the above mentioned example.

II. Greedy Approach:

A. Implementation:

The greedy approach involves choosing the immediately apparent optimal choice as opposed to aggregating the entire input. This leads to inaccuracy in optimization but executes fastest.

For the kidney paired donation problem, the greedy approach involves picking donors serially. For each donor, the most compatible patient (patient with highest match matrix value) is found and assigned to. The patient is then removed from the patient pool, and the next donor is picked and the process is repeated until the donor pool is exhausted. In the greedy approach, the Donor which gets matched to a patient earlier, has an advantage of receiving highest compatibility over the Donors who are matched later.

Considering the previously mentioned example input data:

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	82	83	69	92
Donor_2	77	37	49	92
Donor_3	11	69	5	86
Donor_4	8	9	98	23

We shall pick the donors in order, and for the most compatible patient chosen for each donor, we shall set the compatibility of the patient with all other donors to -∞ to virtually remove the patient from the patient pool:

Donor 1:

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	82	83	69	92
Donor_2	77	37	49	-∞
Donor_3	11	69	5	-∞
Donor_4	8	9	98	- ∞

Donor 2:

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	-∞	83	69	92
Donor_2	77	37	49	-∞
Donor_3	-∞	69	5	-∞
Donor_4	-∞	9	98	-∞

Donor 3:

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	-∞	83	69	92
Donor_2	77	37	49	-∞
Donor_3	-∞	69	5	-∞
Donor_4	-∞	9	98	-∞

Donor 4:

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	-∞	-∞	-∞	92
Donor_2	77	-∞	-∞	-∞
Donor_3	-∞	69	-∞	-∞
Donor_4	-∞	-∞	98	-∞

The weight of the maximum matching as produced by the Greedy approach is 326, which is suboptimal for the given problem. This loss of optimality stems from the subtle linear prioritization of the Donors (Donor 1 over Donor 2, Donor 2 over Donor 3, and so on). Overall, this loss of optimality is unchanged whether we contextualize the Donors or the Patients.

B. Output:

For a bipartite graph of 2n vertices. In other words, for nxn matrix (n donors and n patients), the greedy approach produces a solution in $O(n^2)$ time:

For each donor out of n donors:
Finding a patient with maximum compatibility - O(n) time
Storing the chosen Donor-Patient pair - O(1) time

Removing the patient from the patient pool - O(n) time

The space required by the Greedy Approach is in order of $O(n^2)$ since it must create a copy of the graph to work with.

The accuracy of the greedy approach depends on the order in which the Donors are contextualized. For any given instance of the problem, there exists at least one permutation for which the greedy approach will produce the optimal value.

C. *Comparison of greedy approach with Hungarian algorithm*

The greedy approach provides a faster but suboptimal solution as compared to the Hungarian algorithm. This is because the greedy approach subtly, linearly prioritizes the compatibility of donors over each other. Since impartiality and global maximization are of high significance for the kidney paired donation problem, the greedy approach shall be the less recommended method of optimization, compared to the Hungarian algorithm, which incorporates the said features.

The greedy approach for the assignment problem would be recommended in scenarios where the nodes may have a linear priority associated with them. If the nodes may have a first-come-first-serve based priority, the greedy approach can also work with streamed input data.

III. *Brute Force Approach*

A. *Implementation:*

Brute force is the simplest and most fundamental kind of algorithm. It uses simple problem-solving strategies that rely on raw processing power and exhaustive testing of all the possibilities. This method gets complex for large assignments.

For a n*n matrix, there will be n options for the first assignment and n-1 options for the second assignment.The solution is merely the permutation of other jobs.

Considering the previously mentioned example.

	Patient_1	Patient_2	Patient_3	Patient_4
Donor_1	82	83	69	92
Donor_2	77	37	49	92
Donor_3	11	69	5	86
Donor_4	8	9	98	23

Performing Brute force approach by calculating all the possible permutations we get a maximum matching of 344.

B. *Output:*

For a bipartite graph of 2n vertices. In other words, for nxn matrix (n donors and n patients), brute force approach produces solution in
O(n!)

C. *Comparison of Brute force method with Hungarian algorithm*

Though both Brute force and Hungarian algorithm results in the same maximum matching value, their implementation and hence the time taken are very disparate. Brute force algorithm gives the maximum kidney donor matching but it takes higher iterations as the solution is found recursively by trial and error. This is a tedious approach.

Additionally, as the number of tasks increases the number of calculations increases and the time complexity of n! becomes very inefficient.

3. *CONCLUSION*

The main objective of the project is to deep-dive into the Hungarian algorithm and exploit the algorithm for kidney paired donation. This objective was achieved by considering a matrix where every cell represents the compatibility between the donor and patient. By applying Hungarian algorithm on this matrix maximum patient-donor compatibility is obtained which resolves the kidney paired donation problem. The graph for this is also visually represented (in the code attached).

Additionally, the assignment problem is solved using greedy approach and brute force

approach. These approaches are analysed and compared with the Hungarian algorithm.

	Hungarian algorithm	Greedy Approach	Brute Force approach
Time Complexity	$O(n^3)$	$O(n^2)$	$O(n!)$
Space Complexity	$O(n^2)$	$O(n^2)$	$O(n^2)$

It is found that the Hungarian algorithm is the optimal approach to solve the assignment problem.

The time and space complexity of the Hungarian algorithm is much better than the other two as shown in the table.

Future scope of this project can be to solve this problem when the number of donors and patients are not the same i.e, finding a maximum matching considering a n*m matrix with n donors and m patients.

4. REFERENCES

[1] Tim Roughgarden "Kidney Exchange and Stable Matching", Case study algorithm Game Theory, October 2013.

[2] Alvin E. Roth, Tayfun Sönmez, M. Utku Ünver "Pairwise Kidney Exchange", NBER working paper series.

[3]https://theory.stanford.edu/~tim/f13/l/l10.pdf

[4]https://pats.cs.cf.ac.uk/@archive_file?p=1608&n=final&f=1-report.pdf&SIG=816a4e9cafad024afb0ac91a783ddfdb441fc5435f1aeea18f1d3bc86cb82bc1