Department of Information Technology

IT-203 PROJECT REPORT

Doctor-Patient Combined Matching Problem and its Solving Algorithms

Abhinav Bharali - 191IT201
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
abhinavbharali.191it201@nitk.edu.in

Naveen Shenoy - 191IT134
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
naveenshenoy.191it134@nitk.edu.in

Alimurtaza Mustafa Merchant - 191IT205
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
alimurtaza.191it205@nitk.edu.in

Pratham Nayak-191IT241
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
pratham.191it241@nitk.edu.in

I. Introduction:

The number of doctors in a public hospital in China is limited and hence seeing a doctor is difficult in large hospitals. The phenomenon of patients not being given adequate time for treatment in these hospitals led to many contradictions between patients and doctors. Large numbers of scholars have made many efforts in recent years to alleviate the inconsistencies between doctors and patients and the ‘‘difficulty seeing a doctor’’ problem to improve patients. In terms of medical time, researchers mainly enhance patient satisfaction and clinical efficiency by reducing outpatient waiting times.

The paper implemented in the project proposes a multi-patient treatment mode (MTM). Patients with similar disease symptoms are treated simultaneously to improve patient satisfaction and clinical efficiency, addressing the contradiction between doctors and patients and the ‘‘difficulty seeing a doctor’’. Specifically, first, the patients who come to the clinic are grouped according to the similarity of the symptoms; then, the doctor selects several appropriate groups to complete treatment on a day. The key issue to solve is how to find a solution that maximizes the overall patient satisfaction and where all patients are treated and treated only once. The above problem can be called the doctor-patient combined matching problem (DPCMP), which is essentially a more complex combined auction winner determination problem (WDP). The WDP is an NP-hard problem, and considerable effort has been devoted to the development of various algorithms for the WDP.

Given the DPCMP, the paper proposed an improved ant colony optimization algorithm to improve patient satisfaction and clinic efficiency. The MTM in this paper mainly refers to the sharing of medical resources. Patients with similar disease symptoms can be grouped together according to the patient’s information. In the MTM, doctors share relevant medical knowledge to patients as a group, where patients in the same group have similar disease symptoms.

Therefore, the following assumptions are made for the applicability of the MTM:

1. The disease is relatively stable (not an emergency)
2. The disease is not cross-infectious
3. The experience and knowledge sharing between doctors and patients greatly affect the prevention and treatment of the disease.

We assumed that all the patient groups are known in this paper. Patients have multiple choices depending on the patient disease symptoms, registration time, other information (choices of which treatment mode, which department, and which doctor, etc.), so many different groups are generated. Which groups should be chosen from all the groups to ensure that all the patients are served, no patients are repeatedly served, and the overall patient satisfaction is maximized the main problem that is solved in this paper, called the DPCMP.

II.Work Done

The following is the mathematical model for the Doctor-Patient Combined Matching Problem (DPCMP):-

$$
\begin{aligned}
&max \sum_{1 \le j \le N} b_j \cdot x_j \\
&s.t. \sum_{1 \le j \le N} y_{ij} \cdot x_j \le 1 \quad \forall i \in I \\
&\quad \sum_{1 \le j \le N} x_j \le 1 \quad 1 \le j \le N \\
&\quad \sum_{1 \le j \le N} S_j \cdot x_j = P \quad 1 \le j \le N \\
&\quad x_j, y_{ij} = \{0, 1\} \quad \forall j \in J,\ 1 \le s \le N
\end{aligned}
$$

The goal is to maximize patients' satisfaction. Here $P_i \in P = \{P_1, P_2, \ldots, P_n\}$ where $i \in I$ be the i-th patient, $S_j \in S = \{S_1, S_2, \ldots, S_N\}$ where $j \in J$ denotes the j-th of the all N groups, $b_j \in B = P\ \{b_1, b_2, \ldots, b_N\}$ where $j \in J$ denotes the satisfaction of Sj, bj denotes the satisfaction of group S_j.

The decision variables x_j and y_{ij} in DPCMP are:

$$
x_j = \begin{cases} 1 & select\ group\ S_j \\ 0 & others \end{cases} \quad 1 \le j \le N
$$

$$
y_{ij} = \begin{cases} 1 & P_i \in S_j \\ 0 & others \end{cases} \quad 1 \le j \le N,\ i \in I
$$

Where $x_j = 1$ shows, group Sj was selected to be treated by the doctor, and $y_{ij} = 1$ denotes patient P_i is in group S_j. The constraints mentioned above ensure that all the patients will be treated, and each patient will be treated at most once.

If we don't take any shortcuts, the number of operations in the worst case is (n-1)!. Thus DPCMP is an NP-hard problem. As traditional mathematical programming cannot solve this problem in a reasonable amount of time, the paper implemented an ant colony optimization algorithm.

First, we created an undirected graph. Then implemented the route construction using the ant colony optimization (ACO) algorithm. Finally, we implemented an improvised ACO algorithm using the random-restructure local search operator to improve the quality of the solutions.

Construction of undirected graph:

We constructed an adjacency matrix where edges between groups exist if they do not have any common patients and initialized that particular edge's pheromone. In other words,

$$A_{i,j} = \begin{cases} 1 & \textit{there is an side between } S_i \textit{ and } S_j \\ 0 & \textit{others} \end{cases}$$

Route construction:

The four processes during the route construction are:

1. Choose the next group based on a probability function of two attraction measures.
2. Keep a history of the groups visited during the current route.
3. Update the set of allowed groups selected in the next step.
4. Update the trail intensities of the edges visited. Then, local search procedures are applied to enhance the solution quality.

Initially, initialize all the pheromones for each edge with 1 as we can start at any group with probability 1. After selecting the first group randomly, each next group is selected from the list of all allowed groups that have not been allocated in the current loop. The next group, j, is selected according to the following condition:

$$j = \begin{cases} arg\max_{l \in J_k(i)}\{[\tau(i,l)]^{\alpha}[\eta(i,l)]^{\beta}\}, & if\ q \leq q_0 \\ J, & if\ q \geq q_0 \end{cases}$$

Here $J_k(i)$ is the set of all available groups after visiting group i.
Here q is a randomly drawn number, and q_o is a parameter(usually q_o>0.9 being a good choice).

τ_l is the value of the pheromone trail on group l; $\eta_l \in \eta = \{\eta_1, \eta_2, \ldots, \eta_N\}$ is the inspired value of group l,

$$\eta_p = \frac{b_p}{\sum_{q=1}^{N} b_q}.$$

J is a group selected according to the following probability function:

$$P(J|i) = \frac{\tau(i,J)^{\alpha}\eta(i,J)^{\beta}}{\sum_{l \in J_k(i)} \tau(i,l)^{\alpha}\eta(i,l)^{\beta}}$$

After making a move from group i to j, each ant applies a local pheromone update to update the values of the pheromone trails on the traversed edges(i,j).

$$\tau(i,j) \leftarrow (1-\rho)\cdot\tau(i,j) + \rho \cdot \Delta\tau(i,j)$$

where ρ is the parameter regulating evaporation of pheromone over time, and the pheromone increment is

$$\Delta\tau(i,j) = \gamma \cdot max_{z \in J_k(j)}\tau(i,j)$$

The global pheromone update rule is significant for the algorithm searching for a better solution, which will be done after every ant has completed the construction of their solutions.

$$\tau(u,v) \leftarrow (1-\rho)\tau(u,v) + \Delta\tau(u,v)$$

The global pheromone update rule for all edges is:

$$\tau(u,v) = \begin{cases} (1-\rho)\tau(u,v) + \Delta\tau(u,v) & (u,v) \in \tilde{S} \\ (1-\rho)\tau(u,v) & others \end{cases}$$

Where the pheromone increment $\Delta\tau$ (u, v) = 1 /$\sum b_j$.ρ is defined as trail persistence $0 \leq \rho \leq 1$, the term (1 − ρ) is interpreted as trail evaporation. The significance of the global pheromone update is that it could ensure the best solution has higher probabilities of being selected in the next iterations.

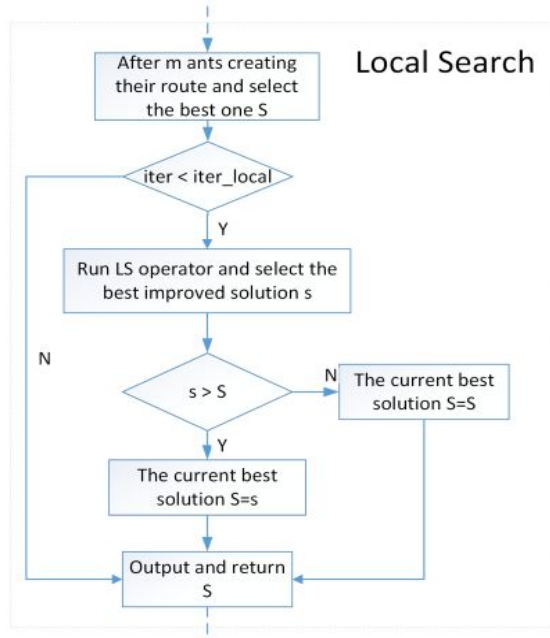
<u>Local Search:</u>

The ACO algorithm is responsible for finding a candidate solution, while the aim of the LS operations is to improve the solution by performing small changes that lead to a neighboring solution of better quality. After m ants create routes, the LS procedure is started. For the implementation of the local search operation in the paper, we have used the Random-restructure operator to improve the solution's quality.

Algorithm 1 Improved Ant Colony Algorithm

Input: S,P,B,Tabu $= \varnothing$, allowed $=$ P.
Output: $S_u, S_v, \cdots, S_\omega$

```
Start ants' the first 'group'
Update 'allowed'
if all patients are assigned then
    Local Search
else if 'allowed' is empty then
    return the step 1
else
    choose the next 'group' and update local pheromone
    return the step 3
end if
Update global pheromone
if iter < iter_Max then
    Terminate(report best solution)
else
    iter = iter + 1 and return the step 1
end if
```



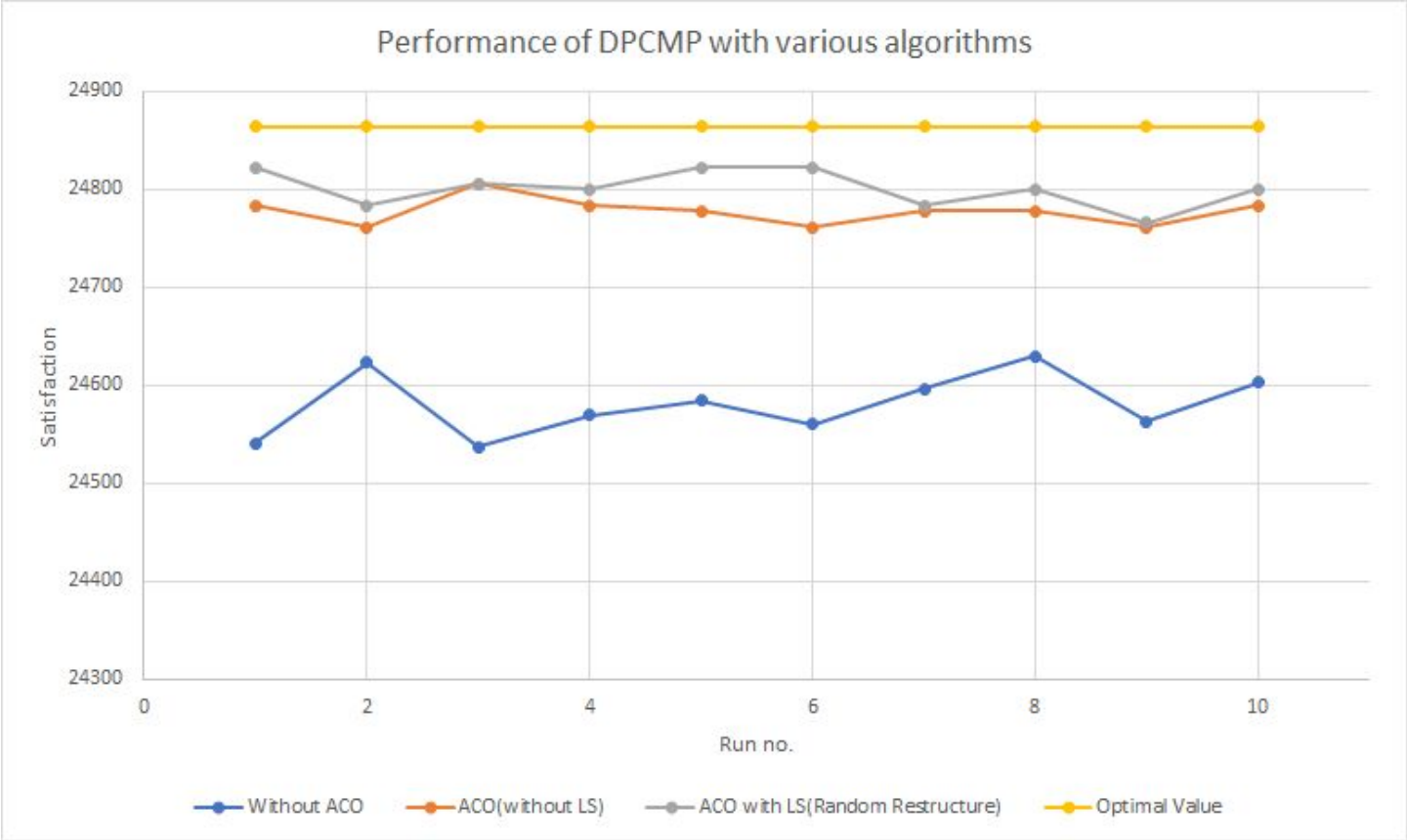
Random-restructure: In this method, an element is randomly selected in the solution sequence. The solution sequence is divided into two parts at the element. The first part of the sequence does not change, and the second part of the sequence is regenerated.

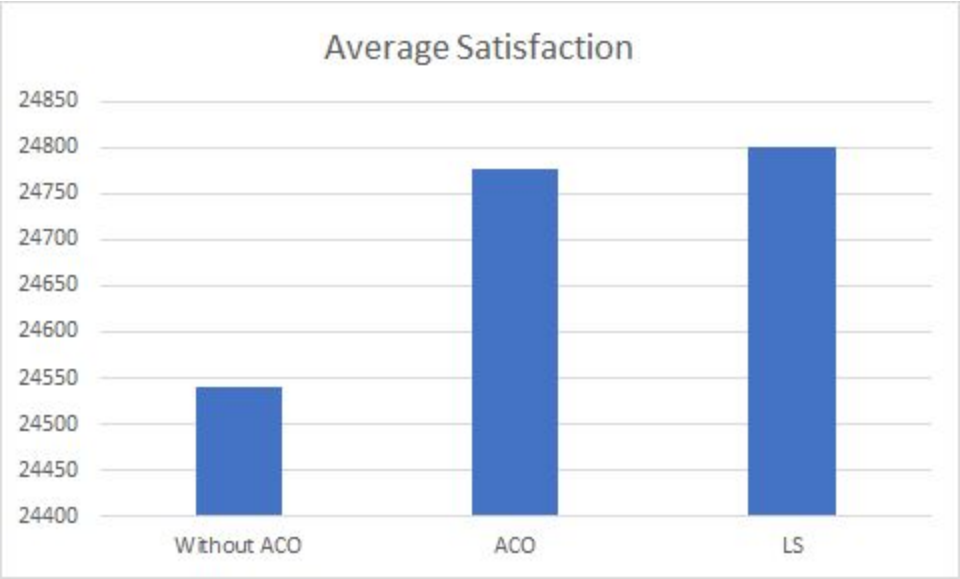
Algorithm 2 The Computing Complexity of ACO Algorithm

```
for i = 1 : m do
    for j = 2 : n do
        update tabu.
        for k = 2 : j − 1 do
            update A.
        end for
        update 'allowed'.
        for s = 1 : length(allowed) do
            update the probability of selecting next 'group'.
        end for
    end for
end for
```

The ACO algorithm's time complexity is $O(mn^2)$, where m is the number of ants in ACO and n is the number of patient groups.

III. Results and Discussion





To see the performance of the algorithm proposed in this paper, we executed three types of algorithms:

1. The naive random algorithm
2. The ACO algorithm
3. The ACO algorithm combined with random restructure local search

The algorithms were executed for a specific amount of time, and the solutions were compared to the actual most optimal solution. The average value for 10 executions was calculated to get a fair idea of the degree of effectiveness of the algorithm proposed in the paper, namely the ACO and ACO with LS.

As shown by the above graphs, the performance of the ACO algorithm was relatively superior as compared to that of the random algorithm. The modified ACO algorithm, i.e., ACO with LS, fared slightly better than the traditional ACO algorithm.

Thus, the results have shown us that the modified ACO algorithm solves the DPMCP quite accurately in a relatively short time.

IV. Conclusion

The experiments show that the ACO algorithms proposed in this paper quickly and effectively can solve the DPCMP. The heuristic operators in this paper could enhance the performances and efficiency of the ACO algorithm. The improvised ACO algorithm improves the time complexity of the given problem from O(n-1)! to $O(mn^2)$. This is a significant improvement as this is an NP-hard problem where $O(mn^2)$ can be considered a good time complexity. The paper implemented in this project can be quite useful for the Indian context. An application portal can be implemented using the proposed algorithm in the backend to make the lives of patients easier in India. The condition of public hospitals in our country is quite similar to our Chinese counterparts as people have to wait in long queues to get their health check-up done. Once this algorithm's potential can be harnessed to its full potential, the conflicts between patients and doctors should reduce.