Profit Optimisation through Employee and Budget Redistribution in a Restaurant Conglomerate

Akshat Nigam (200107006), Noel Tom (200107055)
Department of Chemical Engineering IIT Guwahati

***Abstract*—This paper concerns the redistribution of the number of different types of chefs, and the budget allocated for marketing and restaurant ambiance between a fine dining and a casual dining restaurant owned by a single company, for maximizing profit using meta heuristic techniques. This is a single objective optimization.**

I. Introduction

This is an age where companies try to diversify their businesses in different sectors and cater to different types of audiences for risk management and to reduce overall loses. Many restaurant brands like Landry's Restaurant Corp., Speciality Restaurant Ltd. and Sage Restaurant Group own a plethora of both casual and fine dining restaurants, hence catering to a larger audience and reducing the loses associated with one of the businesses not performing well. Hence, for optimizing profits earned, companies need to think of investment strategies for different types of restaurants depending on the demographic and spending nature of the customers.

Our goal is to develop an effective strategy for allocating resources, such as the number of chefs (Head, Assistant, and Junior), marketing and ambiance budgets, in a casual dining (CD) and fine dining (FD) restaurant owned by the same company in the same location, catering to different subcategories of customers in order to increase profits. We will also devise a plan for reinvesting quarterly profits into marketing and ambiance budgets to further maximize growth. There are several factors that affect profits, including the location's demographics, customer spending limits, budget limitations, employee salaries, marketing and ambiance costs. While a company may operate two different types of restaurants to target specific audiences, it can be difficult to determine the optimal setup for maximum profit or minimum loss. Our aim is to optimize resource allocation and recommend the best employee team and budget for each domain to increase profitability.

II. Literature

Several studies on restaurant revenue management (RRM) have been published over the past three decades. Sheryl E. Kimes (1999,1999,1998) [1] [2] [3] proposed using the revenue per available seat hour (RevPASH) as the restaurant's performance metric. Vakharia (1992) [4] used heuristics to find the best way to optimize wages and hour preferences of part-time employees to minimize cost. D. Bertsimas and R. Shioda (2003) [5] proposed a dynamic table-seating model as opposed to the traditional FCFS model to increase profits.

Most of the studies in this area focus on maximizing profits for one restaurant through the n-number of techniques listed above as opposed to distributing resources and personnel between two different types of restaurants catering to different customers to generate maximum profit, which is what this paper deals with.

III. Problem modeling

A. Input description

We are given the following parameters as input,

1) $P =$ Array of size 4 which denotes the quarterly number of restaurant-going population of the city in the rich, upper middle, lower middle, and lower class respectively.
2) $S =$ Array of size 6 which denotes the quarterly salary of each of the 3 types of chef(Head, Assistant, Junior) for FD and CD restaurant respectively.
3) $F =$ Array of size 4 which denotes the average fee paid by rich at FD, upper middle at FD, lower middle at CD, lower at CD per visit.
4) $S_C =$ Array of size 6 which denotes the maximum serving capacity of one chef in the respective category in a quarter year.
5) $EmpCap =$ Maximum number of chefs that can be employed
6) $SalMax =$ Maximum quarterly budget allocated for salary of chefs
7) $MA_B =$ Initial quarterly budget allocated for marketing and ambiance

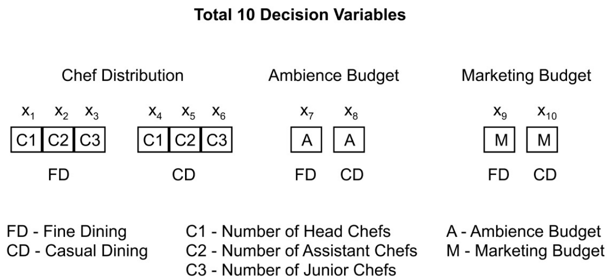
B. Assumptions

Some assumptions were made while solving this problem are:
1) The rich and upper middle class population go solely to FD while lower middle and lower class population go solely to CD.
2) People preferring FD give more weightage to a quality chef (quality food) and the restaurant ambiance and people preferring CD give more weightage to an assistant chef(less-priced food) and marketing of the restaurant.

.

C. Decision Variables

For a restaurant chain, we need 10 decision variables representing 3 classes of chefs in FD, 3 classes of chefs in CD, an ambiance budget in FD and CD and a marketing budget in FD and CD respectively.



D. Objective function

We need to *maximize the quarterly profit* earned by the restaurant company.

PR_k is the probability of different types of customers coming to a restaurant based on number of different types of chefs

$$PR_k = (\sum_i (x_i/xt_j) * w_{ik}) \tag{1}$$

where,
x_i is the number of head, assistant and junior chefs in FD ($i = 1, 2, 3$) and CD ($i = 4, 5, 6$) respectively.
xt_j is the total number of chefs in FD ($j = 1$) and CD ($j = 2$).
$w_{i,k}$ is the weight of a class of population due to the chef in that dining ($k = 1, 2, 3, 4$)

A_{factor} and M_{factor} determine the relative budget allocated to Ambiance and Marketing in FD and CD

$$A_{factor}(j) = \min(x_i/MA_B, 1) * B/UB_i \tag{2}$$

where $i = 7, 8$

$$M_{factor}(j) = \min(x_i/MA_B, 1) * B/UB_i \tag{3}$$

where $i = 9, 10$

A_{loss} and M_{loss} account for the loss of customers due to a lack of investment in Ambiance and Marketing in FD and CD.

$$M_{Loss}(k) = w_{ik} * (1 - M_{factor}(j)) * P(k) \tag{4}$$

$$A_{Loss}(k) = w_{ik} * (1 - A_{factor}(j)) * P(k) \tag{5}$$

The new population after accounting for the losses is given by $NewPop$

$$NewPop(k) = P(k) - A_{Loss}(k) - M_{Loss}(k) \tag{6}$$

The total serving capacity of each restaurant

$$S_{Cj} = \sum_i (Ec_i * x_i) \tag{7}$$

where for $j = 1, i = 1, 2, 3$; for $j = 2, i = 4, 5, 6$

Quarterly revenue is given by,

$$R = \sum_k (F_k * \min(NewPop(k), S_{Cj}) * PR_k) \tag{8}$$

where for $j = 1, k = 1, 2$; for $j = 2, k = 3, 4$

Total salary of chefs for one quarter

$$TotSal = \sum_i (S_i * x_i) \tag{9}$$

Marketing and Ambiance costs are given by,

$$A_{cost} = x_7 + x_8 \tag{10}$$

$$M_{cost} = x_9 + x_{10} \tag{11}$$

The total quarterly profit from both FD and CD is given by,

$$Profit = R - (TotSal + A_{cost} + M_{cost}) \tag{12}$$

20% of profit from each quarter is re-invested in marketing and ambiance for next quarter.

E. Constraints

We have the following constraints-

1) Employee salary constraint:
$\sum_i (S_i * x_i) <= SalMax; i = 1, 2, ...6$

2) Total employee count constraint i.e
$\sum_i (x_i) <= EmpCap; i = 1, 2, ...6$

3) Ambiance and marketing budget constraint
$x_7 + x_8 + x_9 + x_{10} < MA_B$

4) Number of head chefs should be greater than the average of the other two types in FD
$2 * x_1 >= x_2 + x_3$

5) Number of head chefs should be greater than the average of the other two types in CD
$2 * x_4 >= x_5 + x_6$

IV. Using Teaching and learning based optimization algorithm (TLBO)

A. Introduction

The Teaching and Learning Based Optimization (TLBO) is a population-based method that mimics the knowledge flow inside a classroom, presented by RV Rao [6]. The TLBO method is mathematically formulated as follows:
For the teaching phase,

$$X_{new,i} = X_i + r(X_{teacher} - T_F * X_{mean}) \tag{13}$$

$$\text{if } f(X_{new,i}) < f(X_i) \text{ then } X_i = X_{new,i} \tag{14}$$

$$\text{if } f(X_{new,i}) > f(X_i) \text{ then } X_i = X_i \tag{15}$$

where, $X_{new,i}$ is the updated position of i^{th} agent, x_i is its current location, the teaching factor, $T_F = 1$ or 2 Also, X_{mean} is the mean of all agents and it is formulated as follow:

$$\mathbf{X}_{mean} = \left[m\sum_{j=1}^{n_p}(\mathbf{x}_j)^1, m\sum_{j=1}^{n_p}(\mathbf{x}_j)^2, \ldots, m\sum_{j=1}^{n_p}(\mathbf{x}_j)^{nd} \right] \tag{16}$$

in which, np shows the number of students, m(.) returns the mean value of any inputs and nd indicates the problem dimension. The learning phase mathematically is formulated as follows:

$$X^{new,i} = X^i + r.(X^i - X^j) \text{ if } f(X^j) <= f(X^j) \tag{17}$$

$$X^{new,i} = X^i + r.(X^j - X^j) \text{ if } f(X^j) > f(X^j) \tag{18}$$

Where r is the random value and X_i and X_j are two different members of the population. If $X^{new,i}$ improves the objective value, it is accepted otherwise, it is rejected and Xi is maintained.

V. Genetic Algorithm(GA)

A genetic algorithm (GA) []is a metaheuristic inspired by the process of natural selection and genetics. It is used to find approximate solutions to optimization problems. The algorithm operates on a population of potential solutions, applying the principles of selection, crossover, and mutation to evolve the population towards an optimal solution.

In a GA, the selection operator is used to choose individuals from the current population to produce offspring for the next generation. One common selection strategy is tournament selection. In k-way tournament selection, k individuals are chosen at random from the population and the fittest individual among them is selected as a parent.

The crossover operator combines the genetic information of two parent individuals to produce offspring that inherit characteristics from both parents. There are many different crossover techniques that can be used, such as one-point, two-point, and uniform crossover.

The mutation operator introduces random changes into the genetic information of an individual. This helps to maintain diversity in the population and prevents premature convergence to a suboptimal solution.

A. Pseudo-code

Algorithm 1: Pseudo code for GA

1. Initialize population with random individuals
2. Evaluate fitness of each individual
3. Repeat until termination condition is met:
a. Select parents using tournament selection
b. Perform crossover to create offspring
c. Perform mutation on offspring
d. Evaluate fitness of offspring
e. Select individuals for next generation based on their fitness

VI. Using Firefly Algorithm(FA)

The firefly algorithm (FA), introduced by Yang, is a search method that imitates the behavior of fireflies in the natural world. It is based on how fireflies flash and attract others of their species. It is modelled mathematically as follows:

- The firefly algorithm assumes that all fireflies are attracted to each other regardless of their gender.
- The level of attraction is directly proportional to the brightness of the fireflies. Thus, a brighter firefly will be more attractive to others and the less bright one will move towards it.
- The level of attractiveness also depends on the distance between fireflies; the farther apart they are, the less attractive they become.
- If a firefly is isolated from others, it performs random movements.
- The brightness of fireflies is influenced by the objective function landscape, such as in foggy weather where close fireflies may not be able to see each other.

According to the given information, the firefly algorithm relies on two key factors: the light intensity of agents and the way their attraction levels are formulated. The attraction of fireflies to the light intensity of a particular firefly depends on their distance, and this can be mathematically defined using an exponential function, as shown below:

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \tag{19}$$

where, r_{ij} indicate the distance between a pair of fireflies and can be defined as Euclidean distance as $r_{ij} = ||X_i - X_j||$ and $\beta 0$ and γ are used to represent the attractiveness level and light absorption properties of a state. The movement of the fireflies is formulated as follows:

$$\Delta x_i = \beta_0 e^{-\gamma r_{ij}^2} * (x_i - x_j) + \alpha(rand - 1/2) \tag{20}$$

$$x_i^{t+1} = x_i^t + \Delta x_i \tag{21}$$

α is a random number uniformly generated from [0,1] interval and 'rand' provides a vector of random numbers selected from [0,1] interval and β_0=1.

VII. Proof of concept

To show that the algorithm is capable of converging to a correct solution, we take a scaled down version of the problem and find the best solution by going through all possible solutions in an iterative manner and compare it with the solution generated by our algorithm for the first quarter. We will be solving a bigger dataset and compare multiple runs of the algorithm later.

We take the following problem for the following input:

- 10 decision variables
- $2 <= x_i <= 4; i = 1, 2, 3, 4, 5, 6$
- $200 <= x_i <= 240; i = 7, 8, 9, 10$
- Domain constraints:

Algorithm 2: Pseudo code for FA

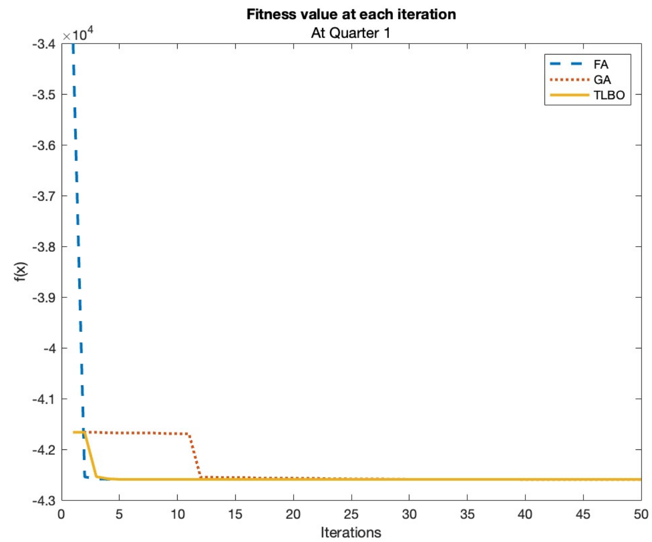
Input: Internal algorithm parameters
Output: Optimized solution
Generate n random fireflies; Ii is the light intensity at Xi determined by f(Xi); **while** *not termination condition* **do**
  for *each agent i* **do**
    for *each agent j* **do**
      if *Ij* > *Ii* **then**
        Firefly i should move towards Firefly j;
      end
    end
    Adjust attractiveness according to distance r via $exp[-\gamma r]$; Evaluate new solutions and update light intensity;
  end
end

1) $\sum_i (S_i * x_i) <= SalMax; i = 1, 2, ...6$
2) $\sum_i (x_i) <= EmpCap; i = 1, 2, ...6$
3) $x_7 + x_8 + x_9 + x_{10} <= MA_B$
4) $2 * x_1 >= x_2 + x_3$
5) $2 * x_4 >= x_5 + x_6$

- Max Employee Count, $EmpCap$ = 20
- Marketing and Ambience Budget, MA_B = 850
- Max Salary, $SalMax$ = 25000
- Population, P = [1000 3000 3000 2000]
- Employee Salary, S = [4000 2500 1500 1000 800 500]
- Average Fee Paid, F = [1000 400 300 150]
- Serving Capacity, S_C = [15 12 10 30 20 10]

The iterative solutions gives us the exact solution, however it took around 50 minutes to complete all iterations, which gave the maximum profit as 43692. Whereas our solution using TLBO, FA and GA ran for less than a second and obtained the following result:

$TLBO-> 42549.4242$
$FA-> 42595.4242$
$GA-> 42549.4242$
As we can see the profit obtained is very close to the solution obtained from the iterative method.



Thus, it proves that the meta-heuristic solution works and gives a solution which is comparable to the iterative solution, with a significant improvement in run time and other resources. **NOTE: The solution converges very quickly because the data set is very small. This is just a proof to show that the algorithm is capable of finding the optimal solution.**

VIII. Solving a Problem

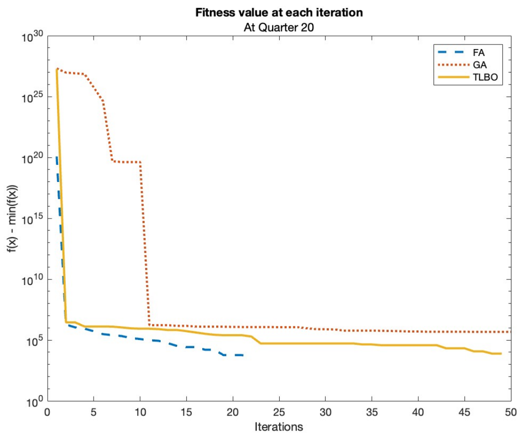
We ran the meta-heuristic code for the following input- 3 types of chefs, Ambiance and marketing cost in each restaurant, Maximum employee cap is 25, Maximum Salary Budget is 4000000, defined population, salary, average spending from each type of customer, and serving capacity of the chefs. We have the following variables and constraints:

- 10 decision variables
- $2 <= x_i <= 10; i = 1, 2, 3, 4, 5, 6$
- $18000 <= x_i <= 1800000; i = 7, 8, 9, 10$
- Domain constraints:
  1) $\sum_i (S_i * x_i) <= SalMax; i = 1, 2, ...6$
  2) $\sum_i (x_i) <= EmpCap; i = 1, 2, ...6$
  3) $x_7 + x_8 + x_9 + x_{10} <= MA_B$
  4) $2 * x_1 >= x_2 + x_3$
  5) $2 * x_4 >= x_5 + x_6$
- Max Employee Count, $EmpCap$ = 25
- Initial Marketing and Ambience Budget, MA_B = 130000
- Max Salary, $SalMax$ = 4000000
- Population, P = [200000 270000 270000 180000]
- Employee Salary, S = [300000 200000 135000 90000 72000 45000]
- Average Fee Paid, F= [1000 400 300 150]
- Serving Capacity, S_C = [1400 1200 900 2700 1800 900]
- Re-investing 20% of profit from each quarter in marketing and ambiance for next quarter.

We ran these these for 20 quarters; that is 5 years and reinvested an amount from the profit of each quarter into marketing and ambiance for next quarter. The results are displayed in the next section.

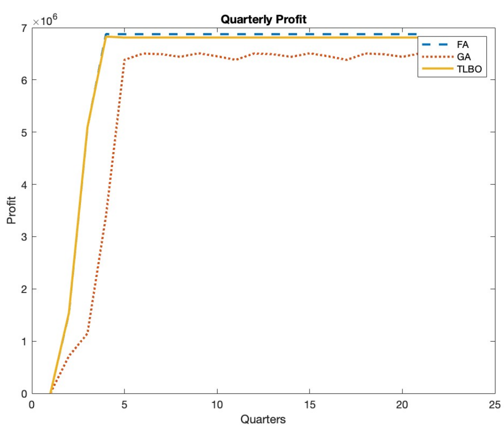
IX. RESULTS

The following graph demonstrates the fitness function of each algorithm (TLBO, FA, GA) at each iteration in the 20th quarter.

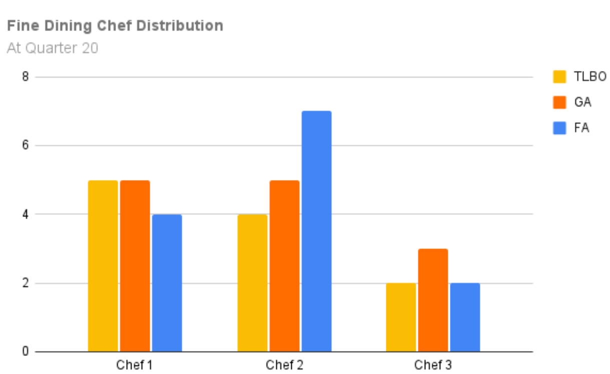


In the below graphs can see the change in various parameters with each passing quarter.

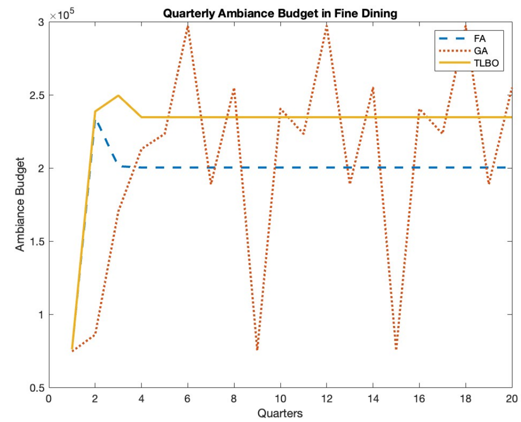
A. Quarterly Profit



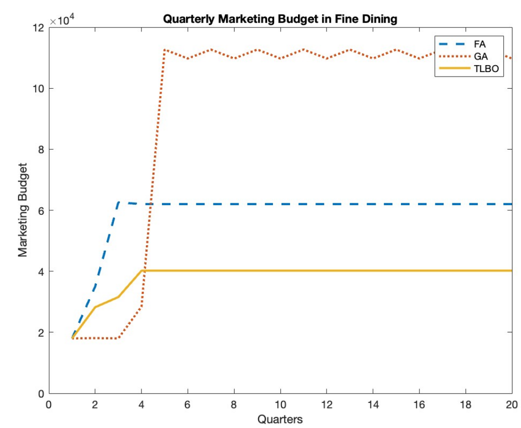
B. Chef Distribution for FD



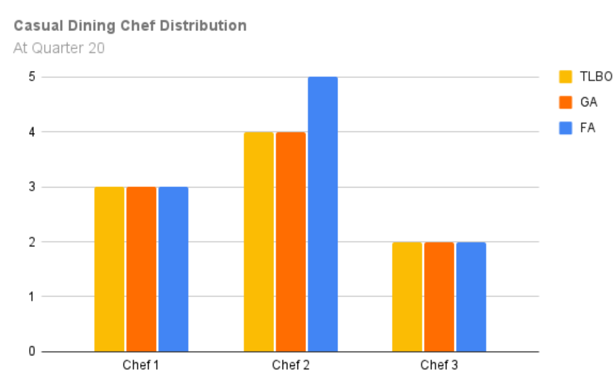
C. Ambience Budget for FD



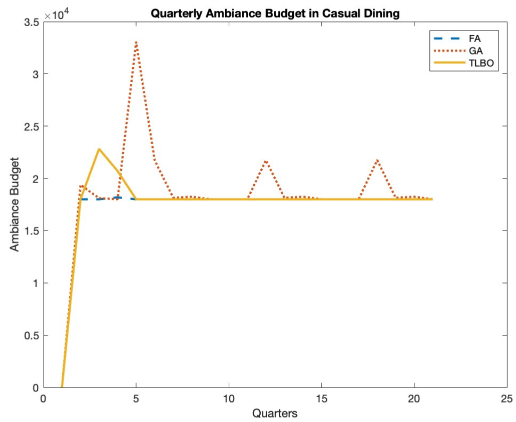
D. Marketing Budget for FD



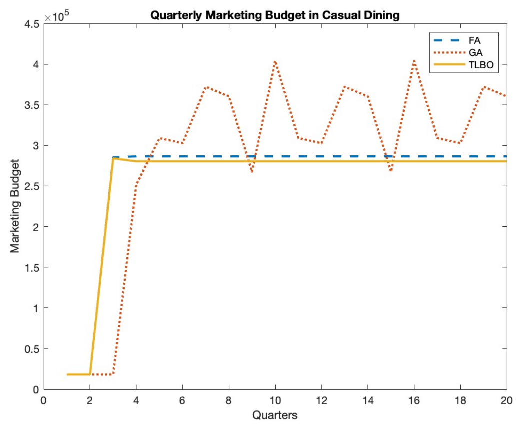
E. Chef Distribution for CD



F. Ambience Budget for CD



G. Marketing Budget for CD



H. Comparison of the 3 Algorithms

Algorithm	Population Size	Iterations**	Execution Time (sec)	Final Fitness*
TLBO	100	50	1.703	-6810537.66
GA	100	300	5.026	-6509412.56
FA	100	25	30.202	-6875468.35

*at Quarter 20
**per quarter

X. Conclusion

Comparing the 3 algorithms used for this problem (TLBO, FA and GA) we recieve comparable results from all the 3 algorithms, however GA showed more deviations in its results due to its exploratory nature. FA gave the best fitness/profit, however it took the longest computational time. TLBO gave the quickest results. Meta heuristic techniques does not guarantee global optimal, however the solution converging for multiple runs indicates that the algorithm is capable of always reaching a optima if it exists.

References

[1] S. E. Kimes, "Implementing restaurant revenue management: A five-step approach," *Cornell Hotel and Restaurant Administration Quarterly*, vol. 40, no. 3, pp. 16–21, 1999.

[2] S. E. Kimes, D. I. Barrash, and J. E. Alexander, "Developing a restaurant revenue-management strategy," *Cornell Hotel and Restaurant Administration Quarterly*, vol. 40, no. 5, pp. 18–29, 1999.

[3] S. E. Kimes, R. B. Chase, S. Choi, P. Y. Lee, and E. N. Ngonzi, "Restaurant revenue management: Applying yield management to the restaurant industry," *Cornell Hotel and Restaurant Administration Quarterly*, vol. 39, no. 3, pp. 32–39, 1998.

[4] A. Vakharia, H. Selim, and R. Husted, "Efficient scheduling of part-time employees," *Omega*, vol. 20, no. 2, pp. 201–213, 1992.

[5] D. Bertsimas and R. Shioda, "Restaurant revenue management," *Operations research*, vol. 51, no. 3, pp. 472–486, 2003.

[6] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-aided design*, vol. 43, no. 3, pp. 303–315, 2011.

Appendix A
Code for Algorithm

```
function [f,profit] = OBJ(x,ub,lb,B)

    % number of higher, upper middle, lower middle and lower class population
    P=[200000 270000 270000 180000]';
    % salary of each type of chef
    S=[300000 200000 135000 90000 72000 45000]';
    % avg spending from each type of customer at each restaurant
    F=[1000 400 300 150]';
    % serving capacity
    SC=[1400 1200 900 2700 1800 900]';

     EmpCap = 25;                           % Maximum number of Employees to be occupied
     SalMax= 4000000;                       % Available Budget for Salary of Employees
     TotalFineChef=x(1)+x(2)+x(3);          % Total number of chefs in fine dining restaurant
     TotalCasualChef=x(4)+x(5)+x(6);        % Total number of chefs in casual dining restaurant
     AmbianceCost1=x(7);                    % Amount allocated for Ambiance in fine dining restaurant
     AmbianceCost2=x(8);                    % Amount allocated for Ambiance in casual dining restaurant
     MarketingCost1=x(9);                   % Amount allocated for marketing in fine dining restaurant
     MarketingCost2=x(10);                  % Amount allocated for marketing in casual dining restaurant

%% Evaluation of probability of customers coming to restaurants based on number of different types of chefs

     PR=[0 0 0 0]';

     PR(1)= 0.5*(x(1)/TotalFineChef)+0.3*(x(2)/TotalFineChef)+0.2*(x(3)/TotalFineChef);
     PR(2)= -0.1*(x(1)/TotalFineChef)+0.5*(x(2)/TotalFineChef)+0.6*(x(3)/TotalFineChef);
     PR(3)= 0.2*(x(4)/TotalCasualChef)+0.5*(x(5)/TotalCasualChef)+0.3*(x(6)/TotalCasualChef);
     PR(4)= 0.1*(x(4)/TotalCasualChef)+0.2*(x(5)/TotalCasualChef)+0.7*(x(6)/TotalCasualChef);

%% Evaluation of New population based on the nature of investment in Ambiance and Marketing

     NewPop=P;       % initially new population equals initial population pool
```

```matlab

    % Following factors determines the relative amounts allocated to
    % Ambiance nad Marketing in Fine and Casual Dining Restaurants

    A_factor(1)=min(AmbianceCost1/B,1)*B/ub(7);
    A_factor(2)=min(AmbianceCost2/B,1)*B/ub(8);
    M_factor(1)=min(MarketingCost1/B,1)*B/ub(9);
    M_factor(2)=min(MarketingCost2/B,1)*B/ub(10);

    % Following factors account for the loss of customers due to lack of
    % investing in Ambiance and Marketing in Fine and Casual Dining Restaurants

    MLoss(1)=0.3*(1-M_factor(1))*P(1);
    MLoss(2)=0.6*(1-M_factor(1))*P(2);
    MLoss(3)=0.5*(1-M_factor(2))*P(3);
    MLoss(4)=0.7*(1-M_factor(2))*P(4);
    ALoss(1)=0.7*(1-A_factor(1))*P(1);
    ALoss(2)=0.4*(1-A_factor(1))*P(2);
    ALoss(3)=0.5*(1-A_factor(2))*P(3);
    ALoss(4)=0.3*(1-A_factor(2))*P(4);
    NewPop(1)=P(1)-MLoss(1)-ALoss(1);
    NewPop(2)=P(2)-MLoss(2)-ALoss(2);
    NewPop(3)=P(3)-MLoss(3)-ALoss(3);
    NewPop(4)=P(4)-MLoss(4)-ALoss(4);

%% Revenue Calculations

   % Calculating serving capacity of chefs
   CasualChefCap=0;         % Serving capacity of chefs in casual dining restaurant
   FineChefCap=0;           % Serving capacity of chefs in fine dining restaurant
   for j=1:length(SC)
       if j>=4
           CasualChefCap = CasualChefCap + SC(j)*x(j);
       else
           FineChefCap = FineChefCap + SC(j)*x(j);
       end
   end

   % Calculating revenue generated
   Revenue=0;
   ServingCap=FineChefCap;
   for j = 1: length(F)
       if j>=3
           ServingCap=CasualChefCap;
       end
       Revenue = Revenue + F(j)*(min(NewPop(j),ServingCap))*PR(j);
   end

%% Cost Calculations

   % Chef salary calculation
   EmpSalary=0;
   for j = 1: length(S)
       EmpSalary = EmpSalary + S(j)*x(j);
   end

%% Penalties and Constraints

   % Employee Salary constraint
   PenaltySalary = 0;                          % penalty due to exceeding employee salary budget
   if  EmpSalary > SalMax
       PenaltySalary = (100*(EmpSalary - SalMax))^2;
   end
   % Total Employee Count constraint
   TotalEmp = TotalFineChef + TotalCasualChef;
   PenaltyEmp = 0;                             % penalty due to exceeding maximum employee count
   if  TotalEmp > EmpCap
       PenaltyEmp = (100*(TotalEmp - EmpCap))^2;
   end

   % Ambiance and marketing budget constraint
   PenaltyExtraBudget = 0;                     % penalty due to exceeding Ambiance and marketing budget
   if  x(7)+x(8)+x(9)+x(10) > B
       PenaltyExtraBudget = (x(7)+x(8)+x(9)+x(10) - B)^2;
   end

   % Number of senior chef constraint in fine dining
   PenaltySenior1 = 0;                         % penalty due to violating senior chef constraint in fine dining
   if 2*x(1) < x(2)+x(3)
       PenaltySenior1 = (10000*(2*x(1) - (x(2)+x(3))))^2;
   end

   % Number of senior chef constraint in casual dining
   PenaltySenior2 = 0;                         % penalty due to violating senior chef constraint in fine dining
   if 2*x(4) < x(5)+x(6)
       PenaltySenior2 = (100*(2*x(4) - x(5)+x(6)))^2;
   end
%% Profit and Fitness Calculations

   profit = Revenue - (EmpSalary + AmbianceCost1 + AmbianceCost2 + MarketingCost1 + MarketingCost2);   % Profit

   f = -profit + 10^15*(PenaltyEmp + PenaltySenior1 + PenaltySenior2 + PenaltyExtraBudget + PenaltySalary);   % Fitness value
```