CL643 Computer-Aided Applied Optimisation

PROFIT OPTIMISATION THROUGH EMPLOYEE AND BUDGET REDISTRIBUTION IN A RESTAURANT CONGLOMERATE

Akshat Nigam (200107006)
Noel Tom (200107055)

TABLE OF CONTENT

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 losses.

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 and fine dining 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, and 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.

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. This value is calculated for each time period of each day, to identify times that the RevPASH is low. 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

PROBLEM MODELING

INPUT DESCRIPTION

We are given the following parameters as input

- 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.
- S = Array of size 6 which denotes the quarterly salary of each of the 3 types of chefs (Head, Assistant, Junior) for FD and CD restaurants respectively.
- F = Array of size 4 which denotes the average fee paid by the rich at FD, upper middle at FD, lower middle at CD, and lower at CD per visit.
- SC = Array of size 6 which denotes the maximum serving capacity of one chef in the respective category in a quarter year
- EmpCap = Maximum number of chefs that can be employed
- SalM ax = Maximum quarterly budget allocated for the salary of chefs
- M AB = Initial quarterly budget allocated for marketing and ambiance

PROBLEM MODELING

ASSUMPTIONS

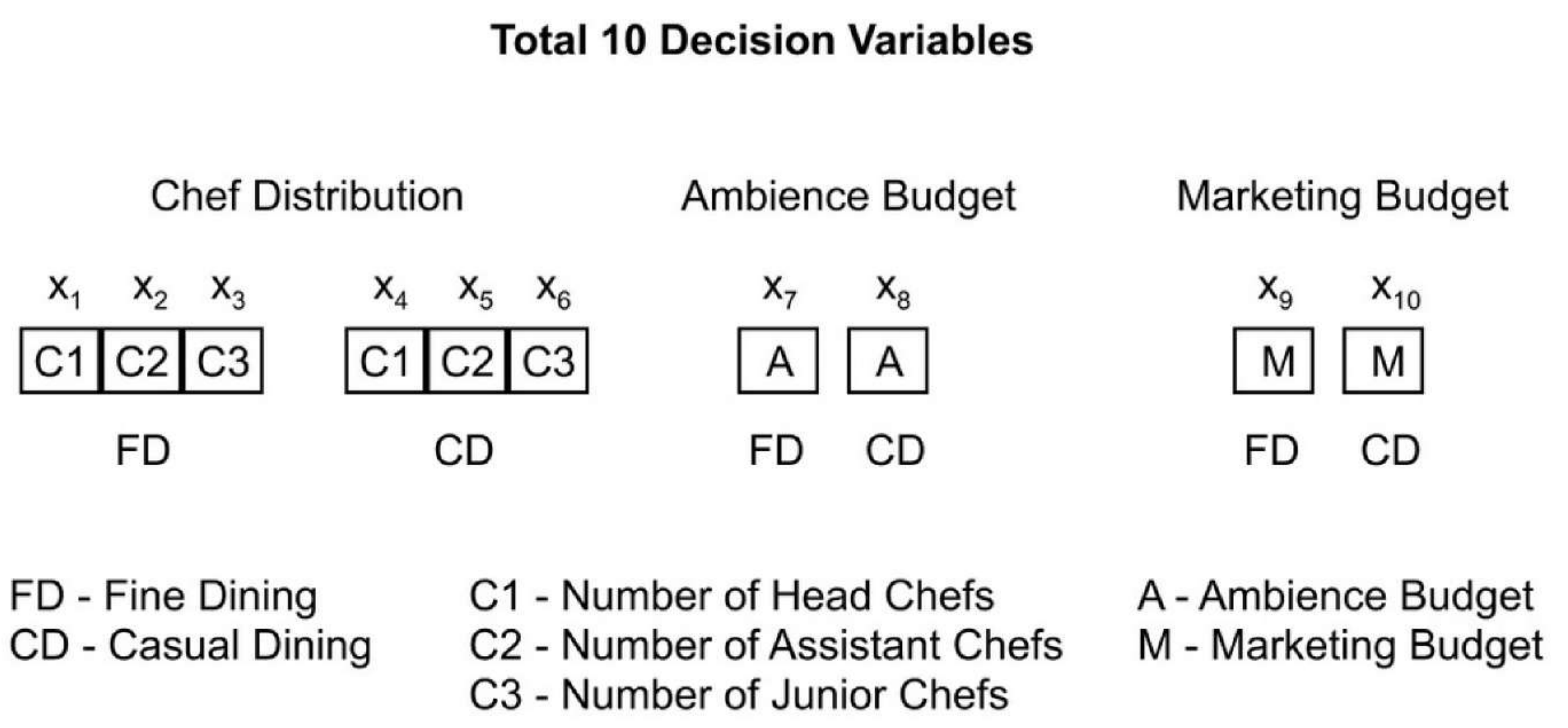
Some assumptions were made while solving this problem are:

1) Rich and Upper middle-class populations go solely to fine dining while Lower middle and Lower class people go solely to casual dining.
2) People preferring fine dining give weightage to a quality chef and ambiance and people preferring casual dining give weightage to an assistant chef(less-priced food) and marketing

PROBLEM MODELING

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



USING TEACHING AND LEARNING BASED OPTIMIZATION ALGORITHM

OBJECTIVE FUNCTION

We need to maximize the quarterly profit earned by the restaurant company.PRk is the probability of different types of customers coming to a restaurant based on number of different types of chefs

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

where,xi is the number of head, assistant and junior chefs in FD (i = 1, 2, 3) and CD (i = 4, 5, 6) respectively. xtj is the total number of chefs in FD (j = 1) and CD (j = 2). wi,k is the weight of a class of population due to the chef in that dining (k = 1, 2, 3, 4) Afactor and Mf actor determine the relative budget allocated to Ambiance and Marketing in FD and CD

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

where $i = 7, 8$

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

where $i = 9, 10$

USING TEACHING AND LEARNING BASED OPTIMIZATION ALGORITHM

OBJECTIVE FUNCTION

Aloss and Mloss account for the loss of customers due to a lack of investment in Ambiance and Marketing in FD and CD

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

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

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

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

The total serving capacity of each restaurant

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

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

USING TEACHING AND LEARNING BASED OPTIMIZATION ALGORITHM

OBJECTIVE FUNCTION

Quarterly revenue is given by

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

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

Total salary of chefs for one quarter

$$TotSal = \sum (S_i * x_i)$$

Marketing and Ambiance costs are given by

$$A_{cost} = x_7 + x_8$$

$$M_{cost} = x_9 + x_{10}$$

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

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

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

PROBLEM MODELING

CONSTRAINTS

We have the following constraints-

1. Employee Salary Constraint: EmploySalary < SalaryBudget
2. Total Employee Count constraint: TotalEmp < EmpCap

   TotalEmp = TotalFineChef + TotalCasualChef
3. Ambiance and marketing budget constraint:

   Ambiance cost + Marketing cost < B
4. Number of senior chef constraints in fine dining:

   $2*x(1) > x(2)+x(3)$
5. The number of senior chef constraints in casual dining:

   $2*x(4) < x(5)+x(6)$

USING TEACHING AND LEARNING BASED OPTIMIZATION ALGORITHM

INTRODUCTION

TLBO is a population based method that starts with a random class which contains possible solution candidates. These random candidates are called learners. This algorithm is two phase method which includes teaching phase and learning phase. The first phase focuses on the knowledge transferring from teacher to the learners while second phase models the learning process among the student through their individual interactions

USING TEACHING AND LEARNING BASED OPTIMIZATION ALGORITHM

The TLBO method is mathematically formulated as follows:

For the teaching phase,

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

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

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

where, Xnew,i is the updated position of ith agent, xi is its current location, the teaching factor, TF = 1 or 2 Also, Xmean is the mean of all agents and it is formulated as follow:

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

USING TEACHING AND LEARNING BASED OPTIMIZATION ALGORITHM

LEARNER PHASE

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

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

Where r is the random value and Xi and Xj are two different members of the population. If Xnew,i improves the objective value, it is accepted otherwise, it is rejected and Xi is maintained.

FIREFLY ALGORITHM

The firefly algorithm (FA) is originally introduced by Yang [29], this method drives its search patterns form the behavior of the fireflies in the real world. It is inspired from the flashing behavior of fireflies and its absorbing effect on their own species. Although the real behavior of fireflies might be complex, to get a mathematical model the idealizations are made as follows: All fireflies are considered as unisex species; it means that all of them regardless to their sexuality they can be attracted to each other. The attractiveness factor directly proportional to their brightness. Therefore, for any pair of fireflies the brighter is more attractive and consequently the less bright firefly will move toward it. Also, this attractiveness depends on the distance between them, so it is decreased when the distance of two fireflies is increased and vice versa. On the other words if a firefly stands so far from others, it is not attracted by any of them and so it performs a random movement. The landscape of the objective function of the optimiza- tion highly affects the brightness of the fireflies (e.g. in foggy weather even close fireflies may not realize each other). Based on this information the two important terms in the firefly algorithm are light intensity of the agents and proper formulation to devote the attraction level of the agents. Since the attraction of other fireflies to the light intensity of current firefly is depended to their distance, it can be defined via exponential function as below

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

where, rij indicate the distance between a pair of fireflies and can be defined as Euclidean distance as rij = ||Xi – Xj || and β0 and γ indicate the attractiveness level for the state and light absorption coefficient, respectively. Consequently, based on the given information the movement of the firefliess mathematically formulated as follow

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

where, superscript t+1 and t show the updated location of the firefly, respectively. Also, α is random number uniformly generated from [0,1] interval and 'rand' provides a vector of random numbers selected from [0,1] interval and β0=1 is set

FIREFLY ALGORITHM

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[−γr]; Evaluate new solutions and update
          light intensity;
    end
end
```

GENETIC ALGORITHM

INTRODUCTION

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.

GENETIC ALGORITHM

PSEUDO CODE

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

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:
  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]

PROOF OF CONCEPT

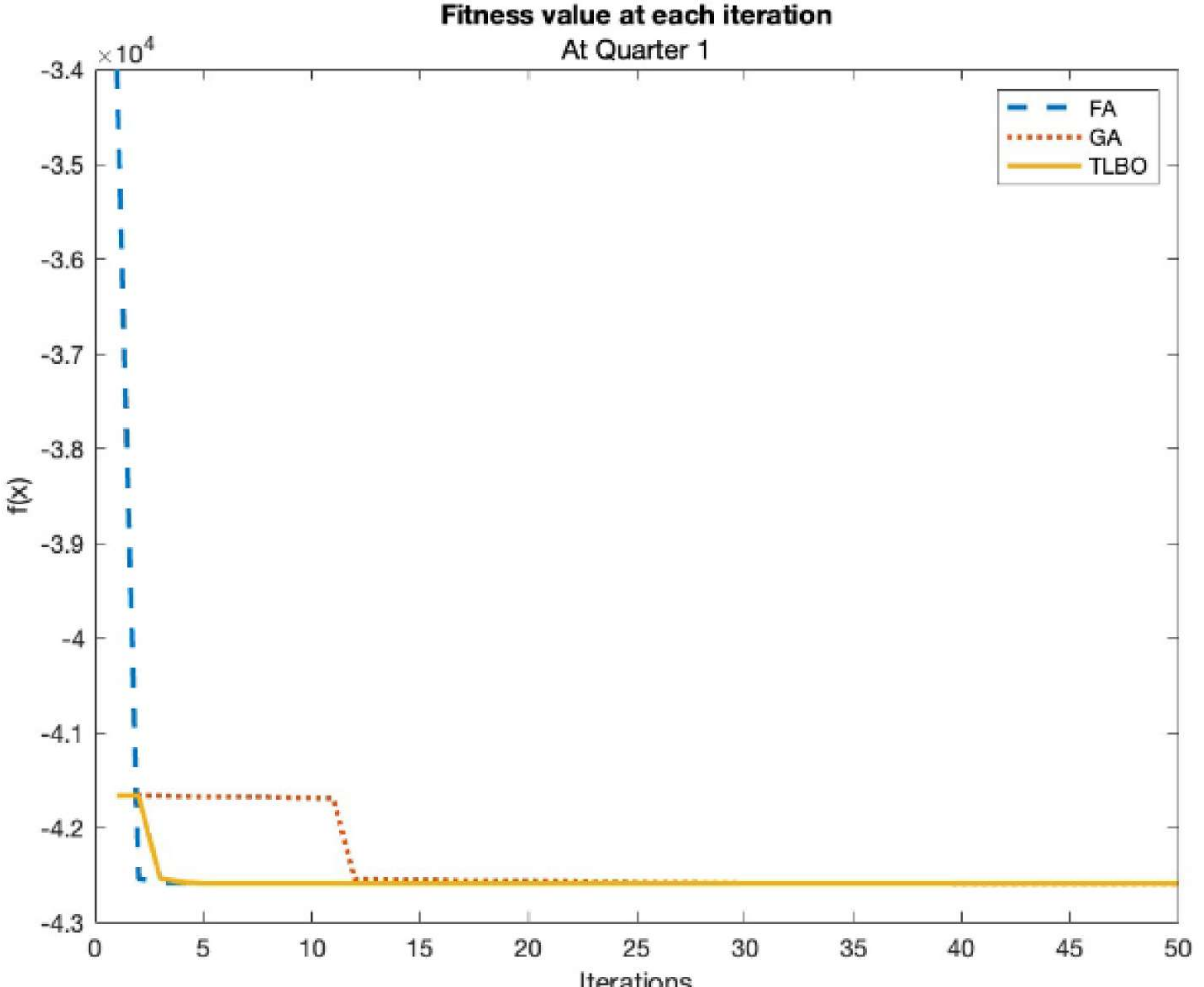
The iterative solutions gives us the exact solution, however it took around 50 minutes to complete all iterations, which gave the maximum profit as 43692. Whereas our solution using TLBO, FA and GA ran for less than a second and obtained the following result:

$TLBO- > 42549.4242$
$FA- > 42595.4242$
$GA- > 42549.4242$

As we can see the profit obtained is very close to the solution obtained from the iterative method.

PROOF OF CONCEPT

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



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

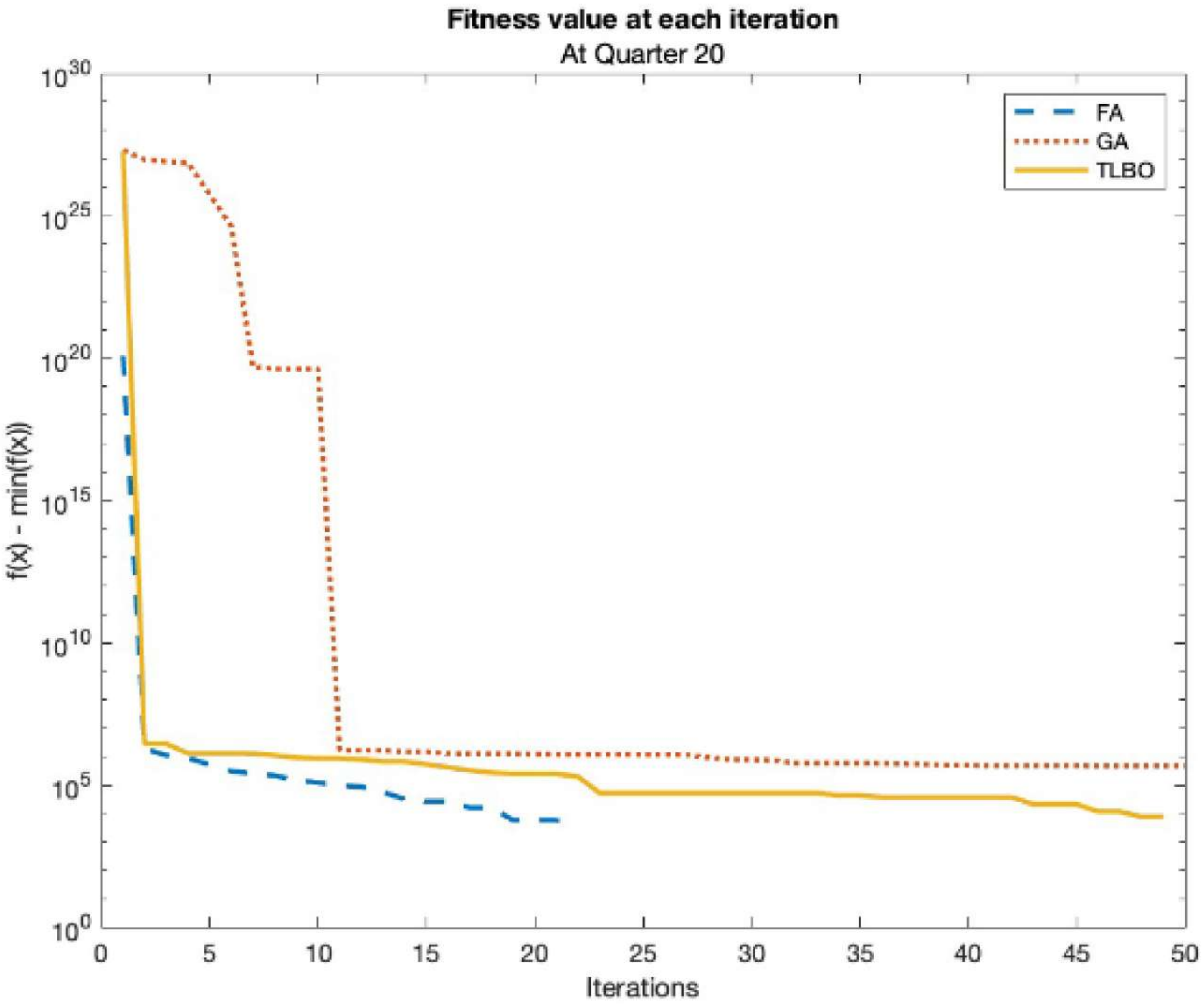
SOLVING A PROBLEM

- 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.

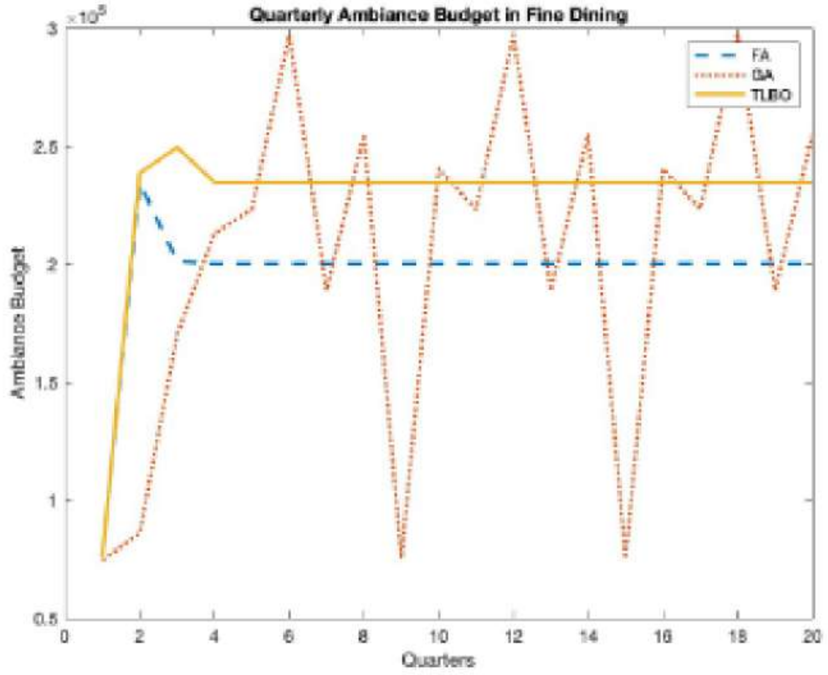
We ran these these for 20 quarters; that is 5 years and reinvested an amount from the profit of each quarter into marketing and ambiance for next quarter. The results are displayed in the next section.

RESULTS

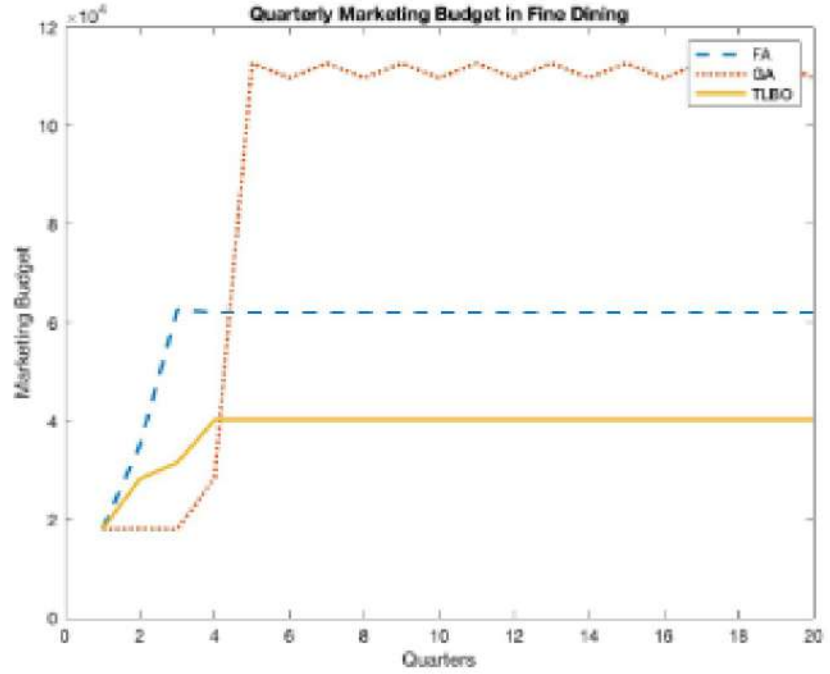
The following graph demonstrates the fitness function of each algorithm (TLBO, FA, GA) at each iteration in the 20th quarter.



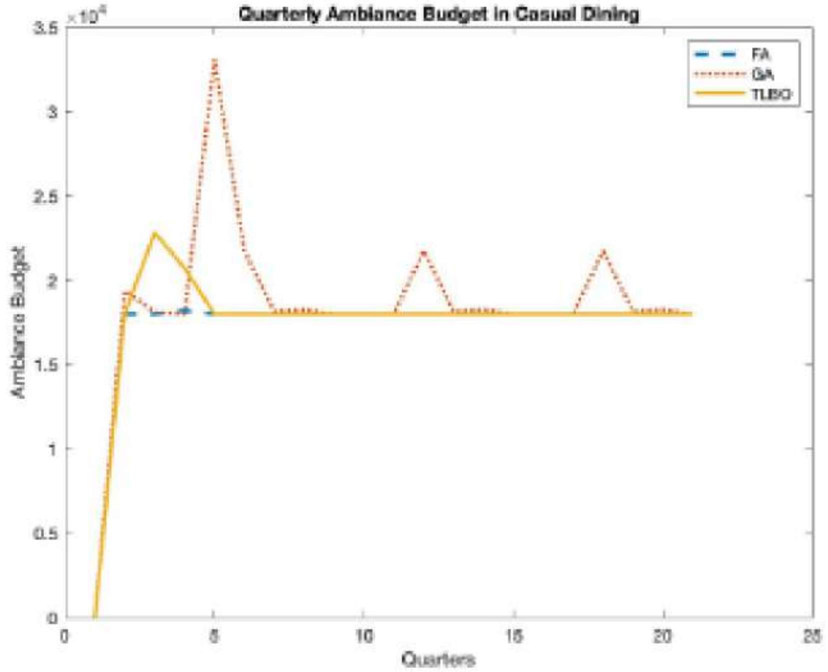
B. Ambience Budget for FD



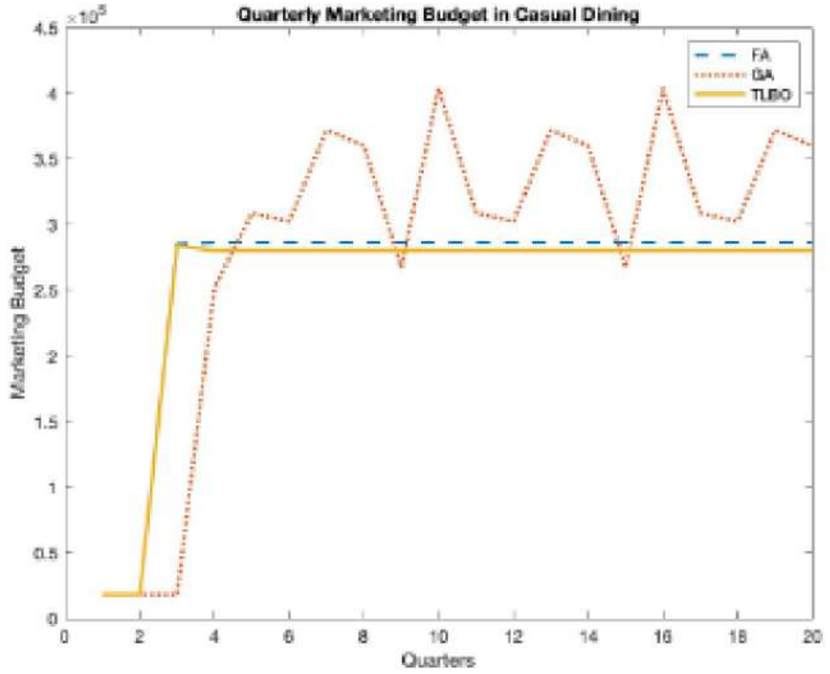
C. Marketing Budget for FD



E. Ambience Budget for CD



F. Marketing Budget for CD



X. CONCLUSION

Comparing the solution of four different runs we can see

THANK YOU