Multi-Objective Optimization Using Deep Reinforcement Learning for Food E-Commerce Industry

Guru Kiran H M[1], Harshit Singh[2], Geetanjali Koya[3], Archit Kumar Raj[4], Jyothi .R[5]

[1]Computer Science and Engineering, Student, PES University, Bangalore, Karnataka, India
Email: *gurukiran1743*[at]*gmail.com*

[2]Computer Science and Engineering, Student, PES University, Bangalore, Karnataka, India
Email: *harshitsingh1999*[at]*gmail.com*

[3]Computer Science and Engineering, Student, PES University, Bangalore, Karnataka, India
Email: *geetanjalikoya*[at]*gmail.com*

[4]Computer Science and Engineering, Student, PES University, Bangalore, Karnataka, India
Email: *arcraj14[at]gmail.com*

[5]Professor, Computer Science and Engineering Department, Assistant Professor, PES University, Bangalore, Karnataka, India
Email: *jyothir[at]pes.edu*

Abstract: ***In this study, we try and come up with an optimal solution for a given set of conflicting multiple objectives, it's in human nature to get the maximum out of a situation, which involves optimization of more than just a single objective at a time, thus we propose a model which can optimize multiple conflicting objectives and provide the best solution which has less trade-off between the objectives. Deep Reinforcement Learning (DRL) for Multi-Objective Optimization (MOO), deep reinforcement learning combines deep learning artificial neural networks with a reinforcement learning agent that uses a trial-and-error method to reach the goal. In this study we mainly consider the food delivery system and work with the Zomato dataset and optimize objectives such as cost per person, estimated time of arrival (ETA), and ratings of the restaurant, we try and find out the best-fit restaurants for the user based on his/her objectives at the time, we imbibe deep-reinforcement learning to achieve the task, by doing so we try to provide better user experience and contribute to the food e-commerce industry.***

Keywords: Multi-objective optimization (MOO), Deep Reinforcement learning (DRL), Particle Swarm Optimization (PSO), Multi-Objective particle swarm optimization (MOPSO), Personal best (Pbest), Global best (Gbest).

1. Introduction

The human brain is a greedy one and aims to get the best out of a situation given all the constraints. We focus on achieving better results that satisfy all our needs, we come across numerous situations where we need to solve a problem with two or more objectives, for instance, we are hungry and are looking for food items to order and look for the best food with more ratings while minimizing the estimated time of arrival and minimizing cost and the list goes on, hence Multi-objective optimization (MOO) has emerged as the preferable approach to solve the problems that require optimizing two or more functions based on one's requirement. We use Deep Reinforcement Learning (DRL) for Multi-Objective Optimization (MOO), deep reinforcement learning combines deep learning artificial neural networks with a reinforcement learning agent that uses a trial-and-error method to reach the goal.

Multi-Objective Optimization (MOO) has the potential to be a game-changer in a lot of fields that involve MOO, it is yet to be studied and explored, previous works on MOO involve solving these problems using genetic/evolutionary algorithms, these algorithms are now saturated and a benchmark accuracy in this field is yet to be set, thus in this study, we try different algorithms to produce the optimal solution with maximum accuracy. In this study we mainly consider the food delivery system and work with the Zomato dataset and optimize objectives such as cost per person, estimated time of arrival (ETA), and ratings of the restaurant, we try and find out the best-fit restaurants for the user based on his/her objectives at the time, we imbibe deep-reinforcement learning to achieve the task, by doing so we try to provide better user experience.

The project focuses on optimizing multiple objectives and providing the best possible solutions without many trade-offs between the objectives. We try to optimize different features such as cost per person, estimated time of arrival (ETA), and rating of the restaurant and provide better options of restaurants for the user to pick from an ocean of restaurants.

2. Related Work

2.1 Literature Survey

Literature survey is an important and essential step while working on a project. A thorough literature survey has been done to gain knowledge and understand the previous approaches taken to solve similar problems, it also helps in identifying gaps and shortcomings of the approaches previously taken, Moreover, a literature survey helps in understanding the algorithm, the accuracy of the algorithm and helps in laying the basement for the approach to be followed and lay out ideas to set a benchmark accuracy, Some of the major approaches and key points collected from the literature survey are as follows:

[1] Particles refers to all the feasible solutions, swarm refers to a group of solutions in consideration and these solutions are optimized to find the best fit solutions i.e., solutions in the Pareto front. It doesn’t have a genetic operator instead pics solutions at random finds its fitness value then finds the pbest and gbest value, it then calculates particle velocity, update its position, find its fitness and do these steps iteratively until the optimal solution set is found or the iteration number equals the maximum iteration value. This model is influenced by the behavior of fish and bird flocking.

[2] Similar to PSO, but includes a global margin ranking (GMR) system along with PSO, GMR: is calculated as the sum of differences of all optimal solutions. Corner points which represent dominated solutions have very high ranks and will be ignored by any other algorithm as it has huge trade-off between multiple objectives while GMR doesn’t discard corner points as it has huge information other than just high rank value. It also calculates GOLBAL DENSITY(GD) which is Euclidean distance between two optimal/non-dominated solutions, greater the GD farther away are the particles i.e., solutions thus greater spread, this makes it evenly distributed and makes it easier for user to choose among different optimal solutions. It also reduces bias.

[3] The self-organizing mapping network is used to collect data regarding population, structure of the distribution and the build corresponding neighborhood. The elite learning strategy avoids premature convergence that might occur and the non-dominated sorting method sorts optimal solutions with specific crowding distance, by doing so solutions with similar fitness value can be mapped into same neighborhood thus helps in optimizing solutions based on distance and obtain non-dominated optimal solution with less or no trade-off between multiple objectives. It performs better than the MO-ring PSO-SCD.

[4] Grey Wolf optimizer (GWO) alongside leaping frog algorithm (LFA) with memeplex structure of the shuffled frog, helps in solving MOO problems with more spread, convergence and diversity. Two approaches followed are, Priori Approach in which multi-objective optimization is solved converting it into single optimization problem using linear programming or weighted sum approach and Posteriori approach in which Independent successful results is obtained and less computational time is required to solve MOO problems using this approach. It’s a population based meta-heuristic algorithm, inspired by frogs, optimal solution set is divided into different groups called memeplex each group has its own information such as local best value which is known as meme, based on local best and global best of memeplexes the position of optimal solution is obtained and thus optimal solution is obtained.

[5] Double Niched Evolutionary Algorithm (DNEA),it’s an algorithm which adopts niche sharing methods, it adopts niche sharing with objective as well as decision spaces. Three polygon based problems were created and Double Niched Evolutionary algorithm was applied on the problems, the three variants, MOEA/D, DN-NSGA-II, NSGA-II, the DNEAobj failed to get multiple Pareto optimal solution sets but DNEA maintains diversity in decision space. Thus we could conclude, crowding distance methods perform poorly as compared to niche sharing method when it comes to maintaining diversity.

[6] A self-organizing MMOPIO algorithm is proposed, which efficiently maintains better Pareto solutions as they could preserve local topological properties, PIO consists two operators namely the landmark operator and the map-and-compass operator. SOM maps the input data to output data. In MMOPIO the neighboring leader pigeon is denoted by nbest, that is obtained by SOM. With increase in complexity of test function The performance of MMOPIO also increases but care must be taken while considering high computational cost.

[7] Formulation of MOTSP- Individual has to obtain a tour of n cities which is a cyclic permutation p to minimize M different cost functions simultaneously. The model: Encoder- used to condense the input sequence into a vector. Decoder-used to unfold the obtained high dimensional vector, which stores the knowledge of inputs into the output sequence. Attention Mechanism determines how much of each input will be used in the subsequent decoding phase. The most important one receives more consideration and may be chosen as the next visitor city.

2.2 Data

In Bengaluru, you may find restaurants which are famous all over the world. You may sample different cuisines, including the United States, Japan, Russia, and Antarctican cuisines. One can find all the options such as delivery, dine-in, pubs, bars, drinks, buffets, and desserts in Bengaluru. Foodies love Bengaluru. Restaurants are springing up all over the place. Approximately 12,000 eateries are currently open. With so many eateries to choose from, this market is still untapped. And more restaurants open each day. However, competing with existing established businesses has become a challenge for them. High real estate prices, rising food costs, a scarcity of qualified labor, a fragmented supply chain, and over-licensing are among the primary factors that continue to be a struggle for them.

The main purpose of analyzing the Zomato dataset is to have a clear idea of factors that influence the initiation of different types of restaurants or cafes which are present throughout Bengaluru, and considering the accumulated rating of each restaurant. Bengaluru, being a fast growing city has more than 12,000 restaurants serving cuisines from around the world, as there is really high demand for food, it's now more important and helpful to conduct a study on demographics of a location. What style of cookery is being enjoyed widely in a certain area? Is there a preference for vegetarian food than other cuisines? If so, take into consideration the population of people that is a particular set of people, such as Jain, Marwari, or Gujarati vegetarians? These kinds of analyses may be done with data by looking at things like • Restaurant location • Approximate food price • Theme-based restaurant or not • Is a specific neighborhood known for a specific type of food?

The information is correct as of March 15, 2021, according to the Zomato website. The information was gathered in two stages from Zomato. After looking through the website's layout, we discovered that there are 6-7 different types of restaurants for each area, including Buffets, Bistros, Delivery, Desserts, and Dining, Nightlife & Drinks, Pubs, and Bars.

3. Methodology

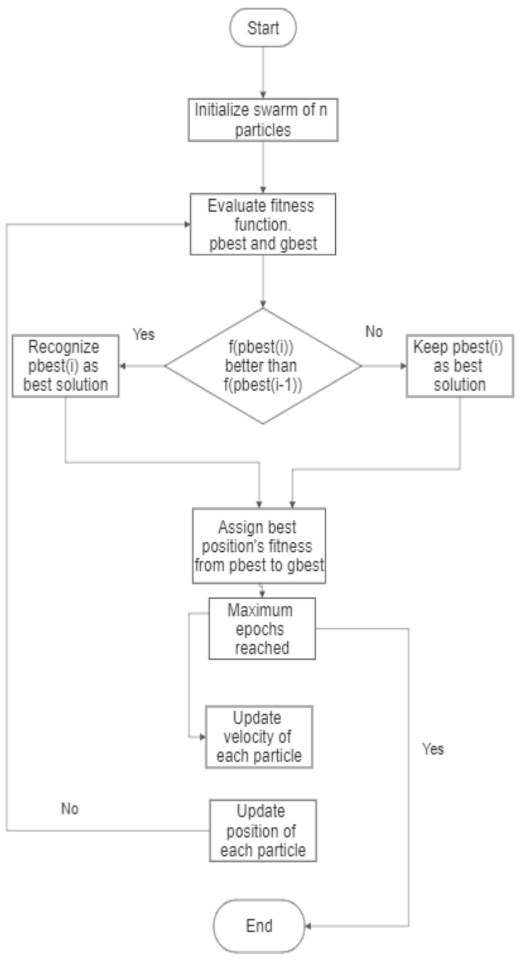
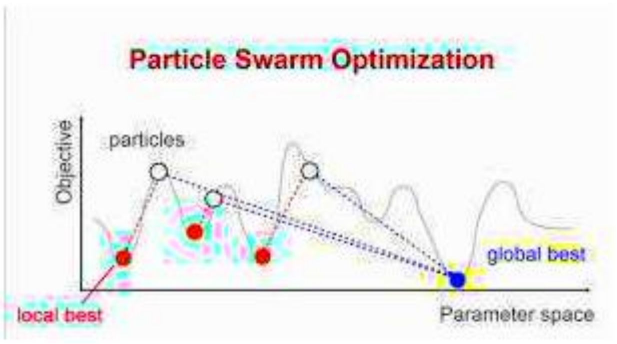

Start
Initialize swarm of n particles
Evaluate fitness function. pbest and gbest
Yes
No
Recognize pbest(i) as best solution
f(pbest(i)) better than f(pbest(i-1))
Keep pbest(i) as best solution
Assign best position's fitness from pbest to gbest
Maximum epochs reached
Update velocity of each particle
Yes
No
Update position of each particle
End

Figure 1: Flowchart Fig.2 - PSO



4. Preprocessing

The dataset was obtained from Kaggle, it included more than 50000 rows that had hotel details and had more than 12 columns with attributes such as hotel name, rating, location of the hotel, address, approx. cost for two, etc. The dataset consists of hotels that are present in Bengaluru only, a thorough exploratory analysis was done on the dataset and unwanted rows and columns were dropped from the dataset, removing duplicate rows, rows that had hotel data repeated were removed or dropped from the dataset, and all the rows with inconsistent data and null or missing values were dropped from the dataset.

Approx. cost for two columns had floating value data which was converted to int data by considering the ceiling value of float data, rating for hotels was stored out of 5 as string data which is not suitable for implementing the algorithm thus the column was converted into float value by ripping of the /5 out of the column and keeping only the float data, The null values and missing values of column rating and approximate cost for two were computed using an appropriate technique such as the mean value of adjacent rows, the final dataset using which the algorithm is implemented consists of 11394 hotels and 10 columns, columns such as hotel name, rating, the approx. cost for two are stored as list using which the MOPSO algorithm is implemented.

Objectives to be optimized

Distance: The distance between the hotel and the user's location determines how long it is going to take for the food to be delivered. In this project, we consider 92 important locations in and around Bengaluru and calculate the distance between these areas and the hotels.

To do so we need the coordinates of the hotels and the 92 prime areas under consideration, We use GEOCODE from awesome table an excel add-on to get the accurate latitudes and longitudes of the hotels and the areas present in the dataset, we store latitude and longitude as 9th and 10th column in the dataset and use this data further while implementing the algorithm and finding optimal solution i.e., best hotels when distance is one of the multiple objectives.

We get the latitudes and longitudes of all the hotels present in the dataset, we obtain coordinates of all the 92 areas under consideration, and we use the formula

$$a = sin(\frac{dlat}{2})^2 + sin\,sin(\frac{dlon}{2})^2 \times cos\,cos(lat1) \times cos\,cos(lat2)$$

$$c = 2(atan2(\sqrt{a}, \sqrt{1-a})$$

r = 6371, where r is the radius of the earth in Kms.

To calculate the distance between all the hotels and the corresponding area in consideration and store the obtained values (in Kms) in an excel sheet with the corresponding location as the name of the excel sheet, ex: Kengeri.xlsx contains the distance between all the hotels and kengeri, 1*11392 matrices are created, 92 such matrices are computed and created, thus we obtain distance data from all the areas (92) to all the hotels present in the dataset after pre-processing.

Cost:
Approximate cost for two column in the datasets gives an idea of how expensive the hotel is, the user can use this data to see if the hotel is in his/her budget and can use this data to filter out the hotels and choose the best one according to his/her needs at the time.

Rating:
The rating column in the dataset indicates how good the hotel's food is and also indicates how good the service of the hotel is, one can use data from this column to make sure they get to eat great food with good customer service, the rating of all the 11392 hotels is been recorded and stored in the excel sheet which can further be used to implement the algorithm when rating is one of the objectives to be optimized.

5. Objectives

We consider cost vs. rating, cost vs. distance and rating vs. distance as three pairs of multiple objectives to be optimized and get a non-dominated optimal solution for the following objectives.

We optimize cost and rating first and find all the hotels with least price and best ratings, we then optimize distance and rating and find out all the hotels within the users budget and fastest delivery time, we then optimize distance and cost to find out all the hotels with best ratings and fastest delivery time i.e., that hotels that are closer to users location.

a) Cost VS Rating

GOAL: To maximize rating while minimizing cost.
For Cost VS Rating, we consider approx. cost for two and rating column from the dataset and implement MOPSO using this two columns, we specify the population size i.e., number of hotels among 11392 hotels to be considered we also mention the repository size which gives as many non-dominated optimal solutions as mentioned, we get the optimal non-dominated solutions we then check for repeated hotel names and discard all the repeated ones, we again provide the user an option to filter out from the optimal solutions to choose a hotel the fits him/her best.

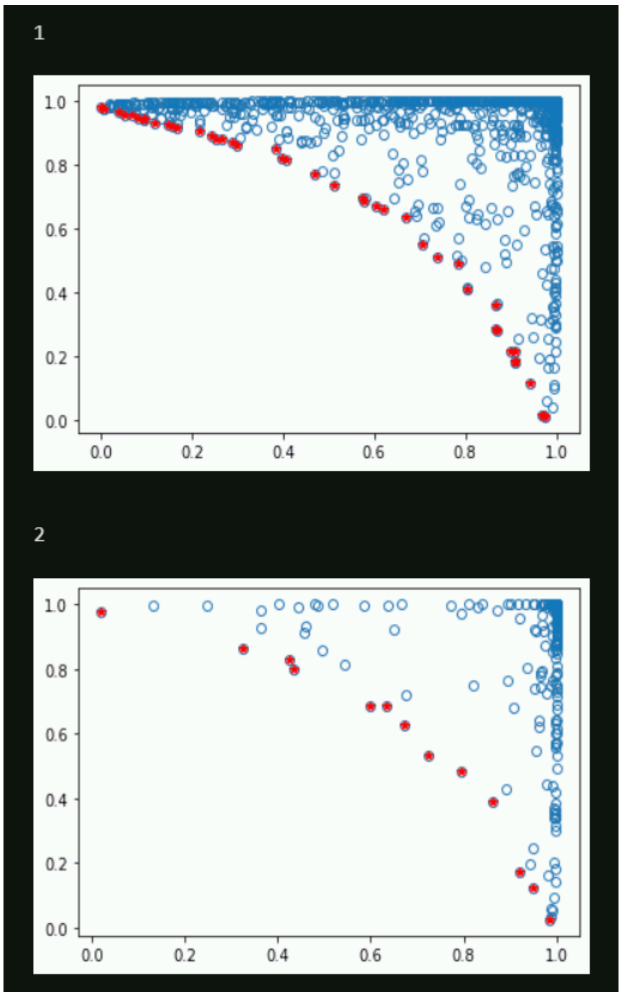


Figure 3: Cost vs Rating

b) Distance VS Cost

GOAL: to minimize distance and minimize cost.
Here the goal is to minimize distance in order to get the food items delivered as soon as possible and minimize cost i.e., to choose a hotel that is budget friendly. To optimize these pair of objectives we use the distance matrix obtained by calculating the distance between the hotel and the 92 areas considered, we calculate distance in Kms using the coordinates, we will have also appended cost and rating to the excel sheet which has distance information, we take distance and cost for two columns into consideration and apply MOPSO on these columns to obtain non-dominated optimal solution, we specify the population and repository size, we then check if there are repeated hotels in the repository and print all the unique values, the user is then given an option to filter out the hotels further based on distance and cost and choose the best fit hotel for his needs at the time.

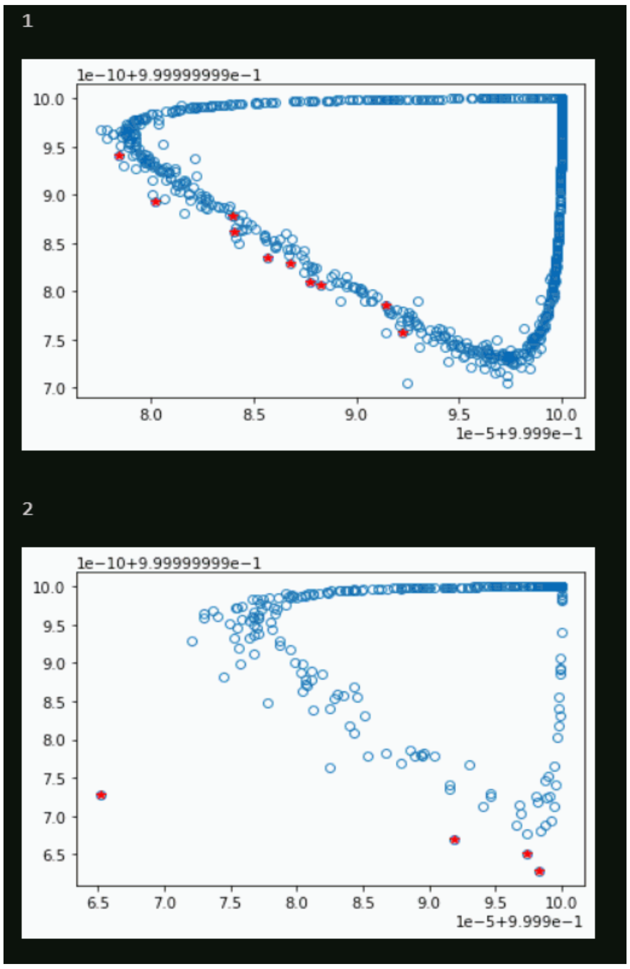
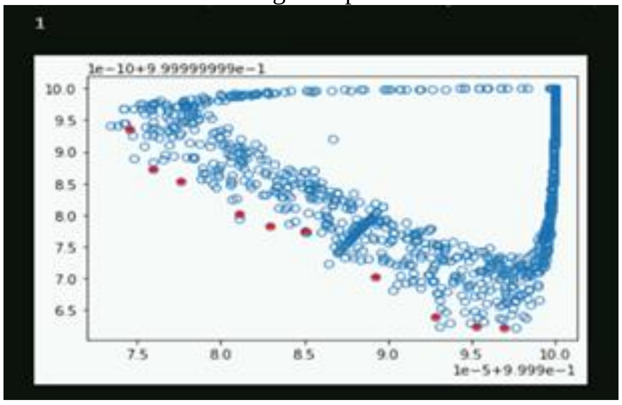


Figure 4: Distance vs cost

c) Distance Vs Rating

Goal: to minimize distance while maximizing rating.
Here the goal is to maximize rating to get best food and service in less time, choosing hotels which are few Kms apart and hotels with maximum/good rating helps achieving this goal, the distance matrix created by computing the distance between hotels and 92 areas is considered it also has the rating information of all the 11392 hotels in the excel which was previously appended while computing distance and creating the excel, thus the distance matrix excel of a location has all the details required i.e., has data regarding both distance and rating, by applying MOPSO on these objectives and specifying population size and repository size, we get non-dominated optimal solutions, the user can further filter the hotels based on distance and rating to choose the best hotel among the optimal hotels.



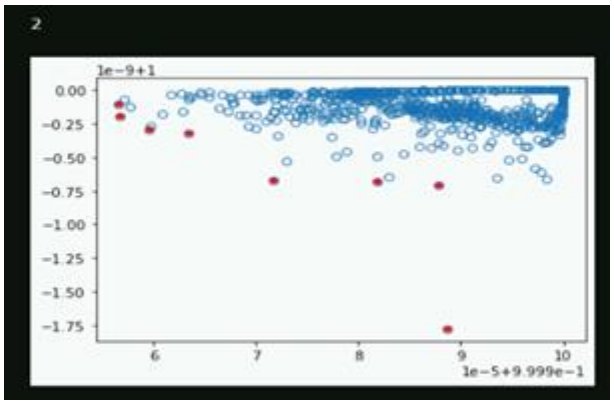


Figure 5: Distance vs rating

6. Mopso Functions

We import NumPy, pandas, matplotlib, random, and math packages which help in implementing MOPSO. We then normalize all the objectives to be optimized as normalization is a technique often applied as part of data preparation for machine learning. The main objective of normalization is to change the dataset's numerical columns and their values to a single scale without losing detail or distorting discrepancies in the values and their respective ranges.

a) Reduce Repository Member function:

This function takes the population size and repository size into consideration, the population size specifies the number of restaurants that are to be taken into account while implementing the algorithm to get the optimal solution while repository size mentions the number of optimal non-dominated solutions expected for the corresponding population size, the function takes the grid indices and position of the particle obtains gbest and pbest which is dominated solution from other functions and keeps deleting one repository member which is not optimal or is a dominated solution.

b) Leader

The leader selection is one of the important processes that take place while implementing MOPSO, at the first iteration a random leader is chosen while in iterations that follow, a leader is chosen based on the gbest and pbest value, the pbest(local best) of small swarms are computed a leader is chosen among those swarms and the global best among all the swarms are computed and the particle with the highest local best is now selected as a global best particle and is chosen as the leader and other particles follow the leader, the agent calculates the risk and reward with each action in each iteration and produces the appropriate value of velocity and particle position which is later used to compute pbest and gbest which is in turn used to select a leader for the swarm of particles.

c) Selection

This function is used to select a particle as a leader, at the first iteration a leader is chosen at random in the iterations that follow the leader is chosen based on pbest and gbest values.

d) Find GridIndices

Particle gridindex and particle gridsubindex are calculated using this function the final particle gridIndex is the sum total of particle.gridIndex and particle gridSubIndex, this is done for all the particles in the population set.

e) CreateGrid:

In this function we take the cost of all the particles in the population set into consideration, min cost and max cost are computed, deltaCost value that is the difference between costMax and CostMin is computed, for all the particles in the population set a grid with upper and lower bounds is created and the grid is returned at the end.

f) Dominates:

This function checks or compares two particles in the population set, it checks if any value of X particle is lesser than the values of Y particle if there exists a value where all or any x<=y it implies x dominates y.

g) Determine Domination:

This function uses the 'dominates' function to check if any particle dominates any other particle, once the domination is known we apply the same function for the whole population and check for all the dominated and non-dominated particles which are later put into the repository which provides all the non-dominated optimal solutions.

h) MOP2

It takes two objectives to be optimized and calculates the fitness value using the formulae

$$z1 = 1 - e^{-\Sigma \quad (\frac{x-1}{\sqrt{n}})^2}$$

$$z2 = 1 - e^{-\Sigma \quad (\frac{x+1}{\sqrt{n}})^2}$$

Fitness functions are used to direct simulations towards the best possible design solutions and are used in generic programming.

We specify population size that is the number of restaurants to be considered for the solution space, we specify repository size which denotes the number of non-dominated optimal solutions in the Pareto front, we specify the number of iterations as maxIt, we also input initial weight w, learning coefficients c1 and c2 where c1 and c2 are personal learning and global learning coefficients we also specify beta and gamma which are leader selection pressure and deletion selection pressure respectively.

i) Initialization

We initialize class particles with attributes such as index, position, isdominated, gridIndex, and gridSubindex,cost, velocity, best position, best cost.

j) GridDim

The function contains lower bounds and upper bounds, for every objective a grid item is a partition of values of objective cost.

The function takes in particles in the population set, in each iteration it calculates the position of the particle in the grid, calculates velocity, cost, best position, and best cost of the particle, it is initially initialized as non-dominated and later in each iteration checks if the particle is been dominated by other particles and updates the isdominated attribute, once all the attributes of the particle are computed in all the iterations, the final values of attributes of the particle is compared and is appended into repository if the particle is a non-dominated optimal solution, length of the repository is also checked at every iteration and only the primary non-dominated optimal solutions are stored and the dominated ones are removed using Reduce Repository Member function, for every iteration a graph is been plotted with particles in its gridindices, all the non-dominated solutions are been represented using red color while the dominated solutions are represented in blue.

7. Final Output

Once the maximum iterations (maxIt) is reached the final optimized solutions that are non-dominated are stored in the repository, the length of the repository is examined to check the number of optimal solutions being obtained, all the optimal solutions are put into a list and list prints all the unique solutions i.e., all the unique hotels that are present in the repository, these unique hotels are the best-fit hotels which are optimal as per user objectives.

Further the project aims at providing the user another filter option to choose the best hotel from the optimized non dominated solution set based on the user's objectives and the corresponding value for the objectives.

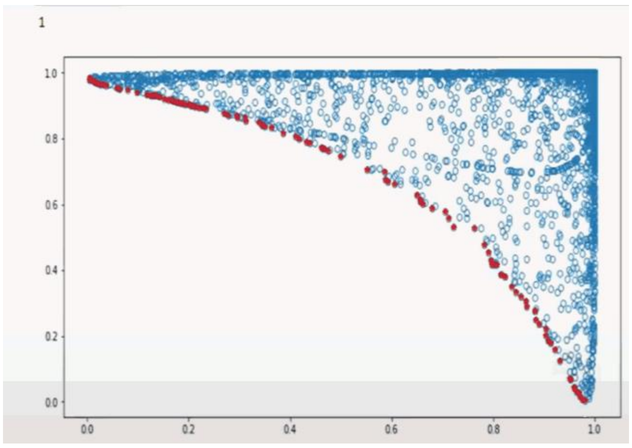


Figure 6: (Iteration 1)

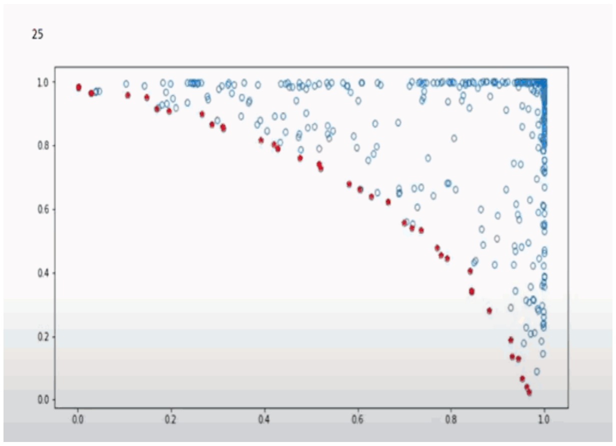


Figure 7: (Iteration 25)

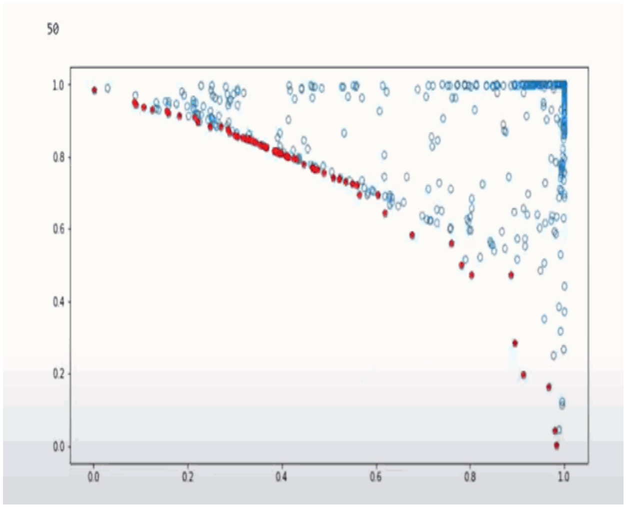


Figure 8: (Iteration 50)

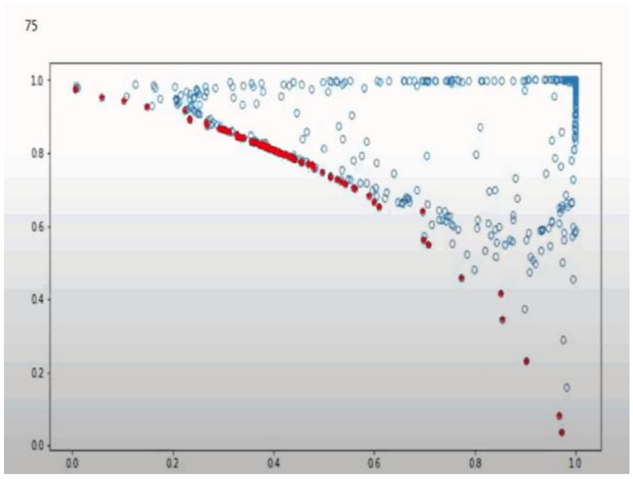


Figure 9: (Iteration 75)

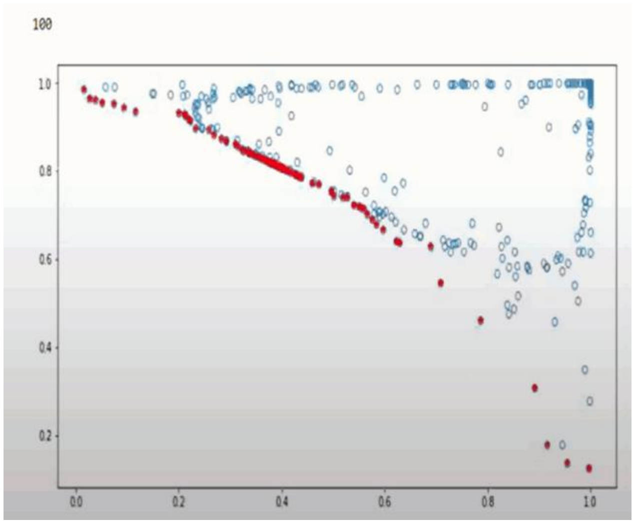
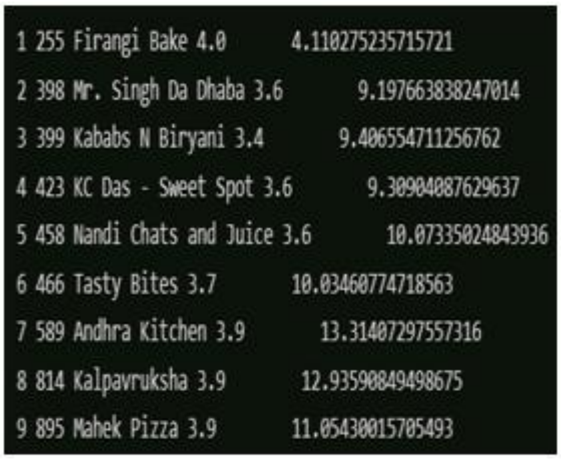


Figure 10: (Iteration 100)

The implemented MOPSO optimizes multiple objectives which are conflicting in nature and provides the optimal non-dominated solution for the given set of data in the population set.

We optimize distance and cost, distance and rating, and cost and rating and obtain the list of best hotels based on the user's objectives. We also provide a further option to the user to filter from the list of optimized hotels based on his/her requirements at the time and help the user choose the best hotel.



```
1 255 Firangi Bake 4.0        4.110275235715721
2 398 Mr. Singh Da Dhaba 3.6        9.197663838247014
3 399 Kababs N Biryani 3.4        9.406554711256762
4 423 KC Das - Sweet Spot 3.6        9.30904087629637
5 458 Nandi Chats and Juice 3.6        10.07335024843936
6 466 Tasty Bites 3.7        10.03460774718563
7 589 Andhra Kitchen 3.9        13.31407297557316
8 814 Kalpavruksha 3.9        12.93590849498675
9 895 Mahek Pizza 3.9        11.05430015705493
```

Figure 11: Final Output (list of hotels optimized with respect to rating and distance)

References

[1] Trivedi, V., Varshney, P. & Ramteke, M. A simplified multi-objective particle swarm optimization algorithm. *Swarm Intell* **14**, 83–116 (2020). https://doi.org/10.1007/s11721-019-00170-1

[2] Li, L., Wang, W., & Xu, X. (2017). Multi-objective particle swarm optimization based on global margin ranking. *Information Sciences*, *375*, 30–47. https://doi.org/10.1016/j.ins.2016.08.043

[3] Liang, J., Guo, Q., Yue, C., Qu, B., & Yu, K. (2018). A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems. In *Lecture Notes in Computer Science* (pp. 550–560). Springer International Publishing.

[4] Karakoyun, M., Ozkis, A., &Kodaz, H. (2020). A new algorithm based on gray wolf optimizer and shuffled frog leaping algorithm to solve the multi-objective optimization problems. *Applied Soft Computing*, *96*(106560), 106560. https://doi.org/10.1016/j.asoc.2020.106560

[5] Liu, Y., Ishibuchi, H., Nojima, Y., Masuyama, N., & Shang, K. (2018). A double-niched evolutionary algorithm and its behavior on polygon-based problems. In *Parallel Problem Solving from Nature – PPSN XV* (pp. 262–273). Springer International Publishing.

[6] Hu, Y., Wang, J., Liang, J., Yu, K., Song, H., Guo, Q., Yue, C., & Wang, Y. (2019). A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm. *Science China Information Sciences*, *62*(7). https://doi.org/10.1007/s11432-018-9754-6

[7] Researchgate.net. Retrieved April 20, 2023, from https://www.researchgate.net/publication/331824516_Multi-objective_Solution_of_Traveling_Salesman_Problem_with_Time

[8] Deb, K. (n.d.). *Lecture 9: Multi-Objective Optimization*. Purdue.edu. Retrieved April 20, 2023, from

https://engineering.purdue.edu/~sudhoff/ee630/Lecture09.pdf

[9] Researchgate.net. Retrieved April 20, 2023, from https://www.researchgate.net/publication/350486652_A_Review_of_Multi_Objective_Optimization

[10] Amidpour, M., &KhoshgoftarManesh, M. H. (2021). Optimization of cogeneration and polygeneration systems. In *Cogeneration and Polygeneration Systems* (pp. 287–323). Elsevier.