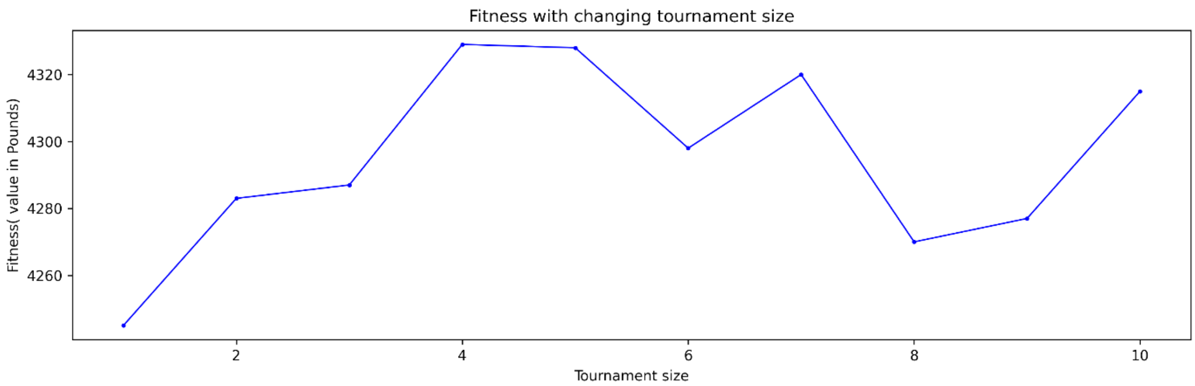
Solving bank problem using evolutionary algorithm

Experiment1: Tuning the Tournament size(t) parameter

Implemented Evolutionary algorithm (EA) was run for different tournament sizes ranging from 1 to 10. Other parameters like mutation rate (1) and population size (10) were kept fixed. Results are tabulated as follows:

Tournament size (t)	Weight (kgs)	Fitness (Pounds)
1	273	4245
2	274.2	4283
3	274.4	4287
4	275.9	4329
5	275.9	4328
6	276.5	4298
7	275.6	4320
8	275.6	4270
9	274.4	4277
10	276.6	4315

Following is the graph depicting the results. X-axis has tournament size and y-axis the fitness values (in Pounds).



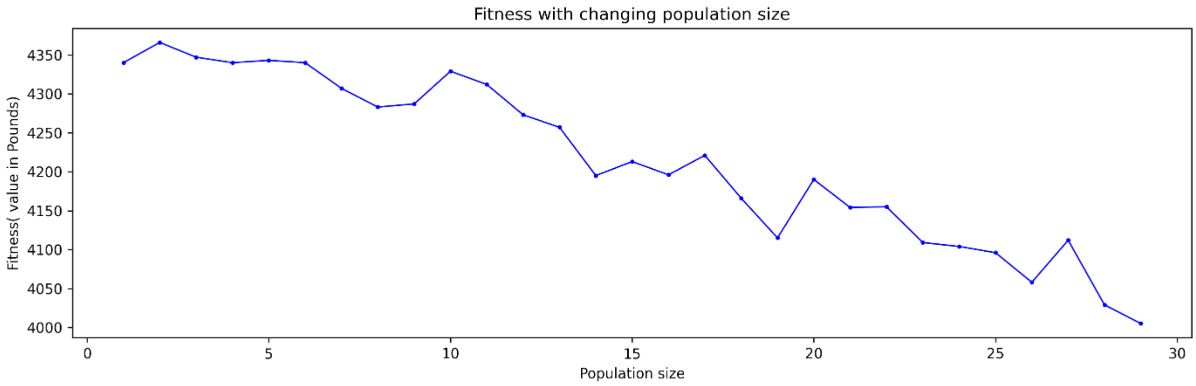
Conclusion: In the range of t=1 to t=4, higher fitness is achieved when we increase the tournament size. So, peak fitness of 4329 is achieved at t=4. However, after t=4, fitness value is not consistent with this trend and there are local minima and maxima in the fitness landscape. Increasing the tournament size is causing a greater number of evaluations at tournament stage, which is leading to an early termination of the EA. This can be the reason for less fitness score at higher t values. So, selection pressure of t=4 seems to be optimum for a population of size 10.

Experiment2: Tuning the Population size (p) parameter

Implemented Evolutionary algorithm (EA) was run for different population sizes ranging from 1 to 29. Other parameters like mutation rate (1) and tournament size (4) were kept fixed. Results for few lower p values are tabulated as follows:

P value	weight	fitness
1	276.3	4340
2	275.8	4366
3	273.7	4347
4	276.5	4340
5	276.3	4343

Following is the graph depicting the results. X-axis has population size and y-axis the fitness values (in Pounds).



Conclusion: With increasing p value, fitness increases only up to p=2 and after that it keeps on decreasing significantly. Thus, maximum fitness of 4366 is achieved at p=2. So, having a population of 2 solutions seems to be sufficient to arrive at an optimal solution.

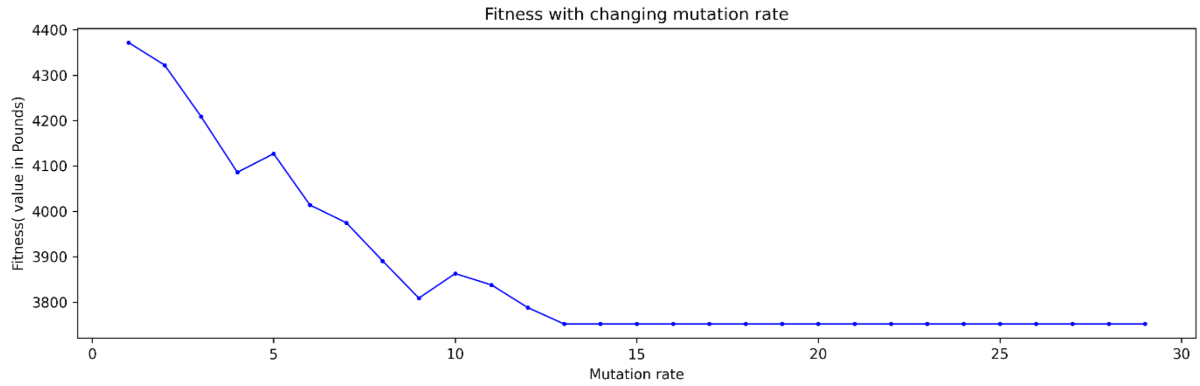
Experiment3: Tuning the mutation rate (m) parameter

Implemented Evolutionary algorithm (EA) was run for different mutation rates ranging from 1 to 29. Other parameters like population size (3) and tournament size (2) were kept fixed.

Results are tabulated as follows:

Mutation (m)	weight	fitness
1	276.8	4372
2	275.3	4322
3	272.3	4209
4	274.5	4086
5	277	4127

Following is the graph depicting the results. X-axis has mutation rate and y-axis the fitness values (in Pounds).



Conclusion: With increasing m value, fitness keeps on decreasing. Maximum fitness of 4372 is achieved at the lowest mutation rate of m=1. For mutation rate higher than 12, there is no change in the fitness score at all.

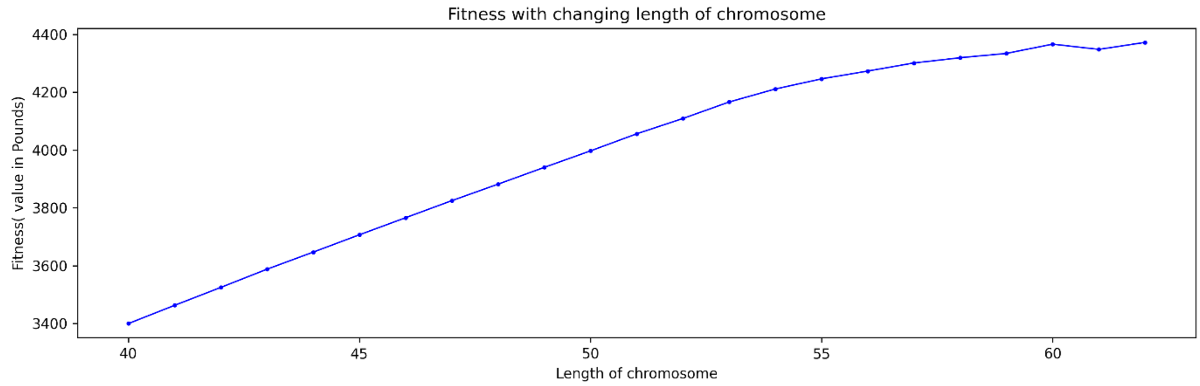
Experiment4: Changing the length of chromosome

Implemented Evolutionary algorithm (EA) was run for different mutation rates ranging from 1 to 29. Other parameters like population size (3) and tournament size (2) were kept fixed.

Following table depicts the fitness score at higher length values.

length	weight	fitness
56	274.8	4273
57	274	4301
58	276.2	4319
59	276.7	4334
60	276.8	4366
61	276.1	4348
62	276.8	4372

Following is the graph depicting the results. X-axis has length of chromosome and y-axis has the fitness values (in Pounds).



Observation: Length of chromosome or solution appears to be an important criterion for arriving at an optimal solution. Fitness score increases continuously with the length of chromosome. Length of 62 gives the highest fitness score of 4372. Post this point, the total weight starts to exceed the maximum van capacity and thus, computation time of EA also increases drastically. This is because with more than 62 bags in a solution, it becomes difficult to randomly generate an initial population of valid solutions (i.e. with total weight less than 277 Kgs).

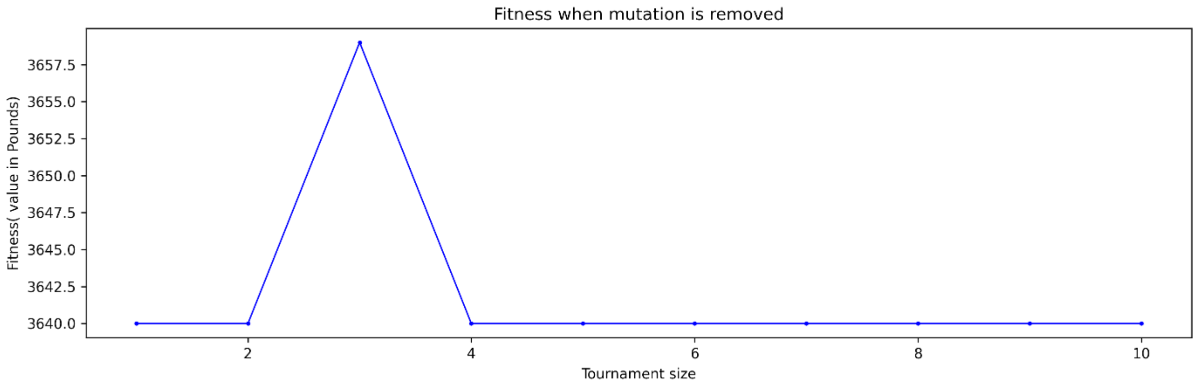
- <u>Which combination of parameters produces the best results?</u>

Population size of 3, tournament value of 2 and mutation rate of 1 gives the highest fitness value of 4376. This is achieved with 63 bags in a solution.

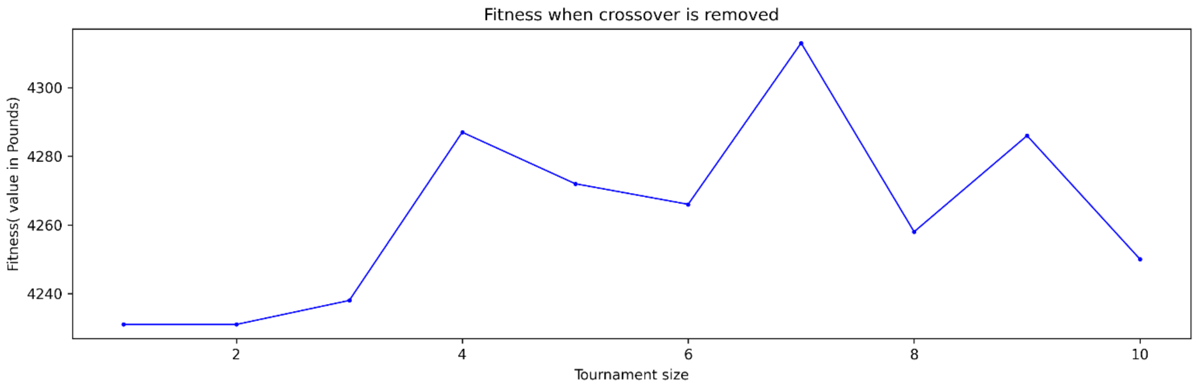
Having a population of 2 solutions seems to be sufficient to arrive at an optimal solution. Also, tournament size of 2 is providing sufficient selection pressure. Mutation rate of 1 is enough to develop good solutions. Higher tournament size can make the EA converge early.

- <u>The effect of removing crossover or mutation</u>

When mutation is removed, there is not much change in the initial population. Hence, there is no progress at all. Weakest replacement operator is not largely able to remove any poor fitness solutions from the initial population because there are no good solutions emerging post crossover to replace with. The crossover is mostly leading to invalid solutions that have the worst fitness. This is shown in graph below.



However, when only crossover is removed, mutation produces good solutions. This mutation combined with weakest replacement leads to overall a good population of solutions. This is shown in graph below.



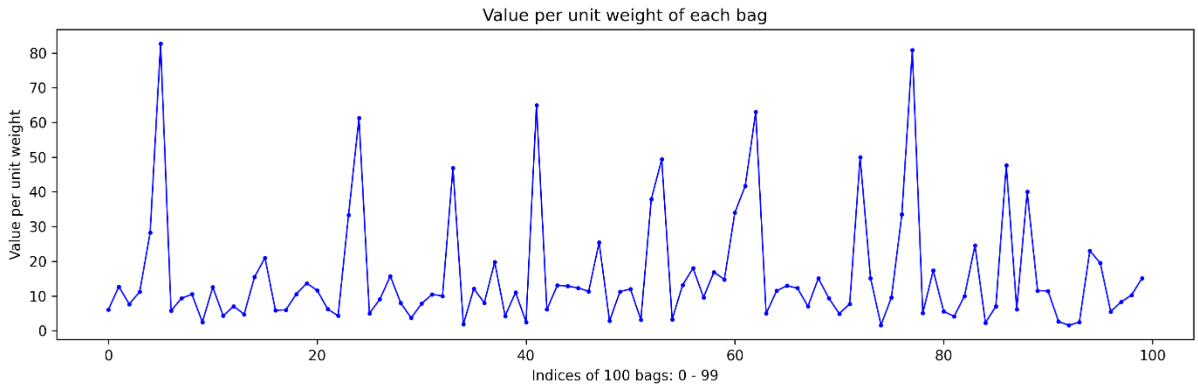
- If you were to extend your EA to work with a multi-objective version of this problem, which functions in your program would you change and why?

To implement a multi-object version of this problem, crossover, mutation and weakest replacement functions can remain the same. However, the current evolutionary algorithm (EA) has a fitness function that calculates single fitness, which is based only on the total money value. Hence, to implement a multi-object EA, we need to change the fitness function to have 2 fitness, one for the objective of maximizing pound value and other for the objective of minimizing weight value.

There is also a need to change the selection operator. Current EA is based on the tournament selection, which would not be of much help in Multi-objective EA because of the insufficient selection pressure. In a Multi-objective problem, the best solutions have a characteristic of non-dominance. These non-dominant best solutions are identified by the Pareto front. Rank-based approach for identifying the best (non-dominated) solutions supplemented with Pareto Domination tournament can be used for selection in MOEA. Pareto Domination tournament facilitates the application of enough selection pressure, which is controlled by the size of randomly selected set for comparison.

Remarks/Analysis of the bank problem

Since there is a variation in value per unit weight of each bag (as shown in the graph below).



Selecting only those bags which have higher values per unit weight (ensuing that total weight does not exceed 277 kgs) can give us a pretty good solution. On this basis, if we select top such 64 bags, we get total value of 4386 Pounds with total weight of 275.6 Kgs. It seems there is still some room for 1 or 2 bags.