ECMM409 Nature-Inspired Computation

Assignment One: Bin-Packing with an Evolutionary Algorithm

For the given problem, I have implemented an evolutionary algorithm which included a total of six experiments which takes different parameters as input and apply it to a bin-packing problem, from which I have obtained outputs with different chromosomes or solutions having lower fitness value.

I have used k-ary representation which has chromosome represented as a list of n numbers whose length is equal to number of items given and with range from 1 to b (no of bins).

The following are the different inputs given to obtain solutions with minimum fitness values. They are:

* Population: The Number of initial chromosomes to be created
* Mutation: Number of mutations to be made on the selected chromosome. If mutation is given as zero then no changes will be made to the selected chromosome.
* Number of bins
* List of weight of items
* Crossover: A Boolean value is passed as parameter, if given true a crossover is made between two randomly selected chromosomes which results in new child chromosomes.
* Number of iterations to be done on population given to obtain a solution of lower fitness

In this problem, the following steps are followed:

* create generate random solutions of given population and calculate fitness of the chromosome in population.
* Created a binary tournament method which selects two random chromosomes from population, passes the random solutions as parameters to function which does crossover of chromosome if the crossover is given as true/required.
* A crossover of two random solutions creates new child solutions/chromosomes.
* If a mutation is required(mutation>0) then these two child solutions under mutation, and fitness is calculated for solutions after mutation.
* If fitness of child solutions is lower than or equal to the weakest solution of population then the weakest is replaced by the new child solution.
* This process is iterated for the number of iterations mentioned in the input parameter, for this algorithm it is given as 10,000.
* Hence, a solution is achieved with lower fitness

I have Used Jupyter notebook with Python Language to structure my algorithm. The results obtained after conducting different experiments are tabulated below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Exp 1 | Exp 2 | Exp 3 | Exp 4 | Exp 5 | Exp 6 |
|  | Pop:M:Crossover | 10|1|True | 100|1|True | 10|5|True | 100|5|True | 10|5|False | 10|0|true |
|  | Average fitness at start | 2921.0 | 3055.775 | 2631.8 | 2974.845 | 2910.0 | 3118.95 |
| BBP1 | Best Fitness | 17.0 | 12.5 | 163.5 | 99.5 | 803.5 | 1442.0 |
|  | Mean | 29.7 | 29 | 336.0 | 126.9 | 930.1 | 1841.5 |
|  | Average fitness at start | 556447.0 | 594169.375 | 563894.45 | 584076.66 | 548156.55 | 580530.95 |
| BBP2 | Best Fitness | 159693.0 | 173710.5 | 251533.5 | 225525.5 | 389095.0 | 420711.5 |
|  | Mean | 168379.5 | 183895.9 | 283031.1 | 240666.8 | 402901.5 | 462638.6 |

The graphs below are plotted to show the fitness for different trials/experiments conducted. We can observe best fitness is achieved when mutation is low and worst is seen when crossover and mutation are removed.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**Questions:**

**1.Which combination of parameters produces the best results for BPP1 and BPP2?**

BBP1: The best results are produced when population=100, mutation=1, Cross over=True.

BBP2: the best results are produced when population=10, mutation=1, Cross over =True.

**2.Why do you think this is the case?**

For BBP1 when population is increased, with less number of bins and a greater number of chromosomes from which selection can be made, after a mutation and crossover, the weakest chromosome can be potentially replaced by selected chromosome most of the time. As we have a smaller number of bins, fitness improves quickly with every iteration. For 10000 iterations and larger population(100)we can achieve a best solution when compared to other combinations.

For BBP2 when population is less and with larger number of bins, I believe more number of iterations can be made against less number of chromosomes which helps to remove the weakest solution. If the population is increased and larger number of bins are used, the number of iterations (10000) may not be sufficient to optimize and get the best solution. Hence, we can achieve a best solution with larger bins and smaller population when compared to other combinations.

Also, when there is no mutation and crossover, we observed the worst results because it resulted in child chromosome being similar to the parent chromosomes which effected in improvement of algorithm. When mutation value is increased, we have observed lesser number of mutations has produced better solutions.

**3.What was the effect when you removed mutation? What about crossover?**

When mutation is removed, worst fitness is generated across the algorithm. As removing mutation resulted in child chromosome being similar to the parent chromosomes after reaching local minimum which effected new chromosome unable to reach global optimum, thus effected on the scope of improvement of algorithm.

Even when crossover is removed, I observed worst fitness is generated across the algorithm. Crossover helps in producing the child chromosome which has better fitness than parent chromosome. As crossover is not done most of the child chromosome has fitness worse than the parent chromosome which effect the scope of improvement of the algorithm.

Hence, it concludes that crossover and mutation play a vital role in improvement of fitness of chromosome.

**4.Which other nature-inspired algorithms might be effective in optimising the BPP problems? Explain your choice(s).**

Ant Colony Optimization also known as ACO can also be effective in optimising BPP problems. In ACO we use graphs for solving the BPP problems. It is a method inspired from ant’s behaviour. Using this optimization, a graph is used where we bind the items to the bins. The overall result is to find the shortest path from the initial problem to the solution. Though we can use ACO for bin packing it is useful when we have large number of bins and items.

**NOTE:**

**Further Experiments Conducted:**

I have conducted a further experiment by doing a **two-point** crossover for the parent chromosomes selected from the population. After running the algorithm across all the experiments, I have observed that every combination of parameters produced the best results for BPP1 and BPP2 when compared to the **single point** crossover.

For BBP1 the best results are produced when population=100, mutation=1, Cross over=True.

For BBP2 the best results are produced when population=10, mutation=1, Cross over =True.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Exp 1 | Exp 2 | Exp 3 | Exp 4 | Exp 5 | Exp 6 |
|  | Pop:M:Crossover | 10|1|True | 100|1|True | 10|5|True | 100|5|True | 10|5|False | 10|0|true |
|  | Average fitness at start | 2781.3 | 3108.42 | 2727.95 | 3116.105 | 3137.4 | 2941.3 |
| BBP1 | Best Fitness | 18.0 | 14.5 | 142.5 | 112.5 | 819.0 | 1129.0 |
|  | Mean | 26.3 | 24.3 | 203.2 | 138.4 | 992.4 | 1483.1 |
|  | Average fitness at start | 559939.0 | 569730.16 | 566649.25 | 584076.66 | 548156.55 | 575733.35 |
| BBP2 | Best Fitness | 152860.0 | 155792.5 | 206923.5 | 225525.5 | 378690.0 | 435896.5 |
|  | Mean | 171613.2 | 169706.3 | 222264.7 | 237981.0 | 409629.0 | 461047.8 |

The graphs below are plotted to show the fitness for different trials/experiments conducted with two point crossover. We can observe best fitness is achieved when mutation is low and worst is seen when crossover and mutation are removed.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**Conclusion:**

* After conducting experiments with different parameters, I conclude that worst solutions are generated when the mutation and crossover are removed
* When given larger number of items, population and bins it requires more number of iterations to get a best solution
* If mutation is removed it resulted in child chromosome similar to the parent chromosomes after reaching local minimum which effected new chromosome unable to reach global optimum, thus effected on the scope of improvement of algorithm
* Crossover of parent chromosome is required to get a better child chromosome
* I observed that two point crossover resulted in achieving a better solution when compared to single point cross over.