Heuristic Methods

Project name: KSP-[GA-TS]

Goal of the project:

Solve the Knapsack Problem (KSP) using genetic algorithms and tabu search.

Statement of the problem:

A knapsack of fixed capacity is filled with objects. Let G be the maximum weight that the knapsack can carry and let N be the number of available objects. Let g_i be the weight of the i-th object and let v_i be the value of the i-th object. Find the best way to load objects in the knapsack to maximize the value of the knapsack. The more value is loaded in the knapsack, the better the solution. Two solutions involving the same loaded value will differ in the weight of the knapsack. The solution with a smaller weight will be considered better. If both value and weight are the same for two different solutions, they are considered equal.

Requirements

1. Define a dataset for the KSP as a 4-tuple of the form DS = {G, N, g[], v[]}, with g[] and v[] being arrays of numbers of size N.
2. Randomly build 4 datasets: DS_8 with N_8 = 8 objects, DS_{10} with N_{10} = 10 objects, DS_{50} with N_{50} = 50 objects and DS_{100} with N_{100} = 100 objects. You should take care when building the datasets, because a small value for G and large values in g[] will have a bad influence on the solution space. We want the solution space very large, so the "random" generation process should not be exactly random, but rather controlled. For example, values in g[] should be chosen in relation to G, so that the size of the solution space remains considerably large.
3. Implement a backtracking solution for the KSP and run it on DS_8 and DS_{10} to obtain the respective optimum solutions S_{B-8} and S_{B-10} (Backtracking should be useless on DS_{50} and DS_{100}).
4. Use a timer to measure the time T_{B-8} required to obtain S_{B-8} and the time T_{B-10} required to obtain S_{B-10}. Compute T_{B-10}/T_{B-8} as opposed to 10/8 = 1.25 and find an explanation. Draw the necessary conclusion about the feasibility of backtracking on larger datasets.
5. Implement neighborhood search for the KSP and run it on all 4 datasets to obtain solutions S_{NS-8}, S_{NS-10}, S_{NS-50} and S_{NS-100} respectively. Compare S_{NS-8} with S_{B-8} and S_{NS-10} with S_{B-10}.
6. Use a timer to measure the time T_{NS-8} required to obtain S_{NS-8} and the time T_{NS-10} required to obtain S_{NS-10}. Compare T_{NS-10}/T_{NS-8} or alternatively, T_{B-8}/T_{NS-8} and T_{B-10}/T_{NS-10}. Draw the necessary conclusion about the feasibility of neighborhood search on larger datasets. Also measure the time T_{NS-50} required to obtain S_{NS-50} and the time T_{NS-100} required to obtain S_{NS-100}.
7. Use the results obtained during solving requirements #5 and #6 to draw a conclusion about advantages/disadvantages of using neighborhood search instead of backtracking.
8. Implement a genetic algorithm for the KSP and run it on all 4 datasets. For each dataset, use 3 different sets of values for the size of the population, number of iterations, crossover rate, mutation rate and cloning rate. You should obtain 4 solution vectors ($S[]_{GA-8}$, $S[]_{GA-10}$, $S[]_{GA-50}$ and $S[]_{GA-100}$), each of them containing 3 solutions. For example, if population size = 1000, number of iterations = 50, crossover rate = 80%, mutation rate = 15% and cloning rate = 5% you should obtain $S[0]_{GA-8}$, $S[0]_{GA-10}$, $S[0]_{GA-50}$ and $S[0]_{GA-100}$ respectively. New parameter values (population size = 50, number of iterations = 1000, crossover rate = 60%, mutation rate = 30%, cloning rate = 10%) would lead to $S[1]_{GA-8}$, $S[1]_{GA-10}$, $S[1]_{GA-50}$ and $S[1]_{GA-100}$ respectively. From now on, by $S[]_{GA-i}$ we will denote all 3 solutions obtained on dataset DS_i using the 3 sets of values that you chose.

9. Implement tabu search for the KSP and run it ten times on each of the 4 datasets. Use a different starting point for each of the ten runs on a dataset. Declare the best solution found during all ten runs to be the official solution found by the algorithm. Use 3 different tabu tenures for each dataset (note that the tabu tenure should be dependent on the number of objects, so you should use 3 smaller tenures on DS_{10} and 3 larger tenures on DS_{100}). You should obtain 4 solution vectors ($S[]_{TS-8}$, $S[]_{TS-10}$, $S[]_{TS-50}$ and $S[]_{TS-100}$), each of them containing 3 solutions. From now on, by $S[]_{TS-i}$ we will denote all 3 solutions obtained on dataset DS_i using the 3 tenure values that you chose.
10. Compare $S[]_{GA-i}$ with $S[]_{TS-i}$ and both of them with S_{NS-i} and S_{B-i} (for i = 8, 10, 50 and 100).
11. Use a timer to measure the times $T[]_{GA-i}$ (3 times for each i = 8, 10, 50 and 100, a total of 12 times) and the times $T[]_{TS-i}$ (3 times for each i = 8, 10, 50 and 100, a total of 12 times). Compare $T[]_{GA-10}/T[]_{GA-8}$, $T[]_{TS-10}/T[]_{TS-8}$, $T_{B-8}/T[]_{GA-8}$, $T_{B-10}/T[]_{GA-10}$, $T_{B-8}/T[]_{TS-8}$ and $T_{B-10}/T[]_{TS-10}$. Draw the necessary conclusions about the feasibility of genetic algorithms and tabu search on larger datasets.
12. Also compare $T[]_{GA-50}$ with $T[]_{NS-50}$, $T[]_{GA-100}$ with $T[]_{NS-100}$, $T[]_{TS-50}$ with $T[]_{NS-50}$, $T[]_{TS-100}$ with $T[]_{NS-100}$, $T[]_{GA-50}$ with $T[]_{TS-50}$ and $T[]_{GA-100}$ with $T[]_{TS-100}$. Correlate each of these times with the optimality of the solution found in that time. Draw the necessary conclusions.
13. For each i = 8, 10, 50 and 100, compare the three elements of $S[]_{GA-i}$. The difference between them must reside in the different sets of control parameters that you've chosen for each of the three runs (population size, number of iterations, crossover rate, mutation rate, cloning rate). Use relevant sets of values to measure the impact of a large population size combined with a small number of iterations, or a small population size combined with a large number of iterations on the quality of the solution. Another direction that you should study refers to the optimal ratio of crossover, mutation and cloning. Should crossover rates be very high compared to mutation rates, or should they be more evenly distributed? Draw the necessary conclusions.
14. For each i = 8, 10, 50 and 100, compare the three elements of $S[]_{TS-i}$. The difference between them must reside in the different tabu tenures that you've chosen for each of the three runs. Draw a conclusion about the influence of the tabu tenure on the final result.
15. Use the results obtained during solving requirements #10, #11, #12, #13 and #14 to draw a conclusion about advantages/disadvantages of using genetic algorithms or tabu search instead of neighborhood search or backtracking.
16. During each run (regardless of the algorithm that runs), store all the intermediate solutions found, together with the best of them so that they can be used later to graphically represent the run. For the genetic algorithm choose a representative from each population, like for example, the best individual in the population. Implement a module that does this graphic representation (for each solution, store and represent only its cost as an integer or a float).
17. Write an article (6-10 pages) entitled "A heuristic approach to solving the knapsack problem. Genetic algorithms and tabu search" where you should put all that you've learned during the development of this project: a description of neighborhood search, a description of genetic algorithms, a description of tabu search, relevant graphics of the different runs that you performed and the conclusions resulting from requirements #4, #5, #6, #7, #10, #11, #12, #13, #14 and #15.
18. You should write the article in a scientific way. One reading the article should find the facts presented in a logical order, starting with an initial statement that presents heuristics as a viable alternative to backtracking on problems with a large solution space, followed by the choice of KSP as the problem to solve and the comparison between backtracking, neighborhood search, genetic algorithms and tabu search in what concerns running times and optimality of solutions found. Major conclusions should come out of mathematic facts or graphics, not intuition or simple common sense.