1)



1a) See code

1b) We have the following suggestions:

1. Make the r value reduce during the evaluation. That is, we can make the r value a function of the current iteration. Effectively, we are reducing the amount of perturbation in later iterations. This is similar to the idea of temperature in simulated annealing, where the later iterations tends to be more “greedy”, fine-tuning the solution in a smaller window.
2. Cut the tails of the normal distribution. In the current implementation, the reflection method simply mirrors the values at the end points, and adds them back to the distribution. This might cause undesired distributions, as if our value is close to the edge, the peak of distribution might not be at our current value. We would instead recommend cutting the tails of the normal distribution. That is, if the value after adding perturbation exceeds the boundaries, we simply redo the perturbation until the value satisfies requirements. It can be shown that this method requires no more than 2 tries in expectation, and thus should be reasonably efficient.

1c) From just looking at the graph, it appears that DDS is better than GA, which is then better than SA.

1d) Perform at least 3 algorithm comparison tests (e.g. boxplot, empirical cdf, hypothesis test, etc.) for 10,000 cost function evaluations for the best solutions at the end of each of the 20 trials for all algorithms SA, DDS and GA. Which algorithm performed the best based on the tests you performed? Is this consistent with what you concluded in the part (c)? Why or why not?