Greg Daniels

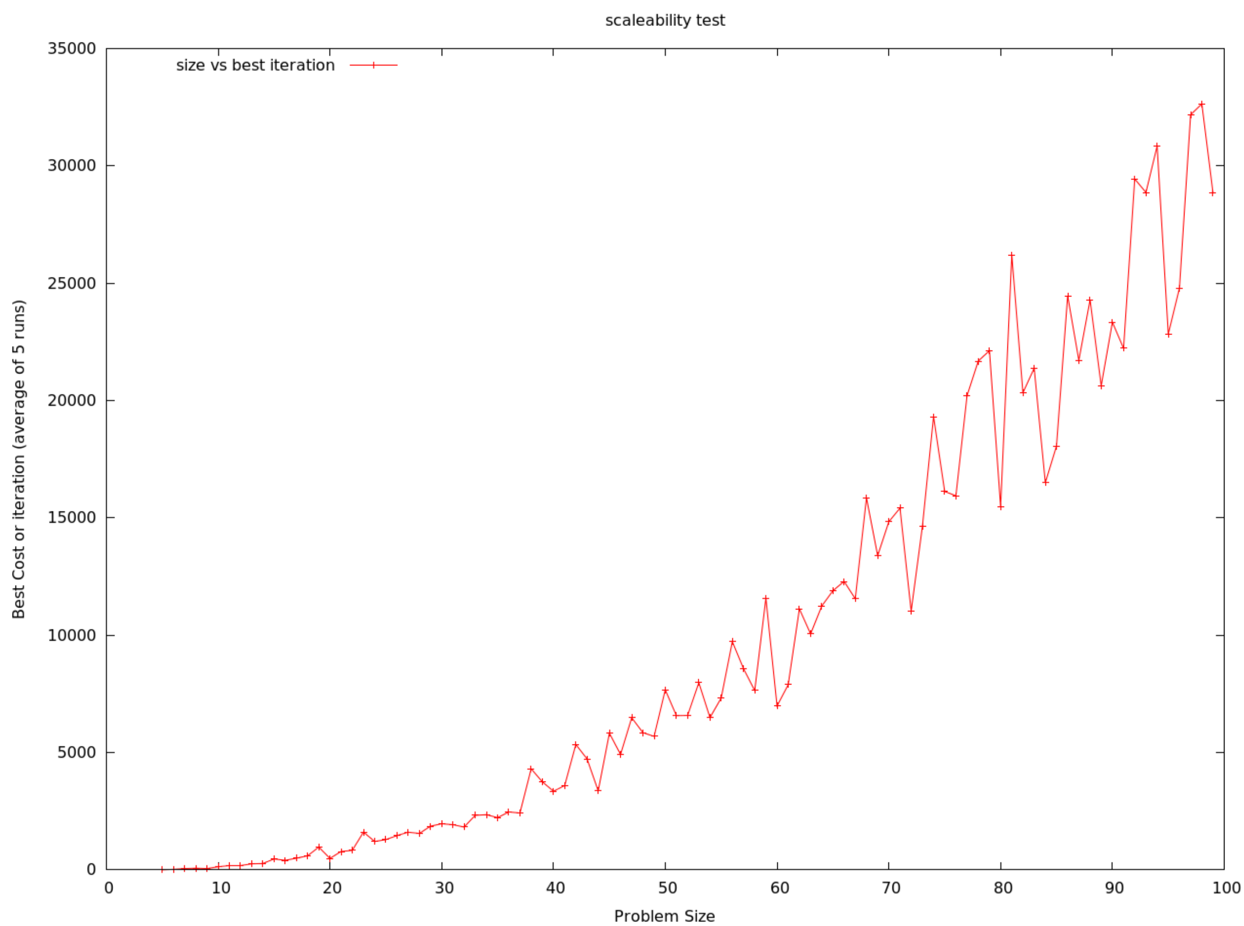
A00798340

Dr Flann CS5600

Simulated Annealing and TSP

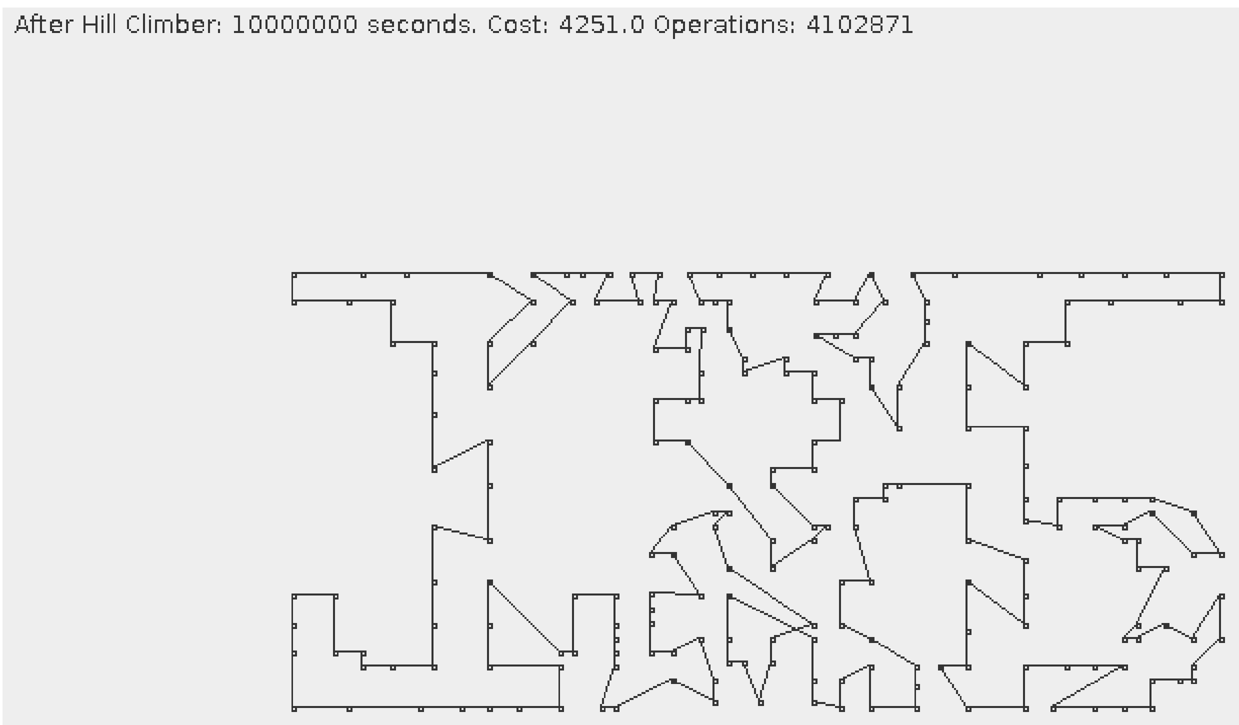
1) How does the performance of these algorithms scale as the number of cities?

Performance scales almost linearly. The only real way to judge performance is the number of iterations before settling on a solution. The more nodes there are the longer it will take because there are more possibilities for it to reverse and a greater likelihood of an increase. To test this the SA algorithm was run on problems of size 5 to 100. Each problem was run 5 times and the average was recorded. The performance has little to do with the final cost since no two graphs are remotely comparable in terms of cost.



2) Given a TSP problem and a fixed amount of time on a single CPU, what would be the most effective way to apply these algorithms?

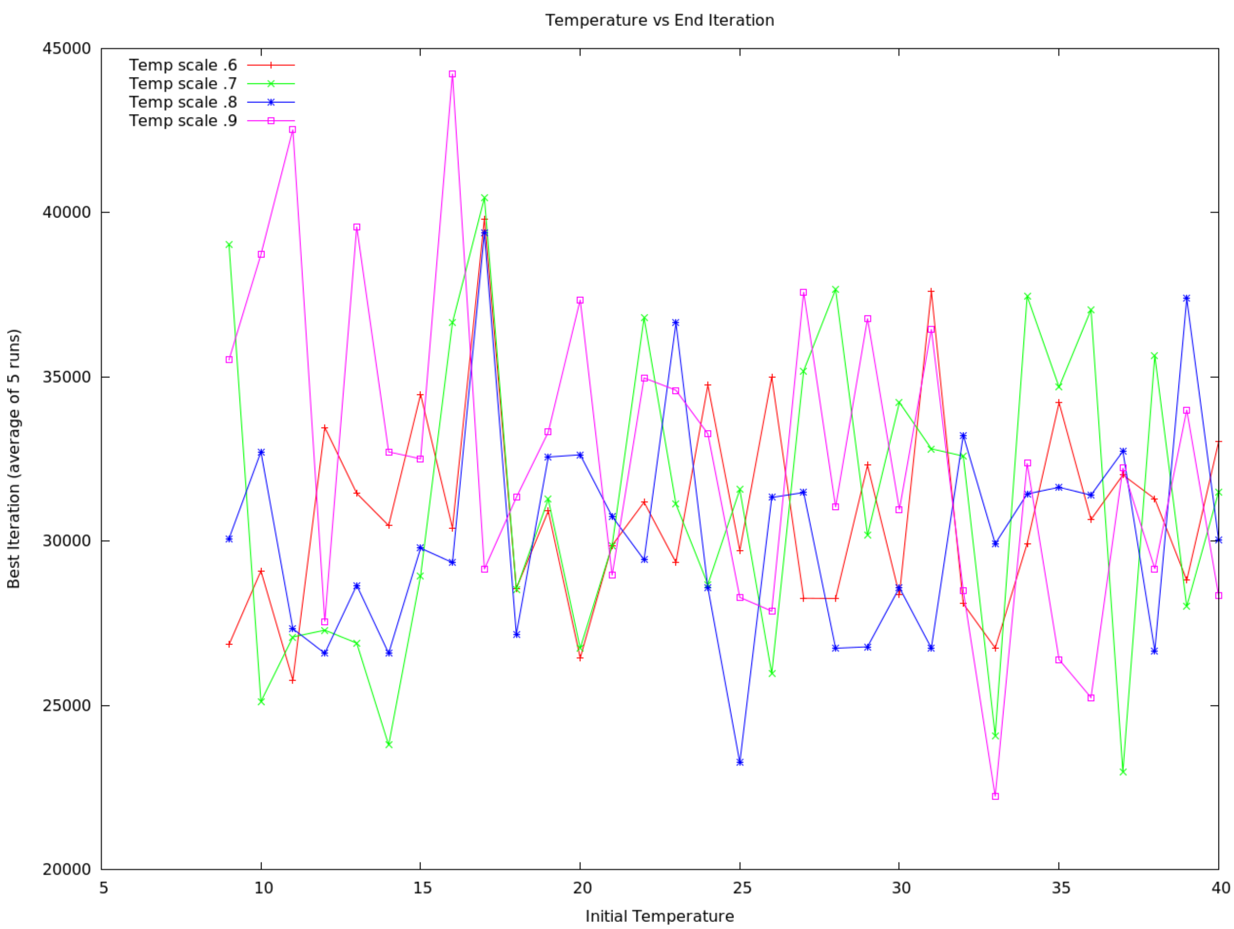
Randomly restarting would probably be the most effective way to find a good solution. I implemented a method of detecting maxima that works based on counting the number of sequential operations the did not result in an improvement. If a predetermined limit was reached it would save that solution shuffle the cities and start over again. after the cpu time is up it returns the best saved solution. The following picture shows a city of cost 4251 that was found at operation 4102871.

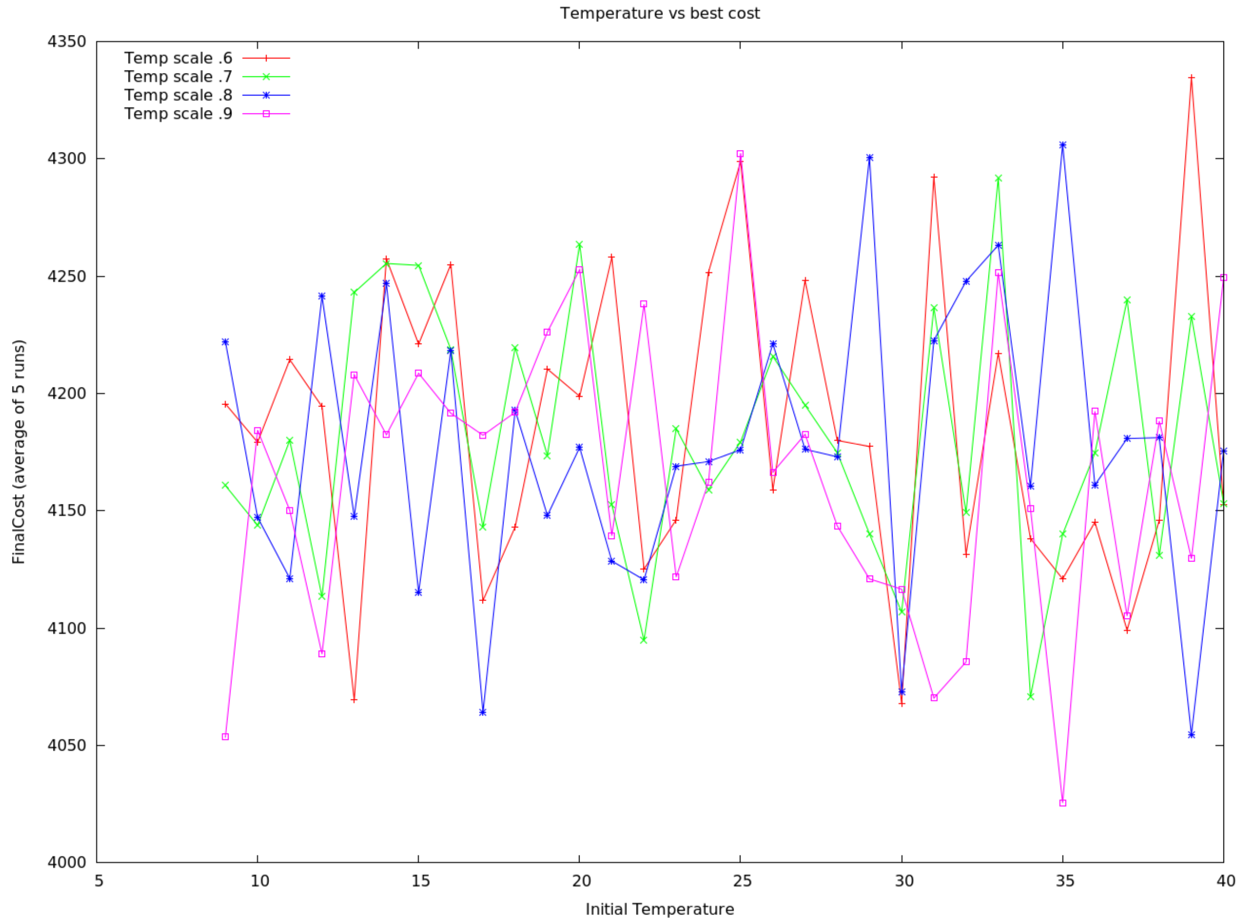


3) What is the effect of the initial temperature and cooling strategy in the performance of SA?

I implemented a geometric cooling strategy and a linear cooling strategy. The geometric outperformed the linear, however no data was collected with the linear method. For this experiment I ran repeated tests on the same graph each time changing the initial temperature. and the four lines each represent the factor by which the temperature was multiplied to reduce it. Initial temperature for each factor was graphed against the average best cost found and the average iteration at which it found that cost.

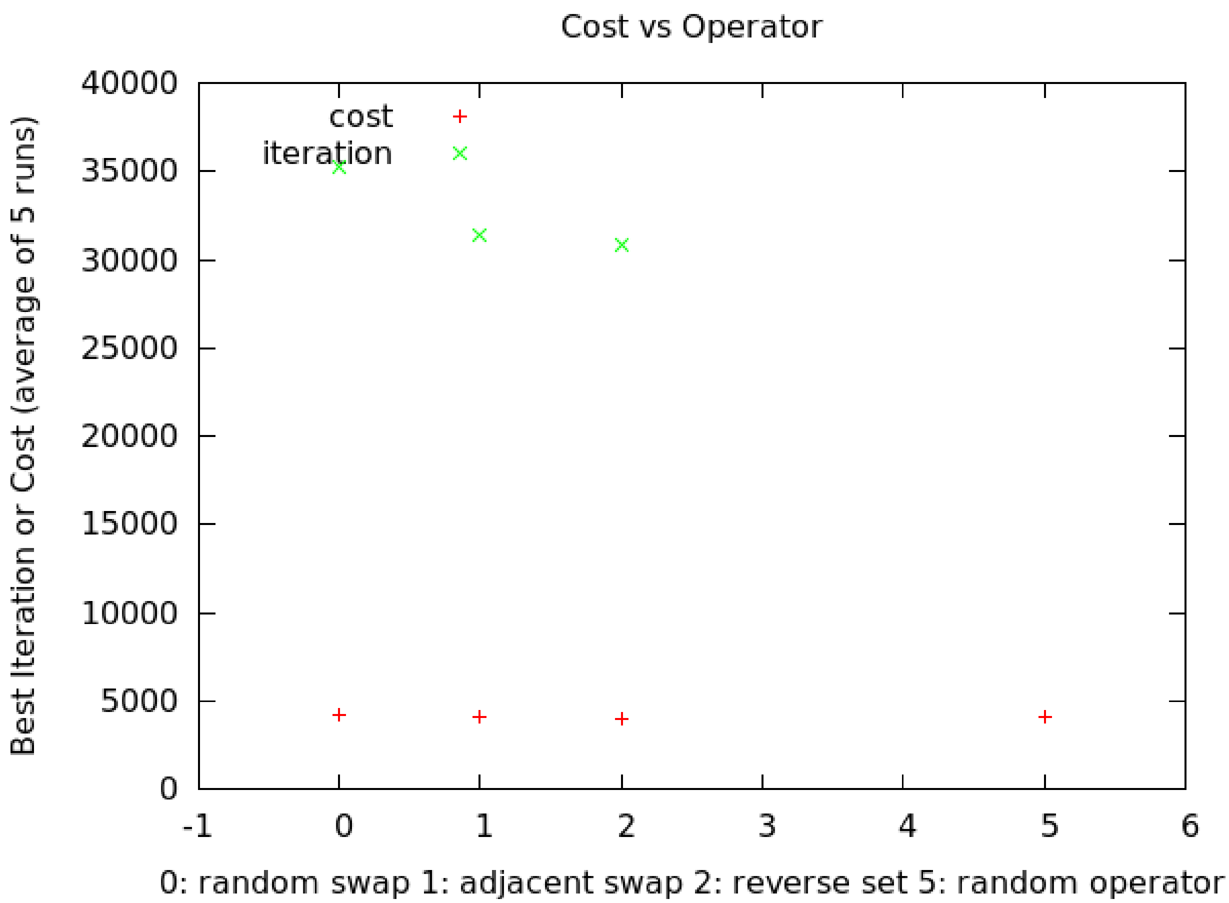
From the data 35 as the initial temperature produced the best convergence and about 25 produced the best average cost.





4) Which operator performs best and why?

All of the operators were tested against the same graph and the average iteration of convergance and the cost were measured. below we see that the reversing of a subset of the cities outperformed the other two. With adjacent swapping coming in second.


Cost vs Operator
cost
iteration
Best Iteration or Cost (average of 5 runs)
40000
35000
30000
25000
20000
15000
10000
5000
0
-1
0
1
2
3
4
5
6
0: random swap 1: adjacent swap 2: reverse set 5: random operator