Name: Thomas Milton User-ID: nvzf61

Algorithm A: 2-Opt

Algorithm B: Particle Swarm

Description of enhancement of Algorithm A:

My basic algorithm implements random 2-opt swaps from a randomly generated tour. It is chosen over my genetic and A* algorithms as it produces better tours despite being much simpler. For my enhanced 2-opt algorithm I followed this article https://towardsdatascience.com/around-the-world-in-90-414-kilometers-ce84c03b8552. The first change is to generate the initial tour using the nearest neighbour algorithm. To improve the quality of initial tours, I amended nearest neighbour to find nearest cities from both ends of the current tour. Second, it performs 2-opt swaps for a given number of iterations before generating a new initial tour. To improve this, I terminate the tour if no 2-opt swaps improve the length of the tour for a designated number of iterations. This number is set by hand as 1,000 for city sets up to 58, 10,000 for 175 and 180 and 100,000 for 535. I also implemented a systematic approach that performs 2-opt swaps between all pairs of edges. It uses random nearest neighbour initial tour generation, as shown in the article, to increase variation in the algorithm, otherwise there would be only num_cities unique tours produced since it will act the same if it is provided the same tour. It performs better than the original random 2-opt swaps with two ended nearest neighbour, however, for large city sets it will not provide good tours quickly given that it has to check every combination of edges multiple times for each initial tour. It is therefore commented out as speed of converge is more important for the secret city sets.

Description of enhancement of Algorithm B:

My basic algorithm implements PSO as shown in the lecture slides. I found that the algorithm struggled to improve tour lengths, this was due to the discretisation used as calculating bubble sort swaps takes a very long time. For my enhanced PSO I implemented the discretisation described by this paper https://www.computer.org/csdl/pds/api/csdl/proceedings/download-article/12OmNCwUmwT/pdf.
It makes 3 changes to the original algorithm: canonical form is removed, the new discretisation is introduced, and the equation for the particle's next velocity is altered. I kept the first 2 changes after experimentation, however the third was replaced, after experimentation, with the original equation from the notes since it has much better control over inertia and cognitive/ social learning factors so can help the particles move towards better tours. I noticed that the new implementation caused the particles to converge too quickly such that their next velocities became 0 when near the current shortest tour found so far. This causes these particles to stop moving and causes the algorithm to terminate early. I altered the values for theta, alpha and beta to prevent the particles heading towards the best tour found too quickly (these values have now been altered to allow for quick converge in the secret city sets). This still caused some particles to stop (especially on smaller city sets), so I implemented a “boredom” feature which gives the particles a small random velocity if they stop moving. This allows particles to continue to search the solution space so the algorithm will never fully converge and terminate. I also experimented with a nearest neighbour initial population, it is commented out.