



COS 314

Assignment 1

Mr. TM Whitaker
U22744968

Question 1:

The two programs I have created are that of Simulated Annealing and Iterated Local search which are both heuristic search algorithms used in this case to solve the Traveling Salesman Problem.

1. Iterated Local Search:

- **Algorithm Configuration:** This algorithm starts with an initial random route and then applies a local search algorithm to it by swapping all consecutive nodes in the route (excluding start and end) and repeating until no improvement is made. This finds the local optima and if this is better than the best route it is saved. A perturbation is then performed on the route by swapping two random nodes in the route to help it escape local optima. This is repeated up to the max iterations value and the best overall route is returned.
- **Experimental Setup:**
 - This experimental setup is to find the best (shortest) route from start through every node (campus) back to start.
 - `currentRoute` is a vector of `CampusVisit` objects representing the current route being worked on and the initial route is passed in as a random route.
 - The `distanceMatrix` represents the distances between campuses as described in the specification.
 - the max iterations is the number of times the algorithm must repeat and is set to 20.

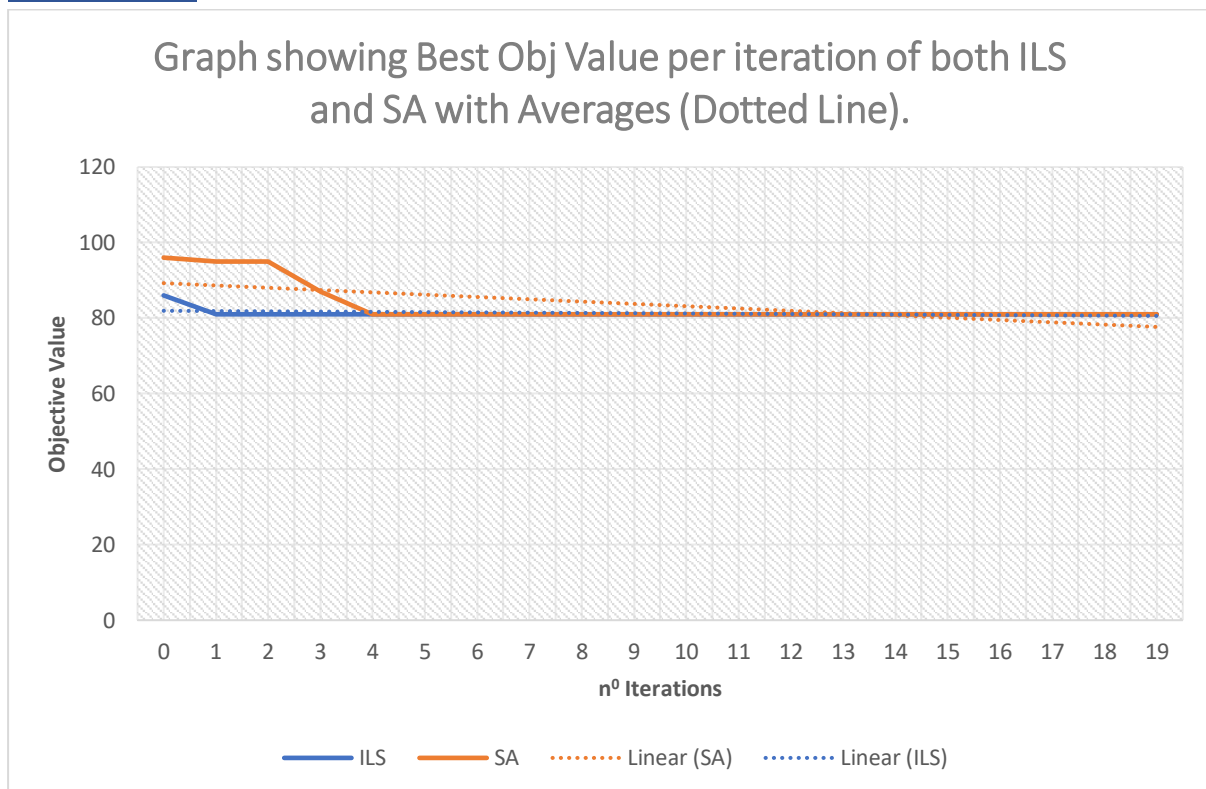
2. Simulated Annealing:

- **Algorithm Configuration:** This algorithm starts with an initial random route and initial temperature it then enters a loop where for each iteration it creates a new route by swapping two random campuses. If the new route is shorter, that will become the new working route, else if the new route is longer, it may still become the current route with a probability that depends on the difference in length and the current temperature this is to break out of accepting only local optima. The temperature decreases in each iteration, reducing the probability of accepting longer routes as the algorithm progresses. The best route found during the iterations is returned as the result.
- **Experimental Setup:**
 - Same as with ILS but with two extra variables.
 - `initialTemperature` is a value which represents the initial temperature that the algorithm will start with set to 1000.
 - `coolingRate` is the rate at which the temperature decreases with iteration set to 0.99.

Question 2:

Problem Set	ILS	SA
Best Solution (route)	Hatfield->Hillcrest->Groenkloof->Prinsof->Mamelodi->Hatfield	Hatfield->Mamelodi->Prinsof->Groenkloof->Hillcrest->Hatfield
Objective Function Val	81	81
Runtime (microseconds)	97	45
Av Obj Func	81.25	83.45

Question3:



Question 4:

From the results, it's clear that both the Iterated Local Search (ILS) and Simulated Annealing (SA) algorithms were able to find the optimal solution to the problem, as they both ended up with the shortest possible distance of 81.

However, there are some differences worth noting:

Execution Time: The SA algorithm was faster than the ILS algorithm, taking only 45 microseconds compared to 97 microseconds for ILS. This could be since SA uses a probabilistic approach to explore the solution space, which can sometimes lead to faster convergence. However, the actual time difference can change with different scenarios.

Convergence Speed: The ILS algorithm was able to find the optimal solution faster than the SA algorithm. ILS reached the optimal distance by the second iteration while SA took until the 5th iteration. This suggests that ILS may be more efficient at exploring the solution space for this problem however this may also be because the local search finds the local optima which also happens to be the global optima in this problem and the solution is therefore found after one run of the local search.

Solution Stability: Both algorithms found the same optimal distance, but they proposed different routes. This shows that there can be multiple optimal solutions to the problem, and the choice of algorithm can influence which one is found.

In conclusion, both ILS and SA are effective algorithms for solving the Traveling Salesman Problem. The choice between them may depend on the specific requirements of the problem, such as the need for speed, the size of the problem, and the importance of solution stability. For larger or more complex problems, it may even be beneficial to use a combination of both algorithms for the optimal solution.