<u>Technical Specification</u>

Iterated Local Search (ILS):

Experimental Setup:

- Algorithm Configuration:
    - Initial Solution Generation: Random selection of campuses in a sequence, excluding the starting point and endpoint which would be Hatfield.
    - Local Search: Swap two consecutive campuses in the solution.
    - Perturbation: Randomly swap any two campuses in the solution.
    - Stopping Criterion: A fixed number of iterations or a situation where there is no improvement after a set number of consecutive iterations.

- Description:
    - The algorithm starts by generating an initial solution by randomly selecting from the 5 campuses in a sequence.
    - It then enters a loop where it applies a local search to the current solution by swapping two consecutive campuses.
    - After the local search, a perturbation operator is applied to the solution by randomly swapping any two campuses.
    - The process of perturbation and local search is repeated for a fixed number of iterations or until a stopping criterion is met.
    - The best solution found is returned as the final route that visits each campus exactly once and returns to the starting point which in this case is Hatfield campus.

2. Simulated Annealing (SA):

Experimental Setup:

- Algorithm Configuration:
    - Initial Solution Generation: Random selection of campuses in a sequence, excluding the starting point which is Hatfield Campus.
    - Cooling: Exponential cooling rate with a starting temperature of 1000 and a cooling rate of 0.003.
    - Neighbour Generation: Swap two random campuses in the solution.
    - Acceptance Probability: Probability function based on the difference in distance between the current and new solutions.
    - Stopping Criterion: Temperature reaching close to 1 or no improvement after a set number of consecutive iterations.

- Description:
    - The algorithm starts by generating an initial solution by randomly selecting campuses in a sequence.
    - It then enters a loop where it generates a new solution by swapping two random campuses in the current solution.
    - The acceptance probability function determines whether to accept the new solution based on the difference in distance and the current temperature.
    - If the new solution is accepted, it becomes the current solution; otherwise, the current solution remains unchanged.
    - The temperature is reduced according to the cooling schedule, and the process is repeated until the temperature reaches close to 1 or a stopping criterion is met.
    - The best solution found during the iterations is returned as the final route that visits each campus exactly once and returns to the starting point.

Presentation of Results

Results were generated using the number of iterations carried out for each algorithm in order to notice or identify trends in the results of each algorithm.

After 10 iterations:

Problem Set	***ILS***	***SA***
Best Solution(route)	[0, 1, 2, 3, 4, 0]	[0, 1, 3, 2, 4, 0]
Objective Function Val	81	87
Runtime (ms)	0,2	0,0
Av Obj Func	81	94

After 100 iterations:

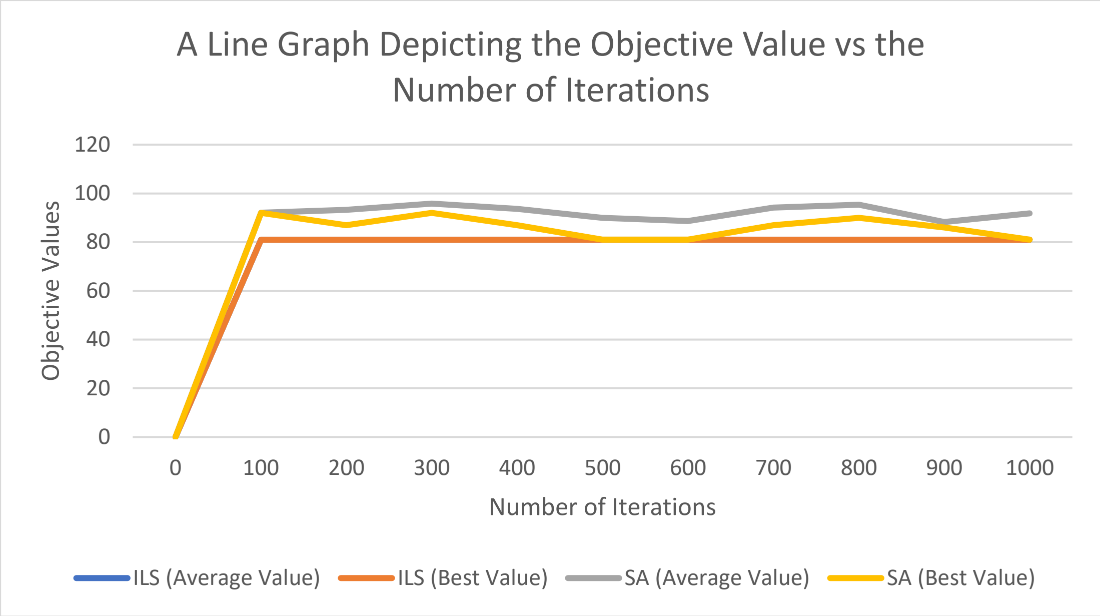
Problem Set	***ILS***	***SA***
Best Solution(route)	[0, 1, 2, 3, 4, 0]	[0, 2, 4, 1, 3, 0]
Objective Function Val	81	92
Runtime (ms)	0,16	0,01
Av Obj Func	81	92

After 1000 iterations:

Problem Set	***ILS***	***SA***
Best Solution(route)	[0, 1, 2, 3, 4, 0]	[0, 3, 1, 4, 2, 0]
Objective Function Val	81	81
Runtime(ms)	0,006	0,003
Av Obj Func	81	91,8

After 10000 iterations:

Problem Set	***ILS***	***SA***
Best Solution(route)	[0, 1, 2, 3, 4, 0]	[0, 4, 3, 2, 1, 0]
Objective Function Val	81	81
Runtime (ms)	0,0085	7,00E-04
Av Obj Func	81	81



Conclusion

The data clearly illustrates the difference in nature of the two algorithms. They are both similar in the sense that they aim to find the most optimal solution possible. However, the data clearly illustrates some of the pitfalls of these algorithms. On one hand, iterated local search has the tendency of getting stuck at a local optimum which in this case is 81 and it does not consider any other solutions after that. On the other, simulated annealing avoids this issue by not getting stuck at a local optimum and instead considers more solutions.

Due to this fact, it does take more iterations to converge to the most optimal solution but we also notice that with simulated annealing that it does offer more than one best path namely [0, 4, 3, 2, 1, 0] and [0, 1, 2, 3, 4, 0] whereas ILS only considers one path when searching for the most optimal solution that being [0, 1, 2, 3, 4, 0].

Interestingly simulated annealing was faster at run-time, but this is most likely due to the fact that it is more likely to accept less than optimal solutions in order to escape local optima which can be seen by the fact that the average objective function values are a lot higher and less consistent than the same values for iterated local search.

The goal of these algorithms is not necessarily to find the perfect solution, but rather to find a good enough solution within a reasonable amount of time. If finding an acceptable solution quickly is more important, then Iterated Local Search might be the preferred choice. However, if finding the global optimum or multiple near-optimal solutions is the priority, then Simulated Annealing would be more suitable, despite its longer convergence time.