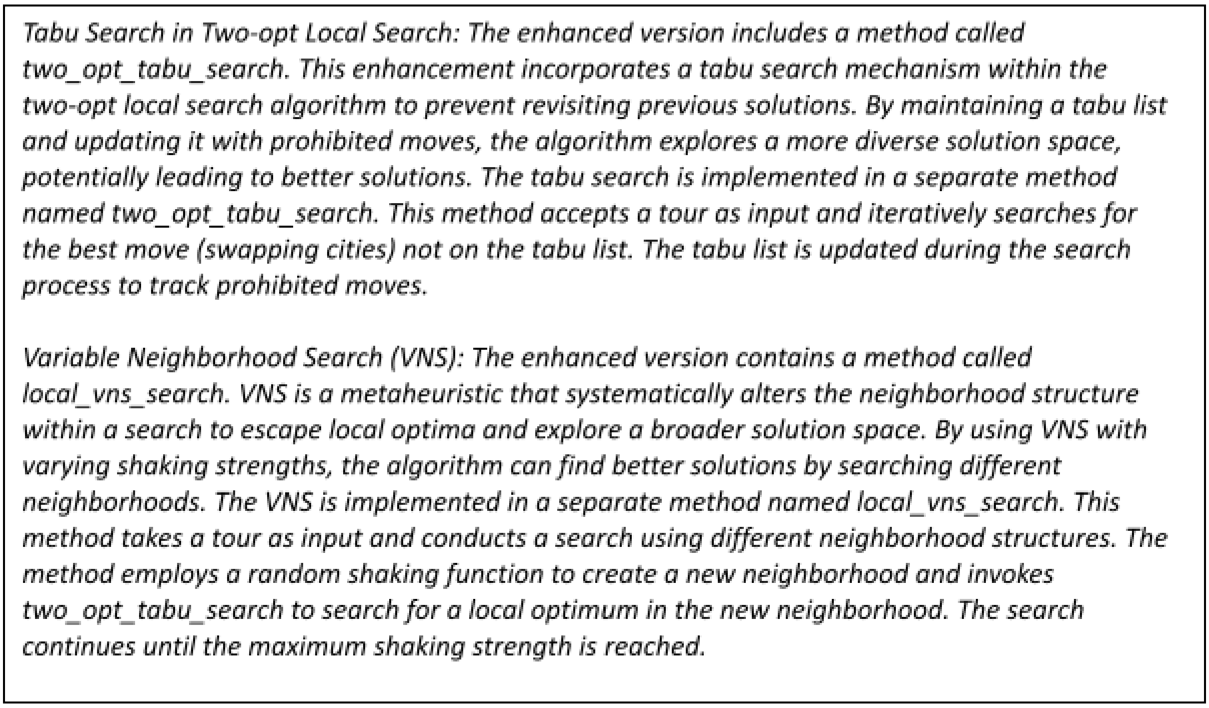
Name: Yosef Berezovskiy User-name: ttrq46

Algorithm A: Ant Colony Optimization

Algorithm B: Basic Greedy Search

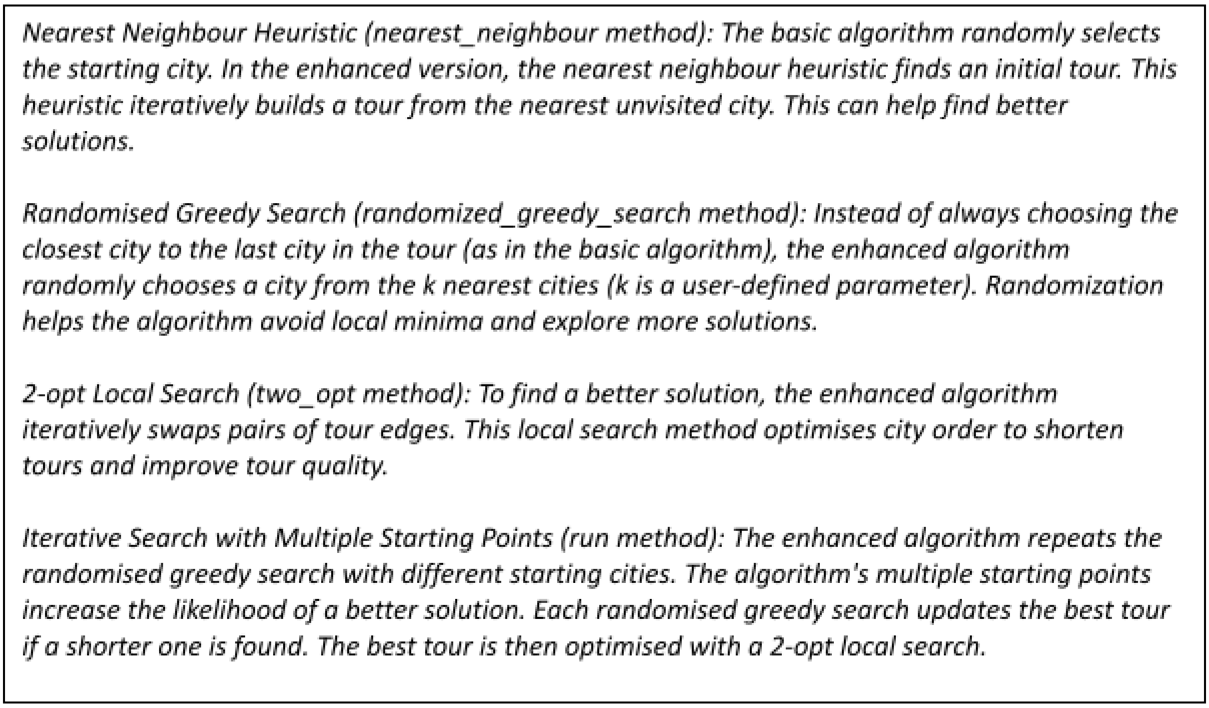
Description of enhancement of Algorithm A:



Tabu Search in Two-opt Local Search: The enhanced version includes a method called two_opt_tabu_search. This enhancement incorporates a tabu search mechanism within the two-opt local search algorithm to prevent revisiting previous solutions. By maintaining a tabu list and updating it with prohibited moves, the algorithm explores a more diverse solution space, potentially leading to better solutions. The tabu search is implemented in a separate method named two_opt_tabu_search. This method accepts a tour as input and iteratively searches for the best move (swapping cities) not on the tabu list. The tabu list is updated during the search process to track prohibited moves.

Variable Neighborhood Search (VNS): The enhanced version contains a method called local_vns_search. VNS is a metaheuristic that systematically alters the neighborhood structure within a search to escape local optima and explore a broader solution space. By using VNS with varying shaking strengths, the algorithm can find better solutions by searching different neighborhoods. The VNS is implemented in a separate method named local_vns_search. This method takes a tour as input and conducts a search using different neighborhood structures. The method employs a random shaking function to create a new neighborhood and invokes two_opt_tabu_search to search for a local optimum in the new neighborhood. The search continues until the maximum shaking strength is reached.

Description of enhancement of Algorithm B:



Nearest Neighbour Heuristic (nearest_neighbour method): The basic algorithm randomly selects the starting city. In the enhanced version, the nearest neighbour heuristic finds an initial tour. This heuristic iteratively builds a tour from the nearest unvisited city. This can help find better solutions.

Randomised Greedy Search (randomized_greedy_search method): Instead of always choosing the closest city to the last city in the tour (as in the basic algorithm), the enhanced algorithm randomly chooses a city from the k nearest cities (k is a user-defined parameter). Randomization helps the algorithm avoid local minima and explore more solutions.

2-opt Local Search (two_opt method): To find a better solution, the enhanced algorithm iteratively swaps pairs of tour edges. This local search method optimises city order to shorten tours and improve tour quality.

Iterative Search with Multiple Starting Points (run method): The enhanced algorithm repeats the randomised greedy search with different starting cities. The algorithm's multiple starting points increase the likelihood of a better solution. Each randomised greedy search updates the best tour if a shorter one is found. The best tour is then optimised with a 2-opt local search.

DESCRIPTION OF ALGORITHM ONLY IF THE ALGORITHM IS NOT COVERED IN LECTURES

Description of *non-standard* Algorithm A:

Describe any non-standard algorithms you have implemented that ***have not been covered in lectures*** *(otherwise these boxes should be blank) You need to convince me that your implementation is indeed that of the named algorithm and you need to* ***provide a full reference to the source for your algorithm****. You should* ***include a pseudocode description****. You can vary the sizes of these boxes but not the font (Calabri), font size (11) or paragraph properties (single space), and everything should fit onto one side of A4. (You can delete these instructions.)*

Description of *non-standard* Algorithm B:

Type here.