Name: Robin Craven User-name: ttsh43

Algorithm A: Simulated Annealing

Algorithm B: Genetic Algorithm

Description of enhancement of Algorithm A:

Same core parameters as basic (initial_temp=10000, final_temp=1, cooling_rate=0.9999, max_iterations=100000). Four enhancements:

-- Enhancement 1: Nearest Neighbour Initialisation: Replaces the random initial tour with a nearest neighbour heuristic (function generate_nearest_neighbor_tour, ~line 400). Starting from a random city, greedily visits the closest unvisited city. Produces a much shorter starting tour so SA spends iterations refining rather than recovering from a poor random permutation. Starting city is randomised to preserve stochasticity.

--Enhancement 2: Multiple Neighbourhood Operators: Adds two operators alongside 2-opt. 1) Insertion (get_neighbor_insertion, ~line 440): removes a city and re-inserts it at a random position, allowing cities to "jump" across the tour. 2) Or-opt (get_neighbor_or_opt, ~line 455): removes a segment of 1-3 consecutive cities and re-inserts elsewhere, reorganising short subsequences. Three structurally different move types reduce the chance of being trapped in a local optimum specific to one operator.

-- Enhancement 3: Adaptive Neighbourhood Selection: Function get_neighbor_adaptive (~line 475) selects among the three operators with probability proportional to their success count (move_success = [1,1,1], incremented on improving moves). Operators that yield more improvements used more frequently, adapts search strategy to each problem instance automatically.

--- Enhancement 4: Reheating Mechanism: Tracks iterations_since_improvement. After 1000 iterations without a new best (reheat_threshold=1000), temperature is multiplied by 1.5 (reheat_factor), capped at initial_temp/2, and the counter resets. res the ability to escape local optima, creating alternating phases of exploitation and exploration. See simulated_annealing_enhar 545).

Description of enhancement of Algorithm B: Genetic Algorithm

Same core parameters as basic (pop_size=50, max_it=500, mutation_rate=0.1, tournament_size=5). Four enhancements:

--- Enhancement 1: Elitism: Top 2 individuals (elite_count=2) are copied unchanged into the next generation, guaranteeing monotonic improvement of the best tour. Addresses the key weakness of full generational replacement where the best solution can be lost. See genetic_algorithm_enhanced (~line 580).

--- Enhancement 2: 2-Opt Local Search (Memetic GA): With probability 0.1 (local_search_prob), a child undergoes 20 random 2-opt improvement attempts (function two_opt_improvement, ~line 530), accepting only improving swaps. This creates a memetic/hybrid GA: crossover and mutation explore broadly, 2-opt refines locally. The low probability balances refinement benefit against computational cost.

--- Enhancement 3: Inversion Mutation: Function mutate (~line 515) applies swap mutation with probability 0.5 and inversion mutation (inversion_mutation) with probability 0.5. Inversion reverses a segment between two random positions, equivalent to a 2-opt move. It preserves adjacency within the segment and only changes two edge connections, making it more TSP-appropriate than swap. Using both types provides structural diversity.

--- Enhancement 4: Nearest Neighbour Seeding: Function initialize_population (~line 430) seeds 5 individuals using the nearest neighbour heuristic (generate_nearest_neighbor_tour), each from a different random city. The remaining 45 are random. This gives the GA high-quality genetic material from generation 1, which crossover can immediately exploit. Combined with elitism, these good solutions are preserved through early generations.