

Artificial Intelligence: Week 05 Assignment Analysis

(2022-2)

Course / Class:

Artificial Intelligence

Class 01

Professor:

Dae-Won Kim

Submission Date:

2022. 10. 10

Student ID:

20184757

Name:

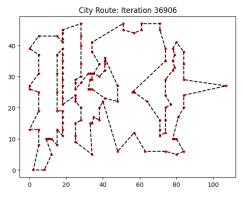
주영석

Objective: Learning and Improving the principle of Genetic Algorithm for TSP Problems

(5.1) Execute and analyze Genetic Algorithm sample code.

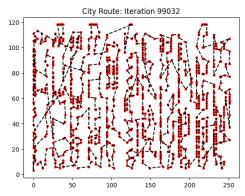
[Level 1: Beginning] Write each execution result (Path Plot, Execution Time, Solution Path, Minimum Cost) for the provided input data.

1) Data 1



- a) Path Plot:
- b) Execution Time: 17.51275324821472
- c) Solution Path: [0, 5, 11, 4, 12, 17, 13, 14, 15, 16, 24, 18, 25, 26, 27, 28, 29, 19, 20, 21, 22, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 48, 47, 46, 54, 53, 45, 44, 52, 73, 63, 67, 74, 76, 77, 80, 88, 92, 97, 111, 122, 129, 120, 117, 113, 123, 124, 125, 130, 126, 127, 128, 121, 116, 119, 115, 118, 114, 108, 107, 112, 106, 105, 104, 99, 100, 101, 98, 93, 91, 109, 110, 102, 103, 96, 95, 94, 89, 90, 72, 71, 79, 75, 78, 82, 83, 84, 85, 87, 86, 81, 68, 64, 61, 65, 69, 70, 66, 62, 55, 49, 50, 51, 56, 57, 58, 59, 60, 43, 42, 41, 40, 23, 10, 3, 9, 8, 2, 1, 7, 6, 0]
- d) Cost: 610.7738222694078

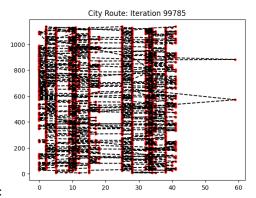
2) Data 2



- a) Path Plot:
- b) Execution Time: 83.99691319465637

d) Cost: 4486.250793642459

3) Data 3



a) Path Plot:

Execution Time: 257.4975690841675

Solution Path: [0, 209, 208, 582, 581, 580, 579, 578, 577, 206, 207, 204, 576, 575, 574, 573, 572, 571, 198, 199, 200, 202, 203, 864, 978, 1216, 1479, 1215, 977, 976, 1214, 863, 975, 1213, 1212, 974, 861, 1478, 1537, 1805, 1804, 1536, 1477, 1211, 973, 860, 972, 1210, 1209, 971, 859, 858, 970, 1207, 1206, 969, 857, 1205, 1204, 968, 967, 1203, 1532, 1533, 1534, 1535, 566, 567, 2528, 2236, 2239, 2240, 2241, 2530, 2529, 2531, 2532, 2242, 2243, 2668, 2900, 3175, 3255, 3174, 2899, 2901, 3176, 3470, 3469, 3634, 3633, 3648, 2523, 2522, 2521, 2520, 1878, 1879, 1742, 1743, 1745, 1746, 1747, 1748, 1749, 820, 819, 818, 817, 510, 508, 509, 816, 814, 813, 812, 952, 1521, 1441, 1169, 1171, 1443, 1444, 953, 1172, 1173, 1445, 1446, 1174, 1175, 1447, 503, 502, 129, 2506, 2877, 3149, 2500, 1200,1747, 1748, 1749, 820, 819, 818, 817, 510, 508, 509, 816, 814, 813, 812, 952, 1521, 1441, 1169, 1171, 1443, 1444, 953, 1172, 1173, 1445, 1446, 1174, 1175, 1447, 503, 502, 129, 2506, 2877, 3149, 2500, 2499, 2498, 2497, 2185, 2186, 2187, 2189, 2190, 2191, 2501, 2502, 2504, 2505, 2194, 2195, 2196, 2197, 2198, 2200, 2201, 2202, 2203, 2204, 2507, 2205, 2206, 2208, 2207, 2509, 2510, 2511, 2209, 2210, 2211, 2512, 2213, 2513, 2515, 2516, 2214, 2215, 2216, 2217, 2218, 2219, 2517, 2518, 2519, 2220, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2525, 2524, 3167, 2664, 2892, 2891, 3166, 3463, 3462, 3461, 3632, 3631, 3460, 3158, 2661, 2662, 2884, 3223, 3169, 3170, 2895, 2666, 2896, 3171, 3172, 2897, 2667, 2898, 3173, 3254, 3471, 3563, 3464, 3474, 3475, 3476, 3639, 3640, 3477, 3478, 3477, 3478, 3479, 3441, 3462, 3464, 3464, 3462, 3461, 3482, 2910, 2673, 2542, 2539, 2540, 2541, 2252, 2251, 2250, 2538, 2536, 2537, 2248, 2247, 2246, 2245, 2244, 2533, 2534, 2535, 2669, 2902, 3177, 3256, 3178, 2903, 2904, 3179, 3257, 3180, 2905, 2670, 2672, 2907, 2529, 3183, 3260, 3483, 3484, 3485, 3185, 2675, 2911, 3188, 3489, 3490, 3491, 3189, 2676, 2912, 3188, 3487, 3492, 3493, 3494, 3495, 3496, 3499, 2392, 2933, 2691, 2692, 2934, 1808, 1809, 1810, 1811, 1543, 1545, 1546, 1547, 1220, 1481, 1221, 1548, 1189, 1248, 1484, 1487, 1522, 1759, 1758, 1755, 1755, 1755, 1755, 1755, 4755, 47153, 1455, 1182, 1181, 1454, 1750, 2230, 2231, 2231, 2233, 2234, 2235, 2236, 2237, 2248, 2263, 2364, 3264, 3264, 3244, 3433, 3244, 3443, 3435, 3244, 2433, 3248, 3489, 3490, 3491, 3189, 2674, 1888, 3487, 3492, 3499, 349 2854, 3245, 3120, 3121, 2855, 2650, 2856, 3122, 3246, 3432, 3431, 3621, 3620, 3521, 3429, 3428, 2851, 3117, 3119, 2853, 2478, 2477, 2153, 2154, 2156, 2157, 2158, 2480, 2481, 2159, 2160, 2161, 2482, 2483, 2460, 2461, 2132, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2465, 2464, 2644, 2843, 3107, 3241, 3106, 2842, 2841, 3105, 3240, 2643, 2840, 3104, 3103, 2839, 2838, 3102, 3101, 3101, 2837, 3647, 2625, 2810, 3070, 3071, 2811, 2626, 3232, 2477, 2426, 2083, 2084, 2085, 2084, 2084, 2429, 2431, 3062, 2803, 3383, 3382, 2513, 3384, 3385, 3386, 3066, 2806, 2805, 3604, 3460, 3081, 3080, 2820, 2439, 2440, 2102, 2104, 2105, 2106, 2107, 2441, 2108, 2109, 2442, 2443, 739, 740, 741, 406, 405, 75, 404, 2122, 2124, 2455, 2456, 2457, 3413, 3412, 3411, 3410, 3095, 2832, 2833, 3096, 3097, 2834, 2835, 3098, 3099, 2836, 2641, 2642, 2462, 2462, 2463, 2466, 2142, 2143, 2144, 2145, 1698, 1853, 1854, 1701, 1857, 3619, 3427, 3618, 3617, 3615, 3616, 3426, 2647, 2849, 3115, 3244, 3116, 2850, 3098, 3099, 2836, 2641, 2642, 2462, 2462, 2463, 2466, 2142, 2143, 2144, 2145, 1698, 1853, 1854, 1701, 1857, 3619, 3427, 3618, 3617, 3615, 3616, 3426, 2647, 2849, 3115, 3244, 3116, 2850, 3098, 3099, 2836, 2641, 2642, 2462, 2462, 2463, 2466, 2142, 2143, 2144, 2145, 1698, 1853, 1854, 1701, 1857, 3619, 3427, 3618, 3617, 3615, 3616, 3426, 2647, 2849, 3115, 3244, 3116, 2850, 3098, 3099, 2836, 2641, 2642, 2462, 2462, 2463, 2466, 2142, 2143, 2144, 2145, 1698, 1853, 1854, 1701, 1857, 3619, 3427, 3618, 3617, 3615, 3616, 3426, 2647, 2849, 3115, 3244, 3116, 2850, 3427, 3427, 3428, 34 3091, 2834, 2835, 3098, 3099, 2830, 2641, 2642, 2402, 2405, 2440, 2405, 2406, 2442, 2143, 2144, 2145, 1098, 1853, 1854, 1701, 1851, 3619, 3612, 3611, 3611, 3611, 3611, 3612, 3611, 3622, 2641, 2444, 3113, 3115, 3244, 3116, 2850, 2464, 2474, 2473, 2150, 2152, 2151, 1855, 1856, 1703, 1857, 1398, 1159, 299, 298, 1118, 1397, 1395, 1506, 1706, 15 1300, 1342, 1080, 1079, 1341, 1340, 1076, 2977, 2976, 3210, 2724, 2584, 2723, 2975, 3311, 3560, 3561, 3313, 3314, 3510, 3562, 2974, 2722, 2583, 3310, 3559, 3558, 1256, 1010, 1011, 1257, 1258, 1012, 1488, 1259, 879, 1013, 626, 625, 624, 623, 423, 428, 429, 520, 627, 247, 622, 1819, 1818, 1817, 1816, 1579, 1484, 1259, 1004, 874, 1249, 1248, 1003, 1002, 1474, 1246, 1001, 1000, 1245, 1244, 999, 238, 1239, 1240, 2963, 3299, 3207, 262, 2714, 2579, 1945, 2316, 2317, 2318, 1946, 1947, 1948, 1949, 1950, 1578, 1577, 1576, 1575, 1574, 1573, 1572, 1571, 997, 1241, 1242, 1243, 998, 2694, 2937, 3282, 1957, 1958, 2327, 2328, 1959, 1960, 1961, 1962, 2329, 2331, 2332, 2333, 1963, 1964, 1965, 1966, 2326, 2972, 2720, 2582, 2721, 2973, 3209, 3309, 3555, 3555, 3308, 3208, 29071, 2719, 2581, 2325, 1956, 1955, 2324, 1954, 1953, 2323, 2322, 1952, 1951, 3300, 3553, 3555, 3308, 3307, 1006, 875, 619, 242, 244, 244, 246, 245, 621, 620, 876, 1007, 1253, 1486, 1254, 1008, 877, 878, 1009, 1255, 1487, 1583, 1820, 1821, 1822, 1823, 1584, 1585, 1586, 1587, 1588, 1824, 1825, 1826, 1827, 1589, 1014, 1260, 1261, 1015, 1015, 881, 629, 628, 251, 252, 253, 254, 255, 265, 261, 261, 265, 259, 260, 261, 262, 15, 263, 264, 265, 267, 266, 16, 17, 18, 19, 274, 273, 272, 271, 270, 269, 268, 641, 640, 639, 638, 637, 884, 1019, 1489, 1268, 1267, 1018, 1017, 1266, 1593, 1594, 1783, 1828, 1829, 1830, 1831, 1597, 1596, 1269, 1020, 1022, 1270, 1271, 885, 1023, 1024, 1272, 1598, 1784, 1981, 1980, 1979, 1978, 2342, 2344, 2345, 2346, 3328, 3327, 3326, 3325, 3324, 3323, 3322, 3320, 33508, 3350, 3320, 3350, 3350, 3350, 3350, 3350, 3351, 3315, 2979, 2786, 2777, 2980, 2312, 12882, 2782, 2729, 2982, 2912, 2943, 2740, 2590, 2741, 2994, 2995, 2742, 2591, 2349, 2348, 2347, 1982, 1983, 1984, 1985, 1986, 2352, 1992, 1991, 1990, 1989, 1988, 2350, 2592, 2997, 3329, 3215, 3331, 3332, 3333, 3334, 3255, 3254, 2353, 2593, 2744, 2594, 2746, 2595, 2599, 2597, 2751, 3004, 3006, 2753, 2754, 3007, 3008, 2755, 2598, 2599, 2000, 2757, 3010, 3000, 2756, 3340, 3339, 3333, 3333, 3300, 2000, 27477, 2748, 3001, 3002, 2749, 2596, 1997, 1994, 1995, 1996, 2357, 2358, 2359, 2360, 2361, 2362, 2365, 2366, 2367, 2368, 2014, 2012, 2011, 2010, 2009, 2008, 2007, 2006, 2005, 2004, 2003, 2004, 2000, 1998, 1786, 1607, 1606, 1605, 1028, 1278, 1279, 1029, 891, 1030, 1281, 1031, 1581, 1031, 1581, 1031, 1 1032, 1282, 1283, 1033, 1034, 1284, 1608, 1609, 1610, 1787, 1611, 1286, 1035, 893, 1036, 1287, 1288, 1037, 1038, 1289, 1290, 1039, 296, 294, 293, 292, 291, 290, 656, 657, 658, 659, 660, 661, 297, 298, 1036, 1287, 1288, 1037, 1038, 1289, 1290, 1039, 296, 294, 293, 292, 291, 290, 656, 657, 658, 659, 660, 661, 297, 298, 1038, 299, 300, 301, 24, 25, 26, 27, 28, 30, 667, 668, 669, 670, 313, 671, 672, 673, 316, 315, 314, 34, 33, 32, 31, 312, 311, 310, 309, 308, 307, 306, 305, 664, 665, 666, 895, 1618, 1619, 1620, 1621, 1622, 1790, 2022, 2380, 2379, 2377, 2376, 2375, 3577, 3578, 3348, 3349, 3021, 3020, 2766, 3019, 3220, 3018, 2765, 2605, 2604, 2764, 3017, 3347, 3219, 3346, 3575, 3574, 3345, 3344, 3343, 3572, 3571, 3342, 3217, 3219 2022, 2580, 2397, 2317, 2316, 2315, 3517, 3518, 3348, 3494, 3021, 3020, 2160, 3019, 3220, 3018, 2705, 2003, 2004, 2164, 3011, 3344, 3219, 3340, 3515, 3514, 3343, 3514, 3543, 3512, 3511, 3344, 2511, 3011, 2758, 2602, 2760, 3013, 3014, 2761, 2762, 3015, 3218, 3016, 2763, 2603, 2017, 2016, 2015, 2369, 2370, 2371, 2372, 2373, 2374, 2020, 2019, 2018, 1788, 1789, 1617, 1616, 1040, 1292, 1293, 894, 1041, 1294, 1295, 1042, 1043, 1296, 1297, 1044, 1045, 1298, 1299, 1046, 1047, 1300, 1301, 1302, 1048, 1049, 1303, 1492, 1304, 1050, 896, 897, 1051, 1305, 1493, 1625, 1626, 1835, 1834, 2028, 2030, 2029, 2386, 2385, 2384, 2027, 2026, 2025, 2024, 2381, 2382, 2383, 2368, 3222, 3027, 3028, 3284, 2027, 2026, 2025, 2024, 2381, 2382, 2383, 2608, 3222, 3026, 2771, 2770, 3025, 3034, 2769, 2606, 2767, 3022, 3221, 3579, 3580, 3351, 3352, 3353, 3581, 3582, 3354, 3223, 3027, 2772, 2609, 2610, 2773, 3028, 3024, 2790, 3370, 3364, 2398, 2395, 2396, 2397, 2615, 2783, 3041, 3040, 3377, 3376, 3375, 3374, 3227, 3051, 2792, 2791, 3050, 3049, 2790, 3372, 3373, 3053, 2794, 2795, 3054, 3055, 2796, 2617, 3228, 3229, 3056, 2797, 2618, 2413, 2063, 2062, 2061, 2060, 2410, 2409, 2408, 2058, 2057, 2056, 2406, 2407, 2055, 2054, 2404, 2403, 2402, 2401, 2400, 2049, 2050, 2051, 2052, 2616, 2788, 3046, 3226, 3045, 2787, 2786, 3044, 3043, 2785, 2784, 3042, 3366, 3367, 3368, 3369, 3589, 3590, 3370, 3048, 2047, 2046, 2045, 2044, 2043, 2042, 2041, 2040, 2039, 2038, 2037, 2394, 2393, 2392, 2391, 2390, 2389, 2388, 2032, 2033, 2034, 2036, 2779, 3035, 3036, 3360, 3361, 3512, 3037, 2613, 2614, 2780, 3038, 3039, 2781, 2782, 3359, 3358, 3357, 3356, 3586, 3585, 3583, 3584, 3355, 317, 674, 1627, 1628, 680, 679, 678, 677, 676, 675, 318, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 335, 336, 337, 40, 39, 38, 37, 36, 334, 332, 331, 682, 681, 1059, 1314, 1313, 1629, 1630, 1631, 1791, 1632, 1494, 1316, 1317, 1635, 1636, 1638, 1639, 1792, 1320, 1321, 689, 690, 691, 692, 693, 344, 345, 346, 347, 348, 349, 52, 342, 47, 49, 535, 532, 531, 608, 607, 608, 53, 54, 55, 65, 68, 363, 533, 532, 531, 350, 609, 701, 702, 573, 599, 703, 1647, 2416, 2413, 379, 320, 3591, 3592, 3593, 3594, 3380, 3381, 2798, 3075, 3058, 2799, 2619, 2621, 2801, 3060, 3061, 2802, 2622, 2417, 2416, 2415, 2064, 2065, 2067, 2068, 2069, 2070, 2071, 2418, 2419, 2420, 2074, 2073, 2072, 1497, 1337, 1793, 1649, 1648, 1839, 1838, 1837, 1836, 2624, 2809, 3069, 3231, 3595, 3596, 3387, 3597, 3598, 3389, 3389, 3390, 3233, 3072, 2812, 2627, 2813, 3073, 3075, 2815, 2628, 2816, 3076, 3077, 2817, 2818, 3078, 3234, 3079, 2819, 2629, 2094, 2093, 2092, 2091,

d) Cost: 12557.245913552035

[Level 2: Approaching] After executing the genetic algorithm, briefly write what you understood and what you thought.

When the genetic algorithm function is executed, the algorithm initializes a population. The initialization function creates a population with a size of globally defined pool size. The first path is generated with the greedy algorithm, and the rest are initialized with random path generation followed by a simulated annealing algorithm. After the initialization of the population is done, selection, crossover, mutation, and replacement of the solution in the pool are made for maximum iteration times.

First, the selection function chooses the solution path randomly with the tournament size. From the chosen solution paths, the function selects solution paths with the smallest costs. This is called a tournament selection algorithm. In this way, the solution paths with lower cost can be selected with higher probability, but not always selected.

Second, the crossover function gets two solution paths selected by the selection function. A child solution path is initialized and a random subpath from the solution path with a smaller cost is copied to the child solution path. The remaining parts of the child solution path are filled with the second selected solution path. From the second solution path, the cities that have been already filled into the child solution path are removed, and the remaining cities are filled into the child solution path that has not been filled yet, preserving the order. This crossover method is called 'ordered crossover' (OX crossover). After generating the child solution path with the crossover function, the GA function calculates its cost.

Third, the mutation function gets a single solution path selected by the selection function. The child solution path generated by the mutation function is the 2-opt transformation of the original path. With 2 points selected randomly, the subpath between the 2 points is flipped from the original path. Also, after generating the child solution path with the mutation function, the GA function calculates its cost.

Lastly, after generating 2 child solution paths with crossover and mutation functions, the GA function replaces 2 solutions in the pool from the largest cost solutions to the generated child solutions.

After repeating these 4 steps for maximum iteration times, the GA function returns the solution path with the smallest cost in the solution pool.

(5.2) Execute and analyze Genetic Algorithm which you proposed.

[Level 3: Proficient] Write your idea that can improve sample genetic algorithm, and briefly explain the reason with example.

A simple way to improve the sample genetic algorithm is making more child solutions using various crossover and mutation methods to increase the probability for a better child to be generated. Also, selecting a better pool for the next generation in the replacement step can improve the genetic algorithm.

First, to generate more child solution paths, we can generate more child solution paths from the sample code. In the sample code, a child solution with a random subpath from the parent solution with a smaller cost is generated. We can make another child solution identically with a random subpath from the second parent solution. For example, there are 2 chosen parent solution paths (a) 0-1-2-3-4-5-6-7-8-9-0 and (b) 0-9-8-7-6-5-4-3-2-1-0. Let's assume solution path (a) has a smaller cost value and the selected 2 indices of crossover point are 3 and 6. The sample code only generates 1 child solution, (c) 0-9-8-3-4-5-7-6-2-1-0. But we can make one more child solution, (d) 0-1-2-7-6-5-3-4-8-9-0. Also, we can reverse the selected random subpath, making 2 more child solutions. From the example above, we can make (c') 0-9-8-5-4-3-7-6-2-1-0, and (d') 0-1-2-5-6-7-3-4-8-9-0. Therefore, we can generate 4 child solution paths from the OX crossover.

Second, we can generate a child solution not only with the OX crossover but also with the PMX crossover method. The PMX crossover method copies a random subpath from one parent solution path, which is the same as the OX crossover method, and copies the remaining from another parent solution. After copying from two parents is done, there will be duplicate cities between the subpath from one parent solution and the other part from the other parent solution. The duplicate cities in the other part from the other parent solution will be replaced with cities in the subpath from the identical parent solution. For example, there are 2 parent solution paths (a) 0-1-2-3-4-5-6-7-8-9-0 and (b) 0-4-5-2-1-8-7-6-9-3, and the selected 2 indices of crossover point is 4 and 8. Then, the subpath from (a) will be 4-5-6-7, and the child solution path will be initialized to (c) 0-4-5-2-4-5-6-7-9-3-0. There are duplicate cities in (c), 4, and 5. The indices of cities 4 and 5 in (a) are 4 and 5, and the corresponding cities in (b) with indices 4 and 5 are 1 and 8. Thus, the duplicate cities in (c) will be replaced as (c) 0-1-8-2-4-5-6-7-9-3. This crossover method is called 'Partially Mapped Crossover' duplicate cities in the child solution path are replaced with the partially mapped cities. We can also generate 4 child solution paths from the PMX crossover.

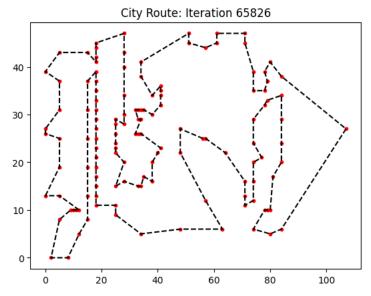
Next, we can generate a child solution path with various mutation methods. First, we can generate a child solution with a single-point mutation method. Selecting a single random point in the path, divide the total path into 2 subpaths except the first and the last city. Then, swap the order of 2 subpaths. For example, there is a solution path (a) 0-1-2-3-4-5-0, and the single point selected is 3. The child solution path with the single-point mutation method is generated as (c) 0-3-4-5-1-2-0. Another mutation method similar to the above is the three-point mutation method. Where the single-point mutation method divided the total path into 2 subpaths, the three-point mutation method first selects a subpath with 2 points and again divides the subpath into 2 subpaths with another point between the previously selected 2 points. Then, swap the order of 2 subpaths. From the above example path (a) and the selected three points are 2, 3, and 5. Then the child solution path will be (d) 0-1-3-4-5-2-0.

Lastly, we can think about the replacement method. The sample code simply replaced 2 solutions with the largest cost to 2 child solutions generated with crossover and mutation. Rather than simply replacing the solutions in the pool, we can also apply the tournament selection algorithm like the selection step. We generate 4 child solutions from OX crossover, 4 from PMX crossover, 1 from 2-opt mutation, 1 from single-point mutation, and 1 from three-point mutation. In the replacement step, first, make a tournament list with solutions that has largest costs, and delete them from the solution pool. Also, add the newly generated child solutions into the tournament list. Then, run a tournament selection algorithm, and add the selected solutions into the pool. In this way, remaining the best solutions in the pool, the child solution path can have a chance of being the next generation, where the solution with a smaller cost has a larger probability and the solution with a larger cost has a smaller probability.

[Level 4: Exemplary] Write the implementation and execution results (Path Plot, Execution Time, Solution Path, Minimum Cost) of the proposed algorithm for the provided input data. Include the results of your analysis comparing the sample algorithm and the proposed algorithm.

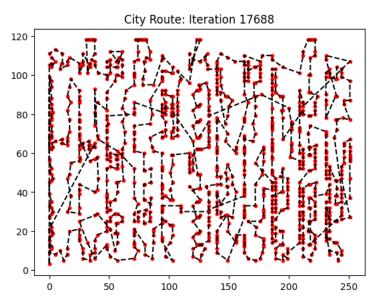
(1) Execution results

1) Data 1



- a) Path Plot:
- b) Execution Time: 44.5940580368042
- c) Solution Path: [0, 6, 7, 1, 2, 8, 9, 3, 10, 23, 40, 41, 42, 43, 60, 59, 58, 57, 56, 55, 51, 50, 49, 48, 47, 46, 54, 45, 53, 63, 67, 74, 76, 77, 80, 81, 68, 64, 61, 69, 65, 62, 66, 70, 75, 78, 82, 83, 84, 85, 79, 71, 72, 90, 89, 94, 95, 96, 103, 102, 110, 109, 115, 119, 116, 121, 128, 130, 129, 122, 111, 113, 117, 120, 123, 124, 125, 126, 127, 118, 114, 108, 107, 112, 106, 105, 104, 99, 100, 101, 98, 93, 91, 87, 86, 92, 97, 88, 73, 52, 44, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 22, 21, 20, 19, 18, 24, 17, 12, 4, 11, 13, 14, 15, 16, 5, 0]
- d) Cost: 593.8203259199624

2) Data 2



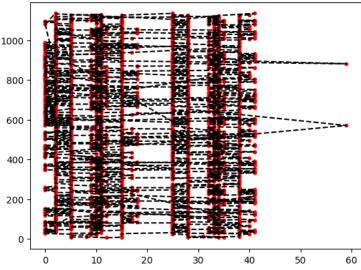
- a) Path Plot:
- b) Execution Time: 58.00222206115723

559, 558, 557, 556, 555, 554, 722, 838, 847, 846, 848, 844, 843, 842, 841, 840, 839, 865, 864, 863, 892, 893, 894, 895, 896, 897, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 902, 901, 1069, 969, 968, 960, 999, 999, 997, 902, 1030, 1042, 1031, 1046, 1045, 1060, 1082, 1068, 1061, 1028, 1093, 1038, 1022, 1035, 1032, 1044, 1059, 1039, 10

d) Cost: 4576.279669823293

3) Data 3





a) Path Plot:

b) Execution Time: 58.00939106941223

1553, 1780, 1812, 1813, 1566, 1567, 1569, 1483, 1238, 996, 873, 995, 1237, 1236, 994, 993, 872, 1234, 1482, 1222, 982, 862, 1208, 1467, 1192, 844, 821, 815, 811, 1170, 1442, 1449, 1177, 116, 108, 2188, 1393, 1703, 1812, 1294, 2075, 2080, 2088, 2436, 2428, 2421, 2103, 2110, 2445, 2121, 2123, 2128, 2133, 2141, 2468, 2479, 2486, 2491, 2496, 2503, 2508, 2514, 2882, 3154, 3441, 3437, 3430, 2162, 2169, 2155, 2035, 2031, 2023, 2378, 2387, 2013, 1999, 1993, 1987, 1977, 1974, 2330, 2319, 2300, 2561, 3250, 3063, 2476, 3287, 3303, 3312, 3511, 2351, 2356, 2364, 1599, 1592, 1580, 1568, 1725, 1499, 856, 794, 601, 617, 880, 1016, 883, 1021, 888, 1277, 642, 633, 20, 21, 22, 23, 295, 303, 29, 319, 59, 587, 1570, 1781, 1815, 1814, 1942, 1943, 1944, 2315, 2314, 2313, 2312, 1940, 1941, 1939, 2311, 2310, 1937, 1927, 2573, 2705, 2950, 2956, 2957, 2709, 2576, 2577, 2710, 2958, 3205, 3282, 2937, 2694, 238, 1239, 1240, 2963, 3299, 3207, 2962, 2714, 2579, 1945, 2316, 2317, 2318, 1946, 1947, 1948, 1949, 1950, 1578, 1577, 1576, 1575, 1574, 1573, 1572, 1571, 997, 1241, 1242, 1243, 998, 999, 1244, 1245, 1000, 1001, 1246, 1247, 1002, 1003, 1248, 1249, 874, 1004, 1250, 1484, 1579, 1816, 1817, 1818, 1819, 622, 247, 627, 250, 249, 248, 1247, 1248, 1249, 623, 624, 625, 626, 1488, 1259, 879, 1013, 1012, 1258, 1257, 1011, 1010, 1256, 3558, 3559, 3310, 2583, 2722, 2974, 3562, 3510, 3314, 3313, 3561, 3560, 3311, 2975, 2723, 2584, 2724, 3210, 2976, 2977 2978, 2725, 1782, 882, 1262, 1590, 1591, 1263, 1264, 1265, 636, 1595, 1976, 1975, 2341, 2340, 2388, 2339, 1973, 1972, 1971, 1970, 1969, 2337, 2336, 2334, 2335, 1968, 1967, 1966, 1965, 1964, 1965, 1964, 1965, 1964, 1965, 1964, 1965, 1964, 1967, 1966, 1967, 1966, 1967, 19 1954, 1953, 2323, 2322, 1952, 1951, 2321, 2320, 3302, 3304, 3305, 3306, 3307, 2970, 2969, 2718, 2717, 2968, 2967, 2716, 2715, 2966, 2965, 2580, 2964, 3509, 3301, 3300, 3553, 3552, 3551, 3550, 3548, 3549, 3296, 3297, 3298, 3206, 2578, 2713, 2961, 2960, 2712, 2711, 2959, 1273, 886, 887, 1025, 889, 1026, 890, 1274, 1602, 1601, 1600, 643, 644, 275, 276, 277, 645, 646, 647, 1490, 1276, 1275, 1027, 2424, 2423, 2623, 2807, 3067, 1657, 3391, 3392, 3600, 3599, 3393, 3394, 3601, 3602, 3396, 3397, 3514, 3398, 3399, 3400, 3082, 2633, 2822, 2632, 2821, 2631, 2630, 3081, 3080, 2820, 2432, 2442, 2443, 739, 740, 741, 406, 405, 75, 404, 2458, 3414, 3415, 3416, 3520, 3611, 3612, 3417, 3418, 3613, 3614, 3419, 3109, 2645, 3110, 3242, 3111, 2845, 2846, 3112, 3113, 2847, 2848, 2646, 3114, 3243, 3425, 3424, 3423, 3422, 3420, 2467, 2469, 2470, 2471, 2472, 2149, 2148, 2147, 2146, 1852, 1851, 1699, 1700, 1506, 1396, 1127, 927, 1126, 1395, 1397, 1128, 928, 929, 1129, 1398, 1507, 1703, 1856, 1855, 2151, 2152, 2150, 2473, 2474, 2475, 2648, 2850, 3116, 3244, 3115, 2849, 2647, 3426, 3616, 3615, 3617, 3618, 3427, 3619, 1857, 1701, 1854, 1853, 1698, 2145, 2144, 2143, 2142, 2466, 2463, 2462, 3647, 341, 46, 45, 43, 339, 42, 41, 685, 686, 687, 688, 1060, 1318, 1319, 1634, 1633, 1315, 903, 683, 684, 1628, 1627, 674, 317, 3355, 3584, 3583, 3585, 3586, 3356, 3357, 3358, 3359, 818, 817, 510, 508, 509, 816, 814, 813, 812, 952, 1521, 1441, 1169, 1171, 1443, 1444, 953, 1172, 1173, 1445, 1446, 1174, 1175, 1447, 1448, 1176, 1178, 1450, 1451, 954, 955, 1179, 1452, 1451, 14 1453, 1180, 956, 822, 823, 523, 824, 825, 526, 154, 155, 156, 157, 160, 159, 158, 528, 831, 830, 829, 828, 782, 7826, 525, 153, 152, 151, 150, 147, 148, 149, 522, 521, 520, 519, 518, 517, 516, 146, 515, 514, 512, 511, 145, 144, 143, 142, 141, 140, 139, 138, 507, 506, 137, 504, 135, 134, 133, 132, 131, 130, 810, 809, 808, 949, 1167, 950, 1168, 951, 1440, 1740, 1739, 1738, 1520, 804, 805, 126, 496, 497, 498, 499, 127, 501, 500, 807, 806, 1166, 1439, 1519, 948, 1165, 1438, 1437, 1164, 947, 946, 1518, 1436, 1163, 1161, 1434, 1433, 1160, 1158, 1431, 1516, 1432, 1159, 944, 491, 492, 801, 802, 803, 495, 493, 124, 123, 122, 121, 120, 119, 118, 489, 807, 98, 1423, 1151, 1152, 1444, 1723, 1721, 1318, 2655, 2184, 2183, 2182, 2181, 2180, 2179, 2177, 2494, 2495, 3133, 3444, 3433, 3248, 2866, 3125, 3440, 3442, 3134, 2654, 2865, 2864, 3133, 3132, 2863, 2653, 2862, 3131, 3626, 3438, 3439, 2860, 3129, 3128, 2652, 3127, 3436, 3625, 3624, 3622, 3623, 3433, 3434, 3435, 3247, 2651, 2859, 3126, 3124, 3125, 2857, 2649, 2854, 3245, 3120, 3121, 2855, 2650, 2856, 3122, 3246, 3432, 3431, 3621, 3620, 3521, 3429, 3428, 2851, 3117, 3119, 2853, 2478, 2477, 2153, 2154, 2156, 2157, 2158, 2480, 2481, 2159, 2160, 2161, 2482, 2483, 2484, 2163, 2485, 2164, 2165, 2166, 2167, 2168, 2487, 2488, 2170, 2171, 2173, 2172, 2489, 2490, 2492, 2174, 2175, 2176, 1866, 1867, 1718, 1720, 105, 104, 468, 469, 784, 783, 937, 1512, 1414, 1143, 1142, 1413, 1412, 1414, 781, 782, 466, 465, 103, 464, 463, 780, 777, 462, 101, 461, 776, 775, 773, 774, 459, 458, 457, 100, 98, 455, 454, 453, 452, 771, 770, 451, 450, 449, 769, 931, 1508, 1401, 1132, 1130, 1399, 1704, 1858, 1859, 1705, 1402, 1133, 932, 933, 1134, 1403, 1509, 1860, 1861, 1706, 1707, 1510, 1511, 1406, 1136, 936, 935, 1135, 1405, 1708, 1862, 1863, 1709, 1710, 1408, 1138, 1425, 1724, 1726, 1727, 1868, 1869, 1728, 1729, 1515, 1730, 1731, 1870, 1871, 1732, 1733, 1734, 1872, 1873, 1735, 1874, 1876, 1736, 1737, 1798, 3459, 3157, 2883, 3156, 3155, 3528, 3456, 3457, 3153 2881, 2880, 3152, 3151, 2660, 2879, 2878, 3150, 3454, 3527, 3526, 3525, 3453, 3524, 3523, 3522, 3452, 3451, 2875, 3147, 3146, 2874, 2659, 2873, 3145, 3449, 3630, 3629, 3448, 3447, 3446, 3628, 3627, 3628, 3629, 3 3445, 3249, 3139, 2867, 2656, 2657, 2868, 3140, 3141, 2869, 2870, 3142, 3251, 3144, 2872, 2658, 2500, 2499, 2498, 2497, 2185, 2186, 2187, 2189, 2190, 2193, 2192, 2191, 2501, 2502, 2504, 2505, 2194 2195, 2196, 2197, 2198, 2200, 2201, 2202, 2203, 2204, 2507, 2205, 2206, 2208, 2207, 2505, 2506, 2508, 2507, 2505, 2506, 2508, 2507, 2505, 2506, 2508, 2507, 2505, 2506, 2508, 2507, 2505, 2506, 2508, 2507, 2505, 2506, 2508, 2507, 2508, 3471, 3635, 3636, 3472, 3473, 3637, 3638, 3474, 3475, 3476, 3639, 3640, 3477, 3478, 3479, 3641, 3642, 3643, 3644, 3480, 3481, 3482, 2910, 2673, 2542, 2539, 2540, 2541, 2252, 2251, 2250, 2538, 2536, 2537, 2248, 2247, 2246, 2245, 2244, 2533, 2534, 2535, 2669, 2902, 3177, 3256, 3178, 2903, 2904, 3179, 3257, 3180, 2905, 2670, 2672, 2907, 3259, 3182, 3183, 2908, 2909, 3184, 3260, 3483, 3484, 3485 3185, 2675, 2911, 2674, 2543, 2544, 2253, 2254, 2546, 2547, 2548, 1473, 1198, 964, 1199, 1474, 1526, 1775, 196, 570, 565, 564, 966, 1202, 1201, 965, 1200, 1476, 1777, 1528, 1529, 1530, 1531, 1779, 1775, 17 3163, 2013, 2911, 2014, 2015, 2014, 2015, 2014, 2015, 2305, 868, 1232, 1561, 1560, 1559, 1558, 1557, 1556, 1555, 1554, 1552, 1551, 1550, 1549, 1223, 1224, 983, 984, 1225, 1226, 1227, 985, 593, 217, 216, 215, 213, 4, 3, 2, 1, 212, 211, 583, 584, 585, 586, 589, 590, 591, 592, 853, 866, 1228, 986, 1229, 987, 988, 1230, 1231, 989, 867, 595, 594, 218, 220, 221, 222, 223, 224, 225, 226, 596, 597, 598, 1235, 871, 992, 870, 991, 869, 990, 1233, 1562, 1563, 1564 . 1965, 1936, 1936, 2309, 2308, 2307, 1935, 1934, 1933, 1932, 1931, 1930, 1929, 1928, 1926, 1925, 1924, 1923, 1922, 1921 3192, 3190, 2913, 2677, 2551, 2552, 2265, 2266, 2267, 2268, 1888, 1889, 2270, 2269, 2560, 2559, 2558, 2557, 2555, 2554, 2553, 2264, 2263, 2262, 2260, 2259, 2258, 2257, 2256, 2550, 2549, 2255, 1882, 1883, 1769, 843, 846, 845, 169, 170, 552, 551, 550, 549, 168, 167, 548, 842, 841, 840, 839, 163, 164, 165, 166, 547, 545, 544, 543, 542, 541, 540, 539, 538, 537, 536, 535, 837, 838, 957, 1463, 1462, 1188, 167, 1882, 1883, 1884, 1187, 1461, 1764, 1765, 1766, 1464, 1189, 958, 959, 1190, 1465, 1466, 1191, 960, 1193, 1468, 1523, 1767, 1880, 1881, 1768, 1524, 1469, 1194, 961, 962, 1195, 1470, 1471, 1196, 963, 1197, 1472, 1525, 1771, 1884, 1885, 1772, 173, 1886, 1887, 1774, 1527, 1475, 850, 849, 848, 847, 553, 554, 171, 172, 173, 174, 175, 176, 557, 558, 559, 851, 852, 179, 178, 560, 561, 562, 563, 180, 181, 182, 183, 184, 186, 187, 188, 189, 190, 568, 569, 195, 194, 193, 192, 191, 567, 566, 1535, 1534, 1533, 1532, 1203, 967, 968, 1204, 1205, 857, 969, 1206, 1207, 970, 858, 859, 971, 1209, 1210, 972, 860, 973, 1211, 1477, 1536,1804, 1805, 1537, 1478, 861, 974, 1212, 1213, 975, 863, 1214, 976, 977, 1215, 1479, 1216, 978, 864, 203, 202, 200, 199, 198, 571, 572, 573, 574, 575, 576, 204, 207, 206, 577, 578, 579, 580, 581, 582, 208, 209, 819, 820, 1749, 1748, 1747, 1746, 1745, 1743, 1742, 1879, 1878, 2520, 2521, 2522, 2523, 2230, 2231, 2233, 2234, 2235, 2526, 2527, 2528, 2236, 2238, 2239, 2240, 2241, 2530, 2529, 2531, 2532, 2242, 2532, 25 2243, 2668, 2900, 3175, 3255, 3174, 2899, 2901, 3176, 3470, 3469, 3634, 3633, 3648, 1750, 1454, 1181, 1182, 1455, 1753, 1754, 1755, 1756, 1757, 1758, 1759, 1522, 1457, 1184, 1183, 1456, 1752, 1760 1761, 1762, 1763, 1459, 1186, 1460, 1458, 1185, 833, 834, 835, 836, 532, 533, 162, 531, 161, 530, 832, 1801, 3186, 3187, 3488, 3489, 3490, 3491, 3189, 2676, 2912, 3188, 3487, 3492, 3493, 3494, 3495, 3497, 3493496, 3498, 3499, 3284, 3543, 3544, 3285, 3286, 3545, 3546, 3288, 3289, 3290, 3291, 3292, 3293, 3294, 3295, 3508, 3547, 2955, 2304, 2303, 2302, 2301, 228, 229, 0]

d) Cost: 12676.959763

(2) Implementation and Comparison

<2022.10.10 14:59> There is a bug in the improved source code, so the genetic algorithm didn't work properly. Until implementing the OX and PMX crossover function, it worked well, but after implementing the mutation function, the algorithm didn't work well.

<2022.10.10 19:19> The implementations of mutation functions were correct, but the PMX crossover function had an error in the previously submitted source code. I've found some exception cases during the PMX crossover steps: chain-mapped cities and cycle-mapped cities. These two exceptional cases were handled in the latest version, and the PMX crossover is made properly. Also, the replacement idea and source code are modified.

The final cost of the improved GA source code decreased on data 1 and data 3. The sample source code doesn't have a time limit and iterates 100000 times, which ran until 83 and 257 seconds on data 2 and 3, but the improved source code has a time limit of 60 seconds. So, for data 2 and 3, the improved source code ran out of time and stopped running the GA function, in which the iteration numbers were roughly 17000 and 3000. Regarding this, although two source codes have some random points, we can consider that the improved GA algorithm has been improved with the implementation of various crossover and mutation methods and replacement with the tournament selection algorithm.