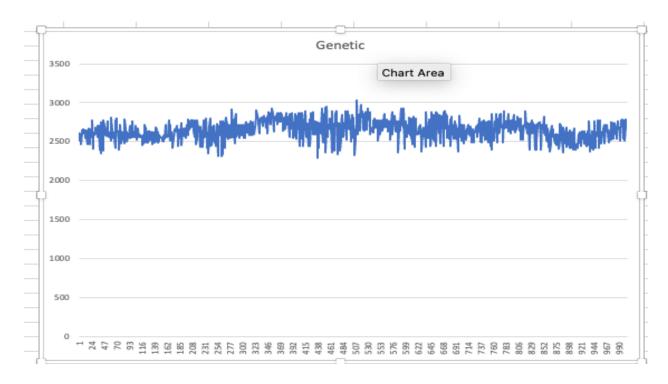
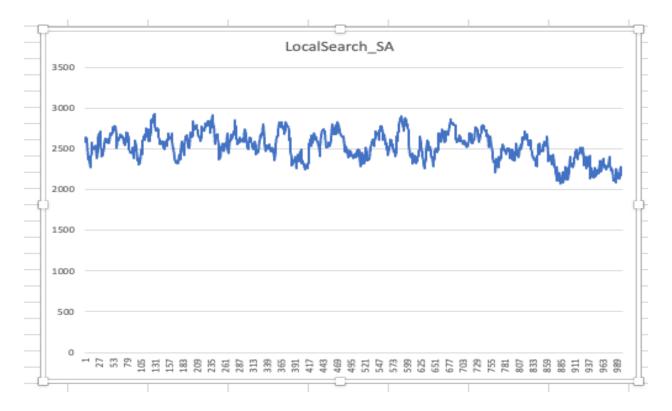
Output:

File1

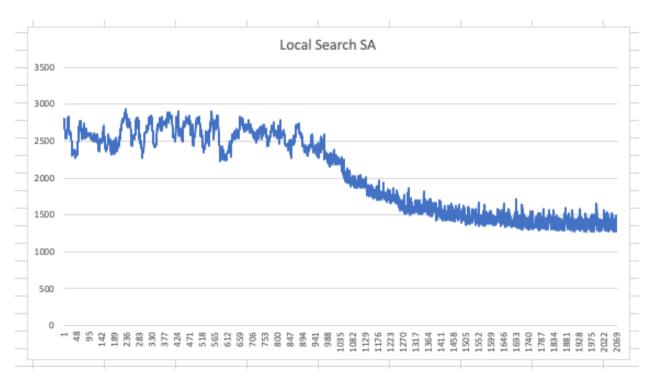
Genetic-file 1



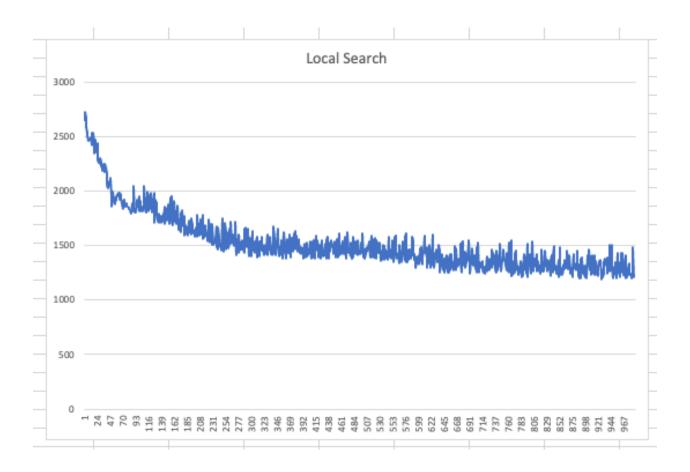
Local search Simulated Annealing - File 1 Case: while (temp > MIN_TEMP && iterations <= maxIterations) {



Local search Simulated Annealing – file 1 Case: while (temp > MIN_TEMP) {



Local Search-file 1



File output 2

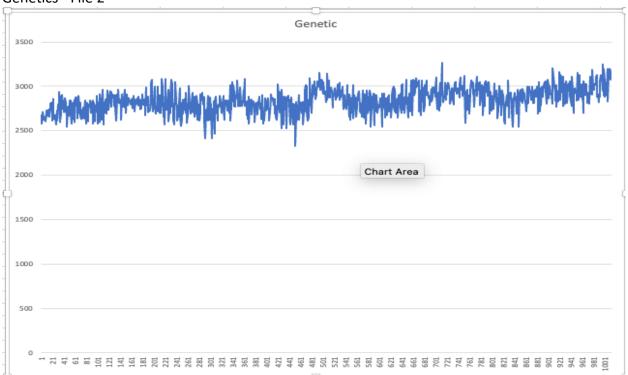
```
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
                                                                                                double temperature = 1000000; // Initial temperature
double coolingRate = 0.99; // Cooling rate
                                                                                                double MAX_TEMP = 10000;
double MIN_TEMP = 0.00001;
double temp = MAX_TEMP;
                                                                                                 Random random = new Random();
                                                                                                //while (temp > MIN_TEMP && iterations <= maxIterations) {
while (temp > MIN_TEMP ) {
   int ranIndex1;
   int ranIndex2;
   // Generate the first random index
          512
513
514
                                                                                                                                                    {
  // Generate the second random index
  ranIndex1 = random.nextInt(path.size());
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 Problems @ Javadoc 🚇 Declaration 🖳 Console 🗶
   Driver_3_4 [Java Application] /Users/xzyloneagora/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.aarch64_17.0.6.v20230204-1729/jre/bin/java (Dec 8, 2023, 10:52:11 PM) [pid: 98737]
Driver_3_4
Grabbing input...
Done
 Using Input File 1
 Local Search Algorithm

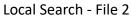
Shortest Path: [[(35,75), (11,32), (6,20), (3,23), (8,14), (4,4), (30,26), (45,22), (47,31), (62,22), (64,14), (71,13), (76,28), (67,46), (56,93), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63,46), (63
     [(35,75), (11,32), (6,20), (3,23), (8,14), (4,4), (30,26), (45,22), (47,31), (62,22), (64,14), (71,13), (76,28), (67,46), (56,93), (63,95), (71,95), (12,32), (12,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32), (13,32),
Local Search Algorithm with Simulated Annealing
Shortest Path: [[(62,22), (89,98), (95,96), (90,79), (71,59), (62,28), (86,16), (81,23), (76,28), (67,46), (59,49), (47,31), (64,14), (53,4), (41,20),
Shortest Path Distance: 1136.779425197609
[(62,22), (89,98), (95,96), (99,79), (71,59), (62,28), (86,16), (81,23), (76,28), (67,46), (59,49), (47,31), (64,14), (53,4), (41,20), (4,4), (8,14),
Genetic Algorithm

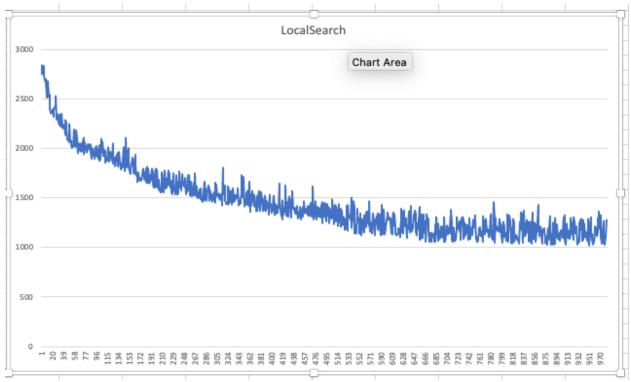
Shortest Path: [[(62,28), (38,90), (35,75), (10,96), (23,38), (41,20), (51,78), (76,28), (86,16), (11,32), (35,97), (4,4), (81,23), (71,59), (62,21), (86,28), (38,90), (35,75), (10,96), (23,38), (41,20), (51,78), (76,28), (86,16), (11,32), (35,97), (4,4), (81,23), (71,59), (62,21), (62,28), (38,90), (35,75), (10,96), (23,38), (41,20), (51,78), (76,28), (86,16), (11,32), (35,97), (4,4), (81,23), (71,59), (62,21), (61,7), (18,42), (76,28), (86,16), (11,32), (35,97), (4,4), (81,23), (71,59), (62,21), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81,27), (81
```

Graphs:

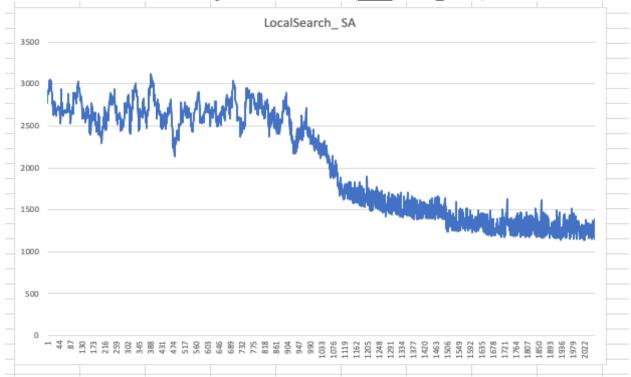
Genetics - File 2





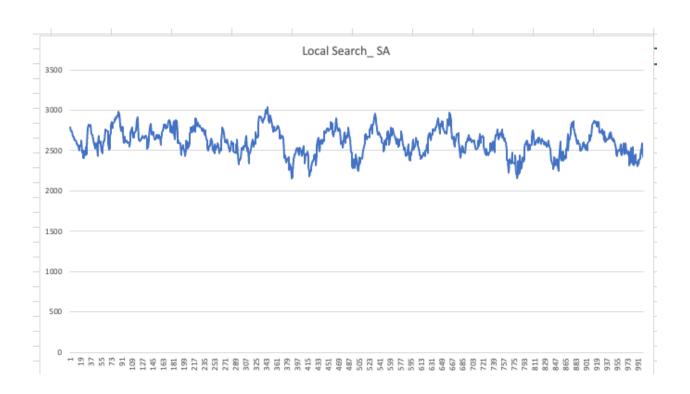




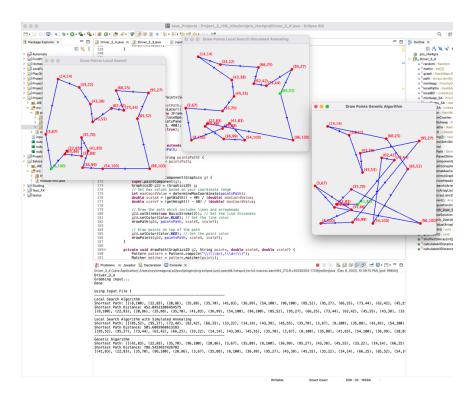


Local search Simulated Annealing - File 2 Case: while (temp > MIN_TEMP && iterations <= maxIterations) {

```
501
502
503
504
505
                                                   double MAX_TEMP = 10000;
double MIN_TEMP = 0.00001;
double temp = MAX_TEMP;
                                                   Random random = new Random();
                                                  while (temp > MIN_TEMP && iterations <= maxIterations) {
    while (temp > MIN_TEMP ) {
        int ranIndex1;
        int ranIndex2;
    }
}
                   //
                                                                 // Generate the first random index
do {
                                                               do {
    // Generate the second random index
    ranIndex1 = random.nextInt(path.size());
    ranIndex2 = random.nextInt(path.size());
} while (ranIndex2 == ranIndex1); // Repeat if the indices are the same
                                                              Svstem.out.println("RANDOM VALUES: ran1- " + ranIndex1 + " ran2- " + ranIndex2):
   518 //
                                                                                                                                                                                                                                                                                                                                                                                               Problems @ Javadoc Q Declaration Q Console X
Driver_3_4 [Java Application] [Users/xzyloneagora/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.aarch64_17.0.6.v20230204-1729/jre/bin/java (Dec 8, 2023, 11:01:01 PM) [pid: 98804]
Using Input File 1
Local Search Algorithm
Shortest Path: [[(4,4), (41,20), (53,4), (45,22), (30,26), (12,38), (14,49), (35,75), (38,90), (39,90), (10,96), (5,44), (23,38), (46,86), (63,95), (75)
Shortest Path Distance: 1146.7057482616967
[(4,4), (41,20), (53,4), (45,22), (30,26), (12,38), (14,49), (35,75), (38,90), (39,90), (10,96), (5,44), (23,38), (46,86), (63,95), (71,95), (95,96),
Local Search Algorithm with Simulated Annealing
Shortest Path: [[(4,4), (95,59), (95,96), (63,95), (35,97), (16,86), (71,95), (59,49), (71,13), (41,20), (81,23), (51,78), (62,22), (23,38), (43,35),
Shortest Path Distance: 2149,95600928456
[(4,4), (95,59), (95,96), (63,95), (35,97), (16,86), (71,95), (59,49), (71,13), (41,20), (81,23), (51,78), (62,22), (23,38), (43,35), (62,28), (52,95)
Genetic Algorithm
Shortest Path: [[(3,23), (18,42), (64,14), (6,20), (63,95), (5,44), (30,26), (89,98), (95,96), (46,72), (8,14), (62,28), (61,7), (95,59), (81,23), (79
Shortest Path Distance: 2176.46109752605
[(3,23), (18,42), (64,14), (6,20), (63,95), (5,44), (30,26), (89,98), (95,96), (46,72), (8,14), (62,28), (61,7), (95,59), (81,23), (79,48), (67,46), (62,28), (64,14), (62,28), (61,7), (95,59), (81,23), (79,48), (67,46), (62,28), (61,7), (95,59), (81,23), (79,48), (67,46), (62,28), (61,7), (95,59), (81,23), (79,48), (67,46), (62,28), (61,7), (95,59), (81,23), (79,48), (67,46), (62,28), (61,7), (95,59), (81,23), (79,48), (62,48), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28), (82,28),
```



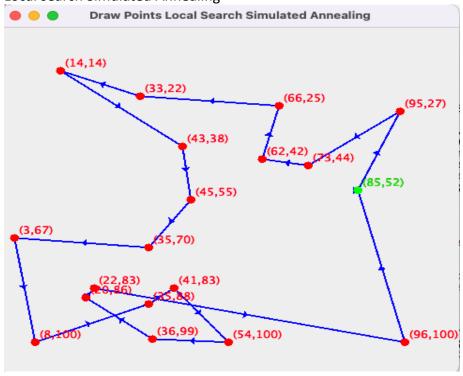
Extra credit:



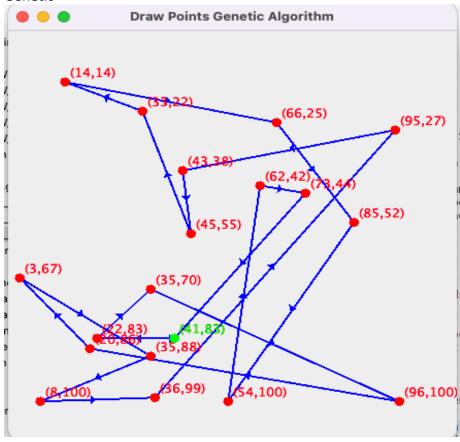
Local Search



Local search Simulated Annealing



Genetic



Conclusion:

Algorithm Speed Comparison

In my experimentation, Local Search consistently emerged as the fastest algorithm. This was followed by the variant that integrated Local Search with Simulated Annealing, and finally, the Genetic Algorithm trailed as the slowest among the three. This speed hierarchy is crucial, particularly in scenarios where computational efficiency is a priority. The stark contrast in execution speed between Local Search and the Genetic Algorithm highlighted the inherent efficiency of Local Search for quick computations.

Modified Iteration Approach in Local Search Simulated Annealing

A notable observation was made when I modified the iteration process of the Local Search Simulated Annealing algorithm. By iterating based solely on the temperature condition (temp > MIN_TEMP) instead of the combined criteria of temperature and maximum iterations (temp > MIN_TEMP && iterations <= maxIterations), I noticed that the algorithm could potentially find shorter paths. However, this adjustment led to longer iteration times for the Simulated Annealing, indicating a trade-off between the speed of execution and the quality of the solution.

Experiment Setup and Graphical Representation

For a comprehensive comparison, I ran the Local Search and Simulated Annealing algorithms for 1000 iterations each. In contrast, for the Genetic Algorithm, I employed a setup of 10 populations across 100 generations. This setup aimed to balance the operational differences among the algorithms. In presenting the results, I chose to graph them using 20 data points to ensure clarity and ease of interpretation. It's worth noting that the graphical representation could handle up to 50 points, demonstrating the scalability of this visualization approach in handling more complex datasets.

```
For graphics
Might need inside a module-info.java
module Project_2_HW_4Gra {
    requires java.desktop;
}
```

```
> Project_2_HW_2
      Project 2 HW 3
                                                                                                                                                                                     /(Globals
static Random random = new Random();
static Random random = new Random();
static HashMap<String, Double> graph = new HashMap<>();
static ArrayList<String> path = new ArrayList<>();
      Project 2 HW 4
        Project 2 HW 4Gra

    Project_2_HW_4Gra
    M_AEE System Library [JavaSE-17]
    M_BE System Library [JavaSE-17]
    M_Broc_Hw4gra
    M
                                                                                                                                                                                  // Min-Heap

Static PriorityQueue-Obuble minHeap = new PriorityQueue<>();

// Found Paths

static HashMap=Obuble, Set<String> localPaths = new HashMap<>();

// BestDistance at iteration

static LinkedListStatic
                    input2.txt

    inputz.txt
    output_Genetic.csv
    output_localSearch_SA.csv
    output_localSearch.csv

                                                                                                                                                                                     // Local search Simulated Annealing variables
                                                                                                                                                                                  // Local Search Dimulated Annua, Albu Valendres
// Min-Heap
static PriorityQueue-Double> minHeap_SA = new PriorityQueue<);
// Found Paths
static HashMap-Double, Set-String>> localPaths_SA = new HashMap
// BestDistance at iteration
static LinkedList-Double> localBD_SA = new LinkedList
();
> Project_Automata
> Salazar_Andrew_Project1
> Studing
> Studing
> Fest_FX
> Fest_FX
                                                                                                                                                                                   // Genetic variables
//Generations
static HashMap<Integer, Set<ArrayList<String>>> population = new HashMap<();
static int generationCounter = 0;
static PriorityQueue-Double> geneticMinHeap = new PriorityQueue<();
// Found Paths
static HashMap<Double, Set<String>> geneticPaths = new HashMap<();
// BestOistance at iteration
static LinkedList<Double> geneticBD = new LinkedList<();</pre>
                                                                                                                                                                                                     System.out.println("Grabbing input... ");
grabInput();
                                                                                                                                             Problems @ Javadoc Q Declaration Console X - terminated-Driver_3_4 [Java Application] /Users/xzyloneagora/p2/pool/plugins/org-eclipse.justj.openjdk.hotspot.jre.full.macosx.aarch64_17.0.6.v20230204-1729/jre/binjava (Dec 8, 2023, 11:12:44 PM – 11:29:57 F Grabbing input... Done
                                                                                                                                             Using Input File 1
                                                                                                                                             Local Search Algorithm
Shortest Path: [(36,99), (54,100), (96,100), (85,86), (42,89), (41,83), (48,85), (76,63), (85,52), (73,42), (73,44), (45,55), (25,48), (28,36), (33,22), (1 Shortest Path Distance: 1063.4647211249433
[(36,99), (54,100), (96,100), (85,86), (42,89), (41,83), (48,85), (76,63), (85,52), (73,42), (73,44), (45,55), (25,48), (28,36), (33,22), (14,37), (7,47), (
                                                                                                                                             Local Search Algorithm with Simulated Annealing
Shortest Path: [[(45,55), (96,100), (85,52), (96,44), (85,86), (54,100), (48,85), (35,70), (28,36), (30,11), (28,4), (43,38), (73,42), (80,23), (95,27), (66
Shortest Path Distance: [268,319840457768
[(45,55), (96,100), (85,52), (96,44), (85,86), (54,100), (48,85), (35,70), (28,36), (30,11), (28,4), (43,38), (73,42), (80,23), (95,27), (66,25), (14,14), (
                                                                                                                                             Genetic Algorithm

Shortest Path: [[(56,43), (6,71), (54,100), (62,42), (63,19), (3,67), (5,90), (48,85), (7,47), (73,44), (22,83), (85,86), (96,44), (75,33), (95,27), (80,23)

Shortest Path Distance: 2017.4063554564548

[(56,43), (6,71), (54,100), (62,42), (63,19), (3,67), (5,90), (48,85), (7,47), (73,44), (22,83), (85,86), (96,44), (75,33), (95,27), (80,23), (55,50), (48,5)
```

pro_Hw4gra.Driver_3_4.java - Project_2_HW_4Gra/src