COIS 4550H

Assignment 3

WInter 2025

Bachelor Of Computer Science, Trent University

COIS-4550H: Artificial Intelligence

Professor: Sheikh Ahmed Munieb

15th March, 2025

Dikshith Reddy Macherla - Student Id: 0789055

Roman Bamrah - Student Id: 0705310

Assignment 3

Question 1

{5,12,8,15,3,18}

S.N	Initial pop	x-value	Fitness f(x) = x^2+2	Prob f(x)	% Prob	Expect count	Actual Count
1	00110	6	38	0.044	4.4%	0.26	0
2	01111	15	227	0.261	26.1%	1.56	2
3	01001	9	83	0.096	9.6%	0.57	1
4	01100	12	146	0.168	16.8%	1.01	2
5	00011	3	11	0.013	1.3%	0.08	0
6	10011	19	363	0.418	41.8%	2.5	2
Sum			868	1		6	
Avg			144.67	0.651		1	
Max			363	0.418		2.5	

Mating Pool {19,19,15,15,12,9}

Parent	Offspring
10011 x 10011	10011 x 10011
01111 x 01100	01110 x 01101
01100 x 01001	01100 x 01000

Offspring = {10011,10011,01110,01101,01100,01000}

Mutation with 13.
Original: 01101
Mutation: 01001

Binary	x-value	Fitness f(x)
10011	19	363
10011	19	363
01110	14	198
01001	9	83
01100	12	146

01000	8	66

Max Fitness = 363

Question 2:

{21,25,28,30}

S.N	Int Pop	x-val	Fitness f(x) = x^2+2	Prob f(x)	% Prob	Expect count	Actual Count
1	10101	21	443	0.161	16.1%	0.64	0
2	11001	25	627	0.227	22.7%	0.91	1
3	11100	28	786	0.285	28.5%	1.14	1
4	11110	30	902	0.327	32.7%	1.31	2
Sum			2758			4	
Avg			689.5			1	
Max			902			1.31	

Mating Pool {30,30,28,25}

Parent	Offspring
11110 x 11100	11100 x 11110
11100 x 11001	11000 x 11101

Offspring = {11100,11110,11000,11101}

Mutation With 17.
Original: 11000
Mutation: 10000

Binary	x-value	Fitness f(x)
11100	28	786
11110	30	902
10000	16	258
11101	29	843

Max Fitness = 902

Question 3:

Analysis of Simulated Annealing Results for Different Temperatures

Introduction:

Simulated Annealing is an optimization technique that balances exploration and exploitation through the gradual cooling of a system. In this experiment, we evaluated the performance of the algorithm for the Traveling Salesman Problem (TSP) with different initial temperatures (1, 100, and 1000) while keeping the population size fixed at 20. Below is an analysis of the results obtained.

a. Analysis of Temperature = 1 and Population Size = 20

I have run the Simulated Annealing algorithm with Temperature = 1 and Population Size = 20. Here's what I observed:

Results:

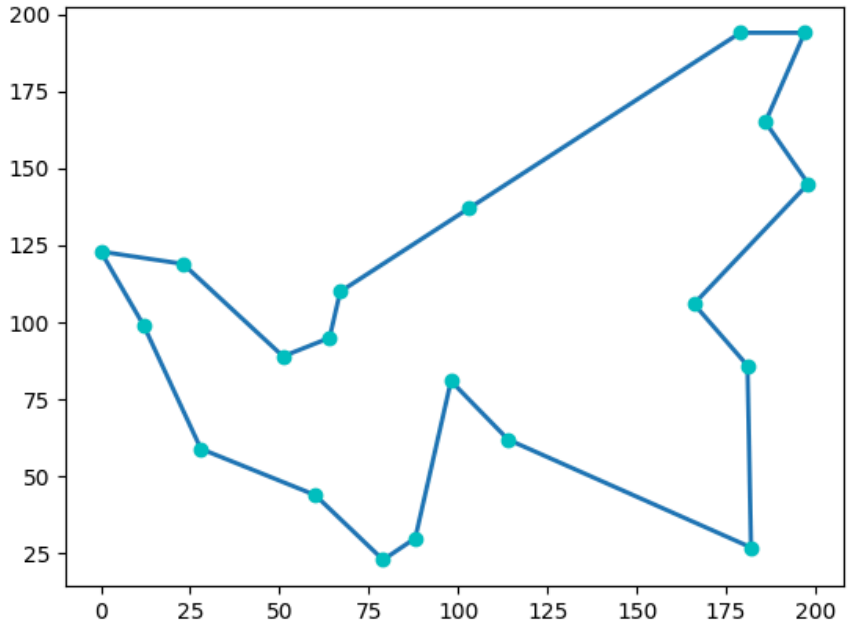
Initial weight: 977.897220116032

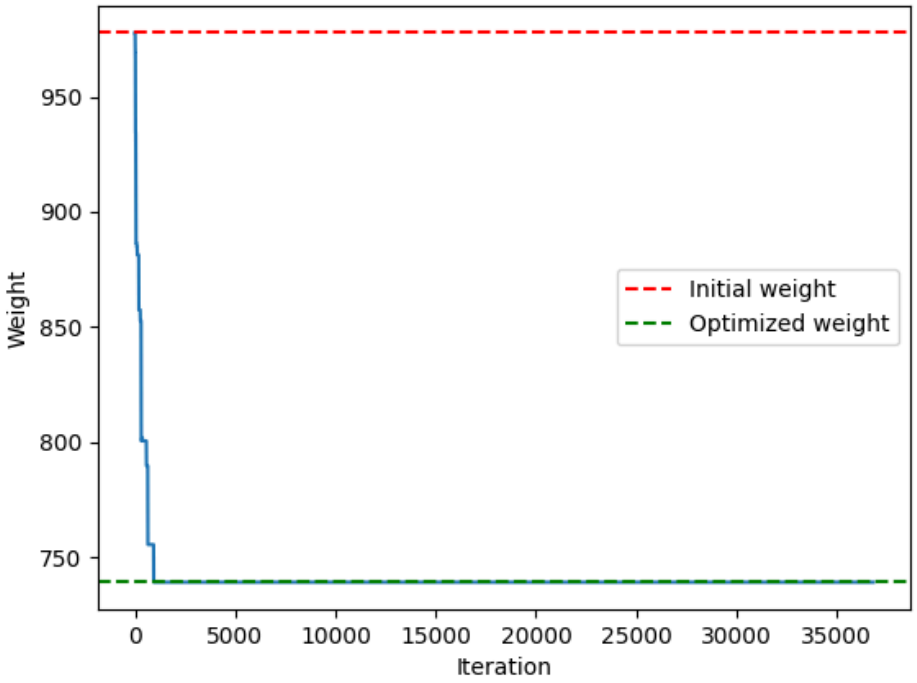
Minimum weight: 739.059155936158

Improvement: 24.42 %

Graph Observations

- The solution improved initially but quickly converged to a local minimum.
- Since the temperature was too low, the algorithm did not accept enough worse solutions, limiting exploration.





b. <u>Analysis of Temperature = 100 and Population Size = 20</u>

I have run the Simulated Annealing algorithm with Temperature = 100 and Population Size = 20. Here's what I observed:

Results:

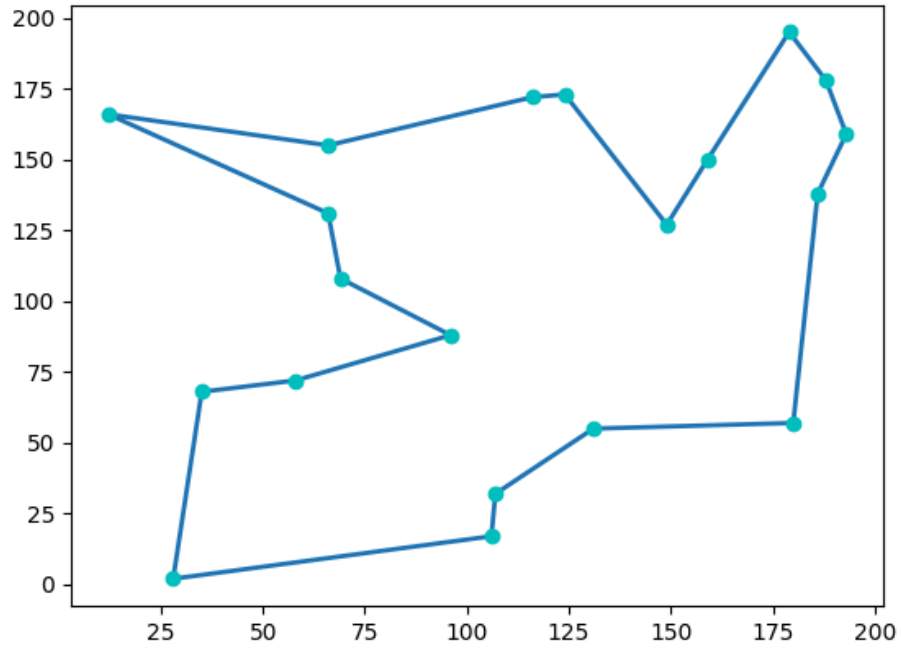
Initial weight: 1128.275086343094

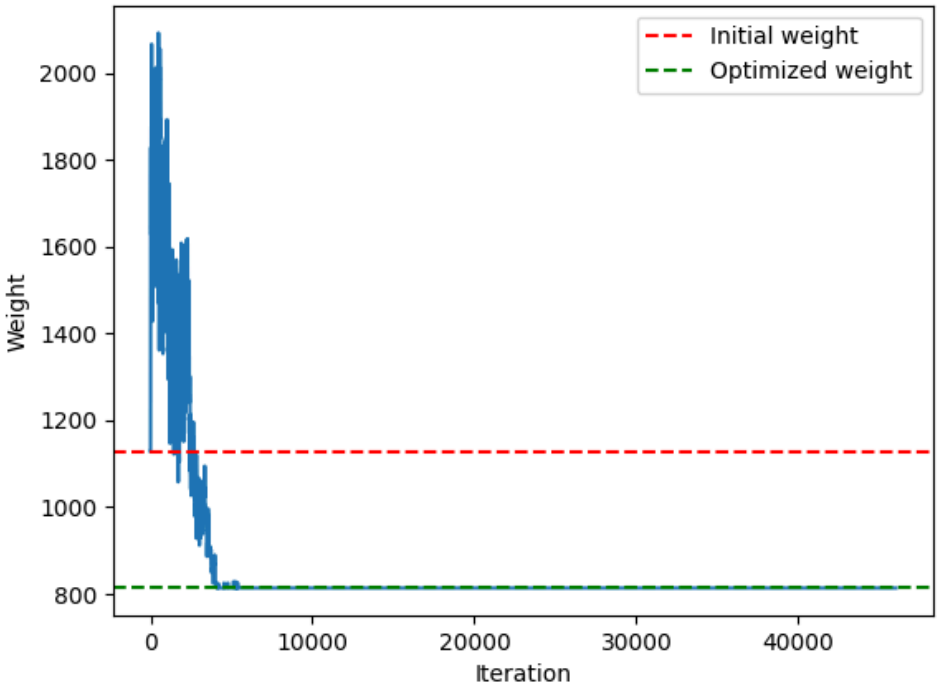
Minimum weight: 813.7384143883382

Improvement: 27.88 %

Graph Observations

- The algorithm explored a wider range of solutions before converging.
- The learning curve shows gradual improvement, indicating a good balance between exploration and exploitation.





c. Analysis of Temperature = 1000 and Population Size = 20

I have run the Simulated Annealing algorithm with Temperature = 1000 and Population Size = 20. Here's what I observed:

Results:

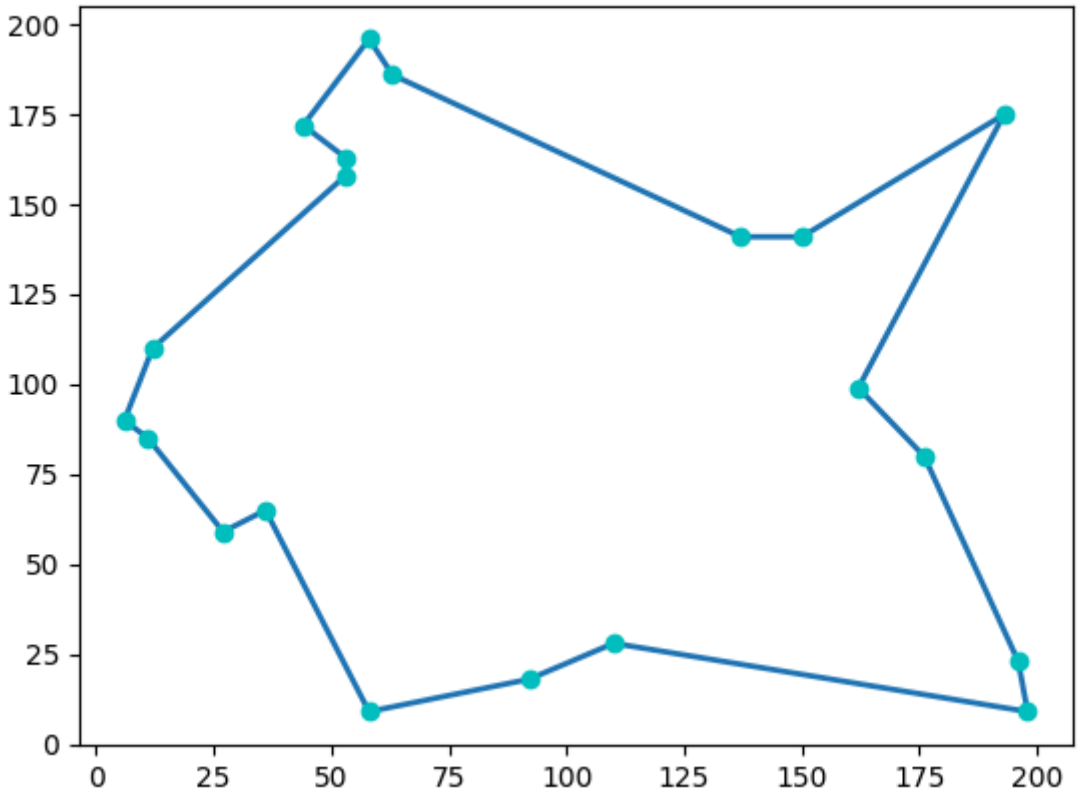
Initial weight: 926.4829684130328

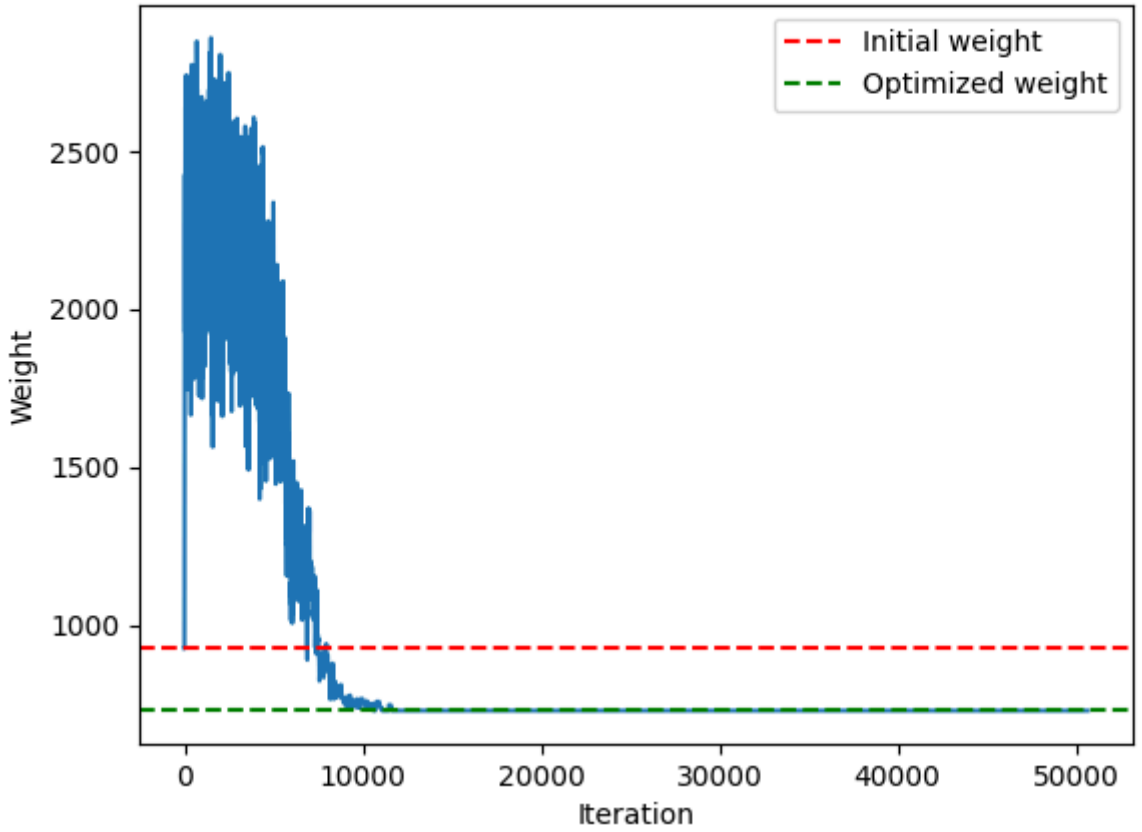
Minimum weight: 729.7288567351541

Improvement: 21.240000000000002 %

Graph Observations

- The algorithm explored excessively, frequently accepting worse solutions.
- The initial weight was better due to the broad search, but the improvement was minimal.





Difference Observed with Different Temperatures and Best Results

In this experiment, we tested Simulated Annealing for the Traveling Salesman Problem (TSP) using three different initial temperatures (1, 100, and 1000) while keeping the population size fixed at 20. The goal was to observe the impact of temperature on optimization performance and determine the best setting.

Observations and Differences

Temperature	Initial Weight	Optimized Weight	Improvement (%)
1	977.90	739.06	24.42%
100	1128.27	813.74	27.88%
1000	926.48	729.73	21.24%

1. **Low Temperature (1)**
    - Good improvement (24.42%), but it gets stuck in a local minimum.
    - The search does not explore enough alternative paths.

- Final solution is decent, but there might be better routes that were missed.

2. **Moderate Temperature (100)**
   - Best improvement (27.88%).
   - The algorithm accepts bad moves occasionally, helping it escape local minima.
   - Balanced exploration and convergence, leading to a better solution overall.
3. **High Temperature (1000)**
   - Least improvement (21.24%), indicating that it explored too much.
   - While it searched a broader solution space, it spent too much time exploring and did not converge effectively.
   - The final optimized solution was good but not significantly better.

Which Temperature Gave the Best Results?

Temperature = 100 provided the best improvement (27.88%).

It explored sufficiently but still converged effectively.

The balance between exploration (searching for better solutions) and exploitation (refining a good solution) was optimal. Temperature = 1 was too restrictive, causing it to settle into a local minimum. Temperature = 1000 was too high, leading to excessive exploration without optimal convergence.

For optimal results in this scenario, Temperature = 100 is ideal.

If needed, tuning the cooling rate (α) and stopping conditions could further refine the results. In general, moderate temperatures (between 100-500) tend to give the best balance.

Question 4:

Analysis of Simulated Annealing Results with Population = 50

After repeating the experiment with a larger population (50 nodes) and the same temperature values (1, 100, and 1000), here are the key findings:

Results Summary (Population = 50)

Temperature	Initial Weight	Optimized Weight	Improvement (%)
1	1397.25	1228.07	12.11%
100	1439.78	1199.36	16.7%
1000	1362.55	1200.96	11.86%

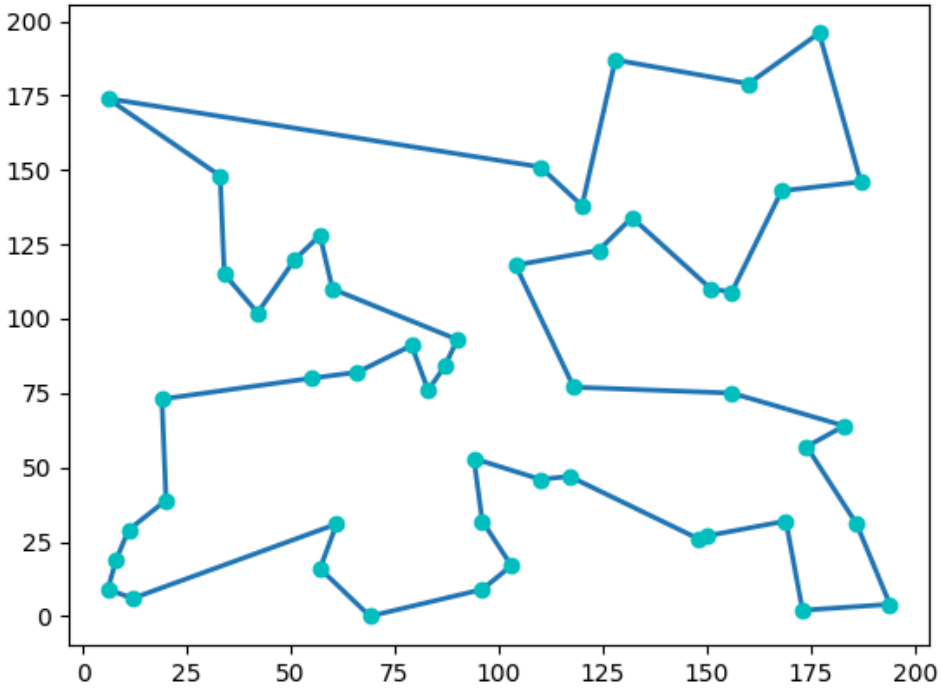
Observations and Differences Compared to Population = 20

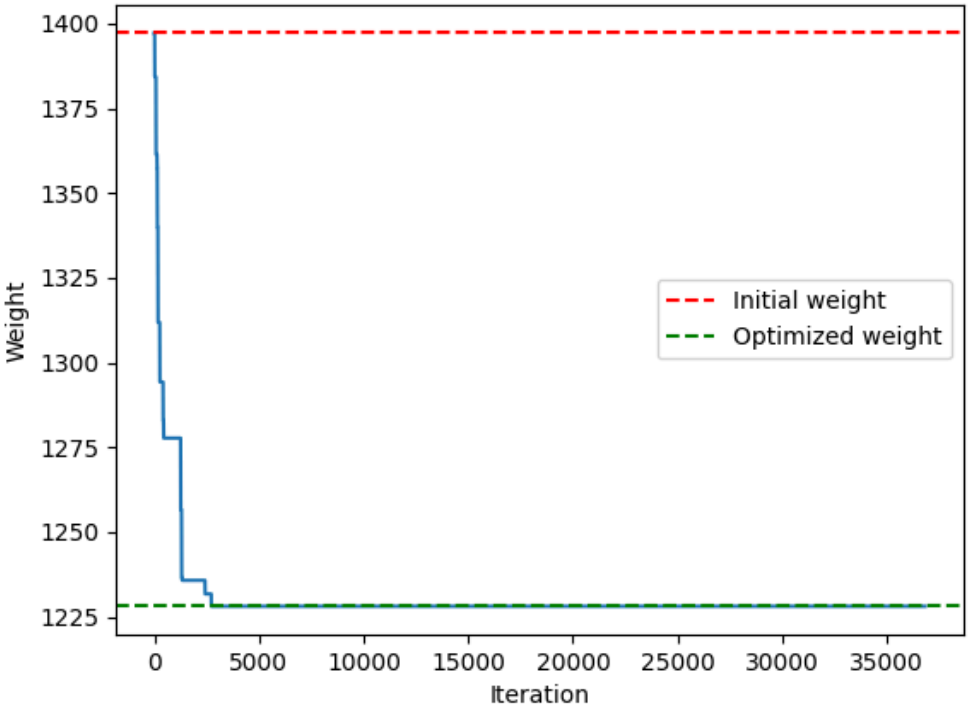
Observation	Population = 20	Population = 50
Best Temperature	100 (27.88%)	100 (16.7%)
Worst Performance	1000 (21.24%)	1000 (11.86%)
Exploration Ability	Moderate is Best	Moderate is Still Best
Improvement Trend	Higher % overall	Lower improvement %

Observations Based on Different Temperatures

1. Low Temperature (1)

- The improvement percentage dropped from 24.42% (Pop=20) to 12.11% (Pop=50).
- A larger problem size required more exploration, but Temp = 1 restricted it.
- The algorithm quickly converged to a local minimum without exploring many alternatives.
- Still provides some improvement, but higher temperatures perform better.

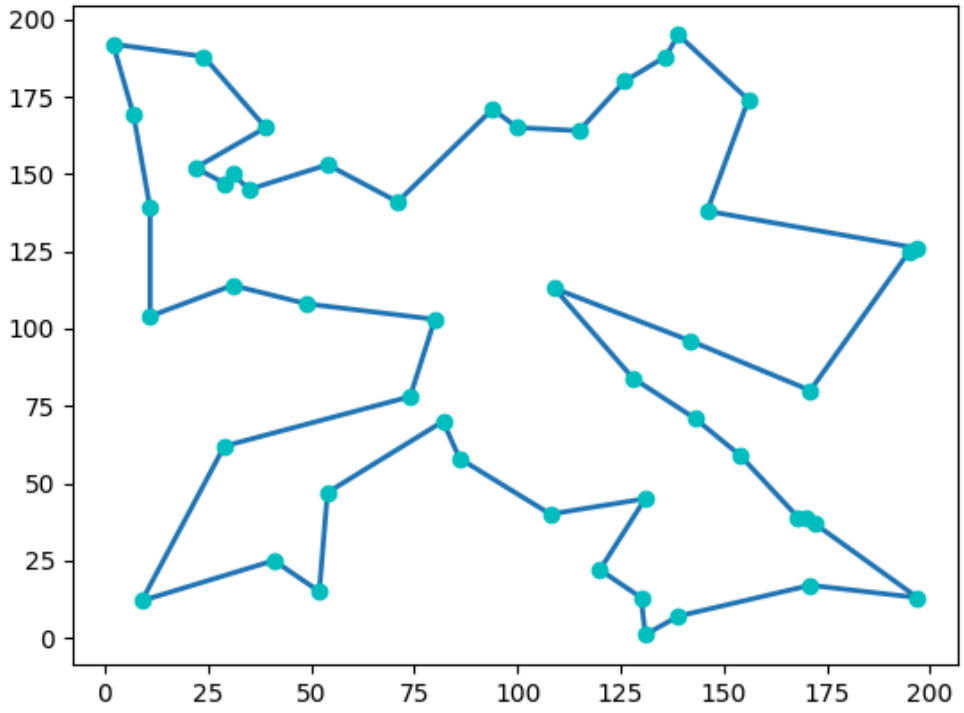


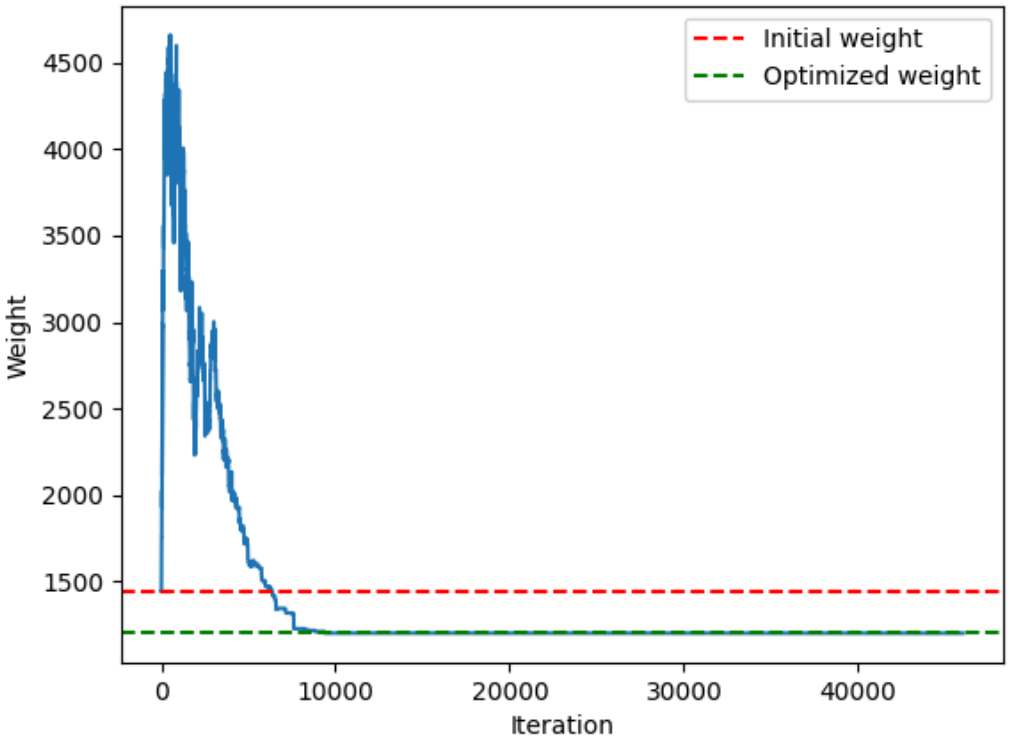


2. Moderate Temperature (100)

- Best improvement for Population = 50 (16.7%).
- Unlike Population = 20, where Temp = 100 had the best balance (27.88%), it still holds the highest improvement in a larger space.

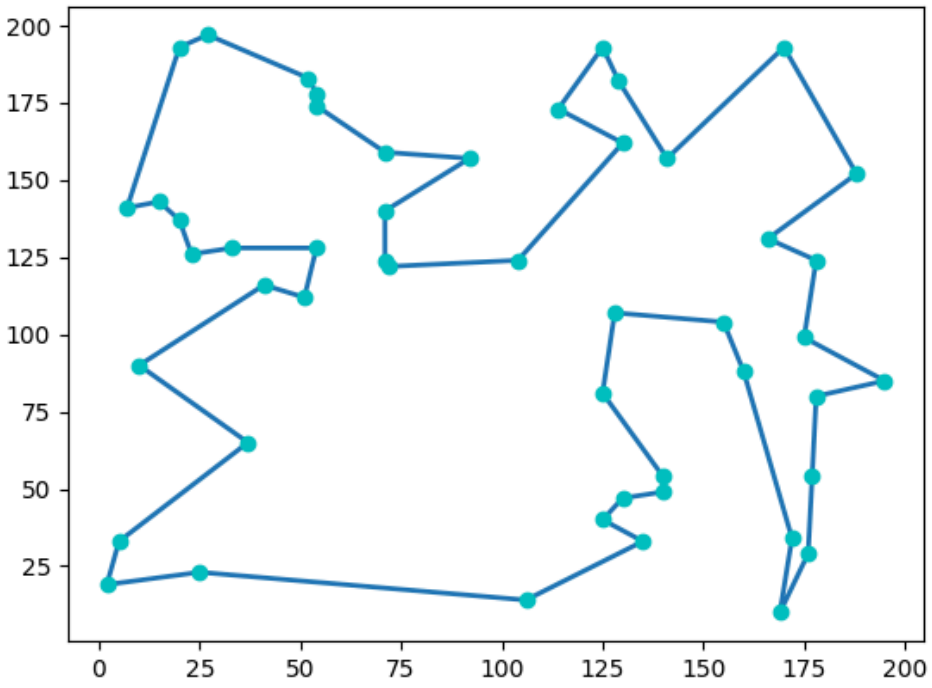
- The learning curve shows gradual optimization, meaning it successfully escaped local minima while converging efficiently.
- Best overall temperature for large populations.

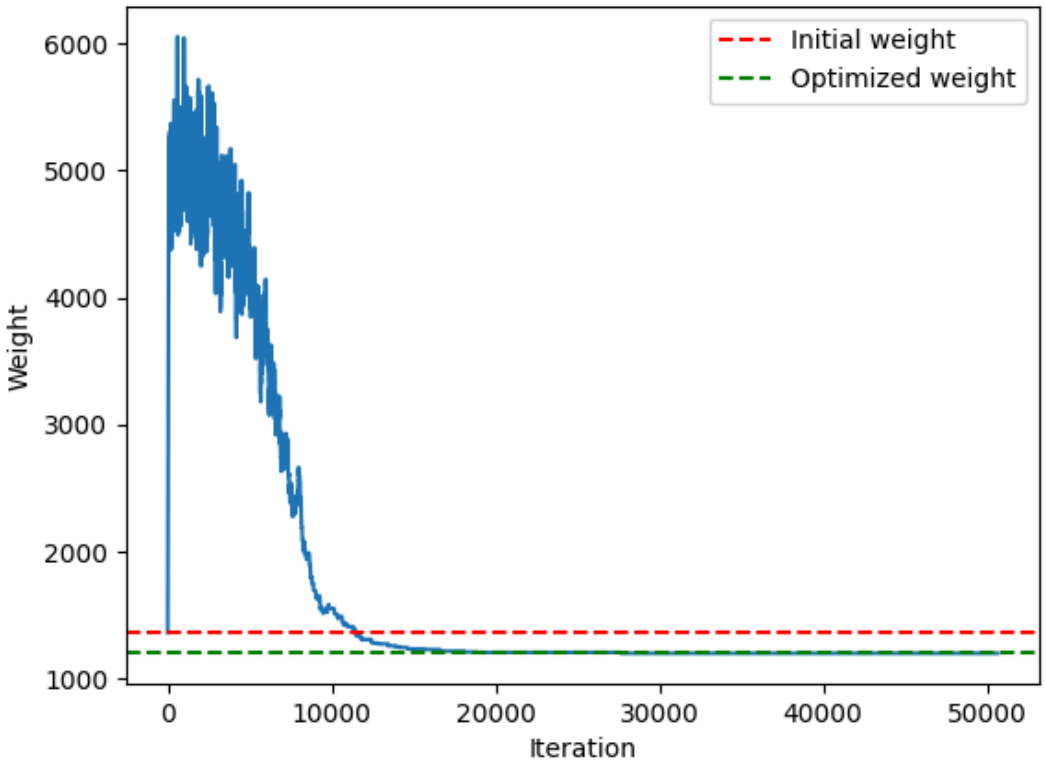




3. High Temperature (1000)

- Improvement dropped from 21.24% (Pop=20) to 11.86% (Pop=50).
- While exploration was extensive, the algorithm struggled to converge efficiently.
- Excessive randomness prevented efficient refinement, leading to slower improvement.
- Too much exploration slows down convergence, making it the least effective choice.





Which Temperature Gave the Best Results for Population = 50?

- Temperature = 100 provided the best improvement (16.7%)
- Still the best balance between exploration and convergence.
- Even though the improvement percentage is lower than for Population = 20, it remains the most effective setting.
- Temperature = 1 was too restrictive, and Temperature = 1000 was too random.

Final Conclusion

1. For small problems (Population = 20) → Temperature = 100 is the best (27.88% improvement).
2. For larger problems (Population = 50) → Temperature = 100 still performs best (16.7% improvement).
3. As population size increases, overall improvement decreases since the problem complexity grows.
4. Lower temperatures perform worse as the problem size increases because they restrict exploration too much.

5. Higher temperatures (1000) lead to excessive exploration, reducing convergence efficiency.