AER501: Assignment 3: Simulated Annealing

Emma Seabrook (1005834563)

December 6, 2023

1 Introduction

The goal of this report is to perform the Simulated Annealing optimization scheme on both the Michalewicz test function as well as the 10-bar truss from Assignment 1. The truss to be used for analysis is depicted in Figure 1.

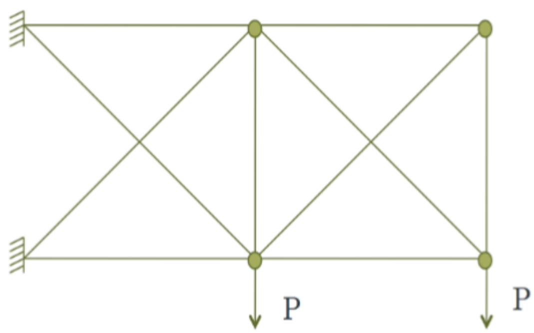

Figure 1: 10-bar truss from A1

2 Matlab Code Structure

The following subsections describe the main analysis functions present in the code.

2.1 Functions

maindriver - This function is the 'main' block that calls the other functions.

SA - This function runs the 'Simulated Annealing' optimization algorithm.

objfcn - This function executes the Michalewicz test function for a given input x.

schedule - This function calculates the cooling schedule based on the starting T, current iteration, and cooling schedule parameter.

move - move generates a random perturbation of the independent variable given bounds.

A1modified - This is a modified version of the maindriver used in Assignment 1 to calculate the stresses on the truss.

The following parameters must be set at the beginning of maindriver.m:

- c = [array] containing the cooling schedule parameters to iterate over.
- epsilon = [float] step-size to control magnitude of perturbations in move.m
- maxiter = [int] max number of iterations for simulated annealing function
- lb = [array] lower bounds for x
- ub = [array] upper bounds for x
- x0 = [array] initial guess for x
- Tstart = [int] start temperature for cooling.

3 Cooling Schedule Parameter

The Simulated Annealing (SA) algorithm makes use of a cooling function expressed in equation 1.

$$T(t) = T_0 c^t \tag{1}$$

Where T_0 is the starting temperature, and c is the cooling schedule parameter.

The following figure demonstrates the effects of the cooling schedule parameter on the result of the algorithm with maxiter set to 5000. For each value of c, 200 trials were run and their average was taken to be the result. From figure 2, we can see that any value greater than 0.99 used for the cooling schedule parameter does not converge to the correct answer. This makes sense as the faster the article cools, the closer the algorithm becomes to a greedy walk algorithm.

Figure 3 demonstrates the effects of the cooling schedule parameter on the number of iterations to reach the correct solution. Again, this was averaged over 200 runs. From the figure, it appears that when the cooling schedule parameter is between 0.6 and 0.8, the average number of iterations to convergence is approximately the same. However, when the cooling schedule parameter is increased beyond 0.8, the number of iterations required to reach convergence increases. For this particular optimization problem, I would select a value of c in the range 0.6-0.8 for rapid convergence.

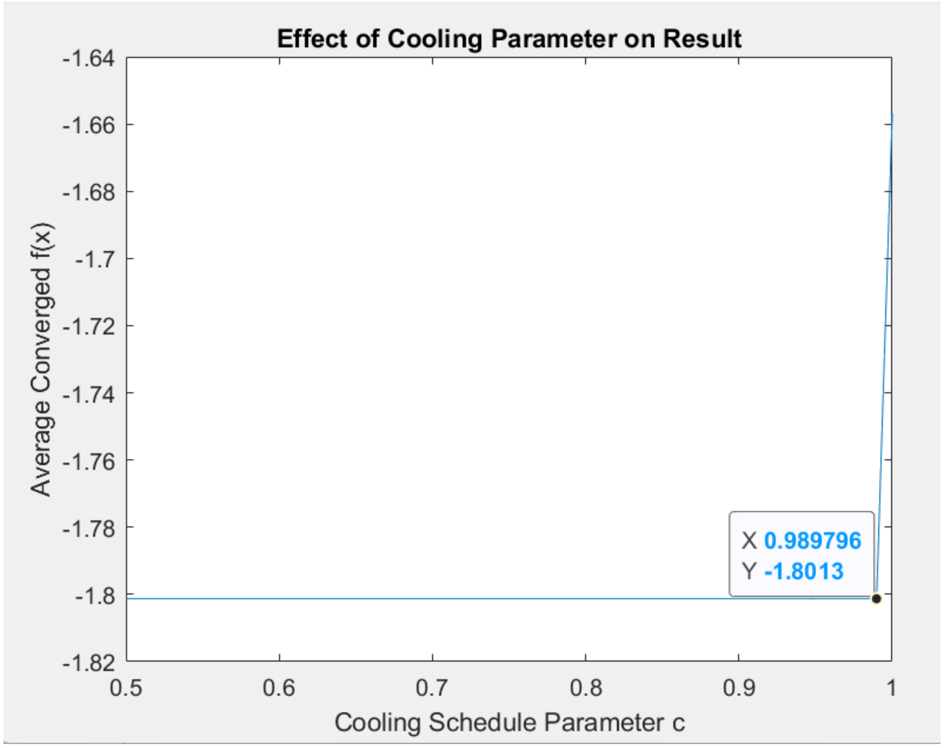

Figure 2: Cooling Parameter vs Solution

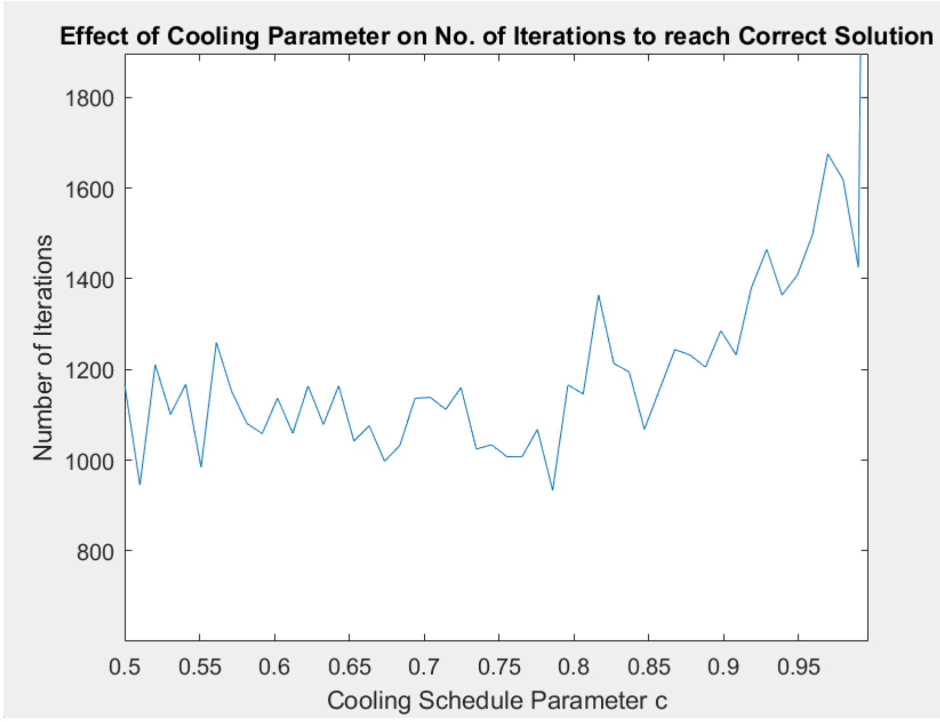

Figure 3: Cooling Schedule Parameter vs No. Iterations

3.1 Results

The best solution results for f(x) and x using the SA algorithm and matlab's fmincon function can be found in Table 1. A cooling parameter value of 0.75 was used for the simulated annealing algorithm. While the SA algorithm ran for the full 5000 iterations, the fmincon function converged in only 12 iterations, making it significantly more efficient than SA. However, fmincon converged to f(x) = -1.214 whereas my simulated annealing function converged to -1.8013 which is a more accurate minimum.

Table 1: SA and fmincon Results for Michalewicz Function

	x1	x2	f(x)
SA	2.202494	1.57145	-1.80128
fmincon	2.202906	2.71157	-1.21410

4 Truss Analysis

In this section, we will analyse the performance of the SA algorithm on the 10-bar truss from Assignment 1. In this case, the mass of the truss is to be minimized using the cross-sectional areas of the beams as input parameters. The mass of each member of the truss can be calculated using Equation 2, where $\rho = 2.7g/cm^3$ is the density of aluminum [1], l is the length of the beam, and a is the cross-sectional area of the beam. The constraint imposed on the system is expressed in Equation 3 where $\sigma_{yield,tensile} = 75MPa$ is the tensile yield stress of aluminum and $\sigma_{yield,compressive} = -30MPa$ is the compressive yield stress of aluminum [1]. The initial value was set to $a_0 = 0.0001cm^3$ which is the value that was used in Assignment 1.

$$m(a) = \rho * a * l \tag{2}$$

$$\sigma < \sigma_{yield} \tag{3}$$

4.1 Cooling Parameter

The cooling parameter selected from part 1 to use for the truss analysis is c = 0.75.

4.2 Penalty Analysis

The truss analysis was conducted using the quadratic penalty function as defined in Equation **??**. Seven different values were tested for the penalty parameter ρ: [0.01, 0.1, 1, 10, 100, 1000]. The results of these trials can be found in Figure 4. As can be seen in the figure, the penalty parameter does not affect the speed at which the algorithm converges.

Table 2 compares the best solution for the six penalty parameters. Despite them converging at approximately the same speed, when $\rho = 1$, the mass of the structure is a minimum. For this reason, 1 is the ideal penalty parameter.

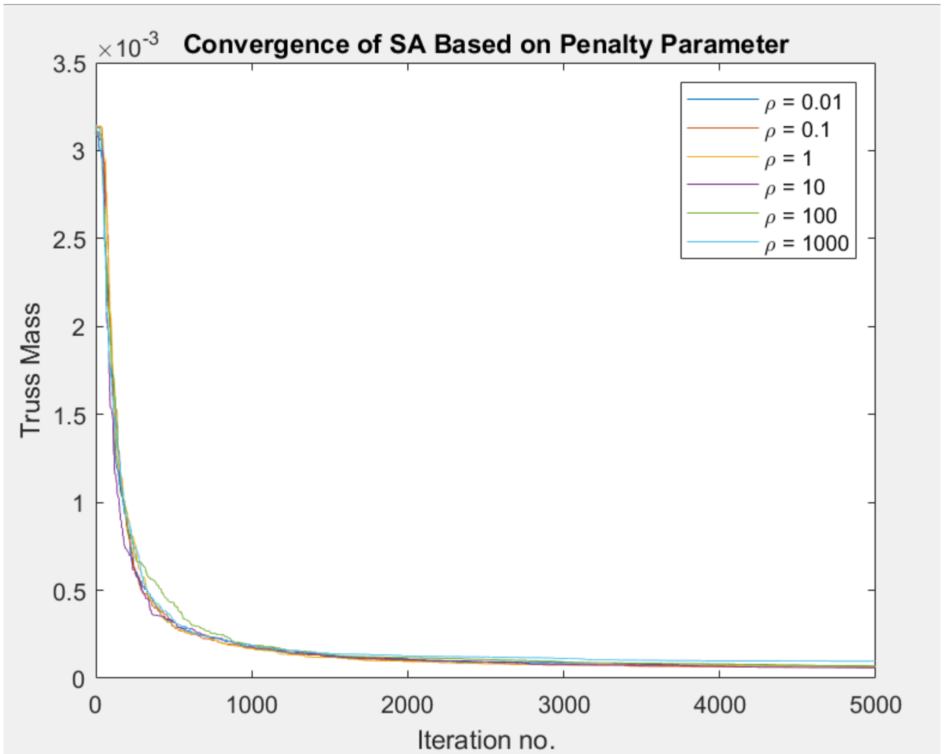

Figure 4: Effects of Changing the Penalty Parameter C

Table 2: Comparing best result for various penalty parameters

ρ	0.01	0.1	1	10	100	1000
fopt	7.90e-5	6.73e-5	6.41e-5	7.43e-5	8.74e-5	7.20e-5

4.3 Results

The results of the SA algorithm for the 10-bar truss problem can be found in Table 3.

Table 3: 10-bar Truss SA algorithm Result using $\rho = 1$

Algo.	areas:	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	Mass (1e-4 kg)
SA (1e-6 cm^3)	1.89	0.87	0.57	8.52	0.26	7.45	0.10	6.78	10.04	6.37	1.34
fmincon (1e-6 cm^3)	3.79	3.90	3.90	5.02	5.92	2.73	3.28	5.61	11.49	4.16	1.53

The following figure (5) demonstrates the proportional thickness of each member according to the Simulated Annealing algorithm. Evidently, to minimize mass but ensure that the members are able to support the loads, the algorithm has thickened the members that must bear more of the load while reducing the mass of the ones that do not.

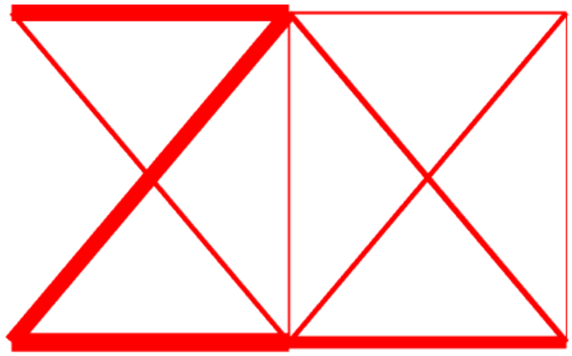

Figure 5: Simulated Beam Thicknesses of SA results

Table 3 presents the results of the truss optimization using Matlab's fmincon function. This function converged on a solution in only 22 iterations making it significantly more efficient than the simulated annealing algorithm. Figure 6 is an illustration of the optimized solution that fmincon converged to. Again, we see the trend of the thicker beams being the ones that need to hold more load. Neither the SA algorithm nor fmincon converged to a solution that violated the constraints.

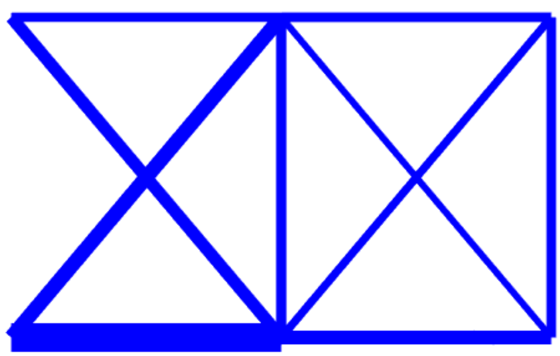

Figure 6: Illustration of Proportion of Truss Members using fmincon

5 Conclusions

In this report, the SA algorithm was used to optimize both the Michalewicz test function and the 10-bar truss problem from Assignment 1. These results were compared to Matlab's fmincon function which outperformed the SA algorithm by converging to the correct solution in fewer iterations. However, in both cases, fmincon converged to a larger value of f(x) than the Simulated Annealing algorithm did. For these reasons, I would recommend fmincon for problems where it is more important to converge quickly than to reach the true minimum.

References

[1] A. Materials, "Aluminum - advantages and properties of aluminum," 2002. [Online]. Available: https://www.azom.com/properties.aspx?ArticleID=1446&fbclid=IwAR1sqtcfo63D8b2rsxg-g9im2fsbnZarBdjyhM2MWVjIbAkA3rPsfEUpbw8