Optimisation Project 1

Tiffany Woodley

9/10/2020

The aim of this project is to optimise a maintance schedule for an array of generating units using simulated annealing. The approch attempted to be recreated is outlined in the paper *An investigation into the effectiveness of simulated annealing as a solution approach for the generator maintenance scheduling problem* [1]. A schedule is optimised for two test benches, a 21 unit and a 32 unit generating unit schedule. The data utilised for the 32 unit testbench is found in the appendix of the forementioned paper[1] and the data utilised for the 21 unit is found in the dissertation *Descicion support for generator maintenance cheduling in the energy sector*[2]. The methodology utilised followed the best implementation found in the paper.

Data

There were two test systems used in the paper, a 32 and 21 unit system, these systems are outlined below,

32 - unit test system

The data used to initialise the 32-unit test syatem can be seen below, where i represents the number of the generating unit, g_i represents the generating capcity of the unit, e_i and l_i represent the earliest and latest starting date for the generating unit's maintenance to commence. And m_i^w indicates the amount of man power required for the maintenance of the generating unit in week w. This system has a maximum of 25 man power hours per week.

Table 1: 32-unit test system data

i	g_i	e_i	l_i	d_i	m_i^1	m_i^2	m_i^3	m_i^4	m_i^5	m_i^6
1	20	1	25	2	7	7	0	0	0	0
2	20	1	25	2	7	7	0	0	0	0
3	76	1	24	3	12	10	10	0	0	0
4	76	27	50	3	12	10	10	0	0	0
5	20	1	25	2	7	7	0	0	0	0
6	20	27	51	2	7	7	0	0	0	0
7	76	1	24	3	12	10	10	0	0	0
8	76	27	50	3	12	10	10	0	0	0
9	100	1	50	3	10	10	15	0	0	0
10	100	1	50	3	10	10	15	0	0	0
11	100	1	50	3	15	10	10	0	0	0
12	197	1	23	4	8	10	10	8	0	0
13	197	1	23	4	8	10	10	8	0	0
14	197	27	49	4	8	10	10	8	0	0
15	12	1	51	2	4	4	0	0	0	0
16	12	1	51	2	4	4	0	0	0	0
17	12	1	51	2	4	4	0	0	0	0
18	12	1	51	2	4	4	0	0	0	0
19	12	1	51	2	4	4	0	0	0	0
20	155	1	23	4	5	15	10	10	0	0
21	155	27	49	4	5	15	10	10	0	0
22	400	1	21	6	15	10	10	10	10	5
23	400	27	47	6	15	10	10	10	10	5

i	g_i	e_i	l_i	d_i	m_i^1	m_i^2	m_i^3	m_i^4	m_i^5	m_i^6
24	50	1	51	2	6	6	0	0	0	0
25	50	1	51	2	6	6	0	0	0	0
26	50	1	51	2	6	6	0	0	0	0
27	50	1	51	2	6	6	0	0	0	0
28	50	1	51	2	6	6	0	0	0	0
29	50	1	51	2	6	6	0	0	0	0
30	155	1	23	4	12	12	8	8	0	0
31	155	1	49	4	12	12	8	8	0	0
32	350	1	48	5	5	10	15	15	5	0

For the 32 unit system, not all of the units can be in maintence simultaneously. The following table indicates how many units in a set of generating units can be in maintenance simultaneously.

Table 2: 32-unit test system exclusion sets

Exclusion set	Max maintenance
1,2,3,4	2
5,6,7,8	2
9,10,11	1
12,13,14	1
15,16,17,18,19,20	3
24,25,26,27,28,29	3
30,31,32	1

The following shows the weekly demand, the appropriate maintenence schedule needs to be chosen in order to ensure these weekly demands can still be met.

Table 3: 32-unit test system weekly demand

Week	Demand	Week	Demand	Week	Demand	Week	Demand
1	2457	14	2138	27	2152	40	2063
2	2565	15	2055	28	2326	41	2118
3	2502	16	2280	29	2283	42	2120
4	2377	17	2149	30	2508	43	2280
5	2508	18	2385	31	2058	44	2511
6	2397	19	2480	32	2212	45	2522
7	2371	20	2508	33	2280	46	2591
8	2297	21	2440	34	2078	47	2679
9	2109	22	2311	35	2069	48	2537
10	2100	23	2565	36	2009	49	2685
11	2038	24	2528	37	2223	50	2765
12	2072	25	2554	38	1981	51	2850
13	2006	26	2454	39	2063	52	2713

21 - unit test system

The following shows the data for the 21-unit system, this test bench however does not include exclusion sets like the 32-unit system. This system has a maximum of 20 man power maintenance hours per week.

Table 4: 21-unit test system data

i	g_i	e_i	l_i	d_i	m_i^1	m_i^2	m_i^3	m_i^4	m_i^5	m_i^6	m_i^7	m_i^8	m_i^9	m_i^10
1	555	1	20	7	10	10	5	5	5	5	3	0	0	0
2	555	27	48	5	10	10	10	5	5	0	0	0	0	0
3	180	1	25	2	15	15	0	0	0	0	0	0	0	0
4	180	1	26	1	20	0	0	0	0	0	0	0	0	0
5	640	27	48	5	10	10	10	10	10	0	0	0	0	0
6	640	1	24	3	15	15	15	0	0	0	0	0	0	0
7	640	1	24	3	15	15	15	0	0	0	0	0	0	0
8	555	27	47	6	10	10	10	5	5	5	0	0	0	0
9	276	1	17	10	3	2	2	2	2	2	2	2	2	3
10	140	1	23	4	10	10	5	5	0	0	0	0	0	0
11	90	1	26	1	20	0	0	0	0	0	0	0	0	0
12	76	27	50	3	10	15	15	0	0	0	0	0	0	0
13	76	1	25	2	15	15	0	0	0	0	0	0	0	0
14	94	1	23	4	10	10	10	10	0	0	0	0	0	0
15	39	1	25	2	15	15	0	0	0	0	0	0	0	0
16	188	1	25	2	15	15	0	0	0	0	0	0	0	0
17	58	27	52	1	20	0	0	0	0	0	0	0	0	0
18	48	27	51	2	15	15	0	0	0	0	0	0	0	0
19	137	27	52	1	15	0	0	0	0	0	0	0	0	0
20	469	27	49	4	10	10	10	10	0	0	0	0	0	0
21	52	1	24	3	10	10	10	0	0	0	0	0	0	0

The 21 unit system has a flat demand as can be seen below

Table 5: 21-unit test system weekly demand

Week	Demand	Week	Demand	Week	Demand	Week	Demand
1	4739	14	4739	27	4739	40	4739
2	4739	15	4739	28	4739	41	4739
3	4739	16	4739	29	4739	42	4739
4	4739	17	4739	30	4739	43	4739
5	4739	18	4739	31	4739	44	4739
6	4739	19	4739	32	4739	45	4739
7	4739	20	4739	33	4739	46	4739
8	4739	21	4739	34	4739	47	4739
9	4739	22	4739	35	4739	48	4739
10	4739	23	4739	36	4739	49	4739
11	4739	24	4739	37	4739	50	4739
12	4739	25	4739	38	4739	51	4739
13	4739	26	4739	39	4739	52	4739

Implementation

In order to initialise a starting solution, a random starting value between the earliest and latest week is chosen for each generating unit with an extension period of 2 weeks on either side. This results in a vector of values which represent the starting week, the indices of the vector represent which generating unit the starting value represents. An example starting solution can be seen below:

Unit	Start Week	Unit	Start Week	Unit	Start Week	Unit	Start Week
1	19	9	7	17	13	25	16
2	17	10	38	18	46	26	32
3	18	11	43	19	51	27	7
4	52	12	3	20	8	28	34
5	3	13	2	21	32	29	23
6	34	14	39	22	10	30	5
7	10	15	12	23	46	31	5
8	44	16	29	24	52	32	16

This initialisation doesn't guarantee a feasible solution, there are four ways in which the solution could be infeasible

1. It falls outside the maintance window,
2. It does not cover the demand in all weeks, and leaves a shortfall in energy,
3. It requires more man power in a week than is available,
4. It allows generating units which can not be down together to go into maintanence simultaneously.

A penalty is calculated for the violations of these constraints and functions are created in order to calculate the infeasibility of a solution.

The first penalty is calculated as a weighted sum of the individual generating unit window constraint violations,

$$P_w = \sum_{i=1}^{n} P_w^i$$

For each generating unit P_w^i can be worked out using the following equation, where x_i is the starting week for the generating unit and e_i and l_i are the earliest and latest weeks.

$$P_w^i = \begin{cases} e_i - x_i & \text{if } x_i < e_i \\ 0 & \text{if } e_i \leq x_i \leq l_i \\ x_i - l_i & \text{if} x_i > l_i \end{cases}$$

The second penalty calculates the total shortfall in covering the demand over the 52 weeks(m)

$$P_l = \sum_{j=1}^{m} P_l^j$$

where P_l^j is the shortfall of the load in the j^{th} week

$$P_l^j = max(-r_j, 0)$$

where

$$r_j = total\ generated_j - demand_j$$

The third penalty calculates the violations against the number of crew available

$$P_c = \sum_{j=1}^{m} P_c^j$$

where

$$P_c^j = max(\sum_{i=1}^{n}\sum_{p=1}^{j} m_{p,i,j}x_{i,p} - M_j, 0)$$

In this case $m_{p,i,j}$ represents the needed manpower to perform maintance on generating unit i in week j if maintenance commenced in week p. $x_{i,p}$ represents a binary variable which indicates whether generating unit's maintanance commenced in week p.

The fourth penalty is for the exclusion constraint, this panalty calculates the exclusion constraint violations.

$$P_e = \sum_{k=1}^{K}\sum_{j=1}^{m} P_e^{k,j}$$

where

$$P_e^{k,j} = max(\sum_{i \in I_k} y_{i,j} - K_k, 0)$$

and $y_{i,j}$ indicates whether the i^{th} unit is in maintenance during the j^{th} week.

The total penalty given due to the different constraint violations is shown below as a weighted combination of the individual penalties. The weights come from the dissertation[2] where they were chosen imperically by determining how many feasible/ infeasible solutions the weight produced when coupled with the square of reserves objective function.

$$P = w_w P_w + w_l P_l + w_c P_c + w_e P_e$$

The penalty weight values were

$$w_w = 40000$$

$$w_l = 1$$

$$w_c = 20000$$

$$w_e = 20000$$

The main objective is to improve the reliablity of the system by attempting to reduce the over production of energy, this is done by minimising the sum of square reserves, where S is a safety factor of 15%

$$sum\ of\ squared\ reserves = \sum_{j=1}^{m}(r_j - demand_j * S)^2 + P$$

Since the aim is to reduce the sum of squared reserves, whilst also reducing the constraint violations, the overall objective is

$$objective = \sum_{j=1}^{m}(r_j - demand_j * S)^2$$

Simulated annealing

The initial solution is optimised using simulated annealing in order to find a new scheduling solution. The initial solution started with a sum of squared reserves of 41754529 MW^2 and penalty values of 5 for the window constraint, 279 for the shortfall constraint, 118 for the crew constraint and 1 for the exclusion constraint.

The function used to generate new candid solutions was an enjection chain pertubation function which makes a sequence of steps altering the solution. These steps are created in such a way that the proceding unit to be changed is chosen if it has the same starting value as the preceding value. The sequence continues until either there is no other units with the starting value or the original starting vlaue is arrived back at.

The starting temperature for the cooling function is found by performing a random walk and altering the initial solution using the enjection chain update and finding the average of the increases in the solution. The starting temperature is then calculated using

$$T_s = \frac{-\Delta E}{ln(\chi_0)}$$

χ_0 the initial acceptance ration, was taken from the paper as 0.5, this meant that the initial temperature value would allow approximatley half the worsening solutions to be chosen.

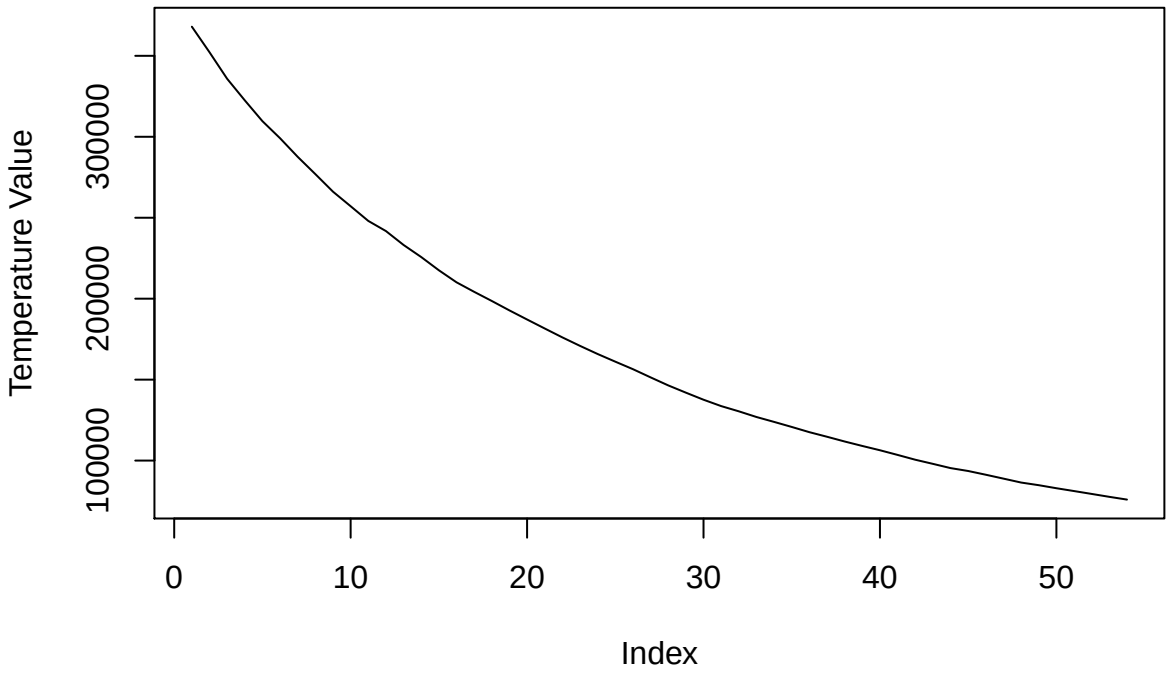
This lead to a starting temperature of 387128.6.

The cooling function used was from *Simulated Annealing: Theory and Applications*[3] were delta was taken as 0.16 and σ_s the standard deviation of the objective values generated using the previous temperature value.

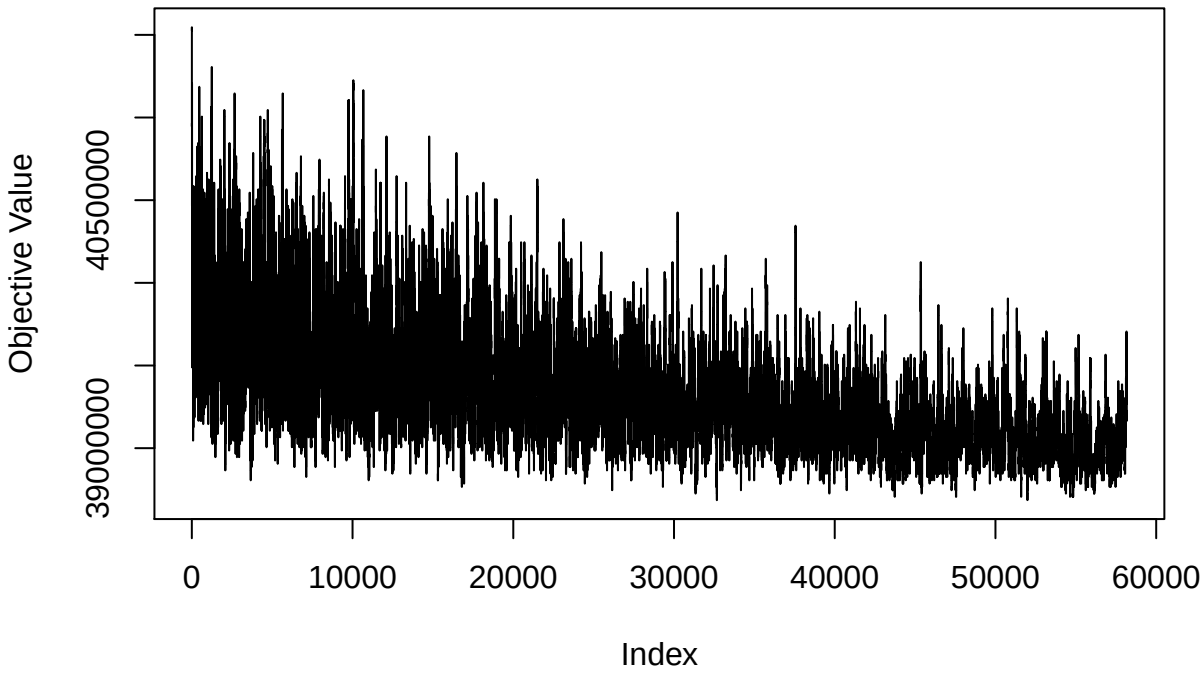
$$T_{s+1} = T_s \frac{1}{1 + \frac{ln(1+\delta)}{3\sigma_s} T_s}$$

The termination of the algorithm was based on the temperature value reaching a certain threshold T_min. In the paper[1] the suggested value for fot T_min was 1, however this did not lead to termination in a practical timeframe as the temperature function started decreasing extremely gradually the further along the algorithm got and so a value of 20% of the initial temperaturevalue was used for T_min in order to allow the algorithm to finish.

The temperature values over all the iterations can be seen below,



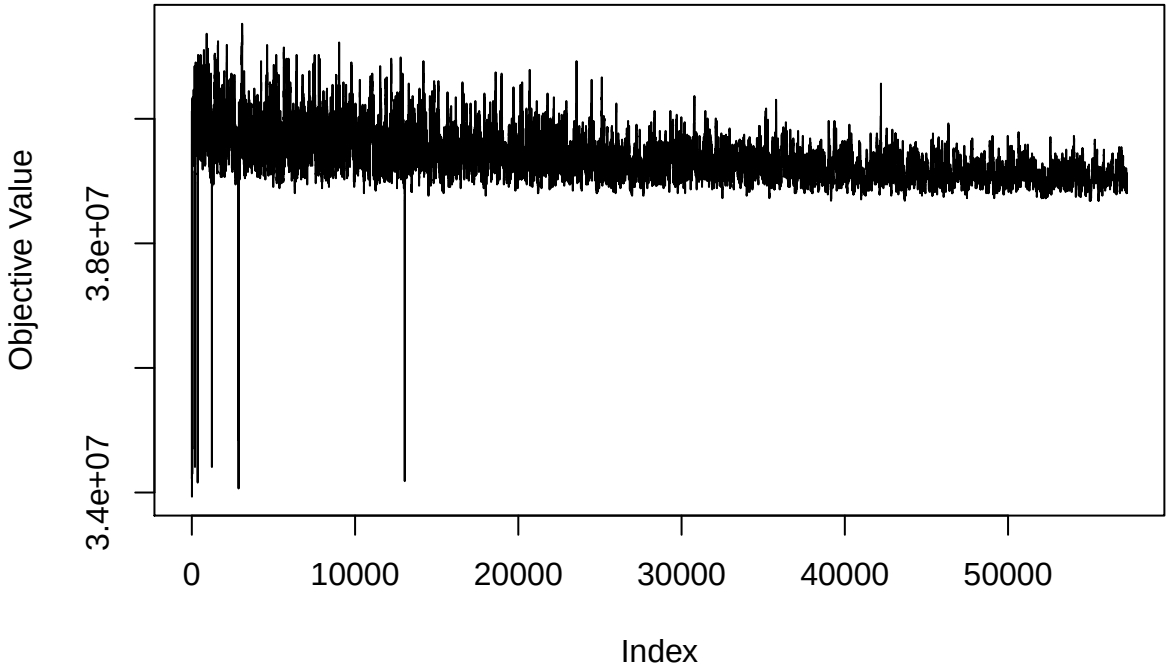
The objective function gradually improves but fails to converge in the given timeframe. The final sum of squared reserves was 35688694 MW^2, an improvement from the initial solution. The final soltion had far fewer constraint violations with there being a penalty of 0 for the window constraint, 0 for the shortfall constraint, 15 for the crew constraint and 2 for the exclusion constraint.



Local search hueristic

The local search hueristic approach outlined in the paper aims to improve upon the previous method, the previous algorithm is implemented with the difference being when an incumbenant solution is found (the lowest solution so far) the algorithm will continue taking greedy moves unil there is no improvement, this new

improved solution is now stored instead of the initial solution and the algorithm continues with it's found solution before the greedy optimisation was applied. This approach doesn't allow the greedy approach to end up stuck in a local optima, but hopes that one of the found local optima is the global optima or close to it.



This approach results in a minimum objective of 33933176, with a sum of squared reserves of 33873176 MW^2 with penalty violations of 1 for the window constraint and 1 for the crew constraint. This a big improvement from the previous implementation.

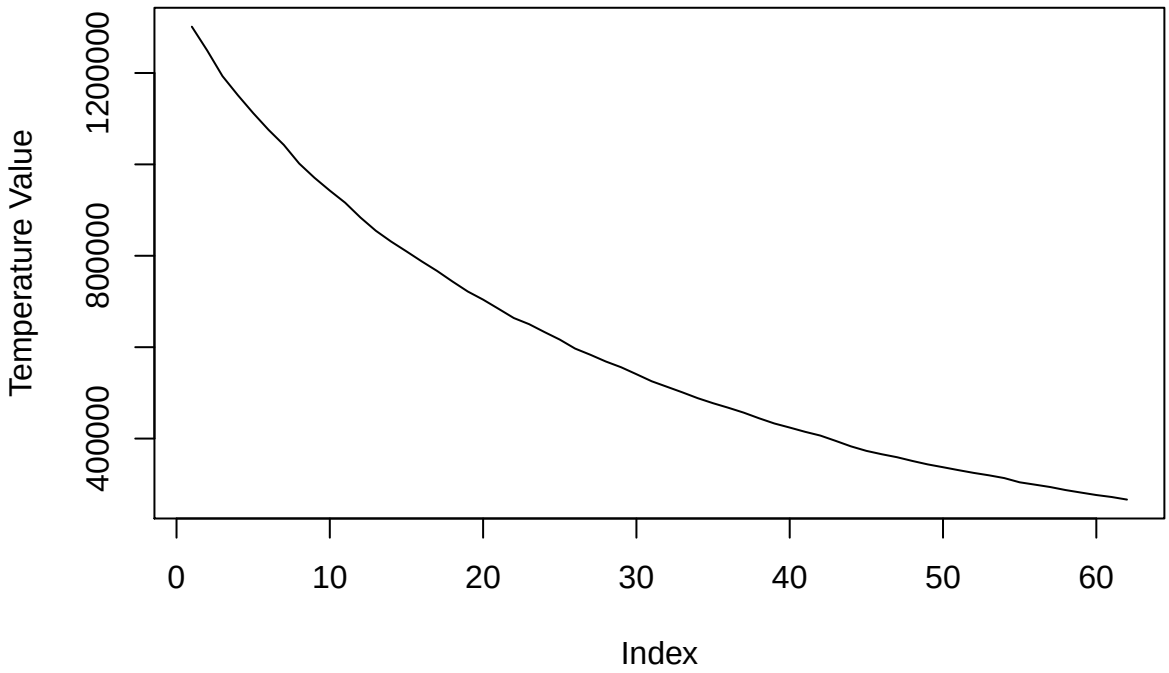
21-unit system

The 21-unit system was approached in the same manner, with the exception of there being no penalty for any exclusion constraints.

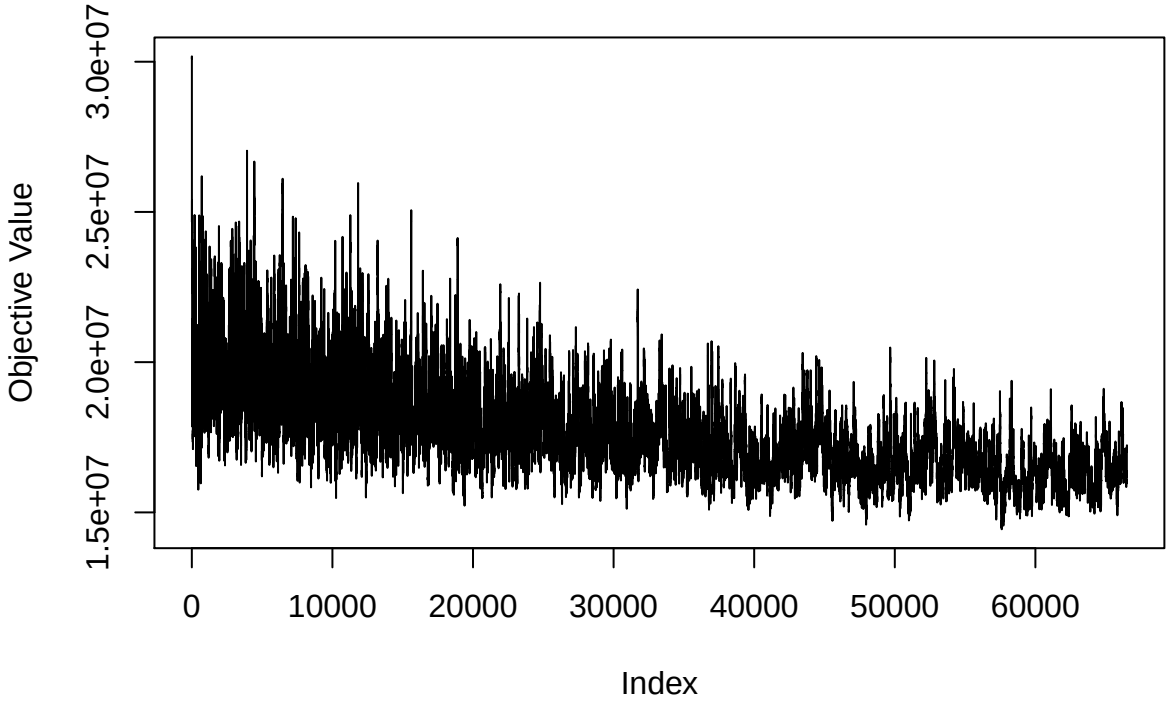
The initial solution had a sum of squared reserves of 27063740 MW^2, with penalty values of 4 for the window constraint, 4041 for the shortfall constraint and 143 for the crew constraint.

The starting temperature for the 21-unit system was 1345932.

The temperature values can be seen below,

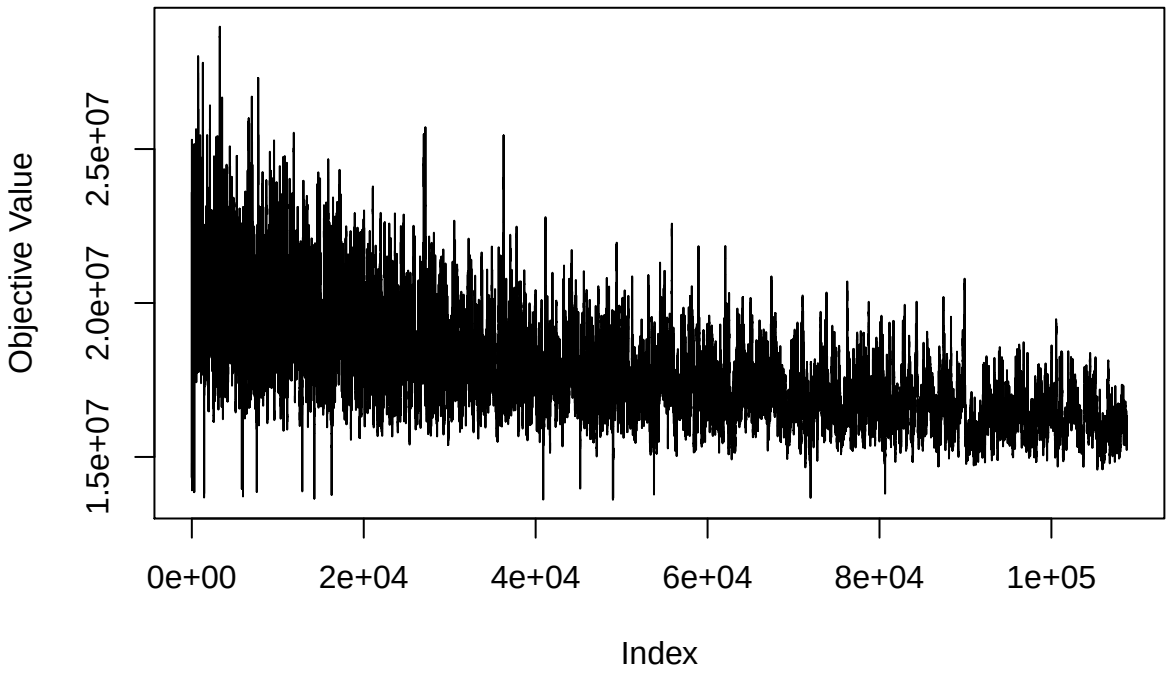


The objective function gradually improves, but fails to converge in the given timeframe. The final sum of squared reserves was 15731813 MW^2, an improvement from the initial solution. The final soltion had a penalty of 1 for the window constraint, 59 for the shortfall constraint and 72 for the crew constraint.



Local search heuristic

This approach results in a sum of squared reserves of 13535635 MW^2 with penalty violations of 1 for the window constraint and 2 for the crew constraint. This a big improvement from the previous implementation.



References

1. International Journal of Electrical Power & Energy Systems, ISSN: 0142-0615, Vol: 53, Issue: 1, Page: 166-174
2. Schlunz EB. Descicion support for generator maintenance cheduling in the energy sector. Master's thesis, Stellenbosch University, Stellenbosch; 2011.
3. van Laarhoven P.J.M., Aarts E.H.L. (1987) Simulated annealing. In: Simulated Annealing: Theory and Applications. Mathematics and Its Applications, vol 37. Springer, Dordrecht. https://doi.org/10.1007/978-94-015-7744-1_2