

5

Case Study

In this chapter, the developed application will be used to design a lattice tower structure that withstands the same loads and has the same geometric constraints as a case study tower provided by Metalgalva.

In the next subchapters, the model will be presented, along with the steps taken to prepare the optimization process.

Finally, the results will be analysed and conclusions about real world implications will be drawn along with future development paths that could be taken to improve the application.

5.1. BASE MODEL

The case study tower is 38 meters high, and carries 7 cables, 3 in each side and one at the tip of the tower (Figure 5.1). The distance between leg members at the base is 5 meters in both directions and, each arm is 2.25 meters long. The steel used is of type S275. The model provided by Metalgalva has a total weight of 7.616 tonnes, distributed according to Table 5.1

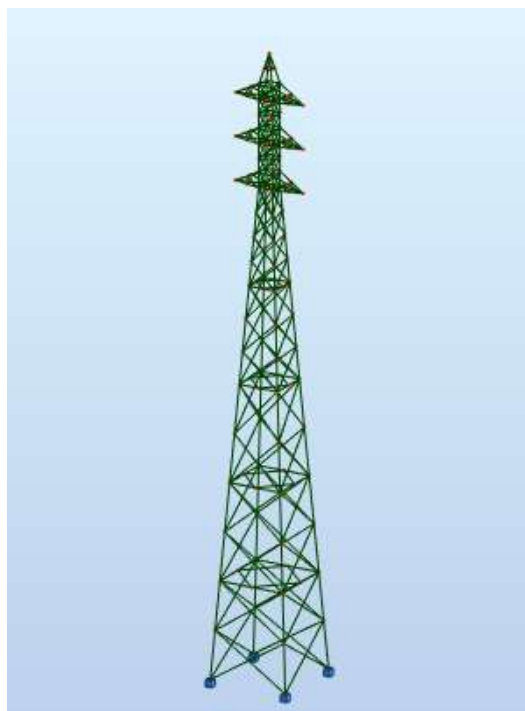


Fig. 5.1 – Base Model

Table 5.1 – Base model weight distribution

Sections	Number	Total weight (kg)
L 50x50x5	262	1873
L 60x60x6	44	699
L 70x70x7	4	123
L 100x100x10	4	379
L 140x140x13	4	660
L 160x160x15	4	870
L 180x180x6	4	1046
Ls 180x180x15	8	1967
		7616

Regarding the loads, the base model provided by Metalogalva contained hundreds of load cases. To improve the run time of the optimisation routine, the critical load cases were identified with the help of Metalogalva's technical department. This interaction with the engineering team, allowed the application to run with only 4 critical load cases added to the self-weight of the structure. Given the way the DNA of the structure is generated, this reduction of load cases adds the need of a final inspection of the output model to ensure symmetry. This is a point of further improvement of the program that will be detailed at the end of the chapter.

Figure 5.2 identifies the arm nodes and Table 5.1 shows the forces applied to each of the four critical load cases (LC1 to LC4).

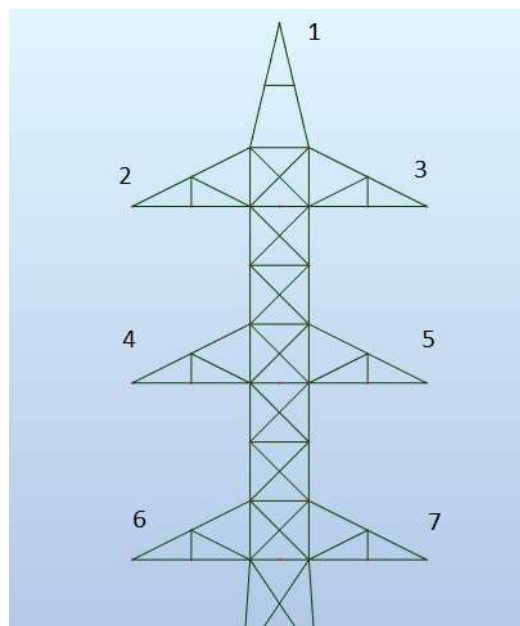


Fig. 5.2 – Arm nodes

Table 5.2 – Load cases

	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7
LC1	FX=13.24	FX=24.02	FX=24.02	FX=24.02	FX=24.02	FX=24.02	FX=24.02
(kN)	FY=1.23	FY=0.49	FY=0.49	FY=0.49	FY=0.49	FY=0.49	FY=0.49
	FZ=-3.43	FZ=7.84	FZ=7.84	FZ=7.84	FZ=7.84	FZ=7.84	FZ=7.84
LC2	FX=0.0	FX=0.0	FX=0.0	FX=0.0	FX=0.0	FX=0.0	FX=0.0
(kN)	FY=12.50	FY=29.41	FY=22.06	FY=22.06	FY=22.06	FY=22.06	FY=22.06
	FZ=-2.45	FZ=-7.84	FZ=-4.90	FZ=-4.90	FZ=-4.90	FZ=-4.90	FZ=-4.90
LC3	FX=1.72	FX=2.45	FX=2.45	FX=2.45	FX=2.45	FX=2.45	FX=2.45
(kN)	FY=12.50	FY=22.06	FY=22.06	FY=22.06	FY=22.06	FY=22.06	FY=22.06
	FZ=-2.45	FZ=-4.90	FZ=-4.90	FZ=-4.90	FZ=-4.90	FZ=-4.90	FZ=-4.90
LC4	FX=0.0	FX=0.0	FX=0.0	FX=0.0	FX=0.0	FX=0.0	FX=0.0
(kN)	FY=12.50	FY=22.06	FY=29.41	FY=22.06	FY=22.06	FY=22.06	FY=22.06
	FZ=-2.45	FZ=-4.90	FZ=-7.84	FZ=-4.90	FZ=-4.90	FZ=-4.90	FZ=-4.90

5.2. PROGRAM SETUP

To configure the application to design an optimised structure that withstands the loads listed above and with similar geometrical characteristics, a few steps are needed.

As it is a small tower (38m), the penalty function, which was discussed on Chapter 4, is changed to 0.15 tonnes. A list of sections needs to be provided to the genetic algorithm before starting the optimisation. The sections considered were the following steel equal angles:

- L50x50x5
- L60x60x6
- L70x70x7
- L100x100x10
- L150x150x15
- L180x180x16

The list has fewer sections than the base model. This decision was based on a previous analysis in which it was concluded that there was no need, based on bar utilization factors and the expected optimization ratio, for two intermediate bars between the L 100 and L 180 sections. A single L 150 section was used instead of the L 140 and L160.

This reduction of available sections also allows the program to run faster as the search space is reduced. However, there is a risk that the output structure still has penalties applied, in other terms, some bars need to have larger sections that are not available. To address this issue, a log file containing the list of bars that need to be strengthened (either by secondary bracing or larger sections) is produced for the final structure.

The load cases described in Section 5.1 were added using the Robot UI and for each load case the self-weight of each bar was added. This additional load is automatically updated by Robot in each iteration.

5.2.1. TEST RUNS AND CALIBRATION

Given the nature of the program, it requires several parameters to be adjusted based on the structure characteristics. This calibration is done to ensure not only a thorough search of the solution space but also a reasonable computational time.

The first parameters to adjust were the threshold values of the U/f that defined over-designed and under-designed bars. This is highly dependent on the variety of sections provided to the GA and how incremental their ultimate resistances are.

The U/f threshold for bars that can be randomly reduced (Section 4.6), needs to be small enough so that a possible reduction or deactivation of a bar does not cause the failure of other bars in its proximities due to load redistribution. Moreover, this U/f value needs to be high enough to remove unnecessary bars from the structure. This is structure dependant as larger structures are likely to have more elements to resist load increases of this type.

To define this value, a test run in debug mode was carried out. Running in debug mode allowed for the U/f threshold to be changed at runtime. The reduced or deleted bars were monitored as well as the remaining bars for changes in U/f . After a few iterations, the lower threshold value of 0.1 was adopted for the majority of the search. In the final iterations, when all the bars are likely to exceed that U/f threshold, the value needs to be increased. In this case, it was increased to 0.3 for the final generations of the GA.

It is important to note that this lower U/f value is used to steer the solution in the right direction and not to limit the solution (Section 4.6). There can be active bars with U/f lower than 0.1 and inactive bars with U/f higher than 0.3 because of the mutation operator.

The higher U/f threshold that splits into two lists over and under-designed bars, was found in the same iterative way to ensure that for bars that failed the EC3 check, the next available cross-section has a high probability – to increase convergence speed – to pass EC3 checks. The value adopted was 0.7 for the entire optimisation run – bars with U/f larger than 0.7 are therefore not reduced.

The remaining adjustable elements regarding the GA algorithm itself are: population size, mutation probability and mutation pressure.

These three parameters are interdependent. For that reason, the calibration requires several test runs to find the ideal combination for the structure to optimize. There are a few rules of thumb useful for the calibration. For example, a low population size can be balanced by a high mutation probability (provided the algorithm still converges); a low selection pressure can be counteracted by a high population size; These rules of thumb have both the same objective, i.e., to maintain search space diversity.

For this case, given that during testing with small structures (200 bars and 3 sections) a population size of 25 provided good results, the population size for the case study was increased linearly – resulting in a population size of 225. From this point, the mutation probability and selection pressure were both adjusted.

With the default mutation probability of 15% (very high for traditional GAs), the search degraded into a random search. The second iteration, with a mutation probability of 10% returned some form of convergence. However, from the several peaks and troughs visible in Figure 5.3, it was apparent that the solution space was not being searched thoroughly.

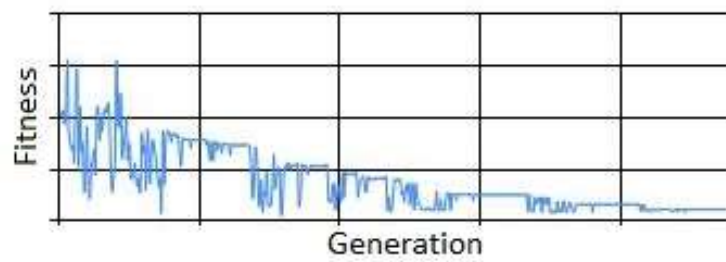


Fig. 5.3 – Mutation adjustment trial

After a few iterations, the final value of 4% of the mutation probability returned a much smoother graph (Figure 5.4). The problem with this configuration was the search time. Such a thorough search took 3 days to complete in the setup described on Section 5.3. At this point the focus shifted to the selection pressure, the last element left to be configured. If this parameter did not deliver a significant reduction of the search time, the entire configuration of the GA would need to start over with a different population size.

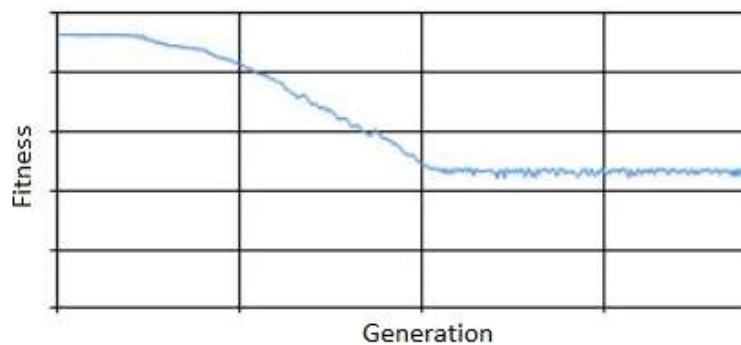


Fig. 5.4 – Ideal mutation probability reached

The selection pressure, is increased with the increase in size of the tournament pool as explained in Section 4.7. With a higher tournament pool, a low-quality individual has lower probability of being selected because there are more individuals he is compared with. In graphical terms, this results in a sharp increase in fitness of the initial generations (Figure 5.5). With a few iterations, it was found that a selection pool with 45 (20 % of total population) returned a good solution with a significant reduction of search time – down to 14 hours.

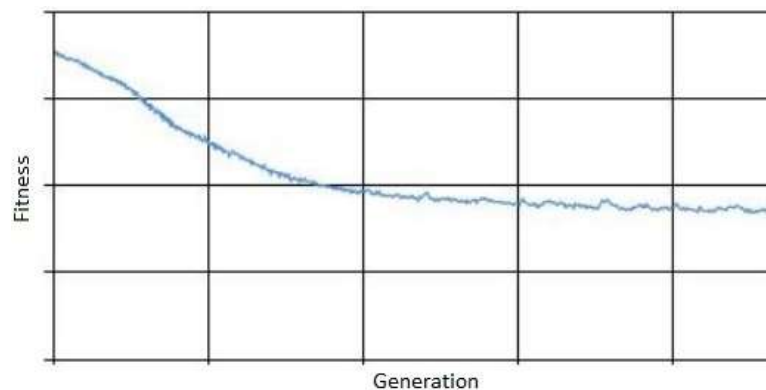


Fig. 5.5 – Final search

5.3. RESULTS

The optimization ran for 14 hours in a computer with 8GB of RAM and an Intel Core I7-6700HQ (8 threads) before the termination criteria of the genetic algorithm was met.

During runtime, several debug logs were collected and they revealed a bottleneck present in the Robot API. The function used to update bar properties for each individual evaluation was responsible for nearly 70% of the runtime of the entire optimization routine. Such delay in this operation, points to a possible limitation in the API to handle fast updates in models with several bars. This is a problem that will need to be addressed if the present software is to be used to optimize larger structures.

5.3.1. POSTPROCESSING

Given the steps taken to reduce the number of load cases, symmetrical load cases were removed. Only loads critical to the upper right quadrant, highlighted in Figure 5.6, of the tower were kept. This meant that the output would be a non-symmetrical structure where that upper right quadrant would need to be reproduced on the remaining three corners of the structure. In Figure 5.7 the critical quadrant is illustrated in isolation, next to the final symmetrical structure before any required strengthening work was performed.

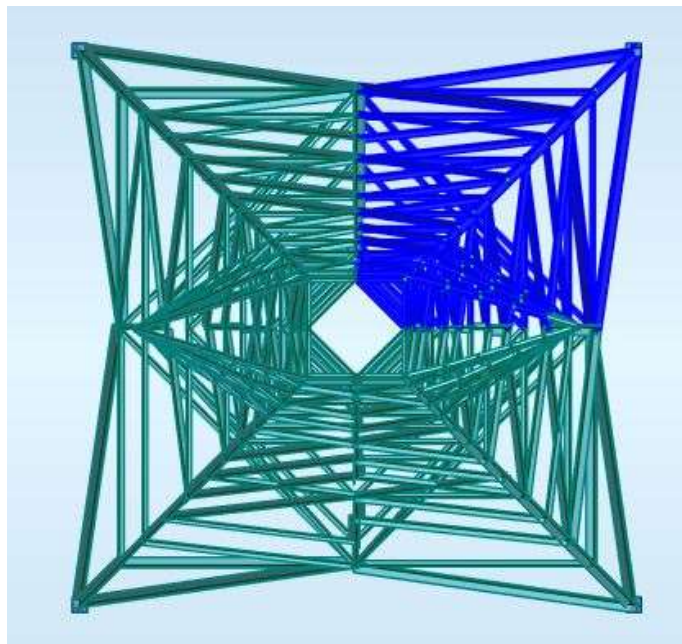


Fig. 5.6 – Plan view of the tower

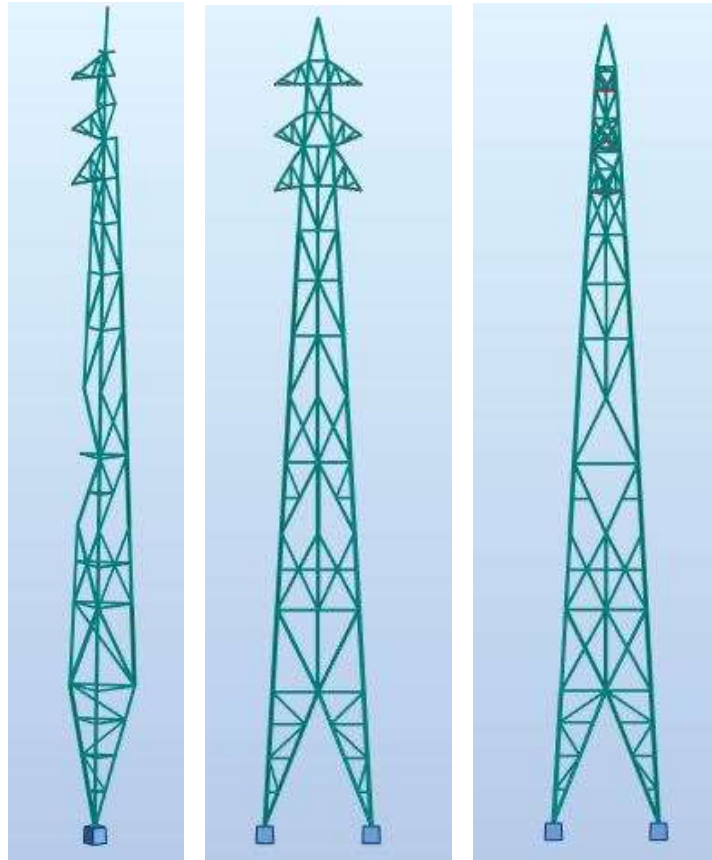


Fig. 5.7 – Critical quadrant, front and side planes

After the symmetry operations, the log file was opened to check if any bars needed additional strengthening. In this case, all the leg members were listed as well as a few bars in the middle of the structure with U/f slightly higher than 1.0. Given the locations of the elements to strengthen, secondary bracing, with section L 40x40x5, was used instead of larger sections.

Figure 5.8 details how the leg members were strengthened. A similar triangulation method was used for the other bars on the list.



Fig. 5.8 – Legs with secondary bracing added

5.3.2. ANALYSIS

The final structure has a total weight 6.8 tonnes, which corresponds to a 10.4% material reduction in comparison to the Metalgalva design. The weight is distributed between the different sections as follows:

Table 5.3 – Final structure weight distribution

Sections	Number	Total weight (kg)
L 40x40x5	72	330
L 50x50x5	281	1041
L 60x60x6	84	858
L 70x70x70	86	1482
L 100x100x10	55	3110
L 150x150x15	0	0
		6821

Adding to the material savings, the list of sections used is also smaller, allowing for more efficient manufacturing.

Given the current concerns about sustainability, savings will be analysed in terms of reduction of CO₂ emissions by using less steel. The national grid of the United Kingdom will be used as the data needed for the analysis is publicly available.

According to the European Strategic Energy Technologies Information System, in 2012 the European steel industry emitted 2.3 tonnes of CO₂ per tonne of steel. The report also identifies paths for improvement that could reduce this value by 70% to 0.7 tonnes of CO₂ per tonne of steel. For this analysis, the best-case scenario (the improvements were successfully applied to the European steel industry) will be used.

Data from the National Grid (UK) website points that there are 88000 electricity pylons in the UK and their average weight is 30 tonnes.

Using this information if a 10% reduction in material usage was applied to every pylon, 2 376 000 tonnes of CO₂ could be saved on the entire grid.

To make some sense of these numbers, data from the Environment Protection Agency (US) was used to compare these emissions with other activities that equate to similar emission levels, 2 376 000 is equivalent to:

- 501 891 Passenger vehicles (with average US yearly mileage) driven for one year;
- 11920 round trips to the moon in a passenger vehicle;
- 754 038 Tons of waste recycled instead of landfilled;
- 1 010 million litres of petrol burned;
- 600 installed wind turbines;
- Energy to power 250 000 homes for a year;
- 70 % of the yearly emissions of a coal-fired power plant.

5.4. CONCLUDING REMARKS

This first version of the program, validated the use of genetic algorithms to automate to a certain degree the design of optimised lattice tower structures, returning gains in productivity and material efficiency.

The returned structure also had fewer cross-section types than the original case study model. Such change is extremely relevant as it implies a reduction of material waste during the fabrication phase.

Having two different truss planes – symmetric in opposite faces – changes the fabrication phase as left and right sections of each bar need to be identified. According to Metalogalva, such a change is not a matter of concern as the fabrication phase is mostly automated. The assembly phase has a slight increase in complexity as each element needs to go not only to the correct place in the truss but also to the correct side of the structure.

5.4.1. FUTURE WORK

During development and testing, some points to improve in a future iteration of the software were also identified.

The Genome class, responsible for the DNA generation should be updated to ensure symmetry of the structure even when loads are reduced, as they were in the case study, to its most critical loads for a specific quadrant. This would shorten the execution time of the optimisation by reducing the number of load cases applied to the structure (by removing the symmetrical LCs). This avoids the need for human operations to re-establish symmetry of the solution. This could be implemented by defining a master quadrant, from which every bar in the remaining three corners of the structure would inherit its properties. The same approach should also be applied to the nodes.

When scaling up to optimise larger structures a constraint in the Robot API was identified. The calls needed to update bar properties between each individual evaluation, take too much time when compared to other calls such as the ones responsible for running and retrieving results from the structural analysis. In fact, when the bar count increases above a certain number, the communication between the developed program and Robot are slow enough to make the application stop responding. In a future version, changing the Robot_call class to work with another structural analysis package such as OpenSees or Oasys GSA will deliver higher performance and will enable more complex optimization problems to run.

Finally, to address the need for human intervention to read the log file and add secondary bracing to the model, a future academic work could study the implementation of a neural network to read input from that file and the structure, and automatically apply the needed bracing elements.

