Summary

By this thesis work, I had an opportunity to study the topic of optimal networks in detail, which are closely related to the problem of Steiner trees. In the meantime, I have recognised the fact, that the problem, which seems to be simple at the first sight is fairly complex, so called NP-hard. Firstly, I have reviewed the commonly used heuristics, which provide acceptable solutions for similar problems. Unfortunately, most of them were unable to obtain the optimal result.

In the aim of more efficient solutions, I have started to examine the agent based heuristics, as the main topic of the thesis work. There were investigated many of the agent based optimisation methods, furthermore 3 frameworks for adapting them to the Steiner problem. Their literature has been reviewed in the work, some of the novel methods have been tried, for instance the AFSO and the EBA methods. In the case of new methods, I have mentioned some deficiencies of the literature and the performance problems of the heuristics.

For the experimentation and measurements, I have designed and implemented a web application. It uses the Angular framework and has been written in the TypeScript programming language. The application makes available the analysis of the mentioned 4 agent based method by various parametrisation. I have studied many optimisation options, and some of them resulted in unexpected outcomes. For instance, in the case of matrix implementation in JavaScript, where the unoptimised source code resulted in the best running time. Moreover, for the shortest path calculations, where large differences have been revealed in the calculation time.

Naturally, there have been remained possibilities for further improvements. For instance, by implementing other algorithms for generating the height maps, by ruling out the water parts of the map from the pathfinding stage, by implementing diagonal steps for the agents, by examining the possibilities of parallelisation, by adding further exit conditions for the optimisation methods or by improving the performance by using WebAssembly.

At the last part of the work, we can find many examples of the usage of the implemented application as test runs. Based on these results, the optimal parameters of the methods have been estimated and compared. Finally, I have provided a guide, which is based on the achieved results, for selecting the proper method in the presented application as a conclusion.