

Project 3 – Nature-Inspired Algorithms

Lawrence Benson, Willi Gierke, Nico Ring

10th July 2017

1 Algorithm Modifications

Most of our algorithm is based on the description given in the task. We adapted the algorithm a bit as described in *Stützle and Hoos paper*. We changed the following things:

- Init all pheromones with τ_{max} instead of τ_{min} . This allows for a larger exploration of the problem space at the beginning.
- We don't use 1 ant per iteration, but 5. This gives a small speed up, because updating the pheromones is an expensive operation.
- Furthermore, we don't always lay the pheromones according to the global best tour (so far), but also use the best tour from an iteration. This allows us to explore more paths instead of running into a local minimum and not getting out again, as the pheromones are too strong.
- If we have not improved for 5000 iterations, we reset the pheromones to get out of the local minimum.
- After 3000 iterations without improvement, we increase the pheromones on all edges for the same reason as initializing all edges with τ_{max} .

2 Evaluation

2.1 Effect of α and β

We evaluated the effect on the convergence to the optimum of the weights α and β for the pheromones and edge weights respectively. Therefore, we run the algorithm on the *gr202* dataset with different values for both parameters and measured the deviation to the optimum after 5000 iterations. We did this five times for each pair of parameters and took the average of the resulting deviations.

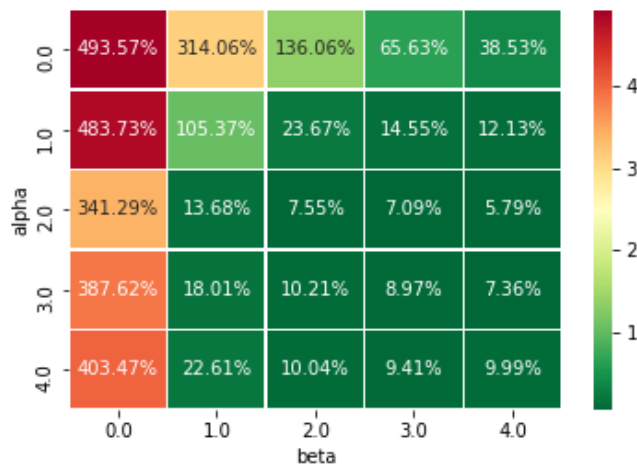


Figure 1: Deviation values for different α and β values after 5000 iterations.

Figure 1 shows the results of testing the effect of α and β on the convergence to the optimum. First of all, we see that if one of both values is zero, thus having no effect on the probability to choose an edge, the deviation

values are much higher compared to the other values. Only for values $\beta \geq 3$ we get a reasonable deviation from the optimum. So only using the edge weight heuristic can produce reasonable results, but only using the pheromone heuristic not. For parameter pairs with $\alpha \neq 0$ and $\beta \neq 0$, $\alpha = 2$ seems to be a good choice for this data set, because for smaller and larger values we get worse results. Furthermore, for this dataset the following holds: the larger beta the better are the results we get. All in all, we conclude that you can achieve the best results if you combine the pheromone and edge weight heuristics.

2.2 Runtime

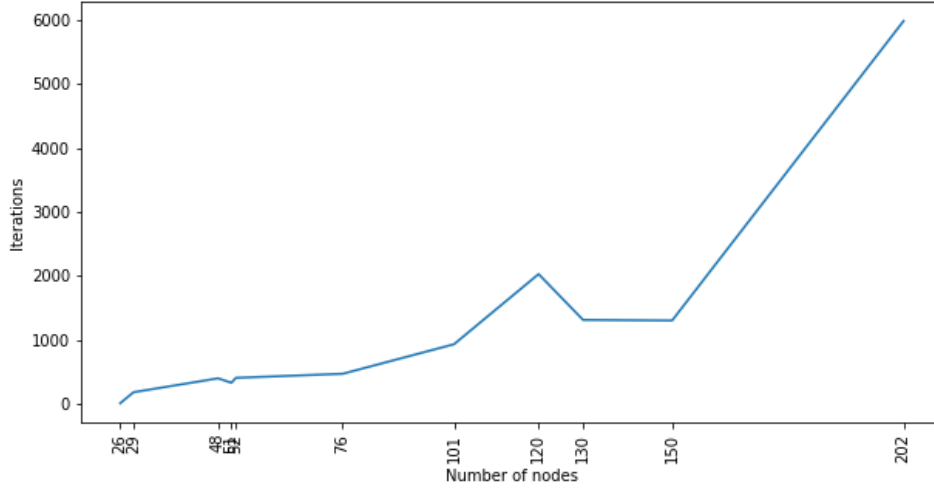


Figure 2: Iterations needed to achieve a deviation of at most 10% from the optimum for different instances.

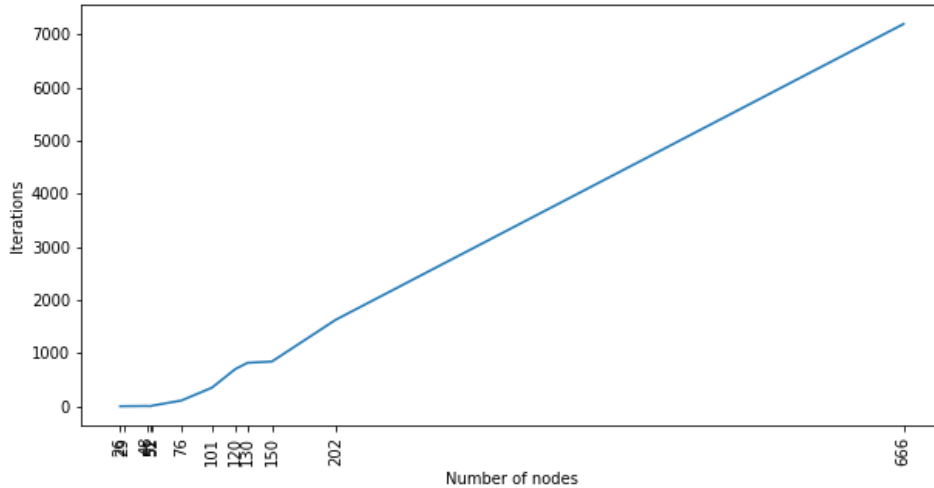


Figure 3: Iterations needed to achieve a deviation of at most 30% from the optimum for different instances.

In Figure 3 we can see that the number of iterations to reach a deviation of 10% from the optimal solution nearly grows linearly in the number of involved nodes. Figure 2 disproves the theory that this might be the general behavior, independent from the number of nodes. Instead, Figure 2 shows that the number of iterations grows exponentially with respect to the number of involved cities. This creates the theory that the number of needed iterations might be heavily dependent on the desired maximum deviation.

Figures 4 and 5 undergrid this thesis. As we can clearly tell from them, the number of iterations grows exponentially with respect to the desired maximum deviation from the optimal solution regarding different instances.

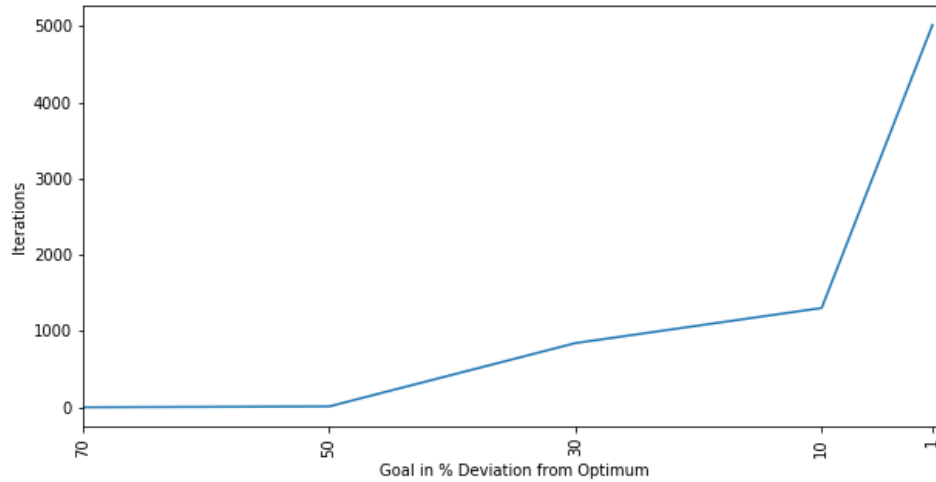


Figure 4: Iterations needed on an instance with 150 cities to achieve different deviations from the optimal solution.

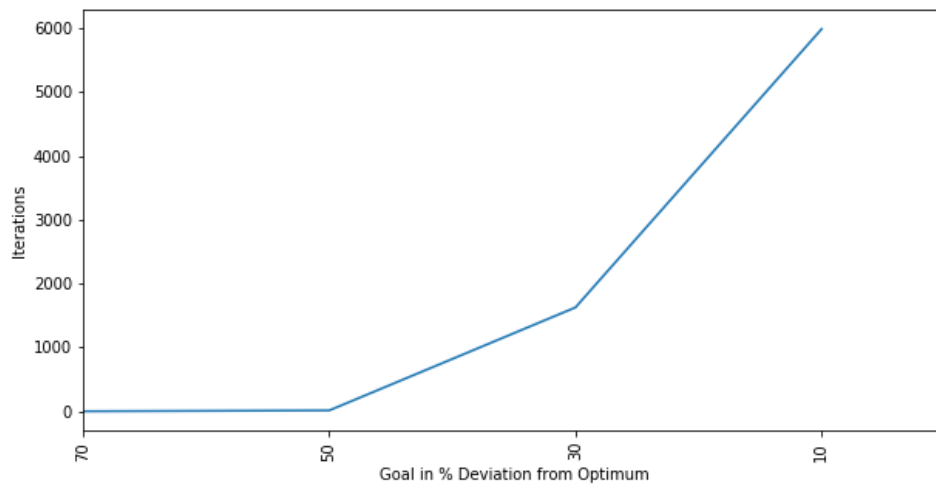


Figure 5: Iterations needed on an instance with 202 cities to achieve different deviations from the optimal solution.

3 Bonus Tasks

Our setup for the following tasks included a MacBook Pro running a 2 GHz Intel Core i7 processor with 8 GB 1600 MHz DDR3 main memory. For bigger instances we used a server hosted by the FSOC lab powering 40 2 Ghz cores and 128 GB RAM. Every task ran on one core with one thread.

3.1 40+ cities optimally

We solved the instance *berlin52* with 52 cities optimally in 4659 iterations. The tour can be found in *berlin52.res* in the zip file.

3.2 100+ cities at 1%

We solved *ch130* with 130 cities within 1% in 6067 iterations. The tour can be found in *ch130.res* in the zip file.

We solved *ch150* with 150 cities within 1% in 5825 iterations. The tour can be found in *ch150.res* in the zip file. It took approximately 2 hours to compute the solutions.

3.3 1000+ cities at 30%

no :(

3.4 Largest instance at 50%

We solved *pla7397* with 7397 cities within 50%. We solved this 3 times within the first iteration. We observed that this instance seems to be easy to solve in general. We observed that it can be solved with heuristic information alone, i.e. we set $\alpha = 0$, so no pheromones were used and also managed to solve this instance in a few iterations. The tour can be found in *pla7397.res* in the zip file. It took us approximately 5 hours to compute the solution.