NI-KOP Homework 4

Patrik Jantošovič

December 8, 2020

1 Assignment

The aim of this homework is to acquaint with one of the advanced heuristics, by solving a simple Knapsack problem. By "playing" with the parameters of the heuristic you will be able to better understand its principles and behavior.
Full text of assignment can be viewed here: https://moodle-vyuka.cvut.cz/mod/assign/view.php?id=89702

2 Implementation

I chose Simulated Annealing and implemented it according to presentations:

```
Solution state = new Solution();
 Solution best = new Solution();
 Double temp = this.initTemp;
 while(!frozen(temp)){
   this.currentSteps = this.steps;
   while(!equilibrium()){
     state = tryNext(state, temp, bag);
     best = better(state, best, bag.MaxWeight);
   }
   temp = cool(temp);
 }
```

with try method as follows:

```
Random random = new Random();
Solution randomNeighbour = getRandomNeighbour(state, bag);
if(randomNeighbour.Price > state.Price) return randomNeighbour;
double delta = randomNeighbour.Price - state.Price;
if(random.nextDouble() < Math.exp(- delta / temp))
  return randomNeighbour;
return state;
```

The full implementation can be viewed here: https://github.com/PatrikJantosovic/NI-KOP-Homeworks

3 Parameters

I tested it on 50 instances of 40 items. I looked at computational time and relative/maximum error on these instances while changing parameters. After some testing, I fixed the value of parameters to:
Fixed steps (equilibrium): 4000
Cooling factor: 0.85
Initial temperature: 1 000 000

Final temperature: 1

And then I changed the values of one parameter at the time, and the results can be seen on the following graphs.

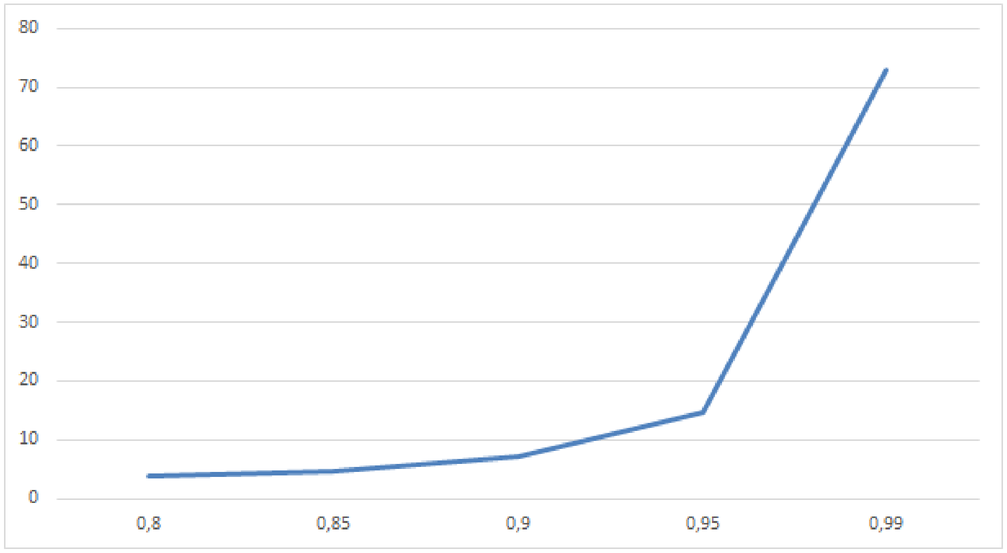

Figure 1: Computational time in seconds for cooling factor.

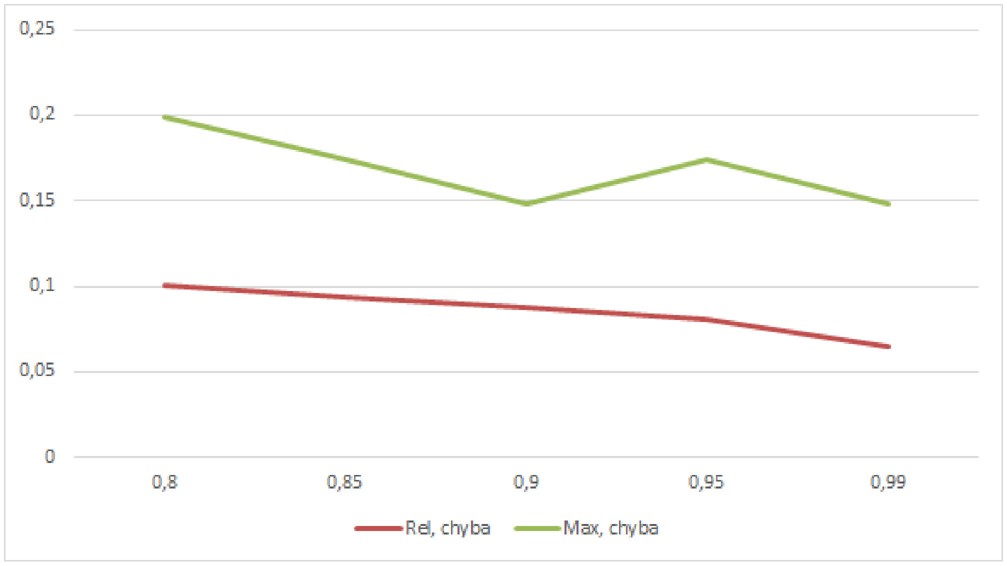

Figure 2: Average and Maximum error for cooling factor.

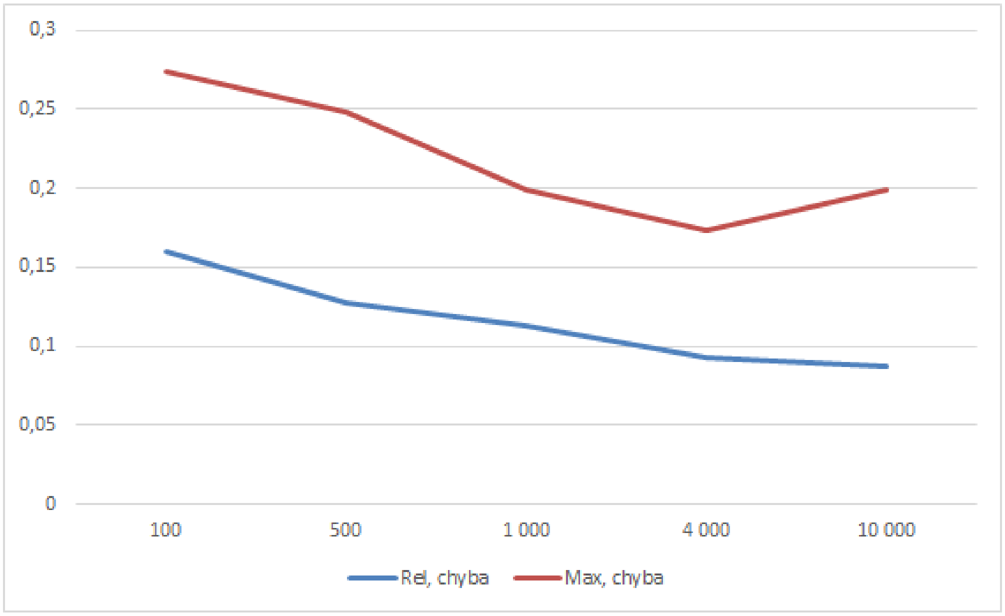

Figure 3: Computational time in seconds for steps.

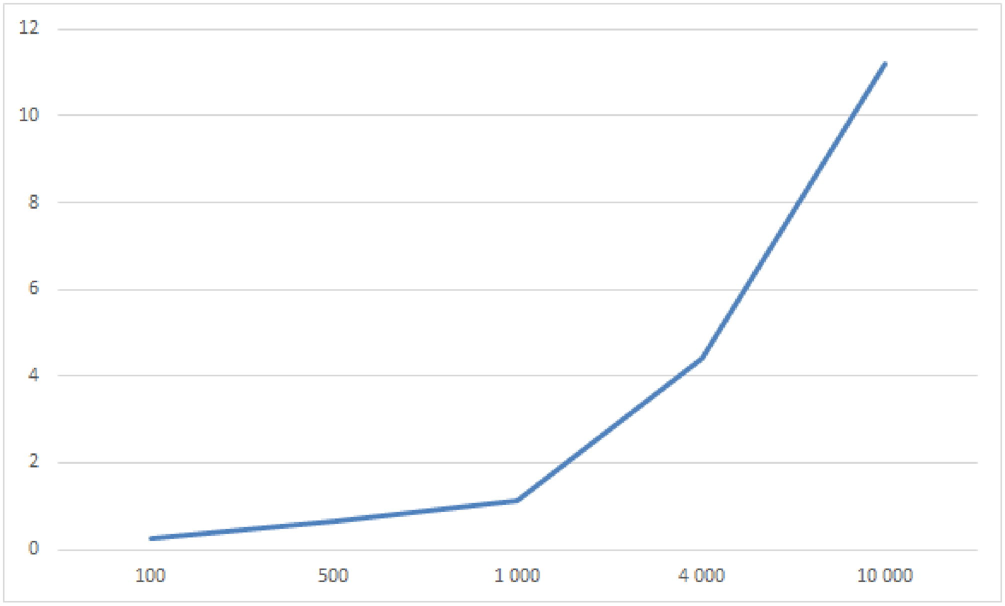

Figure 4: Average and Maximum error seconds for cooling factor.

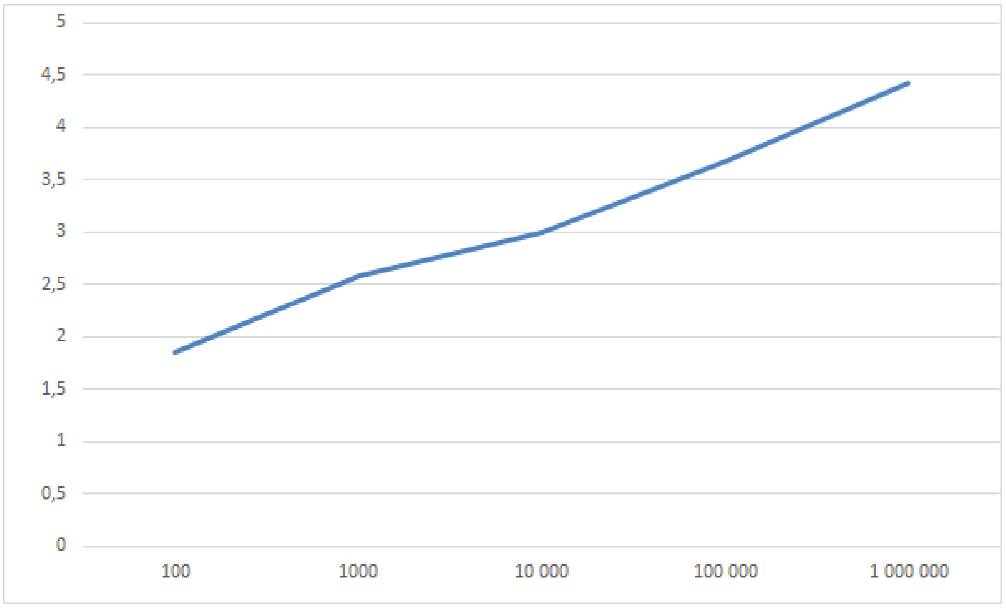

Figure 5: Computational time in seconds for initial temperature.

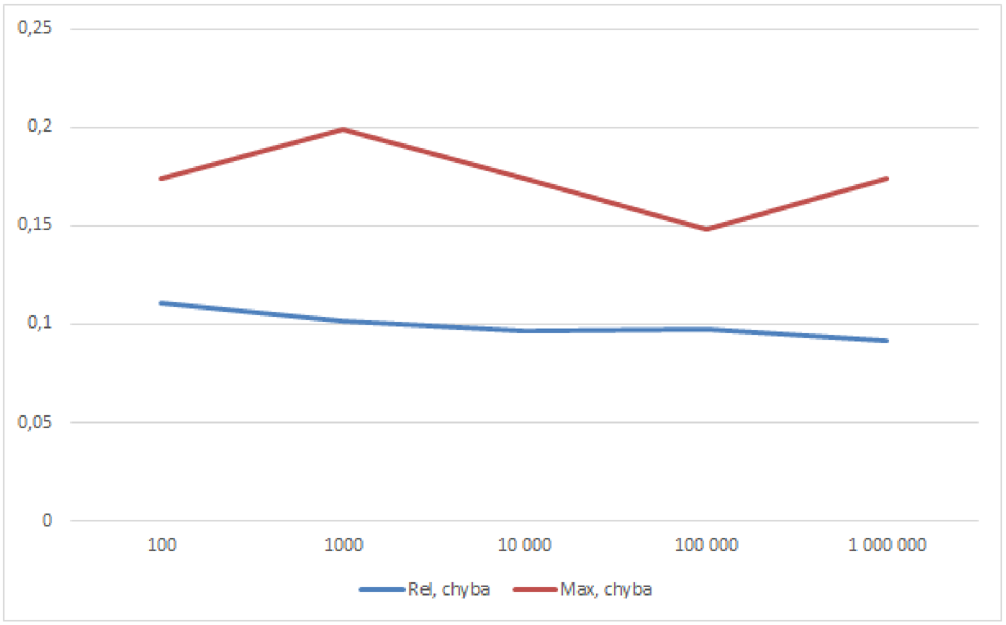

Figure 6: Average and Maximum error seconds for initial temperature.

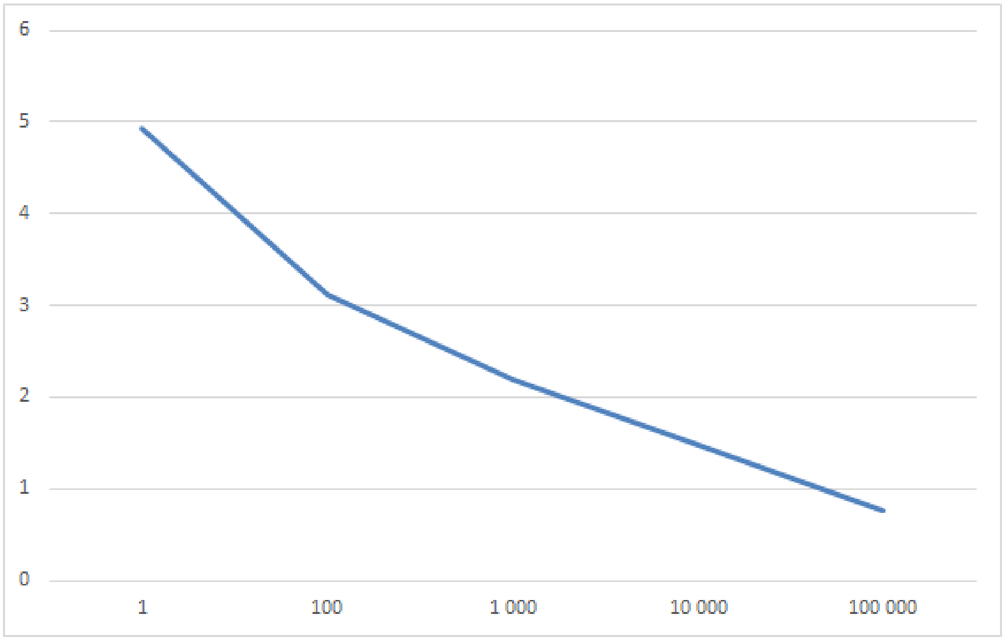

Figure 7: Computational time in seconds for final temperature.

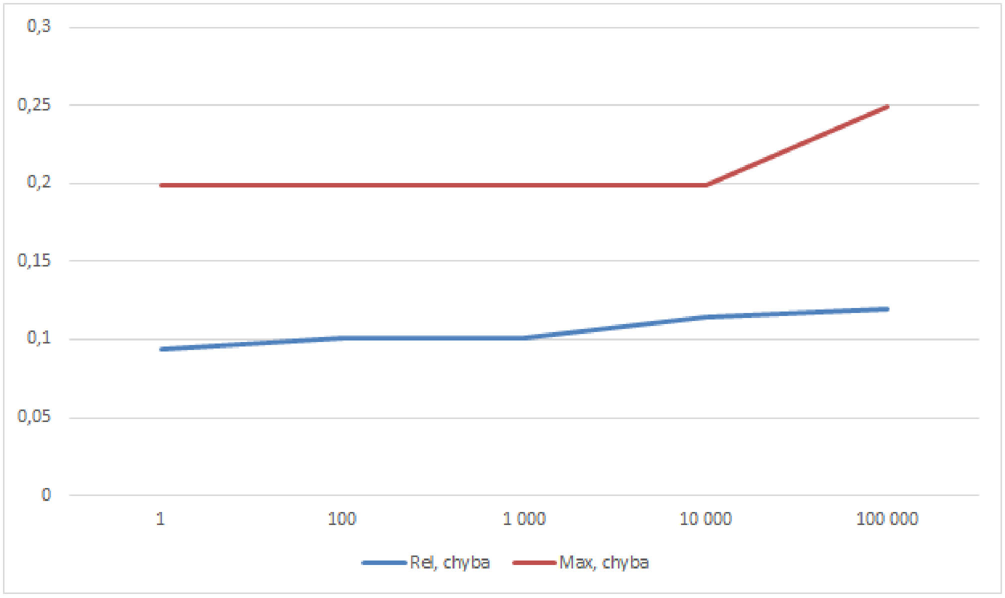

Figure 8: Average and Maximum error seconds for final temperature.

4 Summary and Conclusions

At the end I have to admit that at the start i did not pick the most "optimal" fixed parameters as there were clearly configurations for which algorithm ran faster with comparable success rate.

Anyway, trends can still be clearly seen, showing that with widening temperature range or increasing the number of steps the algorithm visited more states/neighbours hence it was slower but more accurate in both average and maximum error.