Artificial Intelligence for Robotics – Lab

Prof. Dr. Erwin A. Prassler: *erwin.prassler@h-brs.de*

Maximilian Schöbel: *maximilian.schoebel@h-brs.de*

Patrick Nagel: *patrick.nagel@h-brs.de*

Assignment 5

Due Date: Sunday, 4.12.2017, 08:00

1. Programming Assignment: The Traveling Salesman Problem

    - On LEA you will find two input files, which contain cities throughout the world along with their locations. You can assume kartesian distances between the cities.
    - The goal of your agent is to find a cycle (a roundtrip) which visits every city once, while traveling the minimal possible distance. For more information see: `https://en.wikipedia.org/wiki/Travelling_salesman_problem`.
    - In this assignment you will find solutions to the Traveling-Salesperson problem by implementing random-restart hill-climbing and simulated annealing.
    - You can use the smaller file `49_cities.txt` to test your program. Submit the final results on all cities (`cities_full.txt`).
    - There are multiple flavors of the hill climbing algorithm: 1) Russel and Norvig describe the *steepest-ascend variant* of hill climbing. Given a state, **all possible sucessors** are being evaluated and the one with the highest/lowest objective function is chosen as the new state. 2) In the *simple hill climbing variant*, sucessors of the current state are being evaluated until **any better one** is found, which is then selected as the next state. (For more information see "variants" here: `https://en.wikipedia.org/wiki/Hill_climbing`).
    - Implement the two above mentioned versions of hill-climbing and simulated annealing. Compare their performance regarding resulting cycle length. Since the hill climbing algorithm can easily get stuck in local optima, restart each variant at least five times and report the best result. For simulated annealing, report results (cycle lengths) for runtimes of 1 minute, 5 minutes and 20 minutes.
    - Do random-restart hill-climbing in both your implementations and for all target runtimes, that means restart each hill-climbing algorithm at least 5 times and report the best result.
    - Plot your best cycle for each runtime.

    Summary:

    - Implement the steepest-ascent random restart hill-climbing algorithm with (at least) five restarts.
    - Implement the simple random restart hill-climbing algorithm with (at least) five restarts.
    - Implement simulated annealing and report results for 1, 5 and 20 minutes.