Universidade do Minho Academic Year 2019/2020

Mestrado Integrado em Engenharia Informática

Algoritmos Paralelos / Parallel Algorithms

Work 1: Monte-Carlo Travelling Salesman Problem

Introduction Given n points by its (x, y) coordinates, $D(i, j)$ holds the distance between each two points $(x(i), y(i))$ and $(x(j), y(j))$. The problem is to find the shortest route that starts at one of the points (”towns”), randomly choosen, goes once through every other town and finally returns to the starting point.

In the Blackboard, folder ”Evaluation Projects”, you may find the following files:

- *travelingGreedy.m* This Matlab code implements a ”greedy” algorithm. For each journey (but the last one, when the salesman returns to the starting point), the closest town not yet visited, is always choosen.
- *travelingMC.m* This Matlab code implements a Monte Carlo method. It starts with an initial random route (a random permutation of the first n integer numbers) and then reduces the total distance through successive permutation in the order of two neighbouring towns. The new route is accepted only if it reduces the total distance.
- *travelingSA* This is a modification of *travelingMC.m* which implements the so-called “simulated annealing” (to be explained).
- *travelingMAIN* This code randomly generates the points $(x(i), y(i)),\ i = 1, \cdots, n$, (both coordinates are floating point numbers numbers bebetween 0 and 10) and computes the distances $D(i, j)$. Then, it uses *travelingGreedy.m* , *travelingMC.m* and *travelingSA* to get different answers. The code travelingMAIN has two input parameters: the number n of towns and p the number of times to repeat the experience (these p processes are potentially parallel since they are completely independent).

The work to be done by the students (groups of 2)

- To develop codes implementing the different approaches to be tested in the cluster SEARCH. The tests should be carried out using different values of $n(100, 200, 300, \cdots)$, and eventually some of the heuristic parameters in the code may be changed, as well.
- (optional) To develop, code and test different ideas for the TSP.
- To write a short report (maximum 10 pages plus a copy of the code) with a brief description of the problem and its complexity, the algorithms used and a comparison of the results obtained with the different approaches.

Deadline: send pdf file to r_ralha@math.uminho.pt) till march, 16.