Techniques Algorithmiques et Programmation

Traveling Salesperson Problem – Heuristics

Flips and Greedy Algorithm

Given a tour that has already been constructed, we consider a heuristic consisting of removing two independent edges and reconnecting the two pieces of the tour in such a way that they form a new tour of shorter length. This operation is called a "flip" of the edges in question. Independent edges are edges that do not share any common endpoints, meaning they do not follow each other in the tour. The first flip of a tour $v_0 - v_1 - \cdots - v_{n-1}$ is a flip of the edges $v_i - v_{i+1}$ and $v_j - v_{j+1}$ where $i < j$ and where i is as small as possible.

Question 1. *Explain the principle of the flip for edges $v_i - v_{i+1}$ and $v_j - v_{j+1}$ with $i < j$.*

Question 2. *In the case where the independent edges $v_i - v_{i+1}$ and $v_j - v_{j+1}$ cross each other without being collinear, show that their flip yields a strictly positive gain on the length of the tour. (Hint : consider the intersection point and the four segments.)*

We assume that you have already written the function `reverse(T,p,q)` which reverses the part of the integer array `T[p]`...`T[q]` (bounds included) if `p<q`.

Question 3. *Write the function `first_flip(V,n,P)` which, given a tour `P` as input, returns the gain of the first feasible flip (and performs the flip in `P` as output) or 0 if there are none. (Hint : pay attention to the possible indices for i and j.)*

Question 4. *Deduce the function `tsp_flip(V,n,P)` which computes a tour `P` (and returns its length) obtained by applying feasible flips as many times as possible. You may start from the tour of your choice, for example, the trivial tour defined by `P[i]=i`.*

Question 5. *Do your functions always terminate ? Discuss their complexities. Is it possible to obtain a gain after a flip without there being an initial crossing ? If a flip is applied, can we obtain more crossings than before the flip ? Is it possible to have a sequence of flips where the same pair of edges is flipped multiple times ?*

Question 6. *Write the function `tsp_greedy(V,n,P)` providing a tour obtained by choosing at each step the nearest free point to the last point chosen. What is its time complexity ?*

Lab Session

Download (*Save link as...*) the files corresponding to the lab session from the course page available below, and place everything in the same directory as last time :

http://dept-info.labri.fr/~gavoille/UE-TAP/

You will need to edit `tsp_heuristic.c`, comment/uncomment a few lines in `tsp_main.c`, still compile with `make tsp_main`, and run the execution with `./tsp_main`. If your functions enter an infinite loop, remember to add a `(... && running)` in any suspicious test, allowing you to exit the program more easily.

Additionally, you will need to :

1. Complete `reverse()`, `first_flip()`, and `tsp_flip()` in the file `tsp_heuristic.c`.
2. Vary n using the command line. Experiment with `generatePoints()`, `generateCircles()`, and `generateGrid()` to challenge the heuristic. (NB : For a grid where at least one of the dimensions is even, the optimal solution cannot contain any diagonal segments.) You can zoom and move the points with the mouse, and test the `'o'`, `'r'`, and `'t'` keys (as well as the online help with `'h'`). Observe that flips can occur even when there are no crossings.
3. Add the `tsp_greedy()` heuristic. Compare the `tsp_flip()` heuristic alone with `tsp_greedy()` followed by `tsp_flip()`. To do this, modify `tsp_main.c`.