AI CUP 2012

Ioannis Lamprou
University of Lugano

Implementation Information:

The TSP instance read by the the file is converted to an array representation. Hence, two vectors *solution* and *position* are kept and maintained during the course of the program.
Besides the auxiliary functions (most important of which are *swap, invert, shift*), the following algorithms were implemented:

- random permutation (using Knuth shuffle method)
- nearest neighbor heuristic
- nearest insertion heuristic
- farthest insertion heuristic
- 2opt heuristic (using neighbor list idea – to be explained later)
- 3opt heuristic
- random restart (using 2opt)
- iterated local search (using random double bridge & 2opt)

Files delivered:

- TSP.h (header file for all functions created)
- TSP.cpp (implementation of all TSP.h functions)
- main.cpp (main function that uses the TSP functionality)
- results.txt
- Makefile
- README.pdf
- AICupProblems directory including all 10 tsp files
- AI_cup_2012_IoannisLamprou.xls

Compiling and running:

The program was developed in C++ in a Linux environment.
To compile just enter the corresponding directory and type "make".
To remove all object and executable files type "make clean".
The compiler g++ is used for compiling the source code.
To run the program use the "./tsp" executable file together with some arguments:

```
-f <filename>
-a <algorithm> , where <algorithm> = {rand, nn, ni, fi, 2opt, 3opt, ils}
-T <initialTemperature>     (used for iterated local search – default: 100)
-t <frozenThreshold>        (used for iterated local search – default: 1)
-n <number of Iterations>   (used for iterated local search – default: 100)
-s <RandomSeed>             (default: current system time)
-i <firstImprovement>       (give true/false - default: false)
-d <decreaseFactor>         (used for iterated local search – default: 0.95)
-k <neigborListSize>        (default: 15)
```

* -f <filename> -a <algorithm> are two arguments, that without them the program will immediately terminate

Examples of execution:

```
$ ./tsp -f lin318.tsp -a 3opt
$ ./tsp -f rat783.tsp -a ils -n 3000 -d 0.85 -s 1338198609
```

More examples can be found at the results.txt file.

Please perform all your runs on the tsp files I provide you with. They are the original ones, but with some small changes (some white space removed,some info TYPE replaced with COMMENT, etc) that do not affect the substance of each file, but enable secure parsing from my program. (I did not have the time to write a more general way to parse the files, since I emphasized more on the algorithms)

The program will generate an alarm signal after 3 minutes (180 seconds) have passed and will end its execution, returning the best solution found at this point.

Neighbor list idea:

To make 2-opt run faster (O(n) time) a neighbor list idea was implemented. The reference used is the "*The Traveling Salesman Problem: A Case Study in Local Optimization*"(1995) by David S. Johnson and Lyle A. McGeoch and more especially section 3.3 "How to Make 2-Opt and 3-Opt run quickly" (page 25 mostly), where it is demonstrated that "having fixed t1 and t2 , the possibilities for t3 can be limited to those cities that are closer to t2 than is t1" , when you try to find which edge <t3,t4> to exchange with the edge <t1,t2>.
Indeed, applying this idea to 2opt gave much better results in terms of errors (for all 10 instances errors are below 1%) and reduced dramatically the running time, especially for small instances of the TSP problem.

Results:

The most important results come from the execution of an iterated local search using a random double bridge move as a mutation and 2opt as local search.
As aforementioned, all results generated are below 1% error. Results can be found at the results.txt file ,which is of the following format:

```
<program instruction>
<cost of best solution> - <relative error> - <running time>
```

All results were generated at a personal computer with the following characteristics:

Ubuntu Release 10.04(lucid)
Kernel Linux 2.6.32-32 generic
GNOME 2.30.2

3.9GB of RAM
Intel(R) Core(TM)2 Duo CPU T8300 @ 2.40 GHz