`KoptLS` use for JOC experiments

July 5, 2024

Section 5.1 Best move from a random tour

In this section there are experiments on uniform and Euclidean instances on random tours, one move.

Table 1.

- The column CE is computed, since we know exactly how many moves are required by Complete Enumeration
- The column $\mathcal{A}_g$ is obtained by running the command

  ./`KoptLS` -RU n `-A2 4 -NS 1 -NINS 100 -NT 10` [`-SEEDT` s]

  where $n = 2000, 4000, \ldots, 24000$. Optionally, put a seed specification in s (we don't recall what seeds were used in all experiments. When missing, a default `1234` is used)
- Similarly, the column $\mathcal{A}_L$ is obtained by running the command

  ./`KoptLS` -RU n `-A2 3 -NS 1 -NINS 100 -NT 10` [`-SEEDT` s]
- Finally, the column $\mathcal{H}(\delta_n)$ is obtained by running the command

  ./`KoptLS` -RU n `-A2 5 -NS 1 -NINS 100 -NT 10` [`-SEEDT` s]

Table 2.

Results in Table 2 can be obtained in an analogous way:

- The column CE is computed.
- The column $\mathcal{A}_g$ is obtained by running the command

  ./`KoptLS` -RE n `-A2 4 -NS 1 -NINS 100 -NT 10` [`-SEEDT` s]

  where $n = 2000, 4000, \ldots, 24000$.
- Similarly, the column $\mathcal{A}_L$ is obtained by running the command

  ./`KoptLS` -RE n `-A2 3 -NS 1 -NINS 100 -NT 10` [`-SEEDT` s]
- Finally, the column $\mathcal{H}(\delta_n)$ is obtained by running the command

  ./`KoptLS` -RE n `-A2 5 -NS 1 -NINS 100 -NT 10` [`-SEEDT` s]

Table 3.

Table 3 reports results on some TSPLIB instances. TSPLIB instances are publicly available and should be downloaded from the web. The input format for our program is slightly different (simpler) than the TSP format of TSPLIB files.

- The column CE is computed, given n in the 1st column, as $n(n-1)/2$.
- The column $\mathcal{A}_g$ is obtained by running the command

```
./KoptLS -F name -A2 4 -NS 1 -NT 10 [-SEEDT s]
```

where *name* is taken in `rl5915.txt`, ..., `pla33810.txt` (see paper). Each of these files, has been previously created by converting the corresponding TSPLIB file into our format. For example, we obtain `rl5915.txt` by running

```
./convert TSPLIB/euc2d/rl5915.tsp rl5915.txt
```

NOTE: This might create **very large** output files, and might take some time.

- Similarly, the column $\mathcal{A}_L$ is obtained by running the command

```
./KoptLS -F name -A2 3 -NS 1 -NT 10 [-SEEDT s]
```

Times do not include the time spent to generate the instances, which is usually larger for Euclidean than for Uniform instances.

Section 5.2 Best-improvement convergence to a local optimum

Figures 5 and 6.

Figure 5 and Figure 6, left, (or similar figures, depending on the random seed) can be obtained by running

```
./KoptLS -RU 1000 -A2 4 -Wmove nameM -Wtime nameT [-SEEDT s]
```

This will create a text file *nameM*`.00.00` which consists of a list

```
1 m_1
2 m_2
..
N m_N
```

where each m_k is the number of moves that were evaluated by $\mathcal{A}_g$ to find the best one at the k-th step of local search.

The command will also create a text file *nameT*`.00.00` which consists of a list

```
 1 t_1
 2 t_2
 ..
 N t_N
```

where each t_k is the time (in secs) spent by $\mathcal{A}_g$ to find the best move at the k-th step of local search.

To obtain figure 5, one can plot the values m_k from the file *nameM*`.00.00`. To obtain figure 6, one can compute from the file *nameT*`.00.00` some cumulative values T_k defined as

$$T_k := \sum_{i=1}^{k} t_i$$

and then plot the values T_k.
Figures 5 and 6, right, can be obtained in exactly the same way, by replacing, in the command line,

```
-RU 1000
```

with

```
-RE 1000
```

Note that, in Figure 6, there are times also for the algorithms $\mathcal{A}_L$, and CE. To get these times for $\mathcal{A}_L$ it is enough to run the commands (with the same seeds as used before, if any)

```
./KoptLS -RU 1000 -A2 3 -Wtime nameLU [-SEEDT s]
```

and

```
./KoptLS -RE 1000 -A2 3 -Wtime nameLE [-SEEDT s]
```

This will create the two files *nameLU*`.00.00` and *nameLE*`.00.00` containing the times taken for each move by $\mathcal{A}_L$. The uniform case is used for figure 6 left, while the Euclidean case is for figure 6 right. The values to plot can be obtained from these files, as described above for cumulative times.

Finally, the same steps, after replacing in the command line

```
-A2 3
```

with

```
-A2 0
```

will create the files with the times relative to Complete Enumeration (CE).

Figure 7.

Figure 7 is concerned with the TSPLIB instance `pr1002`. We can convert it into our format by

```
./convert TSPLIB/euc2d/pr1002.tsp pr1002.txt
```

Then, the times for CE and for $\mathcal{A}_L$, necessary and sufficient to create the figure, can be obtained by the commands

```
./KoptLS -F pr1002.txt -A2 0 -Wtime nameC [-SEEDT s]
```

```
./KoptLS -F pr1002.txt -A2 3 -Wtime nameL [-SEEDT s]
```

Table 4.

Table 4 is divided into 3 blocks of rows, for uniform, euclidean and TSPLIB instances.

Block UNI. Within a row block, each line has 3 column blocks, corresponding to CE, $\mathcal{A}_L$ and $\hat{\mathcal{A}}_L$ algorithms, and reporting averages for number of moves per step and total convergence time.

Each line correspond to a value of $n = 500, 1000, \ldots, 3000$ and is the average over 10 runs.

The values for CE can be obtained via the command

```
./KoptLS -RU n -A2 0 -NT 10  [-SEEDT s]
```

At the end, the average convergence time and average number of evaluations per step will be displayed on the screen.

In a similar way, one can obtain the values for $\mathcal{A}_L$ by replacing

```
-A2 0
```

with

```
-A2 3
```

in the above command line.

Finally, the values for the hybrid algorithm $\hat{\mathcal{A}}_L$, with switch at $3/4 \times n$, can be obtained via the command

```
./KoptLS -RU n -A2 3 -NT 10 -SWITCH2COEF 0.75  [-SEEDT s]
```

Block EUC. The block for Euclidean instances can be obtained by repeating the exact same steps as for the UNI block, with the exception of replacing

```
-RU n
```

with

```
-RE n
```

Block TSPLIB. This block concerns the set S of TSPLIB instances `rat575`, `pr1002`, `u1432`, `u2152`, `pr2392` and `pcb3038`. We need to convert each $s \in S$ to our format via the command

```
./convert s.tsp  s.txt
```

At this point, we proceed for each instance exactly as in block UNI,with the exception of replacing

```
-RU n
```

with

```
-F s.txt
```

Section 5.3 Experiments with first-improvement

Table 5

The table has 4 blocks of rows, denominated U, E, T, and H. It also has 3 blocks of columns, relative to first-improvement with, respectively, first-improvement with the algorithms CE, $\mathcal{A}_L$ and best-improvement with $\hat{\mathcal{A}}_L$. Each column block is then subdivided into 4 columns, labeled f^*, $\bar{f}$, $\bar{t}$ and "steps".

Block U. The block contains lines for $n = 500, 1000, 1500, 2000$ and 2500.

First, we run best-improvement 10 times, with $\hat{\mathcal{A}}_L$ via the command

```
./KoptLS -RU n -A2 3 -NT 10 -SWITCH2COEF 0.75  [-SEEDT s]
```

On the screen, we get the values for best tour found (column f^*), average tour found (column $\bar{f}$), average time per run (column $\bar{t}$) and average number of LS steps (column "steps").

The total time T needed for all the runs of best-improvement (which is shown on screen, or can be computed as $T := 10\bar{t}$) is then used as a time limit to compute the blocks of first-improvement with CE and $\mathcal{A}_L$. We run the following command, with $x = 0$ for CE and $x = 3$ for $\mathcal{A}_L$:

```
./KoptLS -RU n -A2 x -NT 1000 -DOFINDBEST 0 -TLIM T [-SEEDT s]
```

Notice how we put 1000 (or a high number) as the number of FI convergences, so that the termination will be due to the time limit and not to having reached the number of convergences.

Block E. Block E is computed exactly as Block U, with the exception that

```
-RU n
```

is replaced by

```
-RE n
```

Block T. This block concerns the set S of TSPLIB instances `rat575`, `d657`, `pr1002`, `u1060`, `rl1323`, `u1432`, `u2152` and `pr2392`. We need to convert each $s \in S$ to our format via the command

```
./convert s.tsp  s.txt
```

At this point, we proceed for each instance exactly as in block U,with the exception of replacing

```
-RU n
```

with

```
-F s.txt
```

Block H. This block concerns the set S of "hard tsp instances" (as described in Hougardy and Zhong (2001)) `Tnm511`, `Tnm1021`, `Tnm1501`, `Tnm2011` and `Tnm2521`. The user should download the generator of these instances from the web, as provided by the authors. The authors made available a `.cpp` program which creates these instances, in the TSPLIB `.tsp` format. Once the instances have been generated, they must be converted to our format by using our converter, like, for example,

```
./convert Tnm511.tsp Tnm511.txt
```

At this point, we proceed for each instance exactly as in block T.

Table 6

Table 6 is built exactly as Table 5, the only difference being that each LS convergence is started at a Nearest Neighbor tour rather than at a random tour. In order to start at a Nearest Neighbor tour, one must add to each command line the option `-ST NN`. For instance the command for best-improvement, with $n = 1000$, Uniform instances, would become

```
./KoptLS -RU 1000 -A2 3 -NT 10 -SWITCH2COEF 0.75 -ST NN [-SEEDT s]
```