1 Introductory remarks

We currently have only very limited experiences with different data-sets. Constructing feasible solutions can be, depending on the availability of shuttle vehicles, difficult. In case of the public\_preselection\_1, public\_preselection\_2 and private\_preselection\_instance\_1-instances, a good number of shuttles is present, so that the naïve construction of a first feasible solution is easy. Our current solution approach therefore favors the identification of qualitatively good solutions over the identification of a first feasible alternative. This is a conscious choice in the light of the given instances.

We have tested our algorithm on modified versions of the data-sets, but cannot give a general guarantee whether it always works.

2 Command line options

2.1 Parameters

- **-path string** (optional; specifies the path from where the problem-data is loaded; default value is the directory where the executable is stored)
- **-runningtime int** (optional; specifies the running time in seconds; default value is 600)
- **-solutionfile string** (optional; specifies the path and filename of the solution file; default filename is Solution.csv, default path is the path of the problem-data; note that the program always immediately saves the current best-know solution to disk)
- **-threads int** (optional; specifies the number of threads to be executed in parallel; default value is the number of (logical) processors + 1)
- **-showstats** (optional; forces the program to report some statistics during the optimization run: a counter of the synchronization steps, the worst element of the elite archive, the number of perturbations per second (p/s), the number of evaluations per second (e/s))
- **-info** (displays the info so that the program can be identified)

2.2 Examples

- **CTSSP.exe -runningtime 60 -threads 4**
  runs the program with 4 threads for 60 seconds on the data in the same directory as CTSSP.exe

- **CTSSP.exe -path F:\VSC2015\data1 -solutionfile F:\VSC2015\data1\Sol1.csv -showstats**
  runs the program for 600 seconds on the data in the directory F:\VSC2015\data1, stores the best found solution in F:\VSC2015\data1\Sol1.csv, and displays some statistics during running time; the number of threads is set to number of (logical) processors + 1
3 Identified results

- example-instance: optimal value is 417
  Note that a solution with 417 is found within a running time of a few milliseconds. We believe that it is difficult to assess the performance of a reasonable heuristic based on this instance.

- public_preselection_1: optimal value is at most 298,471.33
- public_preselection_2: optimal value is at most 664,649.08
- private_preselection_instance_1: optimal value is at most 493,554.77

4 Comparison of programs April→May, 2015

We have conducted some experiments on several data-sets. The following results are based on ten independent runs, 600s on an Intel i7-920 CPU (4 native cores, 2.67 GHz, HyperThreading=ON).

The average results of our program from April 2015 are given in blue, the ones of the (final) submission from May 2015 in green. The black line shows that value of the best-know solution (i.e. best-know from our perspective). We compare running time in seconds (horizontal axis) versus costs (vertical axis).

![Figure 1: Results for public_preselection_1](image)
Figure 2: Results for `public_preselection_2`

Figure 3: Results for `private_preselection_instance_1`
5 Other contributions

We have implemented a visual interface that allows the fast inspection of the obtained solutions. While the visualization is not particularly innovative, it is comfortable to use, as solutions can be checked by drag-and-drop.