Hybrid Genetic Search for arc routing problems
HGS-CARP v1.0

Thibaut VIDAL

March 2016

1 License

2 Scope

HGS-CARP can solve Capacitated Arc Routing Problems (CARP) and Mixed Capacitated General Routing Problems (MCGRP) with possible Turn Penalties (MCGRP-TP) The code receives as input CARP instances in the format of `http://www.uv.es/belengue/carp.html`. It can also receive as input the MCGRP instances according to the format of `https://www.sintef.no/projectweb/top/nearp/`, as well as the MCGRP-TP instances included in the archive.

Furthermore, the code can address problem instances with Multiple Depots (MDCARP) or multiple visit periods (PCARP), as well as the Min-Max Windy Rural Postman Problem (MM-kWRPP), which requires to minimize the length of the maximum route in the presence of asymmetric travel costs on edges. The problem instances for the MDCARP and the PCARP are direct extension of the CARP instances with fixed rules (the same for all instances) for allowed day combinations and depot choices. These characteristics are simply initialized in the source code, when reading the files. It is possible, with minor effort, to extend the code dedicating to reading the input, in **`Params.cpp`**, to read different file formats.

Finally, since the PCARP and MM-kWRPP require the ability to minimize the fleet size and length of the maximum route, respectively, we added the option to iteratively decrement the fleet size or the distance constraint as long as a feasible solution is encountered, and stopping in case of infeasibility. This process, visible in the file **`main.cpp`**, allows to solve efficiently these two variants.

This source code is connected to the article: *[1] Vidal, T., 2016. Node, Edge, Arc Routing and Turn Penalties: Multiple problems – One Neighborhood Extension.* This paper should be cited when using the source code for academic purposes. Any suggestion of improvement, comment or bug report is very welcome.

3 Running the algorithm

A makefile is provided with the source code. To compile, go to the directory "Program" and execute the command **make**. Then, the most basic use of the program requires to run the following command:

```
./gencarp pathInstance -type problemType
```

Instance formats are not unified in the vehicle routing literature, it is necessary to specify the type of the problem to be able to read the instance accordingly. The list of available problem types is the follows:

```
type 30:  CARP
type 31:  MCGRP
type 32:  PCARP
type 33:  PCARP
type 34:  MCGRP-TP
type 35:  MM-kWRPP
```

It is also possible to use the following options:

[-sol pathSolution]: This option allows to specify the output location and name for the solution file. The default value is **pathInstance.sol**.

[-t timeLimit]: This option allows to specify the maximum CPU time for the resolution, in seconds. The default value is 300 seconds (5 minutes).

[-s randomSeed]: This option allows to specify the random seed for the run. Specifying **0** allows to generate the seed from the current computer time. This later option is set by default

[-veh nbVehicles]: Some instances do not specify the number of vehicles. In this case, an upper bound should be specified in the line of command.

[-dep nbDepots]: The MDCARP requires to specify the number of depots (from 2 to 6) for the instances used in the paper.

4 Instance Files

The original CARP instance files provided by José Manuel Belenguer in Universitat de Valencia, the NEARP instances provided by SINTEF, the MM-WRPP instances provided by Ángel Corberán, Isaac Plana, and José María Sanchis are included in one folder of the archive in their original form with the permission of their respective authors, along with a description of their format in the file **README.txt** in each folder. The new MCGRP-TP instances are also included, as well as the PCARP instances used and provided by Nacima Labadie, which only differ from the CARP instances in terms of vehicle capacity, and follow a slightly different format.

5 Solution Format

The solution file first includes four lines, which specify respectively the objective function, the number of routes, the CPU time (in seconds) of the run, and the CPU time (in seconds) to reach the best solution of the run.

Then, each line of the solution file corresponds to a route in the solution, indicating the index of the depot, the day, the index of the route, the total demand on the route, the cost of the route, the number of visits in the route (depot visits and services), and finally a sequence of triplets (X i,j,k). In each of these triplets, $X \in \{C, D\}$ stands for a depot visit (D) or a service (S), i is the index of the service (as considered by UHGS), and j and k are the extremities of the edge which is serviced, in the order of the service orientation. To evaluate or draw such a solution, the endpoints of services should be connected via shortest paths.