Benchmark solutions for: "A Multistage Optimisation Algorithm for the Large Vehicle Routing Problem with Time Windows and Synchronised Visits"

Mateusz Polnik[a,*], Annalisa Riccardi[a], Kerem Akartunalı[b]

[a] *Mechanical and Aerospace Engineering, University of Strathclyde, 75 Montrose Street, Glasgow, G1 1XJ, UK*
[b] *Management Science, University of Strathclyde, 130 Rottenrow, Glasgow, G4 0GE, UK*

Abstract

The document describes the dataset of solutions to the popular set of benchmark instances for the Vehicle Routing Problem with Time Windows and Synchronised Visits created by Bredström and Rönnqvist. The solutions were computed using the multistage optimisation algorithm described in the paper "A Multistage Optimisation Algorithm for the Large Vehicle Routing Problem with Time Windows and Synchronised Visits" written by the authors of this document.

The document explains the file format and the structure of the dataset of solutions to the set of benchmark instances for the Vehicle Routing Problem with Time Windows and Synchronised Visits (VRPTWSyn) created by Bredström and Rönnqvist [1]. The solutions were computed using a multistage optimisation algorithm proposed by the authors of the document [3]. The dataset is released on the Open Data Commons Open Database License (ODC-ODbL).

The document is structured as follows. Section 1 briefly explains the design of the benchmark instances. Section 2 describes the naming convention used for files. Section 3 details how files are organised in the data set. The final section of the document explains the file format.

1. Benchmark of the Vehicle Routing Problem with Time Windows and Synchronised Visits

Solving the problem instances from the VRPTWSyn benchmark of Bredström and Rönnqvist [1] became an established practice to demonstrate the effectiveness and to compare solution methods proposed in the literature. The benchmark consists of 50 VRPTWSyn problem instances generated using ten

*Corresponding author
Email addresses: `mateusz.polnik@strath.ac.uk` (Mateusz Polnik), `annalisa.riccardi@strath.ac.uk` (Annalisa Riccardi), `kerem.akartunali@strath.ac.uk` (Kerem Akartunalı)

baseline layouts of visits' locations and five configurations of time windows. The baseline layouts contain either 20 (5 layouts), 50 (3 layouts) or 80 visits (2 layouts). Pairwise synchronisation constraints are defined for 20% of visits. In three out of five configurations of time windows, the precise span of time windows varies considerably between visits. It is progressively expanded in three levels: small, medium or large time windows. In two remaining configurations, the start time for all visits is either fixed (i.e., all time windows have span zero) or the duration of time windows for all visits is equal to the length of the scheduling horizon. For a thorough discussion of current best results, we refer an interested reader to the article [3].

2. Naming Convention for Files

Solution files are named using the following convention: "case_A_B_C_D_E.dat.gexf" where integer numbers replace letters at positions A, B, C, D and E. The prefix "case_A_B_C_D_E.dat" is the name of the problem instance. The number at position A indicates the baseline layout for the given number of visits. The counter always starts from one. The integer value at position B stores the total number of visits. The value at position C is both the number of visits with pairwise synchronisation constraints and the number of vehicles. The value at position D equals the number of visits divided by 10. Finally, the digit at the position E indicates the configuration of time windows (i.e., 1 - no time windows, 2 - small, 3 - medium, 4 - large, 5 - time windows equal to the span of the scheduling horizon).

3. Dataset Structure

The dataset is distributed as a zip archive named "hc_solutions.zip". It contains three folders: case20, case50 and case80, which store solutions to problem instances with 20, 50 and 80 visits, respectively.

Overall, the dataset contains 48 solutions out of which 39 are as of this writing the best known results in the literature. Solutions for five problem instances (i.e., *case_1_80_16_8_2.dat*, *case_1_50_10_5_5.dat*, *case_2_50_10_5_5.dat*, *case_3_50_10_5_5.dat*, *case_1_80_16_8_5.dat*) have been strictly improved. In four cases (i.e., *case_2_80_16_8_2.dat*, *case_2_80_16_8_3.dat*, case_1_80_16_8_4.dat, case_2_80_16_8_4.dat) the results found by the algorithm where worse than the best results published in the literature. The multistage optimisation algorithm failed to find solutions with all visits staffed for *case_1_80_16_8_1.dat* and *case_3_50_10_5_1.dat* instances. To the best of our knowledge, these instances were solved only by [2]. For a summary of computational results and the comparison with the results from the literature, see Table 2 in the article [3].

4. File Format

Solutions are saved as XML files. Their structure is compliant with the schema of the GEXF file format. Every solution is stored in a separate file and can be visualised using Gephi application (https://gephi.org/).

A solution is represented as a directed graph with three classes of nodes: visits, vehicles and users. A node of the type vehicle is named *carer* because the file format was initially developed for storing solutions to the Home Care Scheduling and Routing Problem (HCSRP) which is a specialisation of the VRPTWSyn. Nodes are connected using directed edges. A vehicle node is connected to the first visit node on a route. Subsequent visit nodes are then connected in a path according to the order they are performed. Finally, all visit nodes of the same service user are linked to the user node.

Additional information is associated with nodes and edges using attributes. Some of them are specific to the HCSRP domain and are not relevant in the HCSRP benchmark. A complete list of attributes is displayed in Table 1.

Table 1: Nodes' and edges' attributes

Index	Name	Type	Description
0	id	long	Identifier of a node
1	longitude	double	Longitude of the visit location
2	latitude	double	Latitude of the visit location
3	type	string	Type of a node. Supported nodes' types are: visit, user and carer
4	assigned_carer	long	Carer assigned to the visit
5	dropped	string	Information whether the visit is performed or declined
6	-	-	Legacy - not used
7	user	string	Service user who is being visited
8	start_time	string	Start time of the visit
9	duration	string	Duration of the visit
10	travel_time	string	Travel time to the next visit location
11	sap_number	string	Identifier assigned to the carer for integration with third party software
12	work_relative	double	Proportion of time spent on working
13	work_total_time	string	Total time a carer works a given day
14	work_available_time	string	Total time a carer is available for work a given day
15	work_service_time	string	Total time spent on performing visits
16	work_travel_time	string	Total time spent on travelling
17	work_idle_time	string	Total time spent on waiting before a visit can commence
18	work_visits_count	long	Total number of visits performed
19	-	-	Legacy - not used
20	skills	string	List of skills a given carer possesses
21	tasks	string	List of tasks required to perform during a visit

Nodes' and edges' attributes are referenced using indices instead of names to reduce the size of large graphs.

Listing 1 displays example solution format. Repetitive elements were replaced by "· · ·" for succinctness.

Listing 1: Example solution

```
<?xml version="1.0" encoding="UTF-8"?>
<gexf xmlns="http://www.gexf.net/1.1draft"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.gexf.net/1.1draft␣http://gexf.
        ↪ net/1.1draft/gexf.xsd" version="1.1">
  <meta>
    <description>Cost: 13440
      Errors: 0
      Dropped visits: 0
      Total visits: 20
      Carer utility:
                mean: 0.848907
                median: 0.747559
                stddev: 0.0867957
                total ratio: 0.848907
    </description>
  </meta>
  <graph mode="static" defaultedgetype="directed">
    <attributes class="node" mode="static">
      <attribute id="0" title="id" type="long">
        <default>0</default>
      </attribute>
      ...
    </attributes>
    <attributes class="edge" mode="static">
      <attribute id="10" title="travel_time" type="string">
        <default>00:00:00</default>
      </attribute>
    </attributes>
    <nodes>
      <node id="c0" label="carer␣0">
        <attvalues>
          <attvalue for="0" value="1" />
        </attvalues>
        ...
      </node>
      ...
    </nodes>
    <edges>
      <edge id="ec_c0v1" source="c0" target="v1" />
```

```
      ...
    </edges>
  </graph>
</gexf>
```

The solution file is structured as follows. The *gexf* element wraps the solution. The element *gexf/metadata/description* contains information about the performance of the solution, such as the cost of the solution and the number of declined (a.k.a. dropped) visits.

The solution is stored in the *graph* node. Its content starts with the definition of attributes associated with nodes and edges. For the complete list of the attributes see Table 1. The list of nodes is stored in the element *gexf/graph/nodes*. Finally, the element *gexf/graph/edges* contains the list of edges.

Bibliography

[1] Bredström, D. and Rönnqvist, M. (2007). A branch and price algorithm for the combined vehicle routing and scheduling problem with synchronization constraints. Discussion Papers 2007/7, Norwegian School of Economics, Department of Business and Management Science.

[2] Parragh, S. N. and Doerner, K. F. (2018). Solving routing problems with pairwise synchronization constraints. *Central European Journal of Operations Research*, 26(2):443–464.

[3] Polnik, M., Riccardi, A., and Akartunalı, K. (2020). A multistage optimisation algorithm for the large vehicle routing problem with time windows and synchronised visits. *Journal of the Operational Research Society*.