Data Description for Home Healthcare Routing and Scheduling Problems with task-splitting

Loek van Montfort
Department of Operations Analytics Vrije Universiteit Amsterdam

June 2025

Each instance consists of four files that provide information about the available staff, visits, travel times, and temporal dependencies.

1 Staff

The file `staff.csv` describes the characteristics of the available caregivers.

column	description
`qual_type`	caregiver qualification type
`num`	number of available caregivers
`ear_start`	earliest time a caregiver can work
`lat_end`	latest time a caregiver can work
`wage`	wage parameter

Table 1: Fields in the `staff.csv` file.

2 Visits

The file `visits.csv` describes the characteristics of patient visits.

column	description	comment
`lb_tw`	lower bound time window	α_v
`ub_tw`	upper bound time window	β_v
`dur`	duration	d_v
`Q1`	specifies whether a caregiver of type 1 can perform the visit	1: true, 0: false
`Q2`	specifies whether a caregiver of type 2 can perform the visit	1: true, 0: false
`Q3`	specifies whether a caregiver of type 3 can perform the visit	1: true, 0: false
`split_rel`	specifies whether the visit is splittable or a split part	1: true, 0: false
`split_part`	split part indicator	0: original visit, 1/2: split part number
`org_id`	visit number in the original instance from Bredström and Rönnqvist (2008)	
`id`	visit number in HHCRSP-TS instance	
`ord_qual`	number used to determine the required qualification type of the visit for various visit requirement scenarios	
`qual_relax`	specifies whether a split part can be performed by each caregiver type	1: true, 0: false

Table 2: Fields in the `visits.csv` file.

Table 3 shows an example of (a part of) a `visits.csv` file. The data is structured such that the first and last rows represent the artificial visits that model the start and end of a workday. After the artificial start of a workday, non-splittable visits (`split_rel` = 0) are listed, followed by splittable visits and their corresponding split parts (`split_rel`=1). A splittable visit and both of its split parts are represented by three consecutive rows:

- original visit: `id` = k, `split_part` = 0, and `org_id` = i
- split part one: `id` = $k+1$, `split_part` = 1, and `org_id` = i
- split part two: `id` = $k+2$, `split_part` = 2, and `org_id` = i

`lb_tw`	`ub_tw`	`dur`	`Q1`	`Q2`	`Q3`	`split_rel`	`split_part`	`org_id`	`id`	`ord_qual`	`qual_relax`	
0	718	0	1	1	1	0	0	0	1	0	0	
3	106	77	0	1	1	0	0	1	2	6	0	no task-splitting possibilities
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
481	579	74	1	1	1	0	0	74	6	4	0	
22	120	127	0	0	1	1	0	2	7	12	0	task-splitting possibilities
22	170	77	1	1	1	1	1	2	8	12	1	
22	197	50	0	0	1	1	2	2	9	12	0	
417	587	117	0	0	1	1	0	4	10	11	0	
417	657	47	1	1	1	1	1	4	11	11	1	
417	734	50	0	0	1	1	2	4	12	11	0	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
0	718	0	1	1	1	0	0	0	52	0	0	

Table 3: Example of a `visits.csv` file.

3 Travel times

The file `travel_times.txt` contains a matrix of travel times between visits. The element at position (i, j) specifies the travel time from the visit with `id`=i to the visit with `id`=j.

4 Temporal dependencies

The file `temp_dep.txt` describes the temporal dependencies between visits. A temporal dependency between two visits consists is represented by *two lines*, each specifying the constraint for one possible starting order those visits. Each line has the following format:

```
visit_1   visit_2   order_num   min_diff   max_diff   fixed_order   type
```

Table 4: Format of each line of a temporal dependency.

Each line describes the temporal restriction when `visit_1` starts before or simultaneously with `visit_2`. Specifically:

- `visit_1`, `visit_2`: `id` values of the two visits
- `order_num`: starting order of the visits
- `min_diff`, `max_diff`: permitted minimum and maximum difference in the starting times of `visit_1` and `visit_2` if visit `visit_1` starts before or simultaneously with visit `visit_2`.
- `fixed_order`: 1 if the specified starting order order is mandatory; 0 otherwise.

- `type`: type of temporal dependency
  - 1: strict synchronization
  - 2: precedence
  - 3: disjunction

An example of a precedence relation between visits 8 and 9 is given below:

```
temp dep: 12
  8 9 1 77 718 1 2
  9 8 2 719 719 0 2
```

For $u = 8$ and $v = 9$, the data specifies $\delta_{uv}^{\min} = 77$, $\delta_{uv}^{\max} = 718$, $\delta_{vu}^{\min} = 719$, and $\delta_{vu}^{\max} = 719$. The last two columns indicate that u has to start before v and that precedence is enforced between the two visits.

References

Bredström, D., Rönnqvist, M., 2008. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. European Journal of Operational Research 191, 19–31.