Multi Vehicle Arc Routing Problems

13442 Vehicle Routing and Distribution Planning - Lecture 9

Evelien van der Hurk

Department of Management Engineering, Technical University of Denmark (DTU)
based on previous slides by Roberto Roberti.

Thursday 3rd November, 2016

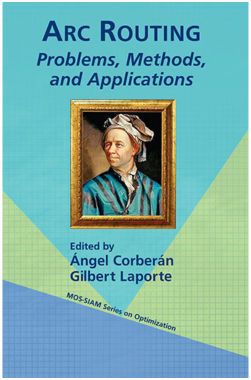
Outline

Learning Objectives

- Define and model the Capacitated Arc Routing Problem (CARP)
- Describe the Path-Scanning algorithm to find heuristic solutions of the CARP algorithm
- Model a real-life application of Waste Collection

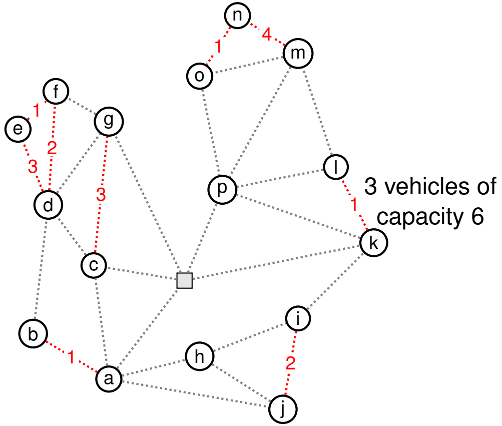
Relevant Literature for This Lecture

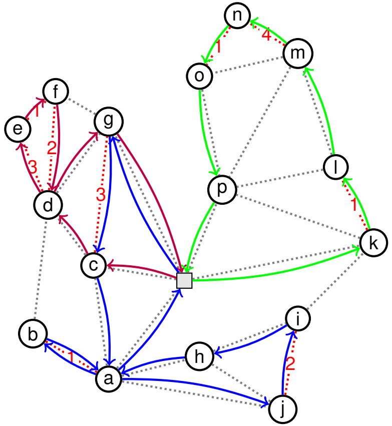
- A. Corberan and G. Laporte
  Arc Routing: Problems, Methods, and Applications
  MOS-SIAM Series on Optimization, 2015


ARC ROUTING
Problems, Methods, and Applications
Edited by
Ángel Corberán
Gilbert Laporte
MOS-SIAM Series on Optimization

Problem Definition

Capacitated Arc Routing Problem (CARP): Finding a set of minimum cost routes for vehicles located at a depot and with limited capacity that service some street segments with known demands, represented by a subset of edges of a graph



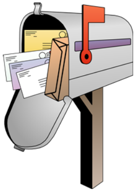


- CARP is NP-hard
- Reduces to the RPP if there is one vehicle with infinite capacity

Real-Life Applications

- Street cleaning
- Snow plowing
- Waste collection
- Salt spreading
- Meter reading
- Mail delivery
- Network inspection

- ... any other application where the focus is on traversing streets of the road network more than visiting vertices



Notation

- $G = (V, E)$: undirected graph of vertices V and edges E
- $0 \in V$: depot
- $R \subseteq E$: set of required edges
- $q_{ij} > 0$: demand of required edge $\{i, j\} \in R$
- s_{ij}: service cost of edge $\{i, j\} \in R$
- d_{ij}: deadhead cost of edge $\{i, j\} \in E$
- Q: vehicle capacity
- K: set of available vehicles
- $E(S) \subseteq E$: set of edges with both endpoints in $S \subseteq V$
- $R(S)$: set of required edges with both endpoints in $S \subseteq V$ (i.e., $R(S) = E(S) \cap R$)
- $\delta_E(S) \subseteq E$: set of edges with exactly one endpoint in $S \subseteq V$ - also called *cutset*
- $\delta_R(S)$: set of required edges with exactly one endpoint in $S \subseteq V$ (i.e., $\delta_R(S) = \delta_E(S) \cap R$)

Transformations into the CVRP

Any CARP instance defined on graph $G = (V, E)$ can be solved by transforming it into a CVRP instance defined on another graph $\widehat{G} = (\widehat{V}, \widehat{E})$ and then applying a solution method for the CVRP. Different transformations have been proposed

- [Pearn et al. 1987]: $|\widehat{V}| = 3|R| + 1$
- [Longo et al. 2006]: $|\widehat{V}| = 2|R| + 1$
- [Baldacci and Maniezzo 2006]: $|\widehat{V}| = 2|R| + 1$

Three-Index Mathematical Formulation

- $x_{ij}^k \in \{0,1\}$: binary variable equal to 1 if vehicle $k \in K$ serves edge $\{i,j\} \in R$
- $y_{ij}^k \in \mathbb{Z}_+$: number of times vehicle $k \in K$ deadheads edge $\{i,j\} \in E$
- $p_i^k \in \mathbb{Z}_+$: auxiliary integer variable for vertex $i \in V$ and vehicle $k \in K$

$$
\begin{aligned}
\min \; & \sum_{k \in K} \left(\sum_{\{i,j\} \in R} s_{ij} x_{ij}^k + \sum_{\{i,j\} \in E} d_{ij} y_{ij}^k \right) & \\
\text{s.t. } & \sum_{k \in K} x_{ij}^k = 1 & \forall \{i,j\} \in R \\
& \sum_{\{i,j\} \in R} q_{ij} x_{ij}^k \leq Q & \forall k \in K \\
& \sum_{\{i,j\} \in \delta_R(S)} x_{ij}^k + \sum_{\{i,j\} \in \delta_E(S)} y_{ij}^k \geq 2 x_{ab}^k \quad \forall S \subseteq V \setminus \{0\} \quad \forall \{a,b\} \in R(S) & \forall k \in K \\
& \sum_{\{i,j\} \in \delta_R(\{i\})} x_{ij}^k + \sum_{\{i,j\} \in \delta_E(\{i\})} y_{ij}^k = 2 p_i^k & \forall i \in V \quad \forall k \in K \\
& p_i^k \in \mathbb{Z}_+ & \forall i \in V \quad \forall k \in K \\
& x_{ij}^k \in \{0,1\} & \forall \{i,j\} \in R \quad \forall k \in K \\
& y_{ij}^k \in \mathbb{Z}_+ & \forall \{i,j\} \in E \quad \forall k \in K
\end{aligned}
$$

Set Partitioning Formulation

- Ω: set of all feasible routes
- c_r: cost of route $r \in \Omega$ equal to the sum of the service and deadhead costs of the traversed edges
- a_{ij}^r: binary coefficient equal to 1 if route $r \in \Omega$ serves edge $\{i, j\} \in R$ (0 otherwise)
- $\lambda_r \in \{0, 1\}$: binary variable equal to 1 if route $r \in \Omega$ is in solution (0 otherwise)

$$
\begin{aligned}
\min \sum_{r \in \Omega} & c_r \lambda_r \\
\text{s.t.} \sum_{r \in \Omega} & a_{ij}^r \lambda_r = 1 && \forall \{i, j\} \in R \\
\sum_{r \in \Omega} & \lambda_r \leq |K| \\
& \lambda_r \in \{0, 1\} && \forall r \in \Omega
\end{aligned}
$$

- The best exact methods for the CARP are based on the Set Partitioning formulation: instances with up to $|V| = 140$ vertices and $|R| = 75$ required edges can be solved, but there exist open instances with $|V| = 77$ and $|R| = 51$

Path Scanning Algorithm [Golden et al. 1983] I

The *Path Scanning* (PS) algorithm is a heuristic proposed by [Golden et al. 1983]

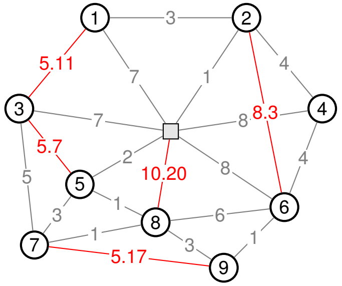
Overall Idea

Path Scanning Algorithm [Golden et al. 1983] II

- Build one route at a time
- For each route, start at the depot, and progressively extend it by adding a required edge $\{i, j\}$, not yet serviced and compatible with vehicle capacity
- Select the next required edge $\{i, j\}$ with one of the following five criteria
  1. Maximize the ratio $\frac{c_{ij}}{r_{ij}}$, where r_{ij} is the remaining demand once $\{i, j\}$ is treated
  2. Minimize the ratio $\frac{c_{ij}}{r_{ij}}$
  3. Maximize the return cost from j to the depot
  4. Minimize the return cost
  5. If the vehicle is less than half-full, apply Rule 3, otherwise Rule 4
- Closed the current route (through the shortest path to the depot) as soon as either no residual required edge fits into the vehicle or all of them have been serviced
- Run PS with the five rules, and take the best solution

Path Scanning Algorithm [Golden et al. 1983] III

Test Instance to Solve with Criterion 2



Vehicles: $m = 3$

Vehicle capacity: $Q = 25$

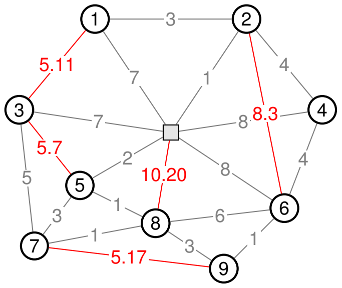
Costs: $c_{01} = 7$ $c_{02} = 1$ ...

Costs: $c_{13} = 5$ $c_{26} = 8$ ...

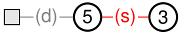
Demands: $q_{13} = 11$ $q_{26} = 3$ ...

Path Scanning Algorithm [Golden et al. 1983] IV

Route 1: Start from 0 - Capacity Left 25

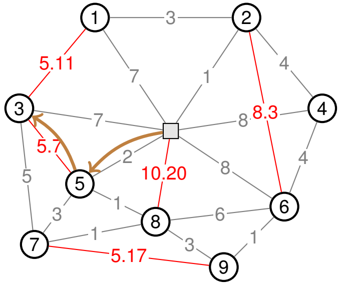


Last Vertex	Edge Served	Additional Cost	Capacity Left	Min Ratio
8	{0,8}	10	5	
3	{1,3}	9	14	
1	{3,1}	12	14	
5	{3,5}	12	18	
3	{5,3}	7	18	⇐
6	{2,6}	9	22	
2	{6,2}	15	22	
9	{7,9}	9	8	
7	{9,7}	11	8	

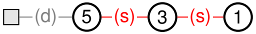


Path Scanning Algorithm [Golden et al. 1983] V

Route 1: Continue from 3 - Capacity Left 18

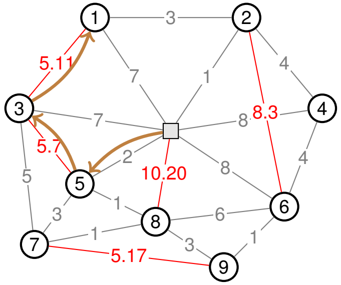


Last Vertex	Edge Served	Additional Cost	Capacity Left	Min Ratio
1	{3,1}	5	7	⇐
6	{2,6}	16	15	
2	{6,2}	18	15	
9	{7,9}	10	1	
7	{9,7}	14	1	

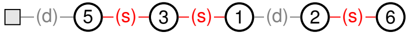


Path Scanning Algorithm [Golden et al. 1983] VI

Route 1: Continue from 1 - Capacity Left 7

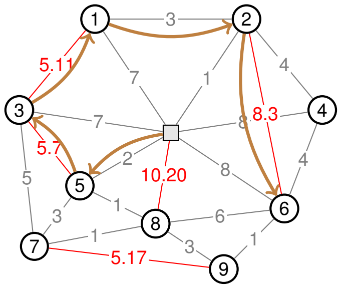


Last Vertex	Edge Served	Additional Cost	Capacity Left	Min Ratio
6	{2,6}	11	4	⇐
2	{6,2}	19	4	

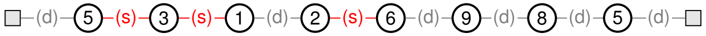


Path Scanning Algorithm [Golden et al. 1983] VII

Route 1: Continue from 6 - Capacity Left 4

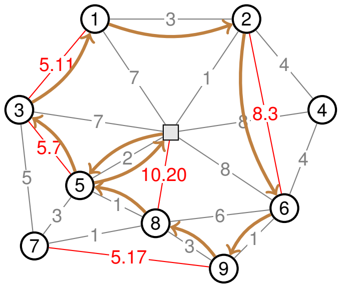


Last Vertex	Edge Served	Additional Cost	Capacity Left	Min Ratio
0	-	7	4	⇐

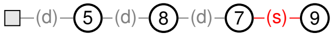


Path Scanning Algorithm [Golden et al. 1983] VIII

Route 2: Start from 0 - Capacity Left 25

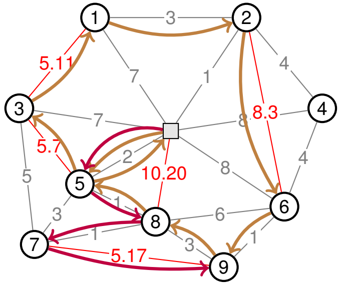


Last Vertex	Edge Served	Additional Cost	Capacity Left	Min Ratio
8	{0,8}	10	5	
9	{7,9}	9	8	⇐
7	{9,7}	11	8	

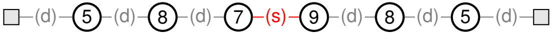


Path Scanning Algorithm [Golden et al. 1983] IX

Route 2: Continue from 9 - Capacity Left 8

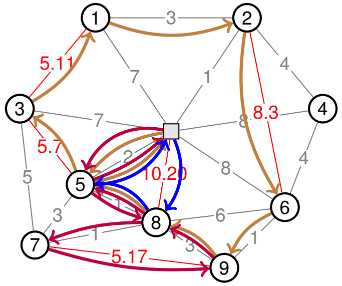


Last Vertex	Edge Served	Additional Cost	Capacity Left	Min Ratio
0	-	6	4	⇐



Path Scanning Algorithm [Golden et al. 1983] X

... Final Solution



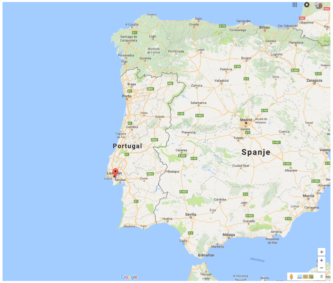
□—(d)—5—(s)—3—(s)—1—(d)—2—(s)—6—(d)—9—(d)—8—(d)—5—(d)—□

□—(d)—5—(d)—8—(d)—7—(s)—9—(d)—8—(d)—5—(d)—□

□—(s)—8—(d)—5—(d)—□

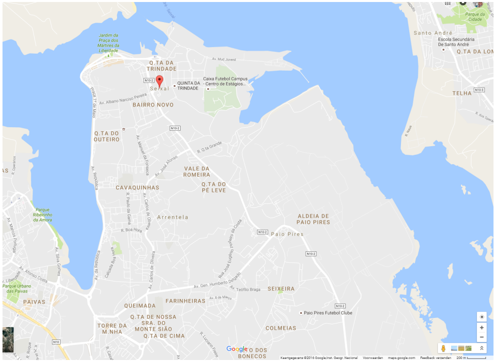
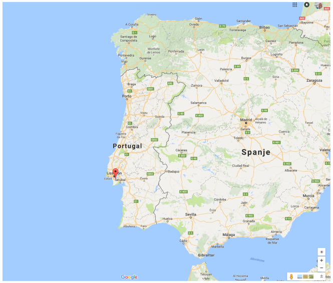
Waste Collection

- What kind of problem is this?
- What kind of constraints are there?



Waste Collection

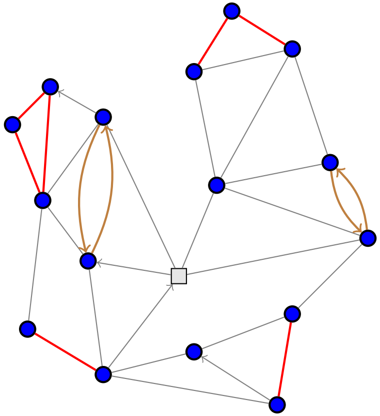
- What kind of problem is this?
- What kind of constraints are there?

Problem Definition

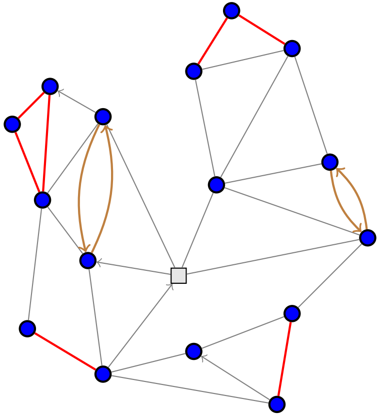
Case study of waste collection in Seixal, one of the 18 municipalities of the Lisbon Metropolitan Area in Portugal

- Road network defined over a mixed graph, where nodes represent intersections and links (i.e., arcs and edges) represent streets
- Links/Streets where waste must be collected are **required links**
- Required arcs are one-way streets or large two-way streets with no zigzag collection
- Required edges are two-way streets where zigzag collection is allowed
- A fleet of homogeneous uncapacitated vehicles are located at a depot
- Vehicles must respect maximum working times
- Collected demands must be balanced among vehicles



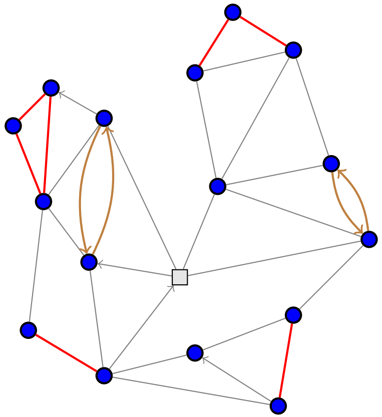
Notation

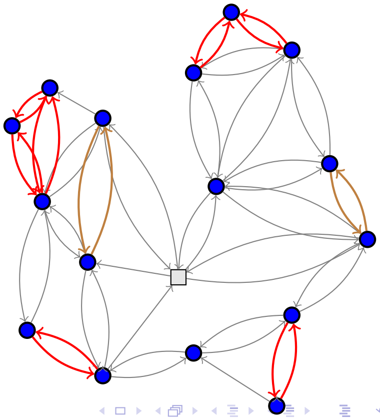
- V: set of vertices (including the depot 0)
- E_R: set of required edges
- A_R: set of required arcs



Graph Transformation

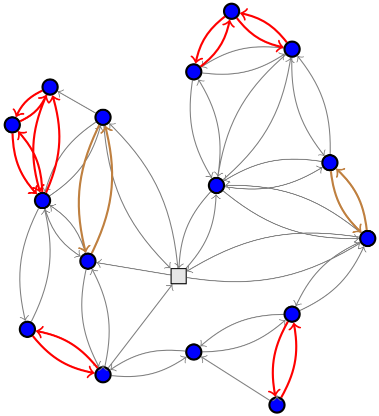
- Replace each required edge $\{i, j\} \in E_R$ with two arcs (i, j) and (j, i)
- Keep each required arc $(i, j) \in A_R$
- Replace any other non-required edge $\{i, j\}$ with two arcs (i, j) and (j, i)
- Keep non-required arcs (i, j)





New Directed Graph

- $G = (V, A)$
- V: vertex set including 0
- A: arc set including required arcs $R \subseteq A$ (notice $|R| = |A_R| + 2|E_R|$)



Additional Notation and Decision Variables

Additional Input Data

- $\mathcal{K}$: set of homogeneous uncapacitated vehicles
- T_{max}: maximum working time per vehicle
- d_{ij}: deadhead time of arc $(i,j) \in A$
- s_{ij}: service time of required arc $(i,j) \in R$
- q_{ij}: demand of required arc $(i,j) \in R$
- $\overline{Q} = \frac{Q_T}{|\mathcal{K}|}$: trip average waste, where $Q_T = \sum_{(i,j)\in R} q_{ij}$
- β: given percentage so that $\beta\overline{Q}$ is the minimum demand per trip

Variables

- $x_{ij}^k \in \{0,1\}$: service variable equal to 1 if required arc $(i,j) \in R$ is served by vehicle $k \in \mathcal{K}$ (0 otherwise)
- $y_{ij}^k \in \mathbb{Z}_+$: deadhead variable equal to the number of times vehicle $k \in \mathcal{K}$ deadheads arc $(i,j) \in A$
- $f_{ij}^k \in \mathbb{R}_+$: flow variable indicating the time left in the route of vehicle $k \in \mathcal{K}$ before traversing arc $(i,j) \in A$

Mathematical Formulation I

$$
\begin{aligned}
\min \sum_{k\in\mathcal{K}} \Big(\sum_{(i,j)\in A} d_{ij} y_{ij}^k \Big) & & (1)\\
\text{s.t.} \sum_{(i,j)\in A} y_{ij}^k + \sum_{(i,j)\in R} x_{ij}^k = \sum_{(j,i)\in A} y_{ji}^k + \sum_{(j,i)\in R} x_{ji}^k & \quad \forall i \in V\setminus\{0\} \quad \forall k\in\mathcal{K} & (2)\\
\sum_{k\in\mathcal{K}} x_{ij}^k = 1 & \quad \forall (i,j)\in A_R & (3)\\
\sum_{k\in\mathcal{K}} (x_{ij}^k + x_{ji}^k) = 1 & \quad \forall \{i,j\}\in E_R & (4)\\
\sum_{(0,j)\in A} y_{0j}^k + \sum_{(0,j)\in R} x_{0j}^k = 1 & \quad \forall k\in\mathcal{K} & (5)\\
\sum_{(j,i)\in A} f_{ji}^k - \sum_{(i,j)\in A} f_{ij}^k = \sum_{(j,i)\in R} s_{ji} x_{ji}^k + \sum_{(j,i)\in A} d_{ji} y_{ji}^k & \quad \forall i \in V\setminus\{0\} \quad \forall k\in\mathcal{K} & (6)\\
\sum_{(0,j)\in A} f_{0j}^k = \sum_{(i,j)\in R} s_{ij} x_{ij}^k + \sum_{(i,j)\in A} d_{ij} y_{ij}^k & \quad \forall k\in\mathcal{K} & (7)\\
\sum_{(i,0)\in A} f_{i0}^k = \sum_{(i,0)\in R} s_{i0} x_{i0}^k + \sum_{(i,0)\in A} d_{i0} y_{i0}^k & \quad \forall k\in\mathcal{K} & (8)\\
f_{0j}^k \le T_{max}(x_{0j}^k + y_{0j}^k) & \quad \forall (0,j)\in A \quad \forall k\in\mathcal{K} & (9)\\
\sum_{(i,j)\in R} q_{ij} x_{ij}^k \ge \beta \overline{Q} & \quad \forall k\in\mathcal{K} & (10)\\
x_{ij}^k \in \{0,1\} & \quad \forall (i,j)\in R \quad \forall k\in\mathcal{K} & (11)\\
y_{ij}^k \in \mathbb{Z}_+ \quad f_{ij}^k \in \mathbb{R}_+ & \quad \forall (i,j)\in A \quad \forall k\in\mathcal{K} & (12)
\end{aligned}
$$

Mathematical Formulation II

$$\min \sum_{k \in \mathcal{K}} \left(\sum_{(i,j) \in A} d_{ij} y_{ij}^k \right) \qquad (1)$$

Minimize the total deadhead of all vehicles over all arcs

$$\sum_{(i,j) \in A} y_{ij}^k + \sum_{(i,j) \in R} x_{ij}^k = \sum_{(j,i) \in A} y_{ji}^k + \sum_{(j,i) \in R} x_{ji}^k \qquad \forall i \in V \setminus \{0\} \quad \forall k \in \mathcal{K} \qquad (2)$$

Flow conservation: the number of times vehicle $k \in \mathcal{K}$ enters $i \in V \setminus \{0\}$ (for either service or deadhead) equals the number of times it leaves i (for either service or deadhead)

$$\sum_{k \in \mathcal{K}} x_{ij}^k = 1 \qquad \forall (i,j) \in A_R \qquad (3)$$

Each required arc must be traversed by exactly one of the vehicles

$$\sum_{k \in \mathcal{K}} (x_{ij}^k + x_{ji}^k) = 1 \qquad \forall \{i,j\} \in E_R \qquad (4)$$

Each required edge must be traversed in any direction by exactly one of the vehicles

Mathematical Formulation III

$$\sum_{(0,j)\in A} y_{0j}^k + \sum_{(0,j)\in R} x_{0j}^k = 1 \qquad \forall k \in \mathcal{K} \qquad (5)$$

Each vehicle must leave the depot

$$\sum_{(j,i)\in A} f_{ji}^k - \sum_{(i,j)\in A} f_{ij}^k = \sum_{(j,i)\in R} s_{ji} x_{ji}^k + \sum_{(j,i)\in A} d_{ji} y_{ji}^k \qquad \forall i \in V \setminus \{0\} \quad \forall k \in \mathcal{K} \qquad (6)$$

For each vertex $i \in V \setminus \{0\}$ and each vehicle $k \in \mathcal{K}$, the sum of the times left of the incoming arcs minus the sum of the times of the outgoing arcs must equal the total service and deadhead times of the selected incoming arcs

---

$$\sum_{(0,j)\in A} f_{0j}^k = \sum_{(i,j)\in R} s_{ij} x_{ij}^k + \sum_{(i,j)\in A} d_{ij} y_{ij}^k \qquad \forall k \in \mathcal{K} \qquad (7)$$

For each vehicle k, the total time left before traversing the outgoing arcs of the depot must equal the total service plus deadhead times of the traversed arcs (either serviced or deadheaded)

Mathematical Formulation IV

$$\sum_{(i,0)\in A} f_{i0}^k = \sum_{(i,0)\in R} s_{i0} x_{i0}^k + \sum_{(i,0)\in A} d_{i0} y_{i0}^k \qquad \forall k \in \mathcal{K} \qquad (8)$$

For each vehicle, the total time left before traversing incoming arcs of the depot is equal to the service plus deadhead times of the traversed incoming arcs of the depot

---

$$f_{0j}^k \leq T_{max}(x_{0j}^k + y_{0j}^k) \qquad \forall (0,j) \in A \quad \forall k \in \mathcal{K} \qquad (9)$$

For each vehicle and each outgoing arc of the depot, the total working time left before traversing such an arc cannot exceed the total working time T_{max}

$$\sum_{(i,j)\in R} q_{ij} x_{ij}^k \geq \beta \overline{Q} \qquad \forall k \in \mathcal{K} \qquad (10)$$

For each vehicle k, the total collected demand must not be less than $\beta \overline{Q}$

$$x_{ij}^k \in \{0,1\} \qquad \forall (i,j) \in R \quad \forall k \in \mathcal{K} \qquad (11)$$

$$y_{ij}^k \in \mathbb{Z}_+ \quad f_{ij}^k \in \mathbb{R}_+ \qquad \forall (i,j) \in A \quad \forall k \in \mathcal{K} \qquad (12)$$

Range of the decision variables

Bibliography I

R. Baldacci and V. Maniezzo
Exact Methods based on Node-Routing Formulations for Undirected Arc-Routing Problems
Networks 47, 52–60, 2006

Á. Corberán and J. M. Sanchis
A Polyhedral Approach for the Rural Postman Problem
European Journal of Operational Research 79, 95–114, 1994

B. L. Golden, J. S. DeArmon and E. K. Baker
Computational Experiments with Algorithms for a Class of Routing Problems
Computers and Operations Research 10(1):47–59, 1983

H. Longo, M. Poggi de Aragão and E. Uchoa
Solving Capacitated Arc Routing Problems using a Transformation to the CVRP
Computers and Operations Research 33, 1823–1837, 2006

W. L. Pearn, A. A. Assad and B. L. Golden
Transforming Arc Routing into Node Routing Problems
Computers and Operations Research 14, 285–288, 1987

Contact

Evelien van der Hurk
Department of Management Engineering, Technical University of Denmark (DTU)

Building 426, Room 042 & Building 115
2800 Kgs. Lyngby, Denmark

evdh@dtu.dk
+45 45254821 phone