A Bidirectional Labeling Algorithm for Solving the Vehicle Routing Problem with Drones

Column Generation 2023, Montreal

Jeanette Schmidt[2], Christian Tilk[1], Stefan Irnich[2]

[1]: Department of Business Decisions and Analytics, University of Vienna
[2]: Chair of Logistics Management, Johannes Gutenberg University Mainz

May 19, 2023

Drones - Advantages and Disadvantages

source: fedex.com

- + not tied to the street network
- + operate faster than trucks
- + environmental-friendly
- – limited capacity
- – limited flight range

Drones - Advantages and Disadvantages

source: fedex.com

- \+ not tied to the street network
- \+ operate faster than trucks
- \+ environmental-friendly
- − limited capacity
- − limited flight range

source: dpdhl.com

- \+ large capacity
- \+ can be used as a drone's charging/replenishment station

Drones - Advantages and Disadvantages

source: fedex.com

+ not tied to the street network
+ operate faster than trucks
+ environmental-friendly
− limited capacity
− limited flight range

source: dpdhl.com

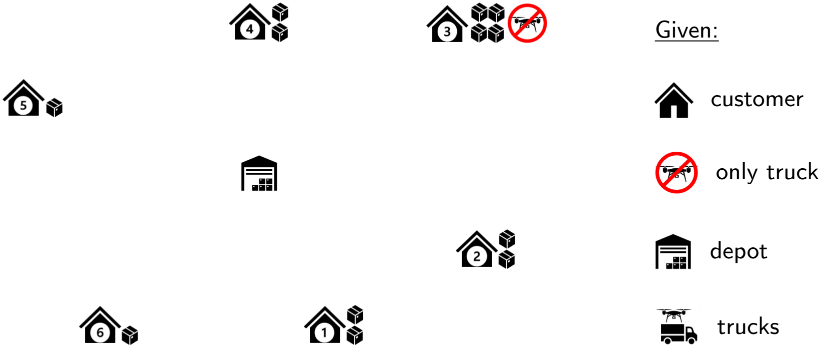
+ large capacity
+ can be used as a drone's charging/replenishment station

⇒ Use trucks and drones together as synchronized working units

Literature

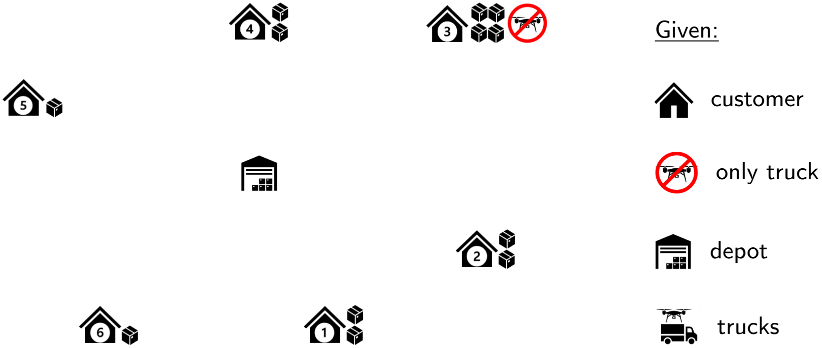
- Vast amount of Literature (Recent surveys:(Otto et al., 2018; Chung et al., 2020; Macrina et al., 2020; Moshref-Javadi and Winkenbach, 2021; Madani and Ndiaye, 2022))
- Heuristics predominant
- Exact approaches for problems with more than one truck:

		Trucks			Drones			
Publication	Obj	Capacity	Revisits	Drive alone	#Drones per truck	Fixed	Range	Deliveries
Wang et al. (2017)	$\sum$dur	✓	✗	✓	≥ 1	✓	✓	1
Bakir and Tiniç (2020)	$\sum$dur	✗	✓	✓	> 1	✗	✓	1
Tamke and Buscher (2021)	$\sum$dur	✗	✗	✓	> 1	✓	✓	1
Li and Wang (2022)	$\sum$cost	✓	✗	✓	> 1	✓	✓	> 1
Zhen et al. (2023)	$\sum$cost	✓	✗	✓	$= 1$	✓	✓	1
Zhou et al. (2022)	$\sum$dur	✓	✗	✗	$> 1^*$	✓	✓	1
Our paper	both	✓	✗	✓	$= 1$	✓	✗	1



- Set of customers N with demand $d_i \quad \forall i \in N$
- Depot 0 with a homogeneous fleet of K trucks with capacity Q^T
- Each truck T is equipped with a single drone D
- Routing costs (travel times) for truck c_{ij}^T (t_{ij}^T) and for drone c_{ij}^D (t_{ij}^D)

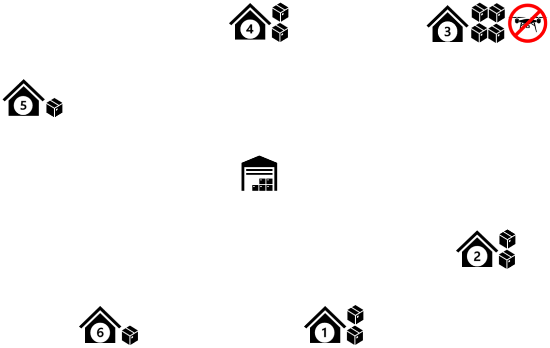
Vehicle Routing Problem with Drones (VRP-D)



Assumptions:

- Fixed assignment of trucks and drones
- A drone can serve only one customer per flight
- Drone release/return: depot or customer locations

Vehicle Routing Problem with Drones (VRP-D)



Given:

customer

only truck

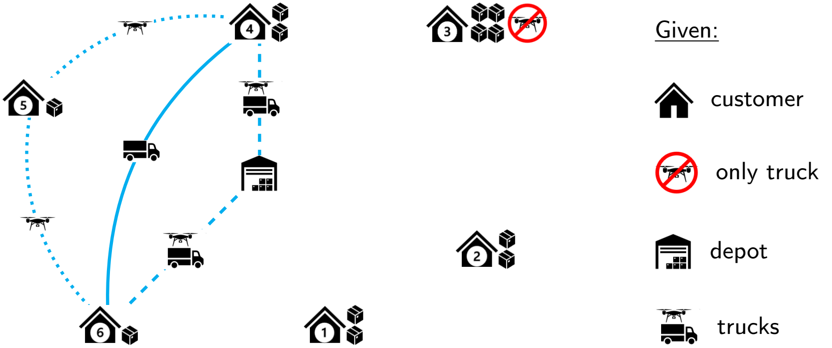
depot

trucks

Task:

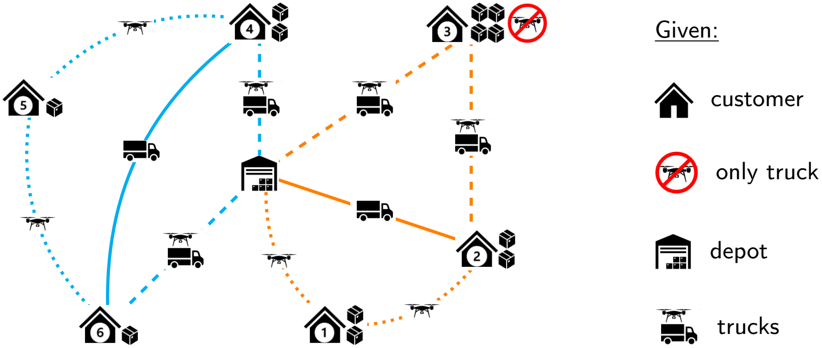
- Each customer is feasibly served
- Capacities are respected
- Minimize sum of routing costs
  or Minimize the sum of the route durations

Vehicle Routing Problem with Drones (VRP-D)



Truck path:(0-4-6-0)

Drone subpaths: $(\langle 4, 5, 6 \rangle)$



Truck path:(0-4-6-0)

Drone subpaths: $(\langle 4, 5, 6 \rangle)$

Truck path:(0-3-2-0)

Drone subpaths: $(\langle 0, 1, 2 \rangle)$

Masterproblem

$$
\begin{aligned}
\min \quad & \sum_{r\in\Omega} c_r\lambda_r & [\text{duals}] & & (1a)\\
\text{subject to} \quad & \sum_{r\in\Omega} a_{ir}\lambda_r = 1 & [\pi_i] & \quad \forall i \in N & (1b)\\
& \sum_{r\in\Omega} \lambda_r \leq K & [\pi_0] & & (1c)\\
& \lambda_r \in \{0,1\} & & \quad \forall r \in \Omega & (1d)
\end{aligned}
$$

Ω a set of feasible routes r

c_r cost (duration) of route r

a_{ir} indicates if customer i is served by route r

Pricing Problem

- Find negative reduced-cost routes with bidirectional labeling on an artificial network
- Artificial network was developed by Roberti and Ruthmair (2021) for the TSP-D

Pricing Problem

- Find negative reduced-cost routes with bidirectional labeling on an artificial network
- Artificial network was developed by Roberti and Ruthmair (2021) for the TSP-D
  - Vertices $i = (i^{tr}, i^{dr})$ represent a combination of truck and drone positions
  - Arcs represent truck and drone operations/movements

Pricing Problem

- Find negative reduced-cost routes with bidirectional labeling on an artificial network
- Artificial network was developed by Roberti and Ruthmair (2021) for the TSP-D
  - Vertices $i = (i^{tr}, i^{dr})$ represent a combination of truck and drone positions
  - Arcs represent truck and drone operations/movements
  - Three different types of arcs:
    1. : Truck and Drone move together and serve a customer

Pricing Problem

- Find negative reduced-cost routes with bidirectional labeling on an artificial network
- Artificial network was developed by Roberti and Ruthmair (2021) for the TSP-D
  - Vertices $i = (i^{tr}, i^{dr})$ represent a combination of truck and drone positions
  - Arcs represent truck and drone operations/movements
  - Three different types of arcs:

    1 : Truck and Drone move together and serve a customer

    2 : Truck drives alone and serves a customer

    $\Rightarrow$ When truck and drone separate, it is not necessary to know in advance which customer the drone will serve and where and when it will return to the truck.

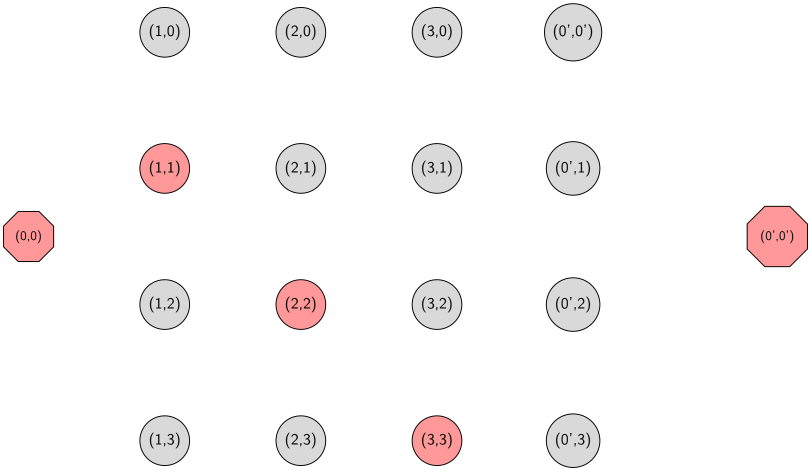
Pricing Problem

- Find negative reduced-cost routes with bidirectional labeling on an artificial network
- Artificial network was developed by Roberti and Ruthmair (2021) for the TSP-D
  - Vertices $i = (i^{tr}, i^{dr})$ represent a combination of truck and drone positions
  - Arcs represent truck and drone operations/movements
  - Three different types of arcs:
    1. : Truck and Drone move together and serve a customer
    2. : Truck drives alone and serves a customer

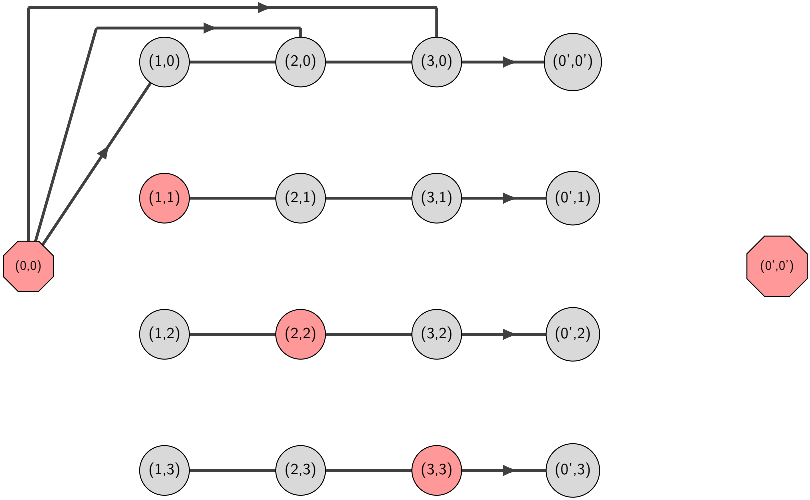
    $\Rightarrow$ When truck and drone separate, it is not necessary to know in advance which customer the drone will serve and where and when it will return to the truck.

    3. : Drone returns to the truck and serves a customer in between

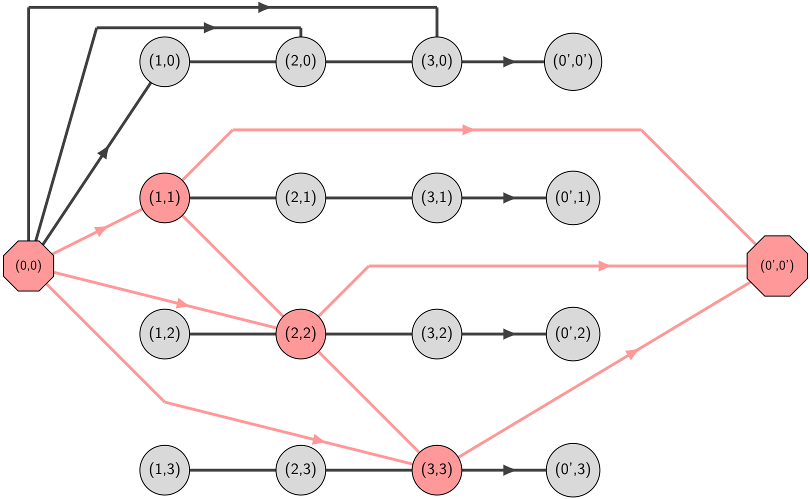
    $\Rightarrow$ When drone returns to the truck, it is decided which customer it had served.


(1,0)
(2,0)
(3,0)
(0',0')
(1,1)
(2,1)
(3,1)
(0',1)
(0,0)
(0',0')
(1,2)
(2,2)
(3,2)
(0',2)
(1,3)
(2,3)
(3,3)
(0',3)

Artificial network (Roberti and Ruthmair, 2021)

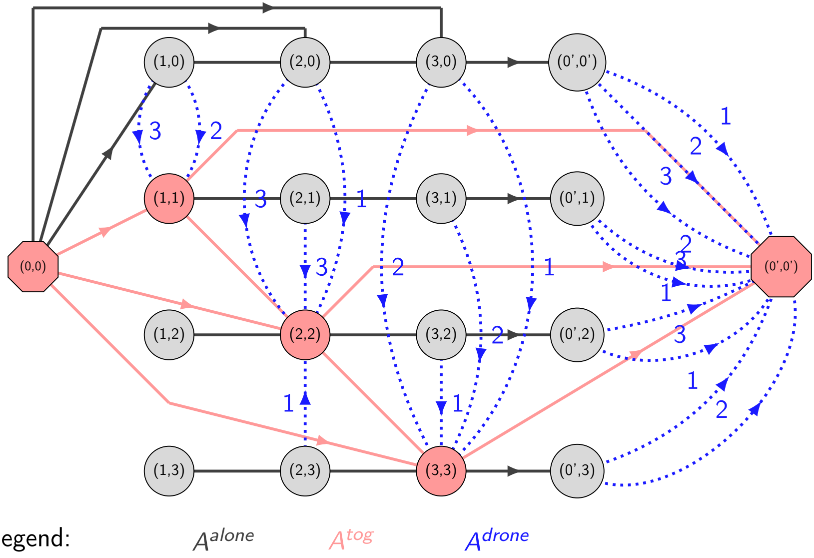


legend: A^{alone}

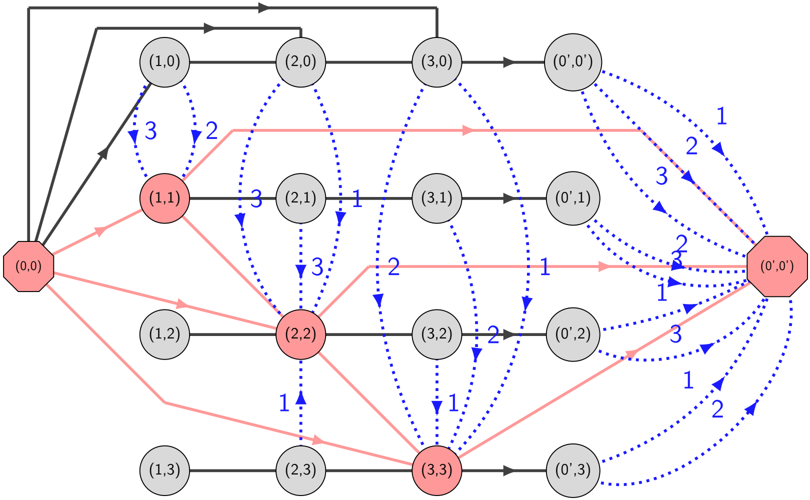


legend: A^{alone} A^{tog}

Artificial network (Roberti and Ruthmair, 2021)



Artificial network (Roberti and Ruthmair, 2021)



legend: A^{alone} A^{tog} A^{drone}

Notation due to parallel arcs: $[(i^{\mathrm{tr}}, i^{\mathrm{dr}}), (j^{\mathrm{tr}}, j^{\mathrm{dr}}), k]$

Resources:

Cost objective:

- Reduced costs C_i^{cost}

- Load Q_i
- Visited customers S_i^n

Forward Labeling algorithm

Resources:

Cost objective:

- Reduced costs C_i^{cost}
- Load Q_i
- Visited customers S_i^n

Duration objective:

- Reduced costs C_i^{dur}
- Duration truck travels alone T_i
- Load Q_i
- Visited customers S_i^n

Labeling algorithm

- Resource updates when using arc $[(i^{\text{tr}}, i^{\text{dr}}), (j^{\text{tr}}, j^{\text{dr}}), k]$ depend on the different arc types $A^{alone}, A^{tog}, A^{drone}$

$$C_j^{cost} = \begin{cases} C_i^{cost} + c_{i^{\mathbf{tr}}, j^{\mathbf{tr}}} - \pi_{j^{\mathbf{tr}}}, & \text{if } a \in A^{alone} \cup A^{tog} \\ C_i^{cost} + c^{\text{dr}}_{i^{\mathbf{dr}}, k} + c^{\text{dr}}_{k, j^{\mathbf{dr}}} - \pi_k, & \text{if } a \in A^{drone} \end{cases}$$

$$Q_j = \begin{cases} Q_i + d_{i^{\mathbf{tr}}}, & \text{if } a \in A^{alone} \cup A^{tog} \\ Q_i + d_k, & \text{if } a \in A^{drone} \end{cases}$$

$$S_j^n = \begin{cases} S_i^n + 1, & \text{if } n = j^{\text{tr}} \text{ and } a \in A^{alone} \cup A^{tog} \\ S_i^n + 1, & \text{if } n = k \text{ and } a \in A^{drone} \\ S_i^n, & \text{otherwise} \end{cases}$$

Labeling algorithm

- Resource updates when using arc $[(i^{\text{tr}}, i^{\text{dr}}), (j^{\text{tr}}, j^{\text{dr}}), k]$ depend on the different arc types $A^{alone}, A^{tog}, A^{drone}$

$$C_j^{cost} = \begin{cases} C_i^{cost} + c_{i^{\text{tr}}, j^{\text{tr}}} - \pi_{j^{\text{tr}}}, & \text{if } a \in A^{alone} \cup A^{tog} \\ C_i^{cost} + c^{\text{dr}}_{i^{\text{dr}}, k} + c^{\text{dr}}_{k, j^{\text{dr}}} - \pi_k, & \text{if } a \in A^{drone} \end{cases}$$

$$Q_j = \begin{cases} Q_i + d_{i^{\text{tr}}}, & \text{if } a \in A^{alone} \cup A^{tog} \\ Q_i + d_k, & \text{if } a \in A^{drone} \end{cases}$$

$$S_j^n = \begin{cases} S_i^n + 1, & \text{if } n = j^{\text{tr}} \text{ and } a \in A^{alone} \cup A^{tog} \\ S_i^n + 1, & \text{if } n = k \text{ and } a \in A^{drone} \\ S_i^n, & \text{otherwise} \end{cases}$$

$$C_j^{dur} = \begin{cases} C_i^{dur} + t_{i^{\text{tr}}, j^{\text{tr}}} - \pi_{j^{\text{tr}}}, & \text{if } a \in A^{tog} \\ C_i^{dur} - \pi_{j^{\text{tr}}}, & \text{if } a \in A^{alone} \\ C_i^{dur} + \max\{t^{\text{dr}}_{i^{\text{dr}}, k} + t^{\text{dr}}_{k, j^{\text{dr}}}, T_i\} - \pi_k, & \text{if } a \in A^{drone} \end{cases}$$

$$T_j = \begin{cases} T_i + t_{i^{\text{tr}}, j^{\text{tr}}}, & \text{if } a \in A^{alone} \\ 0, & \text{if } a \in A^{drone} \cup A^{tog} \end{cases}$$

Labeling algorithm

- Resource updates when using arc $[(i^{\text{tr}}, i^{\text{dr}}), (j^{\text{tr}}, j^{\text{dr}}), k]$ depend on the different arc types $A^{alone}, A^{tog}, A^{drone}$

$$C_j^{cost} = \begin{cases} C_i^{cost} + c_{i^{\text{tr}},j^{\text{tr}}} - \pi_{j^{\text{tr}}}, & \text{if } a \in A^{alone} \cup A^{tog} \\ C_i^{cost} + c^{\text{dr}}_{i^{\text{dr}},k} + c^{\text{dr}}_{k,j^{\text{dr}}} - \pi_k, & \text{if } a \in A^{drone} \end{cases}$$

$$Q_j = \begin{cases} Q_i + d_{i^{\text{tr}}}, & \text{if } a \in A^{alone} \cup A^{tog} \\ Q_i + d_k, & \text{if } a \in A^{drone} \end{cases}$$

$$S_j^n = \begin{cases} S_i^n + 1, & \text{if } n = j^{\text{tr}} \text{ and } a \in A^{alone} \cup A^{tog} \\ S_i^n + 1, & \text{if } n = k \text{ and } a \in A^{drone} \\ S_i^n, & \text{otherwise} \end{cases}$$

$$C_j^{dur} = \begin{cases} C_i^{dur} + t_{i^{\text{tr}},j^{\text{tr}}} - \pi_{j^{\text{tr}}}, & \text{if } a \in A^{tog} \\ C_i^{dur} - \pi_{j^{\text{tr}}}, & \text{if } a \in A^{alone} \\ C_i^{dur} + \max\{t^{\text{dr}}_{i^{\text{dr}},k} + t^{\text{dr}}_{k,j^{\text{dr}}}, T_i\} - \pi_k, & \text{if } a \in A^{drone} \end{cases}$$

$$T_j = \begin{cases} T_i + t_{i^{\text{tr}},j^{\text{tr}}}, & \text{if } a \in A^{alone} \\ 0, & \text{if } a \in A^{drone} \cup A^{tog} \end{cases}$$

- Standard Feasibility rules and dominance

Implicit Bidirectional Labeling

- Bidirectional labeling (Righini and Salani, 2006):
  - Create forward and backward labels
  - Choose a critical resource and extend both labels only up to a halfway point (HWP) (e.g., HWP $= \frac{Q^T}{2}$)
  - Merge suitable forward and backward labels to obtain a feasible route

Implicit Bidirectional Labeling

- Bidirectional labeling (Righini and Salani, 2006):
  - Create forward and backward labels
  - Choose a critical resource and extend both labels only up to a halfway point (`HWP`) (e.g., HWP $= \frac{Q^T}{2}$)
  - Merge suitable forward and backward labels to obtain a feasible route
- VRP-D has a symmetric structure

  $\Rightarrow$ implicit bidirectional labeling (see, e.g., Bode and Irnich, 2012; Goeke et al., 2019; Heßler and Irnich, 2023)

Implicit Bidirectional Labeling

- Bidirectional labeling (Righini and Salani, 2006):
  - Create forward and backward labels
  - Choose a critical resource and extend both labels only up to a halfway point (`HWP`) (e.g., HWP $= \frac{Q^T}{2}$)
  - Merge suitable forward and backward labels to obtain a feasible route

- VRP-D has a symmetric structure

  $\Rightarrow$ implicit bidirectional labeling (see, e.g., Bode and Irnich, 2012; Goeke et al., 2019; Heßler and Irnich, 2023)

  - Only forward labeling up to HWP
  - 'Reversed' forward labels are used backward labels

VRP-D is by definition symmetric but the artificial network is not!

Symmetry Considerations

VRP-D is by definition symmetric but the artificial network is not!

Example:
Truck path $P = (0, 1, 2, 4, 0')$ + drone subpaths $D = (\langle 1, 3, 2\rangle, \langle 2, 5, 0'\rangle)$
Reverse counterparts: $P' = (0, 4, 2, 1, 0')$ and $D' = (\langle 0, 5, 2\rangle, \langle 2, 3, 1\rangle)$.

VRP-D is by definition symmetric but the artificial network is not!

Example:
Truck path $P = (0, 1, 2, 4, 0')$ + drone subpaths $D = (\langle 1, 3, 2\rangle, \langle 2, 5, 0'\rangle)$
Reverse counterparts: $P' = (0, 4, 2, 1, 0')$ and $D' = (\langle 0, 5, 2\rangle, \langle 2, 3, 1\rangle)$.

Corresponding paths in the artificial network:

$$(0,0) = (1,1) - (2,1) \stackrel{3}{-} (2,2) - (4,2) - (0',2) \stackrel{5}{-} (0',0')$$

and

$$(0,0) - (4,0) - (2,0) \stackrel{5}{-} (2,2) - (1,2) \stackrel{3}{-} (1,1) = (0',0'),$$

differ in the vertices $(1,2)$, $(2,1)$, $(4,2)$, $(4,0)$, $(0',2)$, and $(2,0)$.

VRP-D is by definition symmetric but the artificial network is not!

Example:

Truck path $P = (0, 1, 2, 4, 0')$ + drone subpaths $D = (\langle 1, 3, 2\rangle, \langle 2, 5, 0'\rangle)$

Reverse counterparts: $P' = (0, 4, 2, 1, 0')$ and $D' = (\langle 0, 5, 2\rangle, \langle 2, 3, 1\rangle)$.

Corresponding paths in the artificial network:

$$(0,0) = (1,1) - (2,1) \overset{3}{-} (2,2) - (4,2) - (0',2) \overset{5}{-} (0',0')$$

and

$$(0,0) - (4,0) - (2,0) \overset{5}{-} (2,2) - (1,2) \overset{3}{-} (1,1) = (0',0'),$$

Merge at vertex (2,2) is a more or less a standard merge.

VRP-D is by definition symmetric but the artificial network is not!

Example:
Truck path $P = (0, 1, 2, 4, 0')$ + drone subpaths $D = (\langle 1, 3, 2\rangle, \langle 2, 5, 0'\rangle)$
Reverse counterparts: $P' = (0, 4, 2, 1, 0')$ and $D' = (\langle 0, 5, 2\rangle, \langle 2, 3, 1\rangle)$.

Corresponding paths in the artificial network:

$$(0,0) = (1,1) - (2,1) \overset{3}{-} (2,2) - (4,2) - (0',2) \overset{5}{-} (0',0')$$

and

$$(0,0) - (4,0) - (2,0) \overset{5}{-} (2,2) - (1,2) \overset{3}{-} (1,1) = (0',0'),$$

Merge at vertices (4,0) and (4,2) results in correct truck routes but misses the drone visit to customer 5.

Symmetry Considerations

VRP-D is by definition symmetric but the artificial network is not!

Example:
Truck path $P = (0, 1, 2, 4, 0')$ + drone subpaths $D = (\langle 1, 3, 2\rangle, \langle 2, 5, 0'\rangle)$
Reverse counterparts: $P' = (0, 4, 2, 1, 0')$ and $D' = (\langle 0, 5, 2\rangle, \langle 2, 3, 1\rangle)$.

Corresponding paths in the artificial network:

$$(0,0) = (1,1) - (2,1) \overset{3}{-} (2,2) - (4,2) - (0',2) \overset{5}{-} (0',0')$$

and

$$(0,0) - (4,0) - (2,0) \overset{5}{-} (2,2) - (1,2) \overset{3}{-} (1,1) = (0',0'),$$

Merge at vertices (4,0) and (4,2) results in correct truck routes but misses the drone visit to customer 5.

$\Rightarrow$: Merge over a drone arc serving the missing customer leads to additional drone subpath $\langle 0, 5, 2\rangle$

Merge of two labels $\mathcal{L} = (R)$ and $\mathcal{L}' = (R')$ ending at the same artificial vertex (i, i)

Feasibility Check:

$$Q + Q' + q_i \leq Q^T$$
$$S^n + S'^n \leq 1 \qquad \forall i \in N \setminus \{i^{\text{tr}}\}\}$$

Reduced cost of merged path:

$$\tilde{c}_r = C^{cost} + C'^{cost} + \pi_i$$
$$\tilde{c}_r = C^{dur} + C'^{dur} + \pi_i$$

Implicit bidirectional labeling – Merge

Merging two labels with different drone position by adding a drone subpath:

Label $\mathcal{L}$ at $(i^{\text{tr}}, i^{\text{dr}})$ $(i^{\text{tr}} \neq i^{\text{dr}})$ and label $\mathcal{L}'$ at $(i^{\text{tr}}, j^{\text{dr}})$ $(j^{\text{dr}} \neq i^{\text{dr}})$

Added drone subpath: $\langle i^{\text{dr}}, k, j^{\text{dr}} \rangle$ (for some suitable customer k)

Feasibility Check:

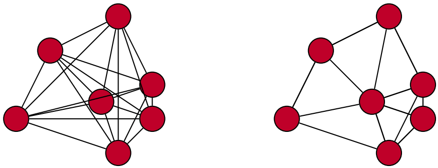
$$\begin{aligned} Q + Q' + q_{i^{\text{tr}}} + q_k \quad &\leq \quad Q^T \\ S^n + S'^n \quad &\leq \quad \begin{cases} 1, & n \in N \setminus \{i^{\text{tr}}, k\} \\ 0, & n = k \end{cases} \end{aligned}$$

Reduced Cost:

$$\tilde{c}_r = C^{cost} + C'^{cost} + \pi_{i^{\text{tr}}} + c^{\text{dr}}_{i^{\text{dr}},k} + c^{\text{dr}}_{k,j^{\text{dr}}} - \pi_k$$

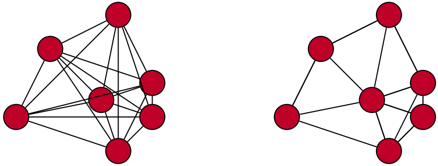
$$\tilde{c}_r = C^{dur} + C'^{dur} + \pi_{i^{\text{tr}}} + \max\left\{T + T', t^{\text{dr}}_{i^{\text{dr}},k} + t^{\text{dr}}_{k,j^{\text{dr}}}\right\} - \pi_k$$

- Reduced artificial networks as pricing heuristics



Pricing Problem - Acceleration techniques

- Reduced artificial networks as pricing heuristics



- *ng*-path relaxation (Baldacci et al., 2011)
  - Allows (specific) non-elementary paths
  - Based on a neighborhood $N_i \subset V \setminus \{0, 0'\}$ for every node $i \in V$
- Acceleration of Merge procedure:
  - Sorting labels according to load resource
  - Precomputation of drone subpaths for merge over arcs

Branching and Cutting

- Hierarchical branching
  1. Number of Trucks K
  2. How many times is a customer $i \in N$ visited by the drone alone?
  3. Truck uses edge $(i, j) \in N \times N$ or not?
  4. Drone uses edges (i, k) and (k, j) when serving customer k?

- Hierarchical branching
  1. Number of Trucks K
  2. How many times is a customer $i \in N$ visited by the drone alone?
  3. Truck uses edge $(i, j) \in N \times N$ or not?
  4. Drone uses edges (i, k) and (k, j) when serving customer k?

- Valid inequalities
  - Subset row inequalities (Jepsen et al., 2008)
  - Non-robust capacity cuts (Baldacci et al., 2007)

Instances and Computational Setting

- Instances from the CVRP library (Augerat et al., 1995)
- Customer: $|N| = \{19, 29, 39, 49\}$
- 20 instances per customer set
- Truck-only customers if $q_i > Q/5$ (10 to 33 % per instance)
- Truck routing costs and travel times based on Manhattan distance
- Drone routing costs and travel times based on Euclidean divided by a given factor β.

Instances and Computational Setting

- Instances from the CVRP library (Augerat et al., 1995)
- Customer: $|N| = \{19, 29, 39, 49\}$
- 20 instances per customer set
- Truck-only customers if $q_i > Q/5$ (10 to 33 % per instance)
- Truck routing costs and travel times based on Manhattan distance
- Drone routing costs and travel times based on Euclidean divided by a given factor β.

- time limit: 3600 seconds
- MIP-based heuristic on 1st and 2nd level and after time out

Acceleration of merge procedure (cost objective, $\beta = 3$)

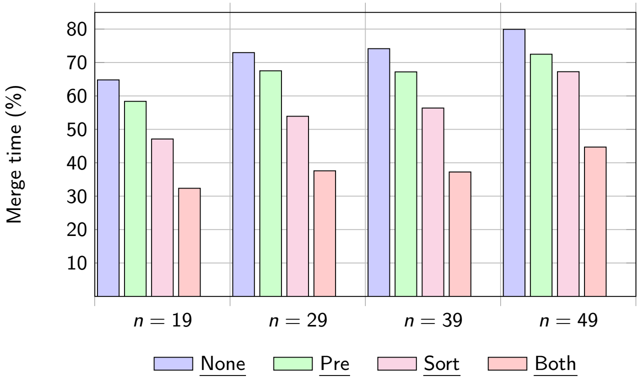


Figure: Average share (in percent) of the total computation time spent in the merge procedure for different acceleration techniques.

Implicit bidirectional labeling (cost objective, $\beta = 3$)

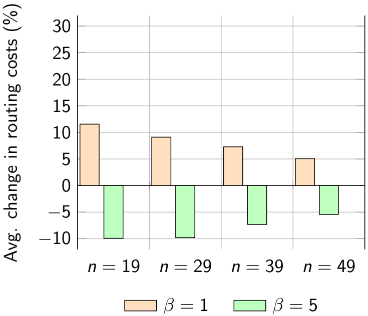
	Forward labeling				Implicit bidirectional labeling			
n	#Opt	Gap	Time	#BB	#Opt	Gap	Time	#BB
19	19	<0.01	295.2	19.0	20	—	41.9	20.5
29	14	0.36	2,160.4	34.2	17	0.07	1,046.8	72.3
39	6	1.55	3,233.0	27.8	11	0.22	2,237.8	64.3
49	2	8.45	3,400.0	8.6	3	2.70	3,215.6	21.0
	41	2.51	2,272.1	22.4	51	0.75	1,635.5	44.5

Table: Comparison of two BPC algorithms equipped with a forward or implicit bidirectional labeling algorithm.

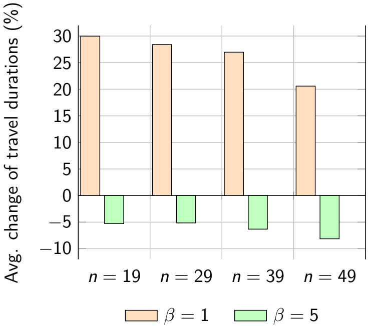
Effect of the Drones' Routing Cost and Speed

Average percentage change in routing costs and travel durations for different β-values.

Cost objective



Duration objective



Effect of the Drones' Routing Cost and Speed

	Cost objective			Duration objective		
n	$\beta = 1$	$\beta = 3$	$\beta = 5$	$\beta = 1$	$\beta = 3$	$\beta = 5$
19	0	27	42	28	51	56
29	0	27	40	27	51	54
39	0	23	37	27	49	52
49	1	22	34	26	47	49

Table: Share (in percent) of drone customers in optimal/best-known solutions.

Conclusion and Outlook

- First exact approach for Vehicle Routing Problem with Drones (VRP-D) on the basis of the work from Roberti and Ruthmair (2021)
- Implicit bidirectional labeling despite asymmetric artificial network
- Instances with up to 49 customers are solved
- Using drones is more beneficial for duration objective

Conclusion and Outlook

- First exact approach for Vehicle Routing Problem with Drones (VRP-D) on the basis of the work from Roberti and Ruthmair (2021)
- Implicit bidirectional labeling despite asymmetric artificial network
- Instances with up to 49 customers are solved
- Using drones is more beneficial for duration objective
- Delayed resource propagation is helpful (load and cuts)
- *ng*-relaxation is improved because of the implicit bidirectional labeling

Thank you for your attention!

Questions or remarks?!

Delayed Resource Propagation

Assume that $q_1 = q_2 = q_4 = 10$ and $q_3 = q_5 = 1$, a vehicle capacity of $Q^T = 38$ and standard propagation of the load resource:

The forward path

$$(0,0) - (1,1) - (2,1) \overset{5}{-} (2,2) - (4,2) - (0',2) \overset{5}{-} (0',0')$$

is feasible.

Delayed Resource Propagation

Assume that $q_1 = q_2 = q_4 = 10$ and $q_3 = q_5 = 1$, a vehicle capacity of $Q^T = 38$ and standard propagation of the load resource:

The forward path

$$(0,0) - (1,1) - (2,1) \overset{5}{-} (2,2) - (4,2) - (0',2) \overset{5}{-} (0',0')$$

is feasible.

But the partial path $(0,0) - (1,1) - (2,1)$ $\overset{5}{-}$ $(2,2)$ pass the HWP allready at vertex (2,1).

Similarly, the partial path $(0,0) - (4,0) - (2,0)$ $\overset{5}{-}$ $(2,2)$ passes the HWP allready at vertex (2,0).

Delayed Resource Propagation

Assume that $q_1 = q_2 = q_4 = 10$ and $q_3 = q_5 = 1$, a vehicle capacity of $Q^T = 38$ and standard propagation of the load resource:

The forward path

$$(0,0) - (1,1) - (2,1) \overset{5}{-} (2,2) - (4,2) - (0',2) \overset{5}{-} (0',0')$$

is feasible.

But the partial path $(0,0) - (1,1) - (2,1) \overset{5}{-} (2,2)$ pass the HWP allready at vertex (2,1).

Similarly, the partial path $(0,0) - (4,0) - (2,0) \overset{5}{-} (2,2)$ passes the HWP allready at vertex (2,0).

Hence the forward path cant be obtained in the merge procedure if resource propagation is not delayed.

Stronger *ng*-path relaxation

Assume $N_3 = N \setminus \{4\}$ and $N_1 = N_2 = N_0 = N_{sink} = N$:

The forward path

$$(0,0) = (1,0) - (2,0) \overset{4}{-} (2,2) - (3,2) - (0',2) \overset{4}{-} (0',0')$$

is *ng*-feasible

Stronger *ng*-path relaxation

Assume $N_3 = N \setminus \{4\}$ and $N_1 = N_2 = N_0 = N_{sink} = N$:

The forward path

$$(0,0) = (1,0) - (2,0) \overset{4}{-} (2,2) - (3,2) - (0',2) \overset{4}{-} (0',0')$$

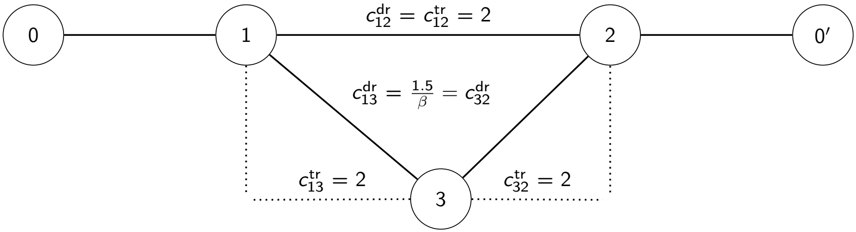
is *ng*-feasible

If we assume the merge is at vertex $(2,2)$ then the two partial paths

$$(0,0) - (1,0) - (2,0) \overset{4}{-} (2,2) \qquad \text{and} \qquad (0,0) - (3,0) - (2,0) \overset{4}{-} (2,2),$$

are *ng*-feasible but cannot be feasibly merged!

Example



0
1
2
0′
3
$c_{12}^{\text{dr}} = c_{12}^{\text{tr}} = 2$
$c_{13}^{\text{dr}} = \frac{1.5}{\beta} = c_{32}^{\text{dr}}$
$c_{13}^{\text{tr}} = 2$
$c_{32}^{\text{tr}} = 2$

Augerat, P., Belenguer, J. M., Benavent, E., Corberán, A., Naddef, D., and Rinaldi, G. (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical report, Institut National Polytechnique, 38 - Grenoble (France).

Bakir, I. and Tiniç, G. Ö. (2020). Optimizing drone-assisted last-mile deliveries: The vehicle routing problem with flexible drones. Optimization online. https://optimization-online.org/?p=16382.

Baldacci, R., Christofides, N., and Mingozzi, A. (2007). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Mathematical Programming, 115(2):351–385.

Baldacci, R., Mingozzi, A., and Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. Operations Research, 59(5):1269–1283.

Bode, C. and Irnich, S. (2012). Cut-first branch-and-price-second for the capacitated arc-routing problem. Operations Research, 60(5):1167–1182.

Chung, S. H., Sah, B., and Lee, J. (2020). Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. Computers & Operations Research, 123:105004.

Goeke, D., Gschwind, T., and Schneider, M. (2019). Upper and lower bounds for the vehicle-routing problem with private fleet and common carrier. Discrete Applied Mathematics, 264:43–61.

Heßler, K. and Irnich, S. (2023). Partial dominance in branch-price-and-cut for the basic multicompartment vehicle-routing problem. INFORMS Journal on Computing, 35(1):50–65.

Jepsen, M., Petersen, B., Spoorendonk, S., and Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. Operations Research, 56(2):497–511.

Li, H. and Wang, F. (2022). Branch-price-and-cut for the truck–drone routing problem with time windows. Naval Research Logistics (NRL), 70(2):184–204.

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., and Laporte, G. (2020). Drone-aided routing: A literature review. Transportation Research Part C: Emerging Technologies, 120:102762.

Madani, B. and Ndiaye, M. (2022). Hybrid truck-drone delivery systems: A systematic literature review. IEEE Access, 10:92854–92878.

Moshref-Javadi, M. and Winkenbach, M. (2021). Applications and research avenues for drone-based models in logistics: A classification and review. Expert Systems with Applications, 177:114854.

Otto, A., Agatz, N., Campbell, J., Golden, B., and Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. Networks, 72(4):411–458.

Righini, G. and Salani, M. (2006). Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Discrete Optimization, 3(3):255–273.

Roberti, R. and Ruthmair, M. (2021). Exact methods for the traveling salesman problem with drone. Transportation Science, 55(2):315–335.

Tamke, F. and Buscher, U. (2021). A branch-and-cut algorithm for the vehicle routing problem with drones. Transportation Research Part B: Methodological, 144:174–203.

Wang, X., Poikonen, S., and Golden, B. (2017). The vehicle routing problem with drones: several worst-case results. Optimization Letters, 11(4):679–697.

Zhen, L., Gao, J., Tan, Z., Wang, S., and Baldacci, R. (2023). Branch-price-and-cut for trucks and drones cooperative delivery. IISE Transactions, 55(3):271–287.

Zhou, H., Qin, H., Cheng, C., and Rousseau, L.-M. (2022). A branch-and-price algorithm for the vehicle routing problem with drones. Optimization online. https://optimization-online.org/?s=8801.