# Branch-Price-and-Cut for the Active-Passive Vehicle-Routing Problem

## Column Generation 2016, Búzios, Brazil

Christian Tilk[1,c], Nicola Bianchessi[1,2], Michael Drexl[1,3], Stefan Irnich[1] and Frank Meisel[4]

[1]: Chair of Logistics Management, Gutenberg School of Management and Economics
[2]: Department of Quantitative Methods, University of Brescia, Italy
[3]: Fraunhofer Centre for Applied Research on Supply Chain Services SCS, Nuremberg, Germany
[4]: Professur für Supply Chain Management, Christian-Albrechts-Universität zu Kiel, Germany
[c]: Corresponding Author: tilk@uni-mainz.de

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

May 22-25, 2016

# Outline

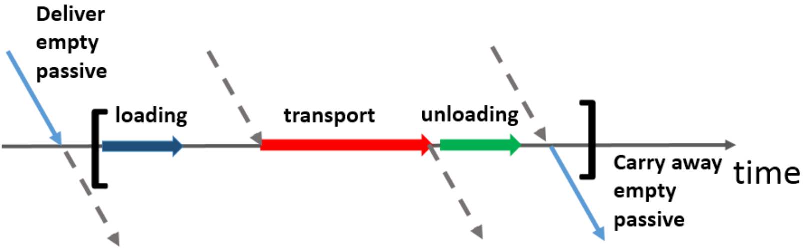
- Problem Description
- Extended Network Model
- Column Generation Formulation
- Pricing Problem
- Computational Results
- Conclusion & Outlook

The **APVRP** was first described by Meisel and Kopfer (2014)

**Given:** a set $R$ of pickup-and-delivery requests with 3 tasks per request:

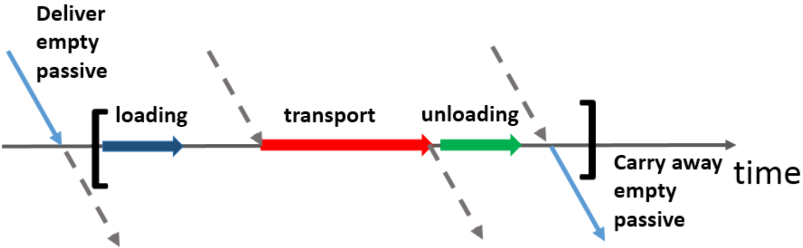
1. Provide an empty passive vehicle at the pickup location
2. Transport the loaded passive vehicle from the pickup to the delivery location
3. Carry away the empty passive vehicle from the delivery location

Synchronization:




Up to three different active vehicles can be involved in performing a request

Synchronization:




Up to three different active vehicles can be involved in performing a request

# Problem Description 3/4

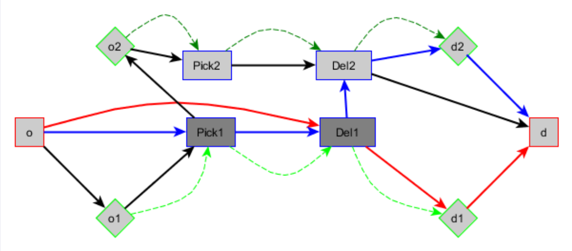
- $R$ set of pickup-and-delivery requests
  - Time Window in which a request can be fulfilled
  - Service times for loading and unloading a request
- $A$ set of classes of active vehicle
- $P$ set of passive vehicles
  - (Different) Origin and destination depot for all active/passive vehicles
- Distances and travel times between each pair of locations
- Objective: Minimize a weighted sum of the total distance traveled, the total completion time of the routes, and the number of unfulfilled requests

# Problem Description 4/4

- Each passive vehicle can load only one request at a time
- Each active vehicle can transport only one passive vehicle at a time
- But an active vehicle can be associated with different passive vehicles during its journey
- A passive vehicles can be associated with different active vehicles during its journey
- Compatibility restriction
  - between active and passive vehicle
  - between passive vehicles and requests

# Example

## Three active and two passive vehicles performing two requests

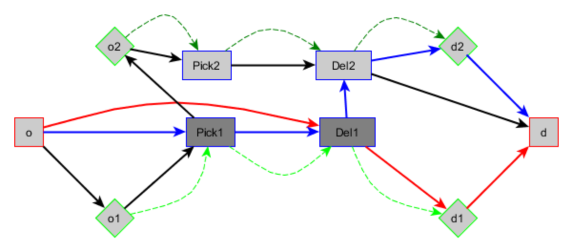



- — : Journey of active vehicle 1
- — : Journey of active vehicle 2
- — : Journey of active vehicle 3
- - - - : Journey of the passive vehicle 1
- - - - : Journey of the passive vehicle 2

- Synchronization in time and space is required at the pickup and delivery location of each request
- Routes of active and passive vehicles have to be determined separately

# Example

## Three active and two passive vehicles performing two requests




- — : Journey of active vehicle 1
- — : Journey of active vehicle 2
- — : Journey of active vehicle 3
- - - - : Journey of the passive vehicle 1
- - - - : Journey of the passive vehicle 2

- Synchronization in time and space is required at the pickup and delivery location of each request
- Routes of active and passive vehicles have to be determined separately

# Extended Network

**Goals:**

1. Modeling synchronization in an effective manner
2. The journeys of all passive vehicles should be fully described by the journeys of all active vehicles

Following ideas from Meisel and Kopfer (2014) and Drexl (2007):

- Define a network for each class of active vehicles
- Both pickup and delivery location of request $r \in R$ are each identified by two different nodes:

$v_r^-$: delivery of empty passive at pickup location

$w_r^+$: pickup of loaded passive at pickup location

$w_r^-$: delivery of loaded passive at delivery location

$v_r^+$: pickup of empty passive at delivery location

- Duplicate all request nodes for each class of passive vehicles $p \in P$
- Time Windows for each node can be derived easily

# Extended Network

**Goals:**

1. Modeling synchronization in an effective manner
2. The journeys of all passive vehicles should be fully described by the journeys of all active vehicles

Following ideas from Meisel and Kopfer (2014) and Drexl (2007):

- Define a network for each class of active vehicles
- Both pickup and delivery location of request $r \in R$ are each identified by two different nodes:

$v_r^-$: delivery of empty passive at pickup location

$w_r^+$: pickup of loaded passive at pickup location

$w_r^-$: delivery of loaded passive at delivery location

$v_r^+$: pickup of empty passive at delivery location

- Duplicate all request nodes for each class of passive vehicles $p \in P$
- Time Windows for each node can be derived easily

# Extended Network

**Goals:**

1. Modeling synchronization in an effective manner
2. The journeys of all passive vehicles should be fully described by the journeys of all active vehicles

Following ideas from Meisel and Kopfer (2014) and Drexl (2007):

- Define a network for each class of active vehicles
- Both pickup and delivery location of request $r \in R$ are each identified by two different nodes:

$v_{rp}^{-}$: delivery of empty passive $p$ at pickup location

$w_{rp}^{+}$: pickup of loaded passive $p$ at pickup location

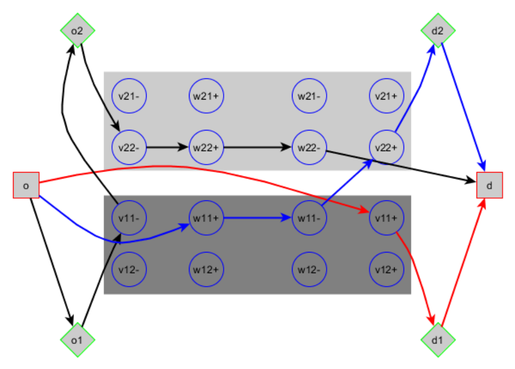
$w_{rp}^{-}$: delivery of loaded passive $p$ at delivery location

$v_{rp}^{+}$: pickup of empty passive $p$ at delivery location

- Duplicate all request nodes for each class of passive vehicles $p \in P$
- Time Windows for each node can be derived easily

# Example

## Three active and two passive vehicles performing two requests




- — : Journey of active vehicle 1
- — : Journey of active vehicle 2
- — : Journey of active vehicle 3

- Synchronization is still required at the four request nodes, but can now be handled by the unique arrival times at these nodes
- Journey of passive vehicles are fully specified by the routes of all active vehicles

Variables:

- $\lambda^{aq}$ : $q$th route (path+schedule) for class $a \in A$ with attributes:

  $T_i^q$ : The point in time when route q visits node $i$
  $X_{ij}^q$ : The number of times route $q$ uses arc $(i, j)$
  $b_i^q$ : The number of times route q visits node $i$
  $c^q$ : The cost of route q

- $u_r$ : Indicating whether or not request $r$ remains unfulfilled
- $x_{ij}^a$ : Indicating the number of times arc $(i, j)$ is traversed by active $a$

# Column Generation Formulation 2/3

$$\min \sum_{a\in A}\sum_{q\in\Omega^a} c^q\lambda^{aq} + \gamma\sum_{r\in R} u_r \tag{1}$$

$$\text{s.t.} \sum_{a\in A}\sum_{q\in\Omega^a}\sum_{p\in P^r\cap P^a} b^q_{v^-_{rp}}\lambda^{aq} + u_r = 1 \qquad r\in R \tag{2}$$

$$\sum_{a\in A^p}\sum_{q\in\Omega^a}\left(b^q_{v^-_{rp}} - b^q_{w^+_{rp}}\right)\lambda^{aq} = 0 \qquad r\in R,\ p\in P^r \tag{3}$$

$$\sum_{a\in A^p}\sum_{q\in\Omega^a}\left(b^q_{w^-_{rp}} - b^q_{v^+_{rp}}\right)\lambda^{aq} = 0 \qquad r\in R,\ p\in P^r \tag{4}$$

$$\sum_{a\in A}\sum_{q\in\Omega^a}\sum_{p\in P^r\cap P^a}\left(T^q_{w^+_{rp}} - T^q_{v^-_{rp}}\right)\lambda^{aq} + s^+_r u_r \geq s^+_r \qquad r\in R \tag{5}$$

$$\sum_{a\in A}\sum_{q\in\Omega^a}\sum_{p\in P^r\cap P^a}\left(T^q_{v^+_{rp}} - T^q_{w^-_{rp}}\right)\lambda^{aq} + s^-_r u_r \geq s^-_r \qquad r\in R \tag{6}$$

- Constraints (2)-(4) imply reduced cost for arcs
- Constraints (5) and (6) imply linear node cost (depending on the time at that the node is visited)

# Column Generation Formulation 2/3

$$\min \quad \sum_{a \in A} \sum_{q \in \Omega^a} c^q \lambda^{aq} + \gamma \sum_{r \in R} u_r \tag{1}$$

$$\text{s.t.} \quad \sum_{a \in A} \sum_{q \in \Omega^a} \sum_{p \in P^r \cap P^a} b^q_{v^-_{rp}} \lambda^{aq} + u_r = 1 \qquad r \in R \tag{2}$$

$$\sum_{a \in A^p} \sum_{q \in \Omega^a} \left( b^q_{v^-_{rp}} - b^q_{w^+_{rp}} \right) \lambda^{aq} = 0 \qquad r \in R,\ p \in P^r \tag{3}$$

$$\sum_{a \in A^p} \sum_{q \in \Omega^a} \left( b^q_{w^-_{rp}} - b^q_{v^+_{rp}} \right) \lambda^{aq} = 0 \qquad r \in R,\ p \in P^r \tag{4}$$

$$\sum_{a \in A} \sum_{q \in \Omega^a} \sum_{p \in P^r \cap P^a} \left( T^q_{w^+_{rp}} - T^q_{v^-_{rp}} \right) \lambda^{aq} + s^+_r u_r \geq s^+_r \qquad r \in R \tag{5}$$

$$\sum_{a \in A} \sum_{q \in \Omega^a} \sum_{p \in P^r \cap P^a} \left( T^q_{v^+_{rp}} - T^q_{w^-_{rp}} \right) \lambda^{aq} + s^-_r u_r \geq s^-_r \qquad r \in R \tag{6}$$

- Constraints (2)-(4) imply reduced cost for arcs
- Constraints (5) and (6) imply linear node cost (depending on the time at that the node is visited)

# Column Generation Formulation 3/3

Fleet constraints:

$$\sum_{q \in \Omega^a} \lambda^{aq} \le K_a \qquad a \in A \tag{7}$$

$$\sum_{a \in A^p} \sum_{q \in \Omega^a} b^q_{o_p} \lambda^{aq} = 1 \qquad p \in P \tag{8}$$

Coupling constraint:

$$x^a_{ij} = \sum_{q \in \Omega^a} X^q_{ij} \lambda^{aq} \qquad a \in A, (i,j) \in E^a \tag{9}$$

Variable domains:

$$\lambda^{aq} \ge 0 \qquad a \in A, q \in \Omega^a \tag{10}$$

$$x^a_{ij} \in \{0,1\} \qquad a \in A, (i,j) \in E^a \cap E^R \tag{11}$$

$$x^a_{ij} \in \mathbb{Z}^0_+ \qquad a \in A, (i,j) \in E^a \setminus E^R \tag{12}$$

$$u_r \in \{0,1\} \qquad r \in R \tag{13}$$

# Branching

- No integer requirement for route variables:
  Two or more fractional route variables with the same path and different schedules are combined to one route
  (Jans (2010), Desaulniers *et al.* (1998))
- Branching on arcs ensures integrality
- Branching on arc sets strengthen the procedure, i.e. branching on
  $\sum_{a \in A} \sum_{p \in P} x^a_{v^-_{\mathbf{rp}}, w^+_{\mathbf{rp}}} = 1$ or $0$

Also possible:

- Branching on number of served requests or on a single request
- Branching on the number of all active vehicle or the number of active vehicle in a class

# Subproblem

The subproblem is an (elementary) shortest path problem with time windows and linear node cost (ESPPTW-LNC)

Algorithms to solve the Subproblem:

- Labeling
- Discretization
- Branch-and-Cut (MiP Formulation)
- Branch-and-Price

# Subproblem

The subproblem is an (elementary) shortest path problem with time windows and linear node cost (ESPPTW-LNC)

Algorithms to solve the Subproblem:

- Labeling
- Discretization
- Branch-and-Cut (MiP Formulation)
- Branch-and-Price

# Subproblem

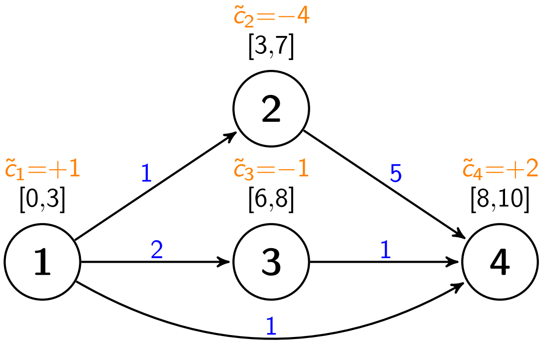
The subproblem is an (elementary) shortest path problem with time windows and linear node cost (ESPPTW-LNC)

Algorithms to solve the Subproblem:

- Labeling
- Discretization
- Branch-and-Cut (MiP Formulation)
- Branch-and-Price

# Example

**Example:** For simplicity $t_{ij} = \tilde{c}_{ij}$ on all arcs:

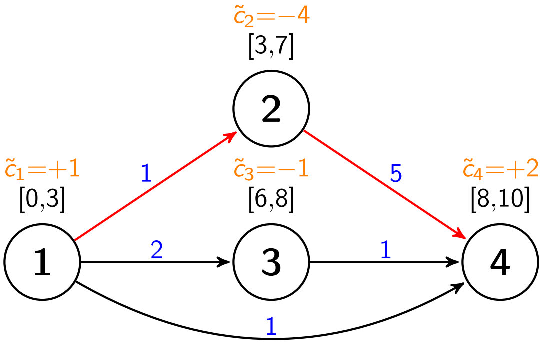



Path $(1,2,4)$: Schedule $(T_1 = 0,\ T_2 = 5,\ T_4 = 10)$;
Cost $(1+5) + 1 \cdot 0 + (-4) \cdot 5 + 2 \cdot 10 = 6$

Path $(1,3,4)$: Schedule $(T_1 = 0,\ T_2 = 7,\ T_4 = 8)$;
Cost $(2+1) + 1 \cdot 0 + (-1) \cdot 7 + 2 \cdot 8 = 12$

Path $(1,4)$: Schedule $(T_1 = 0,\ T_4 = 8)$;
Cost $1 + 1 \cdot 0 + 2 \cdot 8 = 17$

**Example:** For simplicity $t_{ij} = \tilde{c}_{ij}$ on all arcs:

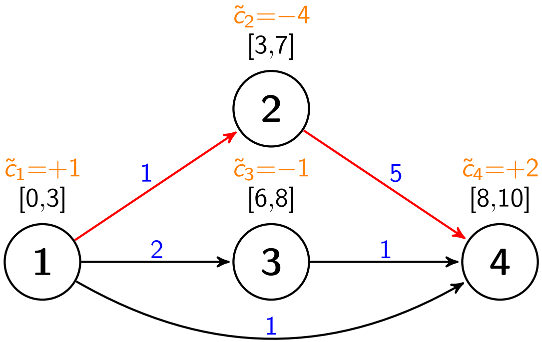



Path $(1,2,4)$: Schedule $(T_1 = 0,\ T_2 = 5,\ T_4 = 10)$;
Cost $(1+5) + 1 \cdot 0 + (-4) \cdot 5 + 2 \cdot 10 = 6$

Path $(1,3,4)$: Schedule $(T_1 = 0,\ T_2 = 7,\ T_4 = 8)$;
Cost $(2+1) + 1 \cdot 0 + (-1) \cdot 7 + 2 \cdot 8 = 12$

Path $(1,4)$: Schedule $(T_1 = 0,\ T_4 = 8)$;
Cost $1 + 1 \cdot 0 + 2 \cdot 8 = 17$

**Example:** For simplicity $t_{ij} = \tilde{c}_{ij}$ on all arcs:

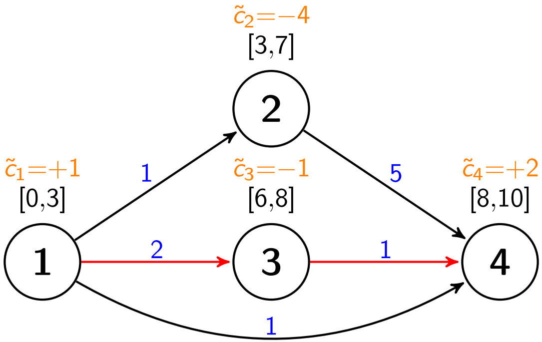



Path $(1,2,4)$: Schedule $(T_1 = 0,\ T_2 = 5,\ T_4 = 10)$;
Cost $(1+5) + 1 \cdot 0 + (-4) \cdot 5 + 2 \cdot 10 = 6$

Path $(1,3,4)$: Schedule $(T_1 = 0,\ T_2 = 7,\ T_4 = 8)$;
Cost $(2+1) + 1 \cdot 0 + (-1) \cdot 7 + 2 \cdot 8 = 12$

Path $(1,4)$: Schedule $(T_1 = 0,\ T_4 = 8)$;
Cost $1 + 1 \cdot 0 + 2 \cdot 8 = 17$

**Example:** For simplicity $t_{ij} = \tilde{c}_{ij}$ on all arcs:



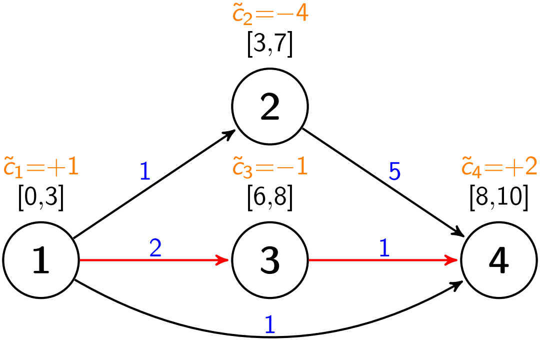

Path $(1,2,4)$: Schedule $(T_1 = 0,\ T_2 = 5,\ T_4 = 10)$;
Cost $(1+5) + 1 \cdot 0 + (-4) \cdot 5 + 2 \cdot 10 = 6$

Path $(1,3,4)$: Schedule $(T_1 = 0,\ T_2 = 7,\ T_4 = 8)$;
Cost $(2+1) + 1 \cdot 0 + (-1) \cdot 7 + 2 \cdot 8 = 12$

Path $(1,4)$: Schedule $(T_1 = 0,\ T_4 = 8)$;
Cost $1 + 1 \cdot 0 + 2 \cdot 8 = 17$

# Example

**Example:** For simplicity $t_{ij} = \tilde{c}_{ij}$ on all arcs:



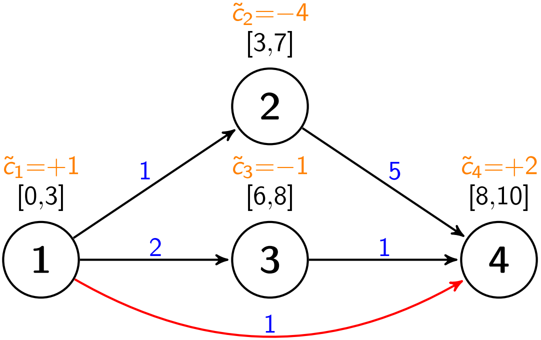

Path $(1, 2, 4)$: Schedule $(T_1 = 0,\ T_2 = 5,\ T_4 = 10)$;
Cost $(1 + 5) + 1 \cdot 0 + (-4) \cdot 5 + 2 \cdot 10 = 6$

Path $(1, 3, 4)$: Schedule $(T_1 = 0,\ T_2 = 7,\ T_4 = 8)$;
Cost $(2 + 1) + 1 \cdot 0 + (-1) \cdot 7 + 2 \cdot 8 = 12$

Path $(1, 4)$: Schedule $(T_1 = 0,\ T_4 = 8)$;
Cost $1 + 1 \cdot 0 + 2 \cdot 8 = 17$

# Example

**Example:** For simplicity $t_{ij} = \tilde{c}_{ij}$ on all arcs:



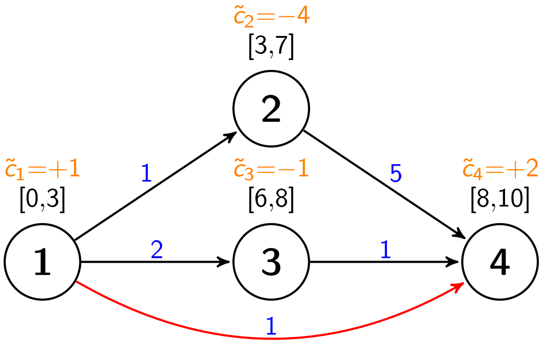

Path $(1,2,4)$: Schedule $(T_1 = 0, T_2 = 5, T_4 = 10)$;
Cost $(1+5) + 1 \cdot 0 + (-4) \cdot 5 + 2 \cdot 10 = 6$

Path $(1,3,4)$: Schedule $(T_1 = 0, T_2 = 7, T_4 = 8)$;
Cost $(2+1) + 1 \cdot 0 + (-1) \cdot 7 + 2 \cdot 8 = 12$

Path $(1,4)$: Schedule $(T_1 = 0, T_4 = 8)$;
Cost $1 + 1 \cdot 0 + 2 \cdot 8 = 17$

# Example

**Example:** For simplicity $t_{ij} = \tilde{c}_{ij}$ on all arcs:



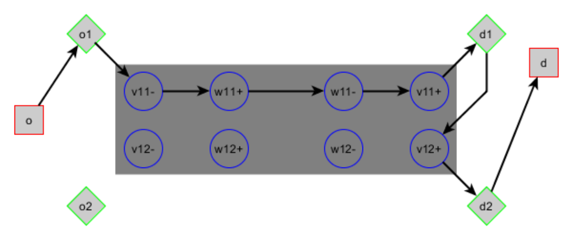

Path $(1, 2, 4)$: Schedule $(T_1 = 0,\ T_2 = 5,\ T_4 = 10)$;
Cost $(1 + 5) + 1 \cdot 0 + (-4) \cdot 5 + 2 \cdot 10 = 6$

Path $(1, 3, 4)$: Schedule $(T_1 = 0,\ T_2 = 7,\ T_4 = 8)$;
Cost $(2 + 1) + 1 \cdot 0 + (-1) \cdot 7 + 2 \cdot 8 = 12$

Path $(1, 4)$: Schedule $(T_1 = 0,\ T_4 = 8)$;
Cost $1 + 1 \cdot 0 + 2 \cdot 8 = 17$

# Elementarity

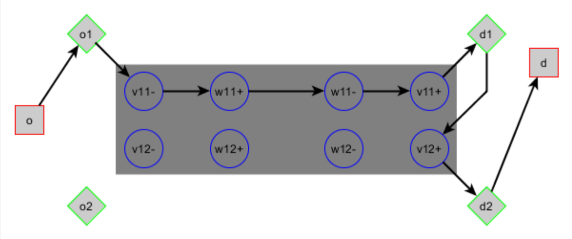
## Elementary path




Path is elementary with respect to the visited nodes, but not feasible:

⇒ Elementarity w.r.t. tasks

## Elementary path




Path is elementary with respect to the visited nodes, but not feasible:

$\Rightarrow$ Elementarity w.r.t. tasks

# Elementarity

Tasks from the view of an active vehicle

- For all requests
  1. Carry passive vehicle to pickup-location of the request
  2. Transport loaded passive vehicle directly from the pickup- to the delivery-location of the request
  3. Carry the empty passive vehicle away from the delivery location of the request
- For all passive vehicles
  1. Pickup the passive vehicle at its origin
  2. Deliver the passive vehicle at its destination

# Precedences

There are also some precedences between the tasks that help to speed up the labeling algorithm

Chain of precedences for each pair $(p, r) \in P \times R^p$

| Node $i$ | $o_p$ | $v_{rp}^-$ | $w_{rp}^+$ |
|---|---|---|---|
| Associated task | $\tau_p^o$ | $\tau_r^1$ | $\tau_r^2$ |
| Tasks to test $\mathcal{T}_i^{\text{test}}$ | $\tau_p^o, \tau_p^d$ | $\tau_r^1, \tau_r^2, \tau_r^3, \tau_p^d$ | $\tau_r^2, \tau_r^3, \tau_p^d$ |
| Tasks to set $\mathcal{T}_i^{\text{set}}$ | $\tau_p^o$ | $\tau_p^o, \tau_r^1$ | $\tau_p^o, \tau_r^1$ |

| Node $i$ | $w_{rp}^-$ | $v_{rp}^+$ | $d_p$ |
|---|---|---|---|
| Associated task | $\tau_r^2$ | $\tau_r^3$ | $\tau_p^d$ |
| Tasks to test $\mathcal{T}_i^{\text{test}}$ | $\tau_r^3, \tau_p^d$ | $\tau_r^3, \tau_p^d$ | $\tau_p^d$ |
| Tasks to test $\mathcal{T}_i^{\text{test}}$ | $\tau_p^o, \tau_r^1, \tau_r^2$ | $\tau_p^o, \tau_r^1, \tau_r^2, \tau_r^3$ | $\tau_p^o, \tau_p^d$ |

# Forward Label

**Idea:** Using cost functions as labels (Ioachim *et al.*, 1998)

A partial path is represented by a Label with the following attributes:

$i$ : last visited node

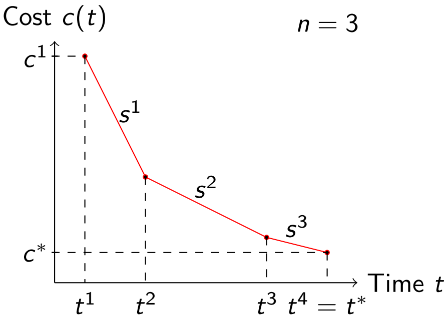
$S$ : the tasks performed on the path

$C$ : the Tradeoff Curve

Properties of the Tradeoff Curve (Ioachim *et al.*, 1998)

- piecewise linear
- convex
- number of linear pieces is at most the number of nodes in the path
- positive slope pieces can be replaced by a piece with slope zero

# Cost Function



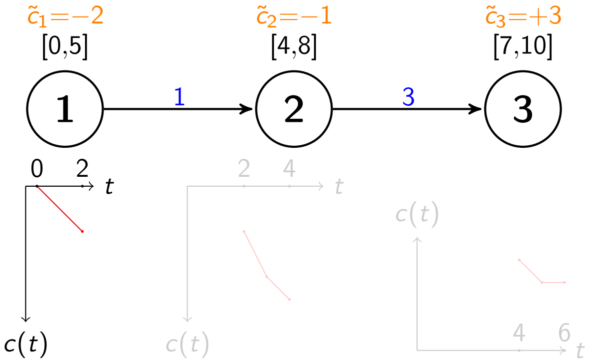

Attributes:

| | |
|---|---|
| $n$ | number of pieces |
| $(t^p, s^p)_{p=1}^n$ | the $n$ pieces |
| $c^1$ | (reduced) cost at start time of piece 1 |
| $t^{n+1}$ | end time of last piece $n$ |

Derivable values:

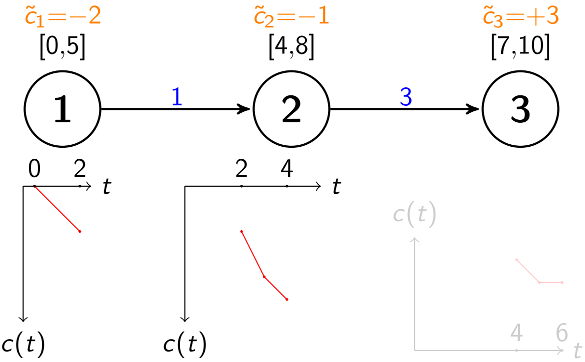
| | |
|---|---|
| $c^*$ | the optimal (reduced) cost |
| $t^*$ | the earliest time to obtain cost $c^*$ |
| $C(T)$ | (reduced) cost at time $T$ |

# Example



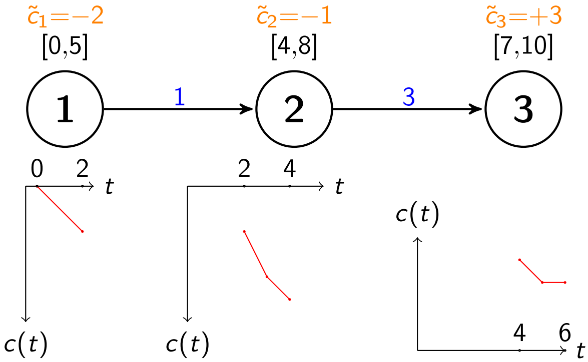
$\tilde{c}_1 = -2$
[0,5]
$\tilde{c}_2 = -1$
[4,8]
$\tilde{c}_3 = +3$
[7,10]
1
1
2
3
3
0
2
t
c(t)
2
4
t
c(t)
c(t)
4
6
t

# Example



$\tilde{c}_1 = -2$
[0,5]
$\tilde{c}_2 = -1$
[4,8]
$\tilde{c}_3 = +3$
[7,10]
1
1
2
3
3
0
2
t
c(t)
2
4
t
c(t)
c(t)
4
6
t

# Example



$\tilde{c}_1 = -2$
[0,5]
$\tilde{c}_2 = -1$
[4,8]
$\tilde{c}_3 = +3$
[7,10]
1
2
3
1
3
0
2
t
c(t)
2
4
t
c(t)
c(t)
4
6
t

# Forward Label Extension

- Update the performed tasks
- Update the tradeoff curve (Ioachim *et al.*, 1998)
  - Extending some of the existing pieces may be obsolete
  - Maybe a new piece must be created
  - non-obsolete pieces $(t, s)$ can easily be extended

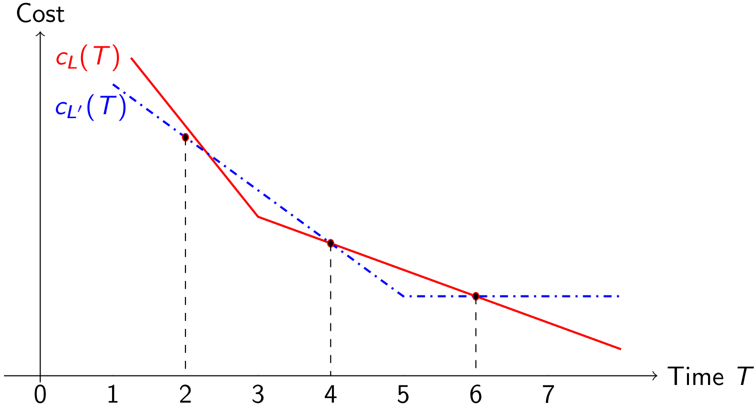
# Dominance

- Pointwise dominance for each point on the tradeoff curve between labels residing at the same node (Liberatore *et al.*, 2011)
- Label $L$ dominates Label $L'$ at time $T$ if

$$S_{L'} \subseteq S_L \text{ and } c_{L'}(T) \leq c_L(T)$$

- Dominated parts of the curve are stored
- If the complete tradeoff curve of a label is dominated, the label itself is dominated and can be discarded
- In most cases: several other labels together are needed to Łmake one label obsolete

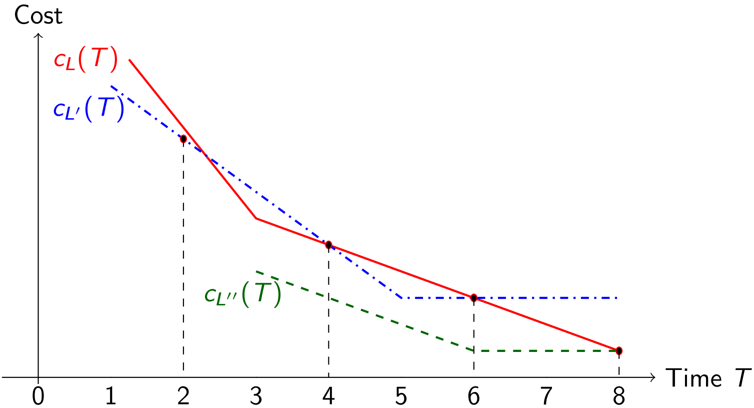
$S_{L'} \subsetneq S_L$



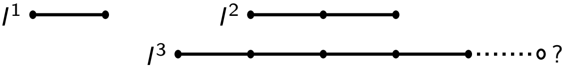

1. $L'$ dominates $L$ $I^1$ $I^2$

# Example 2/3

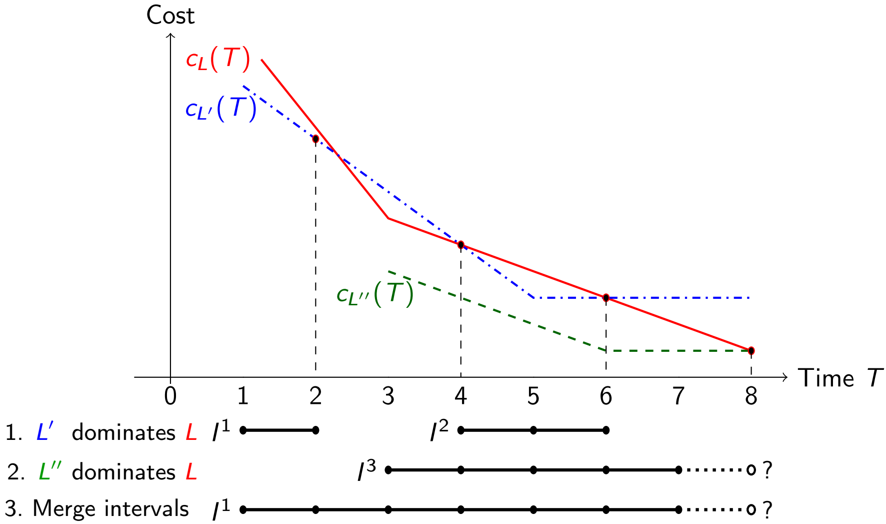
$S_{L''} = S_L$




1. $L'$ dominates $L$
2. $L''$ dominates $L$



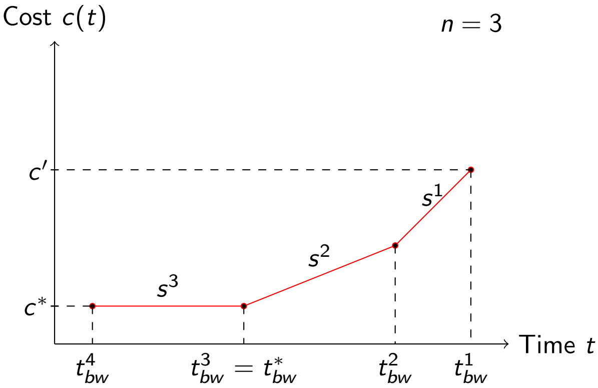
# Example 3/3



Cost
$c_L(T)$
$c_{L'}(T)$
$c_{L''}(T)$
Time $T$
0 1 2 3 4 5 6 7 8
1. $L'$ dominates $L$ $I^1$ $I^2$
2. $L''$ dominates $L$ $I^3$ ?
3. Merge intervals $I^1$ ?

# Backward Labeling

- Invert the time windows and linear node cost
- Same rules as in the forward case are applicable

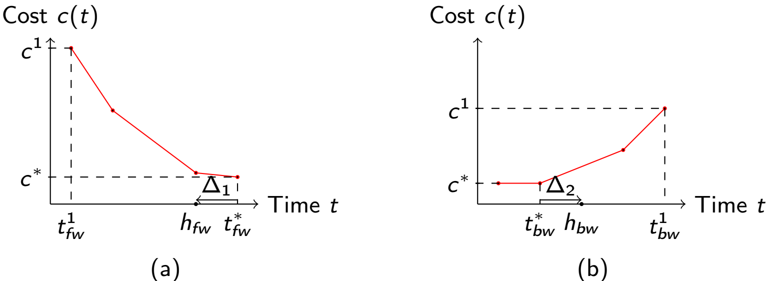


# Bidirectional Labeling

- Labels are propagate up to half-way point (Salani, 2005)
- $L_{fw}$ and $L_{bw}$ are mergeable over arc $(i, j)$, if

$$S_{fw} \cap S_{bw} = \emptyset \text{ and } t^1_{fw} + t_{ij} \leq t^1_{bw} \text{ and } t^1_{fw} \geq t^{max}/2.$$

- Optimal mergepoints $h_{fw}$ and $h_{bw}$ can be determined by distributing $\Delta = \max\{0, t^*_{fw} + t_{ij} - t^*_{bw}\}$ such that $\Delta = \Delta_{fw} + \Delta_{bw}$ , $h_{fw} = t^*_{fw} - \Delta_{fw}$ and $h_{bw} = t^*_{bw} + \Delta_{bw}$
- Optimal distribution can be computed by iteratively choosing the smallest slope



# Acceleration techniques

- *ng*-path relaxation (Baldacci *et al.*, 2011)
- Heuristic Pricing
  - Limited Discrepancy Search (Feillet *et al.*, 2007)
  - Heuristic dominance rule: $L_1$ fully dominates $L_2$ if $c_1^* \leq c_2^*$ and $t_1^1 \leq t_2^1$
- Subset-row inequalities, defined on tasks

# Computational Results

## Instances

- 80 Instances with 10 requests, 4 passive and 2 active vehicle and time horizon 1000 (38 tasks)
- 80 instances with 20 requests, 8 passive and 4 active vehicle and time horizon 1000 (76 tasks)
- There are 4 classes in both sets with different time window widths (25, 50, 100, 200)

## Configuration

- 2 hour time limit
- Pricing heuristics
- *ng*-neighborhood of size 15
- Subset-row inequalities: $Cut_{max} = 5, Cut_{task} = 2$

# Computational Results

Results on 38-task instances without cuts

| TW flex. | # Solved | Time [sec] | | | Gap at root [%] | | | Gap closed [%] | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | min | avg | max | min | avg | max | min | avg | max |
| 25 | 20/20 | 36 | 191 | 931 | 0.00 | 0.43 | 1.65 | 100.00 | 100.00 | 100.00 |
| 50 | 20/20 | 43 | 362 | 2485 | 0.00 | 0.55 | 2.11 | 100.00 | 100.00 | 100.00 |
| 100 | 19/20 | 38 | 1076 | 7200 | 0.00 | 0.99 | 3.00 | 64.29 | 98.21 | 100.00 |
| 200 | 14/20 | 109 | 3466 | 7200 | 0.00 | 2.67 | 9.89 | 8.94 | 79.94 | 100.00 |
| All | 73/80 | | 1274 | | | 1.16 | | | 94.54 | |

Results on 76-task instances without cuts

| TW flex. | # Solved | Time [sec] | | | Gap at root [%] | | | Gap closed [%] | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | min | avg | max | min | avg | max | min | avg | max |
| 25 | 17/20 | 131 | 3594 | 7200 | 0.00 | 0.80 | 2.39 | 46.52 | 90.54 | 100.00 |
| 50 | 8/20 | 282 | 5419 | 7200 | 0.28 | 1.55 | 4.30 | 21.26 | 67.96 | 100.00 |
| 100 | 2/20 | 2814 | 6882 | 7200 | 0.47 | 3.53 | 6.94 | 10.70 | 34.11 | 100.00 |
| 200 | 0/20 | 7200 | 7200 | 7200 | 6.10 | 8.48 | 13.47 | 1.79 | 6.27 | 15.35 |
| All | 27/80 | | 5775 | | | 3.59 | | | 49.72 | |

# Computational Results

Results on 38-task instances with subset-row cuts

| TW flex. | # Solved | Time [sec] | | | Gap closed by cuts [%] | | | Gap closed overall [%] | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | min | avg | max | min | avg | max | min | avg | max |
| 25 | 20/20 | 37 | 264 | 1696 | 3.13 | 67.57 | 100.00 | 100.00 | 100.00 | 100.00 |
| 50 | 20/20 | 49 | 266 | 1210 | 0.00 | 60.97 | 100.00 | 100.00 | 100.00 | 100.00 |
| 100 | 19/20 | 26 | 1398 | 7200 | 0.00 | 31.50 | 100.00 | 66.52 | 98.33 | 100.00 |
| 200 | 14/20 | 68 | 3896 | 7200 | 2.81 | 31.50 | 100.00 | 7.22 | 78.76 | 100.00 |
| All | 73/80 | | 1456 | | | 47.88 | | | 94.27 | |

Results on 76-task instances with subset-row cuts

| TW flex. | # Solved | Time [sec] | | | Gap closed by cuts [%] | | | Gap closed overall [%] | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | min | avg | max | min | avg | max | min | avg | max |
| 25 | 17/20 | 101 | 2550 | 7200 | 0.00 | 35.90 | 100.00 | 67.96 | 92.48 | 100.00 |
| 50 | 7/20 | 289 | 5395 | 7200 | 3.23 | 20.39 | 58.70 | 22.06 | 73.26 | 100.00 |
| 100 | 2/20 | 1118 | 6678 | 7200 | 1.03 | 11.64 | 77.50 | 9.58 | 35.92 | 100.00 |
| 200 | 0/20 | 7200 | 7200 | 7200 | 0.66 | 2.73 | 6.56 | 3.29 | 5.82 | 10.48 |
| All | 26/80 | | 5456 | | | 17.66 | | | 51.87 | |

# Conclusion and Outlook

Conclusion

- First Branch-Price-and-Cut Algorithm for a vehicle routing problem with synchronization in time and space
- First algorithm that solves a linear node cost problem with *ng*-tour relaxation
- Algorithm is able to solve moderate size problem instances

Outlook:

- Alternative solution approaches for ESPPTW-LNC

# Conclusion and Outlook

Conclusion

- First Branch-Price-and-Cut Algorithm for a vehicle routing problem with synchronization in time and space
- First algorithm that solves a linear node cost problem with *ng*-tour relaxation
- Algorithm is able to solve moderate size problem instances

Outlook:

- Alternative solution approaches for ESPPTW-LNC

**Thank you for coming!**

**Questions?!**

# References


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.

Desaulniers, G., Desrosiers, J., Ioachim, I., Solomon, M., Soumis, F., and Villeneuve, D. (1998). A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. G. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 57–93. Kluwer, Boston.

Drexl, M. (2007). *On Some Generalized Routing Problems*. Ph.D. thesis, Faculty of Business and Economics, RWTH Aachen University.

Feillet, D., Gendreau, M., and Rousseau, L.-M. (2007). New refinements for the solution of vehicle routing problems with branch and price. *INFOR*, **45**(4), 239–256.

Ioachim, I., Gélinas, S., Soumis, F., and Desrosiers, J. (1998). A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. *Networks*, **31**(3), 193–204.

Jans, R. (2010). Classification of Dantzig-Wolfe reformulations for binary mixed integer programming problems. *European Journal of Operational Research*, **204**(2), 251–254.

Liberatore, F., Righini, G., and Salani, M. (2011). A column generation algorithm for the vehicle routing problem with soft time windows. *4OR*, **9**(1), 49–82.

Meisel, F. and Kopfer, H. (2014). Synchronized routing of active and passive means of transport. *OR Spectrum*, **36**(2), 297–322.

Salani, M. (2005). *Branch-and-Price Algorithms for Vehicle Routing Problems*. Ph.D. thesis, Faculty of Mathematical, Physical and Natural Sciences, University of Milan.

# Feasible Solution in the extended network

A feasible solution to the APVRP is a set of scheduled routes (for the active vehicles) in the extended network that fulfills:

- Routes starts at node $o$ and terminates at node $d$
- All Nodes are visited within their time windows
- The four nodes of a request $r$ are either all visited exactly once or none of them is visited
- If a request is served, all request nodes are visited by the same passive vehicle
- There is a feasible temporal synchronization of tasks within and between vehicles
- All passive vehicles are picked up at their initial location and are placed at their final locations
- Each active vehicle performs at most one feasible route

# Forward Label Extension

- Extending some of the existing pieces may be obsolete Ioachim *et al.* (1998) $\Rightarrow$: Indices $f$ and $g$ of the first and last new piece to be kept:

$$f := \max\{0, p \in \{0 \ldots n+1\} : t_i^p + t_{ij} \leq e_j\}$$

$$g := \min\{n, p \in \{0 \ldots n\} : s_i^p + \tilde{c}_j \geq 0 \text{ or } t_i^{p+1} + t_{ij} \geq l_j\}$$

- Maybe a new piece must be created Ioachim *et al.* (1998) $\Rightarrow$: new-piece indicator $\delta$

$$\delta := \begin{cases} 1 & \text{if } (g = n_i, t_i^* = t_i^{n+1}, \text{ and } t_i^* + t_{ij} < l_j) \text{ or } (f = n+1) \\ 0 & \text{otherwise} \end{cases},$$

- Second, non-obsolete pieces $(t, s)$ are extended using the function

$$f_{ij}(t, s) := (\{\max\{e_j, t + t_{ij}\}, \min\{0, s + \tilde{c}_j\})$$

- cost at the start time $t_j^1$ of the new pieces can be expressed as

$$c_j^1 = c_i(\min\{t_i^*, t_j^1 - t_{ij}\}) + \tilde{c}_{ij} + \tilde{c}_j t_j^1.$$

- the remaining attributes are

$$S_j := (S_i \cap \mathcal{N}_j) \cup \mathcal{T}_j^{\text{set}}$$