A branch-and-price method for the pickup and delivery problem with truck driver scheduling

Magnus Stålhane, Ole Johannes Lindseth, Simen Sørum

Norwegian Univerisity of Science and Technology

Outline

- Motivation
- Problem definition
- Solution Methodology
- Computational Results
- Summary

Haste AS

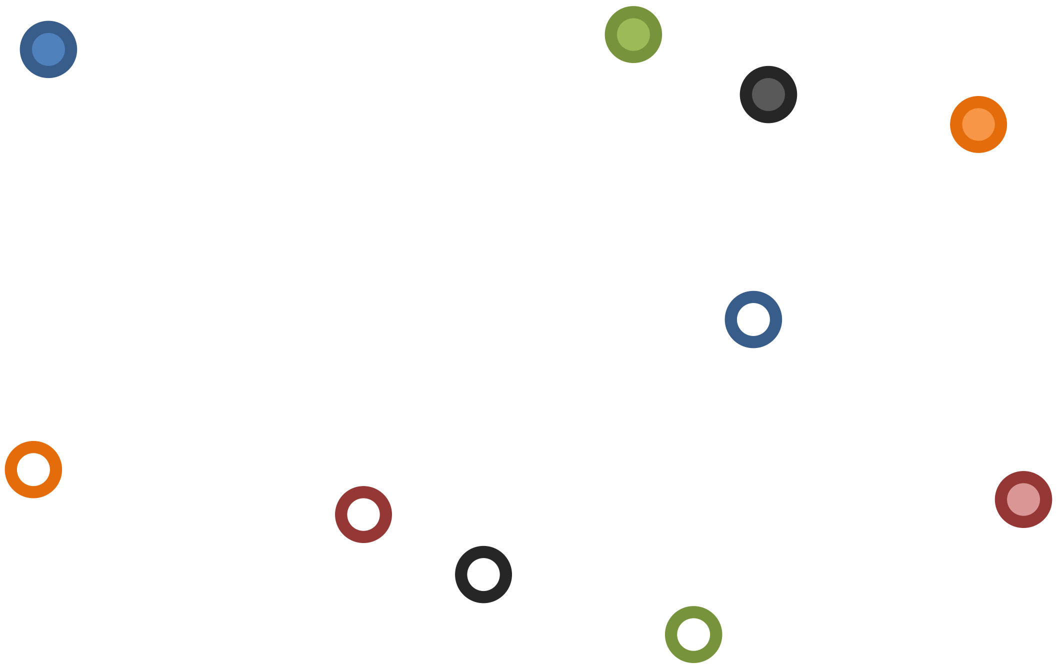
- Start up company
- Portal for planning freight orders
- Market place for transporters and shippers
- Targeting Norwegian/ Scandinavian market

Haste AS

- Main goal: Developed a Heuristic
  - ALNS
- Need to benchmark this
  - Exact CG method

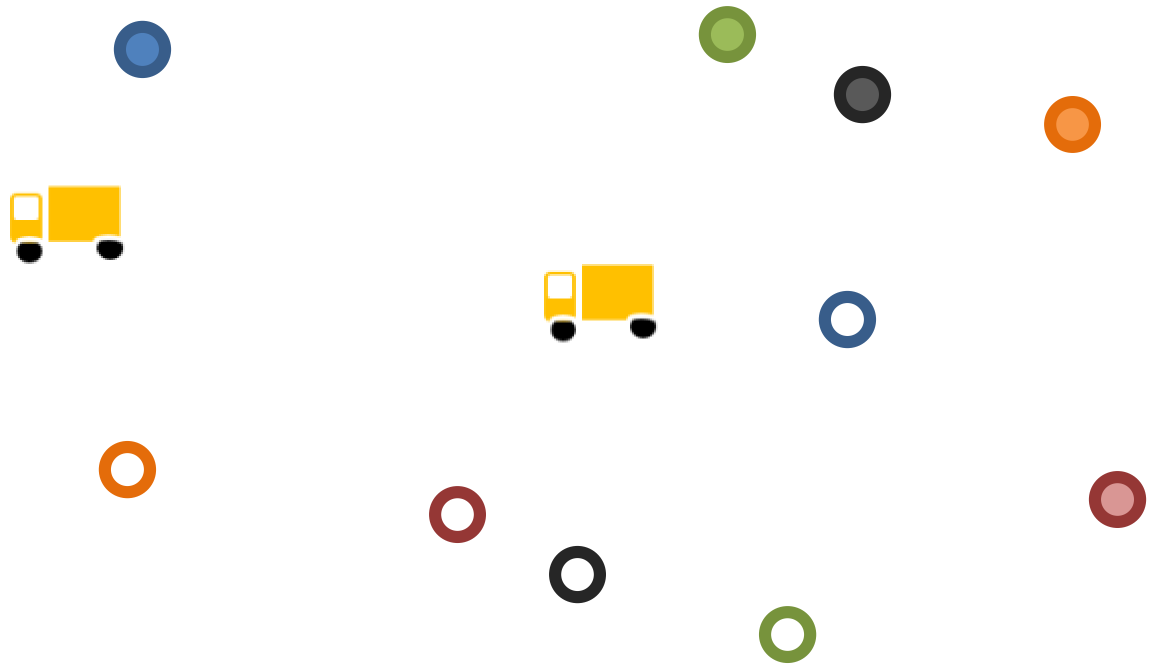
Problem definition

Problem definition



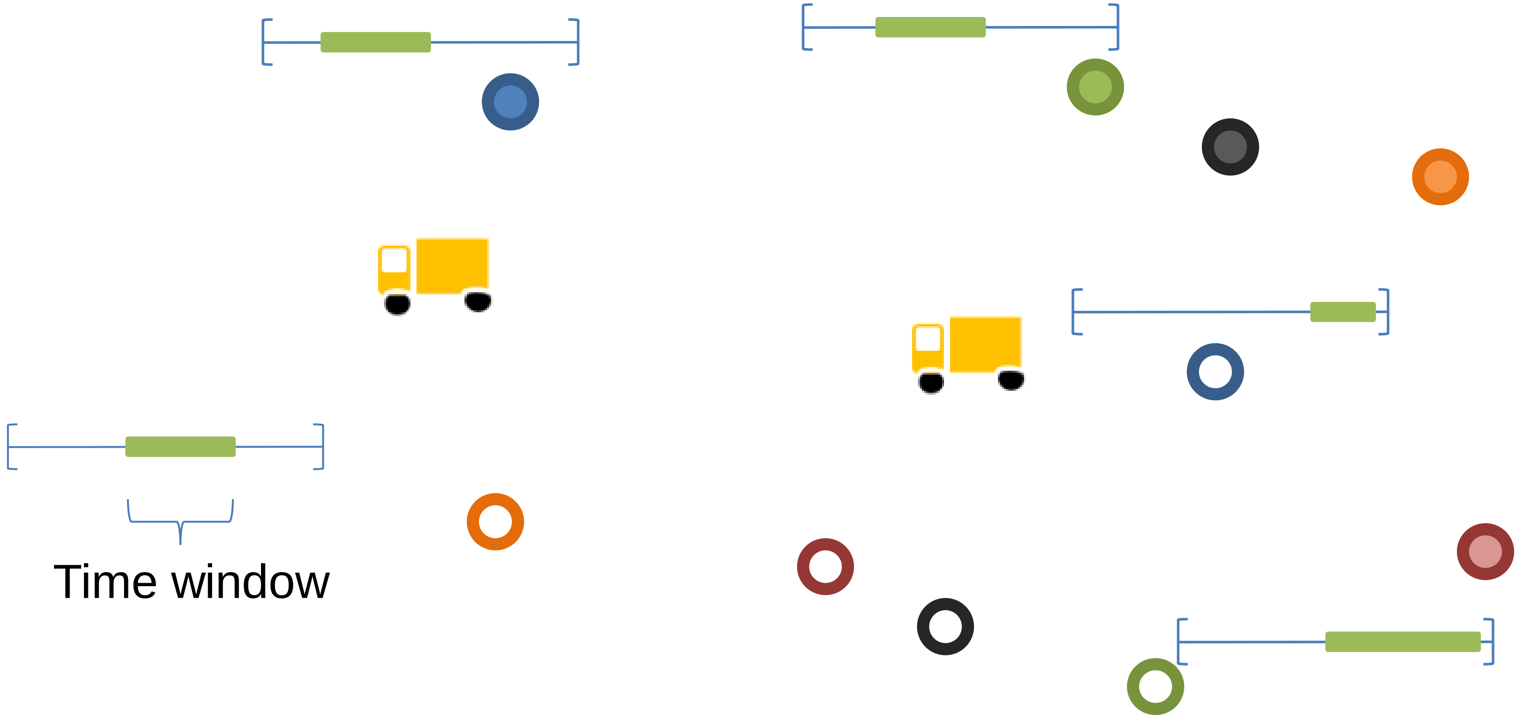
- A set of transportation requests with given:
  - pickup and delivery locations
  - weight/Volume
  - Revenue (if transported)

Problem definition



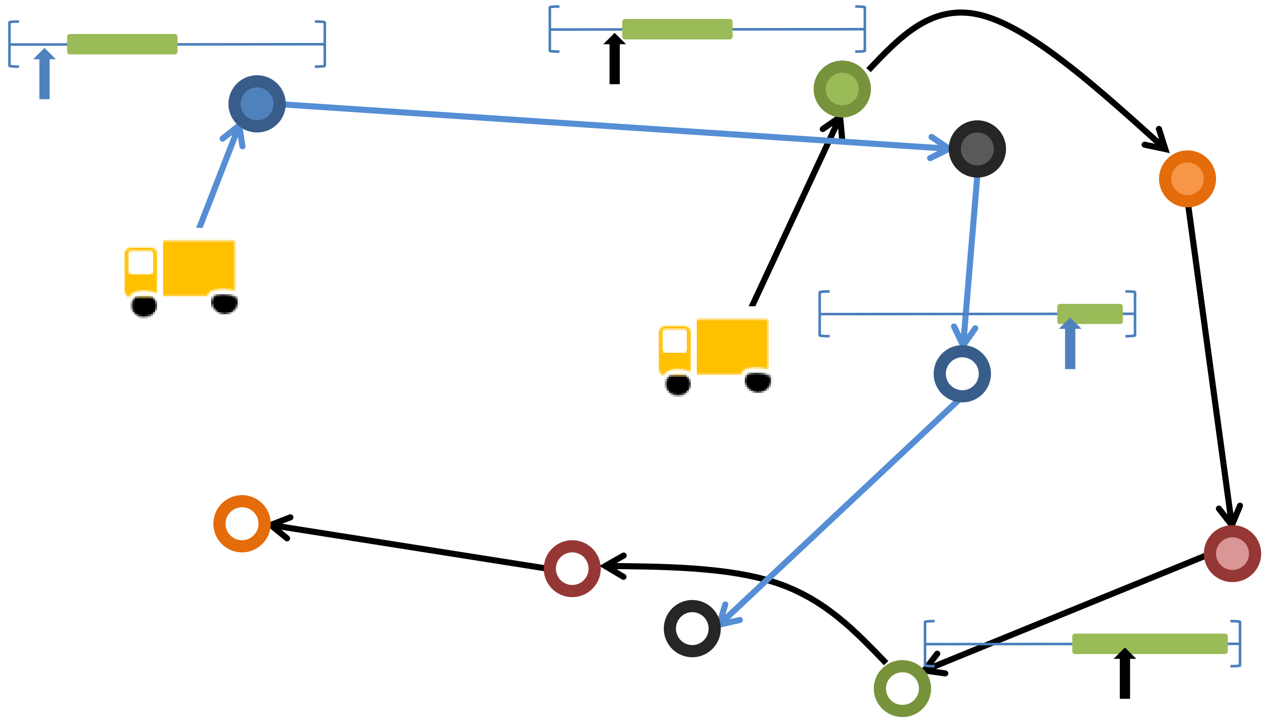
- A set of vehicles with:
  - Individual starting location (no depot)
  - Open ended
  - All vehicles are identical

Problem definition



- Time windows at pickup and delivery

Problem definition



- Pickup and Delivery Problem with time windows (PDPTW)

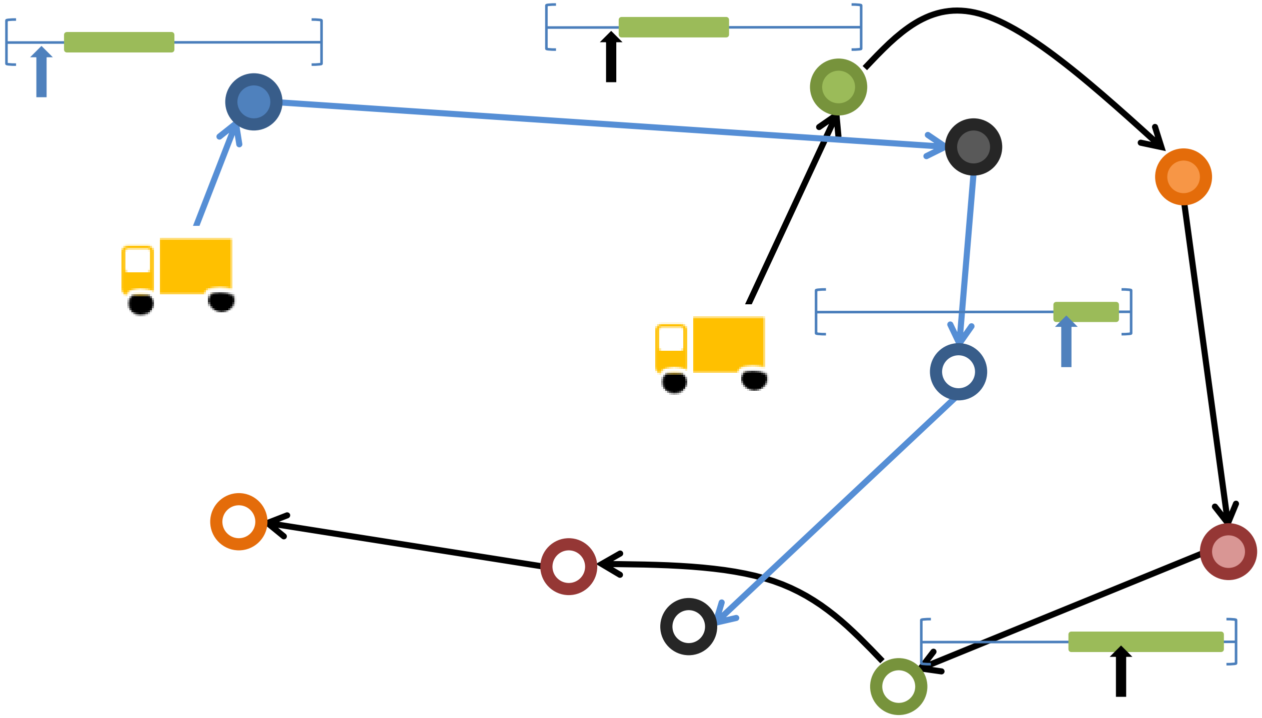
- No depot
- Open ended
- Only optional requests

NTNU

Relevant literature on PDPTW– CG stuff only

- Dumas et al (1991) – First BP alg.
- Røpke and Cordeau (2009) – First BPC alg.
- Baldacci et al. (2011) – Route enumerations
- Gschwind et al (2018) – Bi-directional
- Homsi et al (2020) – Ship routing and scheduling

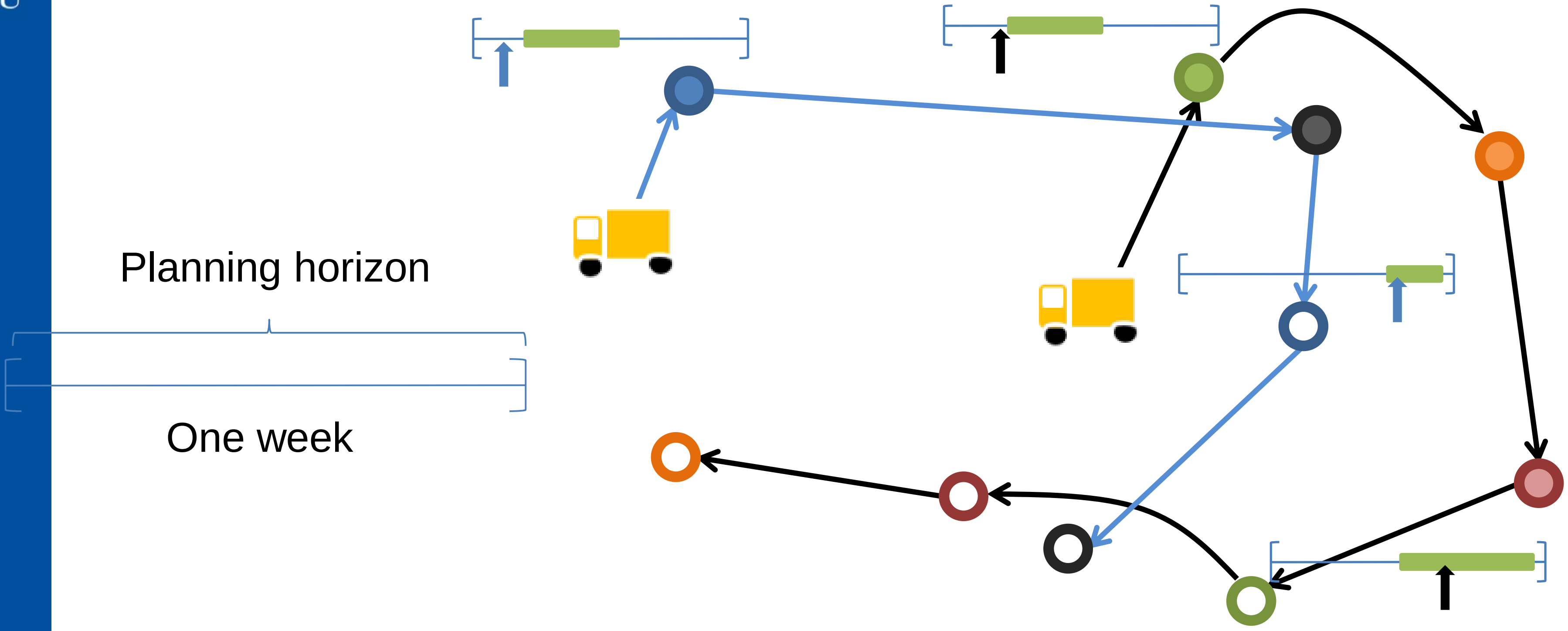
Problem definition



- Pickup and Delivery Problem with time windows (PDPTW)

- No depot
- Open ended
- Only optional requests

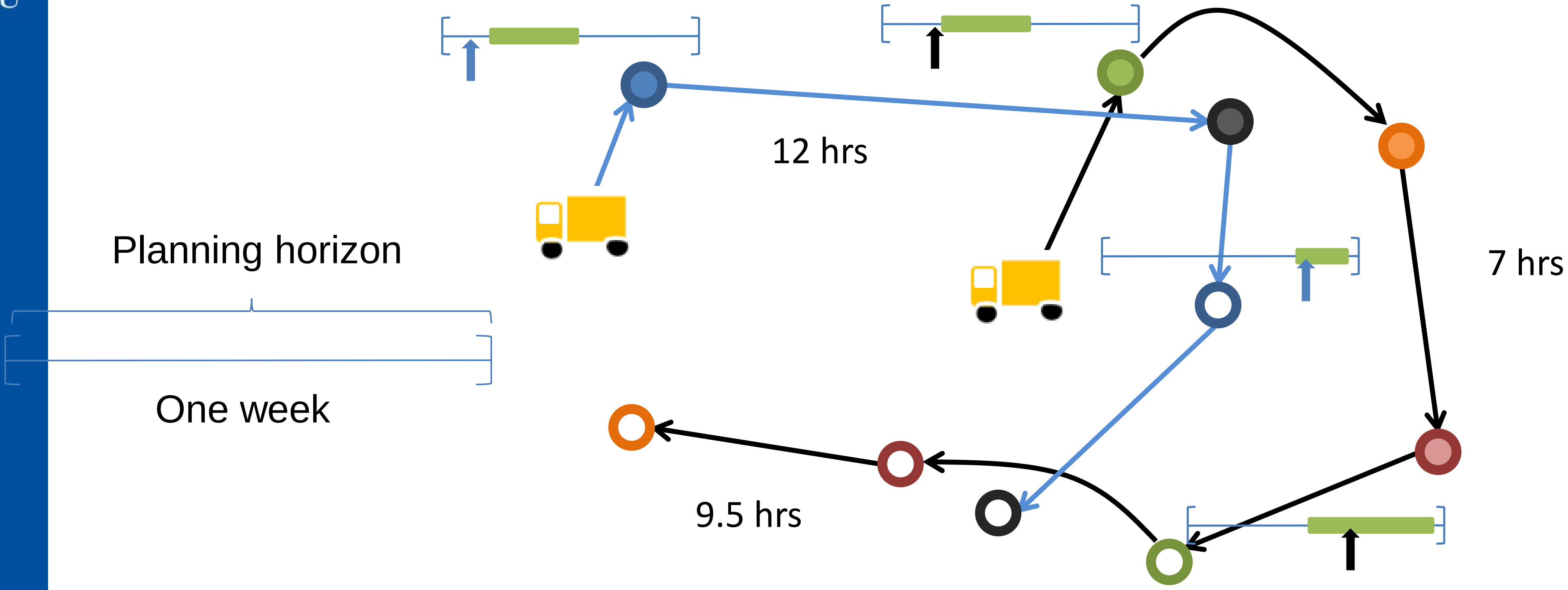
Problem definition



- Pickup and Delivery Problem with time windows (PDPTW)

- No depot
- Open ended
- Only optional requests

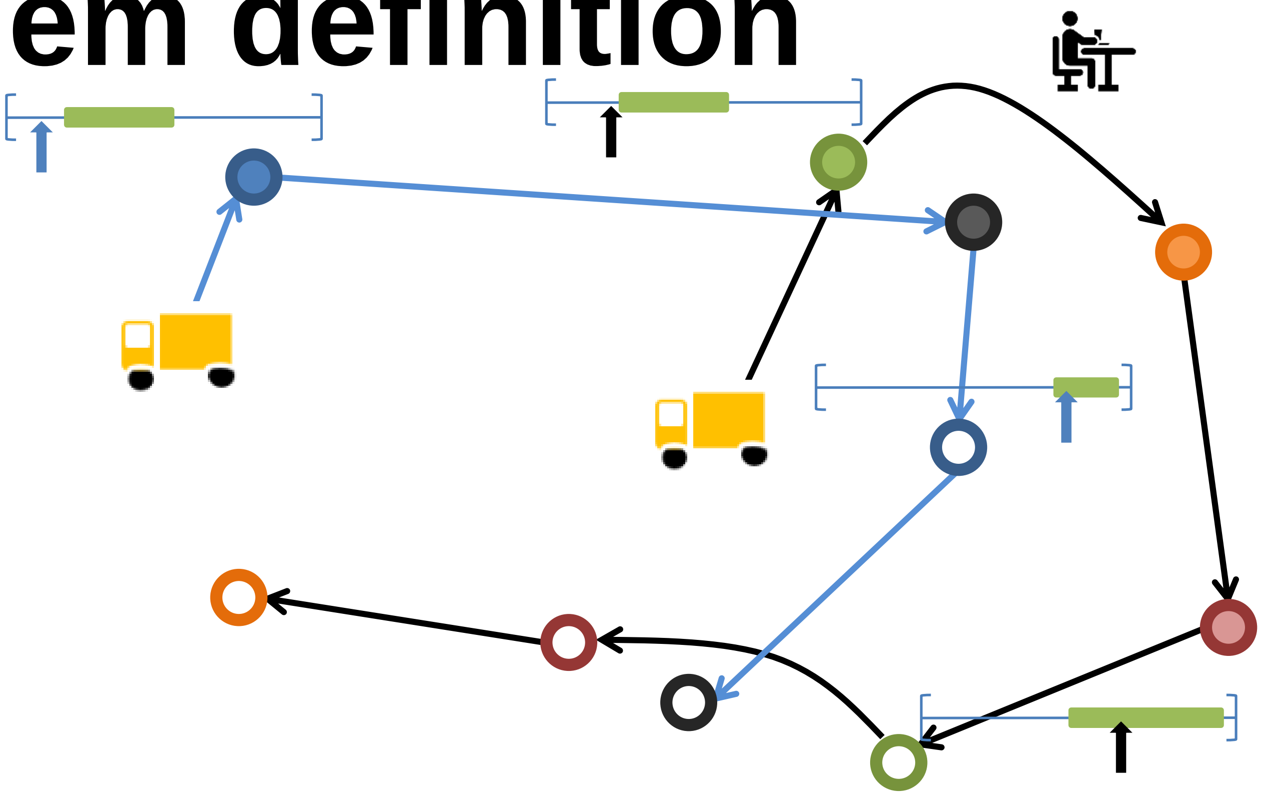
Problem definition



- Pickup and Delivery Problem with time windows (PDPTW)

- No depot
- Open ended
- Only optional requests

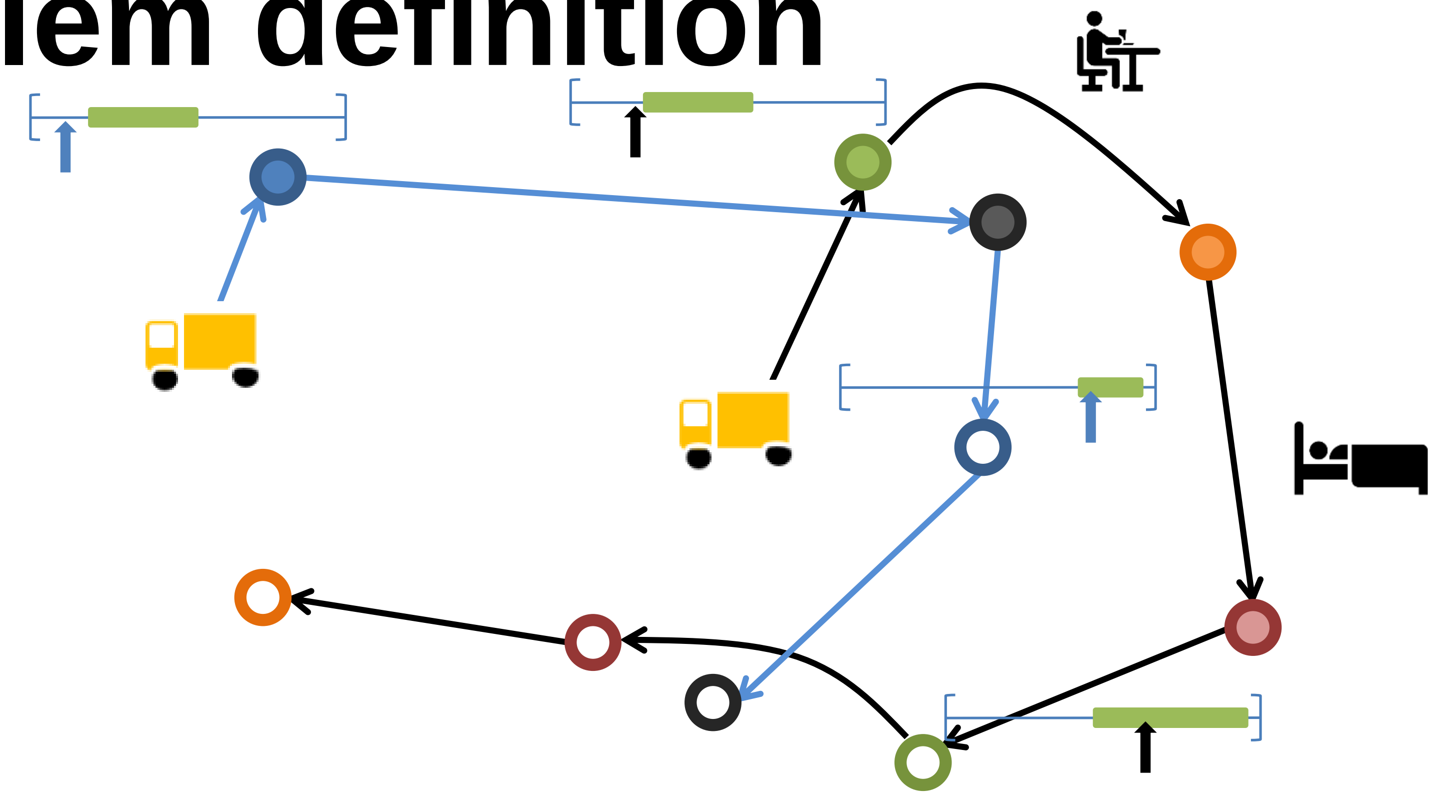
Problem definition



- Pickup and Delivery Problem with time windows (PDPTW)

- No depot
- Open ended
- Only optional requests
- Breaks

Problem definition



- Pickup and Delivery Problem with time windows (PDPTW)

- No depot
- Open ended
- Only optional requests
- Breaks
- Rests

European Hours of Service regulations

	Break - 45 min	Rest - 11 hrs	Weekly rest
Drive	4.5 hrs	9 hrs	56 hrs
Work	6.0 hrs	13 hrs	60 hrs

NTNU

Relevant literature – CG stuff only

- VRP with Truck driver scheduling
  - Goel and Irnich (2017) - First BP alg.
  - Tilk and Goel (2020) – Bi-directional

Solution method

NTNU

Master problem

$$\max \quad z = \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}_v} P_{vr} \lambda_{vr},$$

$$\sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}_v} A_{ivr} \lambda_{vr} \leq 1, \qquad i \in N^P,$$

$$\sum_{r \in \mathcal{R}_v} \lambda_{vr} = 1, \qquad v \in \mathcal{V},$$

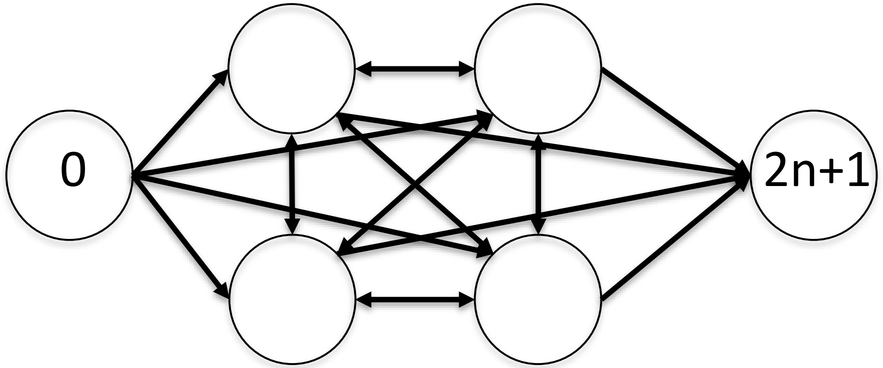
$$\lambda_{vr} \in \{0, 1\}, \qquad \forall v \in \mathcal{V}, r \in \mathcal{R}_v.$$

Subproblem

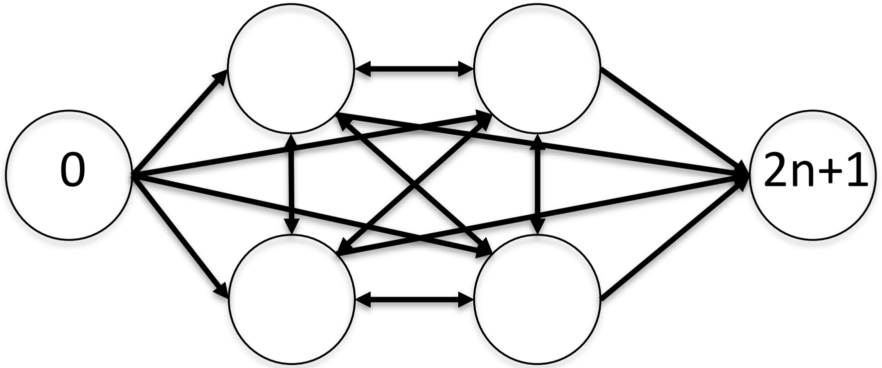
- Resource constrained shortest path problem
- Combination of:
  - Røpke and Cordeau (2009)
  - Goel and Irnich (2017)
  - With some modifications

One subproblem pr vehicle

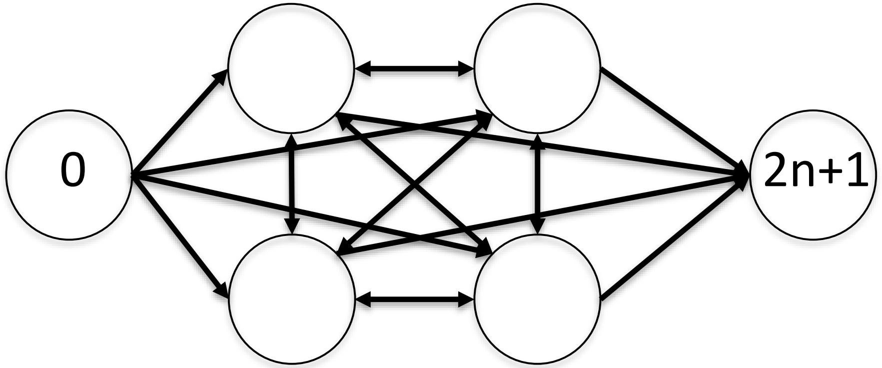
- Vehicle 1:



- Vehicle 2:

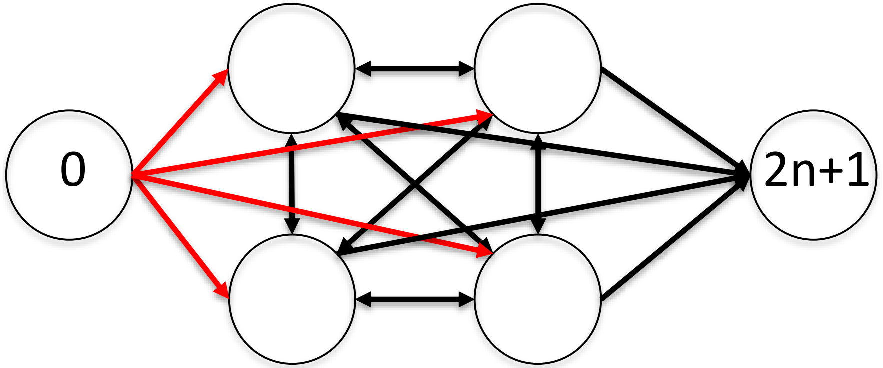


- Vehicle n:

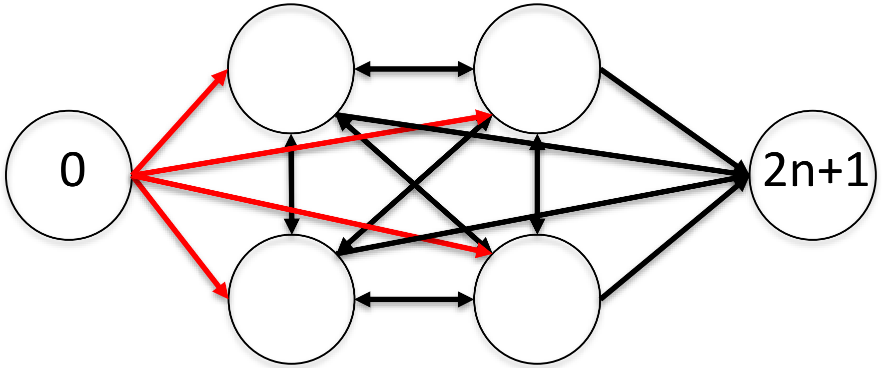


One subproblem pr vehicle

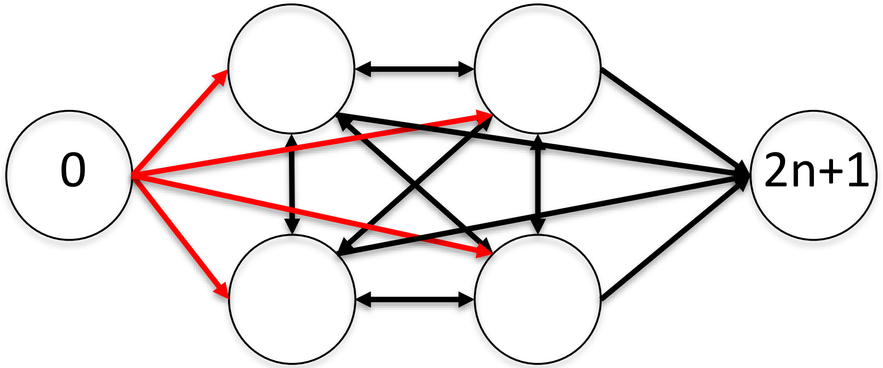
- Vehicle 1:



- Vehicle 2:

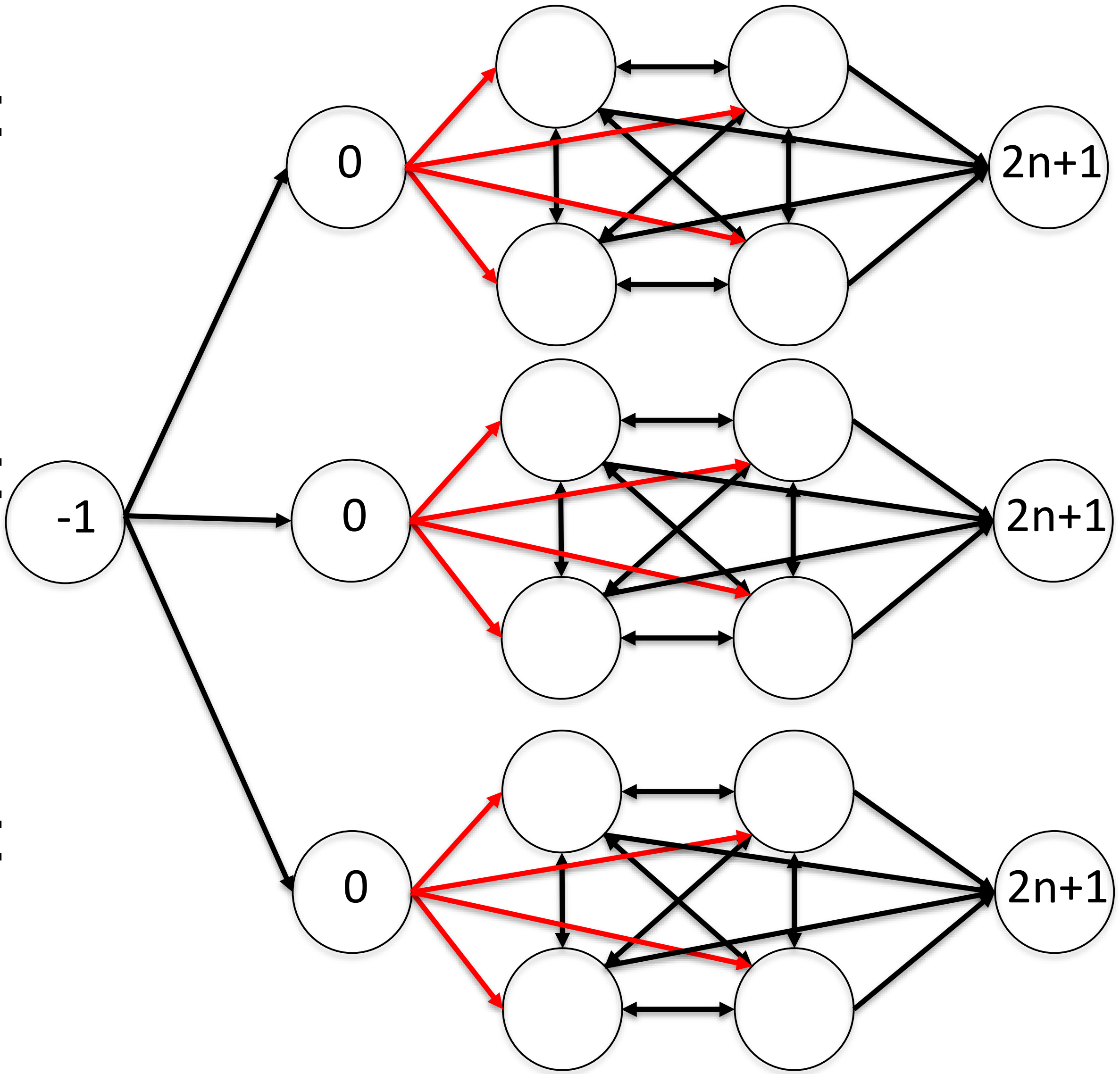


- Vehicle n:



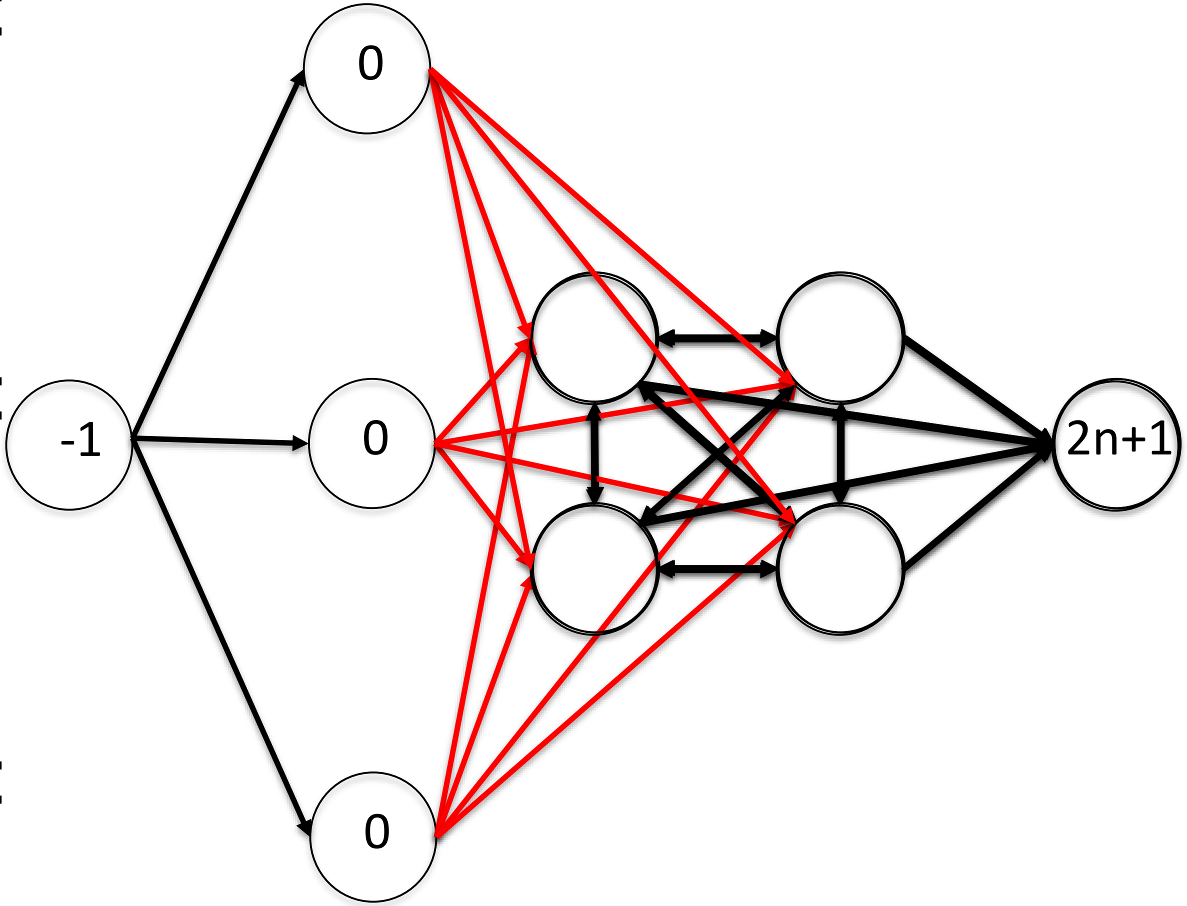
One subproblem pr vehicle

- Vehicle 1:
- Vehicle 2:
- Vehicle n:



One subproblem ~~pr vehicle~~

- Vehicle 1:
- Vehicle 2:
- Vehicle n:



Resources needed for the PDPTW

- Based on Røpke and Cordeau (2009)

Resource	Resource Description	Resource Window
$\overline{p}$	Accumulated reduced cost after visiting node i	$[-\infty, \infty]$
l^W	Load of the vehicle after visiting node i in terms of weight	$[0, W^C]$
l^V	Load of the vehicle after visiting node i in terms of volume	$[0, V^C]$
t^{time}	Time elapsed since start of route	$[\underline{T_i}, \overline{T_i}]$
$\mathcal{U}$	Set of unreachable nodes on the route	$\mathcal{U} \subseteq N^P$
$\mathcal{O}$	Set of requests started but not completed on this route	$\mathcal{O} \subseteq N^P$

Resources needed for TDS

- Based on Goel and Irnich (2017)

Resource	Resource Description	Resource Window
t^{dist}	Remaining driving time to the next node, j	$[0, 0]$
$t^{drive\|R}$	Accumulated driving time since the end of the last rest	$[0, T^{drive\|R}]$
$t^{elapsed\|R}$	Time elapsed since the end of the last rest	$[0, T^{elapsed\|R}]$
$t^{latest\|R}$	Latest time for when a rest must end	$[0, \infty]$
$t^{drive\|B}$	Accumulated driving time since the last break or rest	$[0, T^{drive\|B}]$
$t^{elapsed\|B}$	Time elapsed since the end of the last break or rest	$[0, T^{elapsed\|B}]$
$t^{latest\|B}$	Latest time for when a break must end	$[0, \infty]$
$t^{drive\|W}$	Total accumulated weekly driving time	$[0, T^{drive\|W}]$
$t^{elapsed\|W}$	Time elapsed since the end of the last weekly rest	$[0, T^{elapsed\|W}]$
$t^{latest\|W}$	Latest time for when a weekly rest must end	$[0, \infty]$

Network modification for the TDS

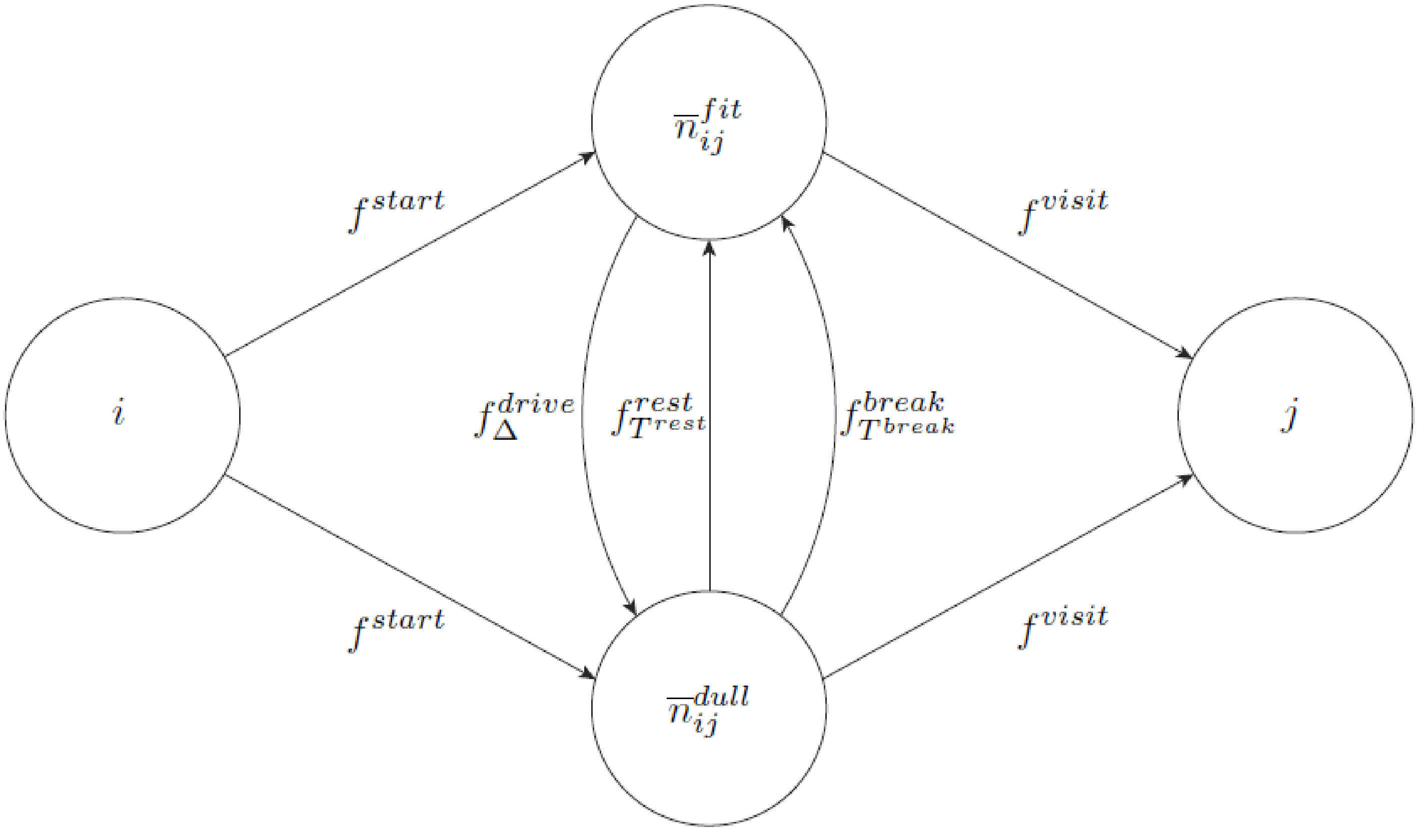


Figure 1: Auxiliary Network proposed by Goel and Irnich (2017). The network describes the possible extensions of a label by traversing the arc (i, j).

Resource extension functions

Resource	Resource Extension Function			
	f^{drive}	f^{break}	f^{rest}	f^{visit}
$\overline{p}(L')$				$\overline{p}(L) + \overline{C}_{ij}$
$l^W(L')$				$l^W(L) + D_j^W$
$l^V(L')$				$l^V(L) + D_j^V$
$t^{time}(L')$	$t^{time}(L) + \Delta$	$t^{time}(L) + T^{break}$	$t^{time}(L) + T^{rest}$	$max\{t^{time}(L), \underline{T}_j\} + S_j$
$t^{dist}(L')$	$t^{dist}(L) - \Delta$			
$t^{drive\|R}(L')$	$t^{drive\|R}(L) + \Delta$		0	
$t^{elapsed\|R}(L')$	$t^{elapsed\|R}(L) + \Delta$	$t^{elapsed\|R}(L) + T^{break}$	0	$max\{t^{elapsed\|R}(L), \underline{T}_j - t^{latest\|R}(L)\} + S_j$
$t^{latest\|R}(L')$			∞	$min\{t^{latest\|R}(L), \overline{T}_j + S_j - t^{elapsed\|R}(L)\}$
$t^{drive\|B}(L')$	$t^{drive\|B}(L) + \Delta$	0	0	
$t^{elapsed\|B}(L')$	$t^{elapsed\|B}(L) + \Delta$	0	0	$max\{t^{elapsed\|B}(L), \underline{T}_j - t^{latest\|B}(L)\} + S_j$
$t^{latest\|B}(L')$	$min\{t^{latest\|B}(L), t^{latest\|R}(L) + t^{elapsed\|R}(L) - t^{elapsed\|B}(L) - \Delta\}$	∞	∞	$min\{t^{latest\|B}(L), \overline{T}_j + S_j - t^{elapsed\|B}(L)\}$
$t^{drive\|W}(L')$	$t^{drive\|W}(L) + \Delta$			
$t^{work\|W}(L')$	$t^{work\|W}(L) + \Delta$			$t^{work\|W}(L) + S_j$
$\mathcal{U}(L')$				$\mathcal{U}(L) \cup \{j\} \cup \mathcal{U}_{j,t}$
$\mathcal{O}(L')$				$\begin{cases} \mathcal{O}(L) \cup \{j\} & \text{if } j \in \mathcal{P} \\ \mathcal{O}(L) \setminus \{j - N^P\} & \text{if } j \in \mathcal{D} \end{cases}$

Improvements

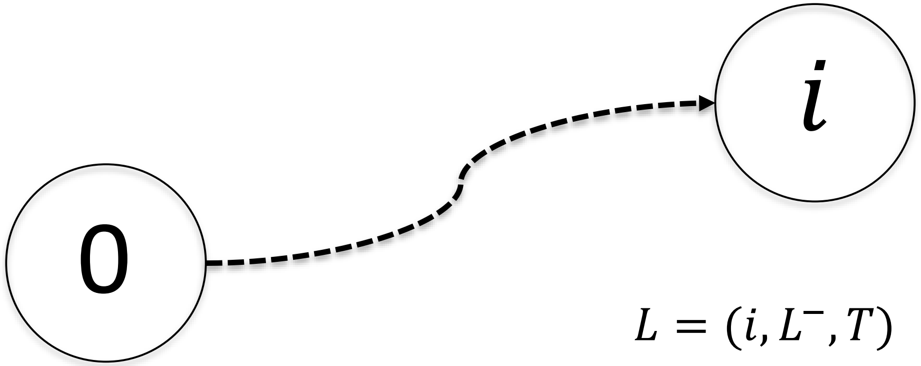
- Combining existing methods work, but can we improve?
- Lots of resources, and nodes gives lots of labels
- What can we do:
  - Relax the subproblem
  - Discard labels earlier
  - Strengthen dominance

Relaxing the subproblem

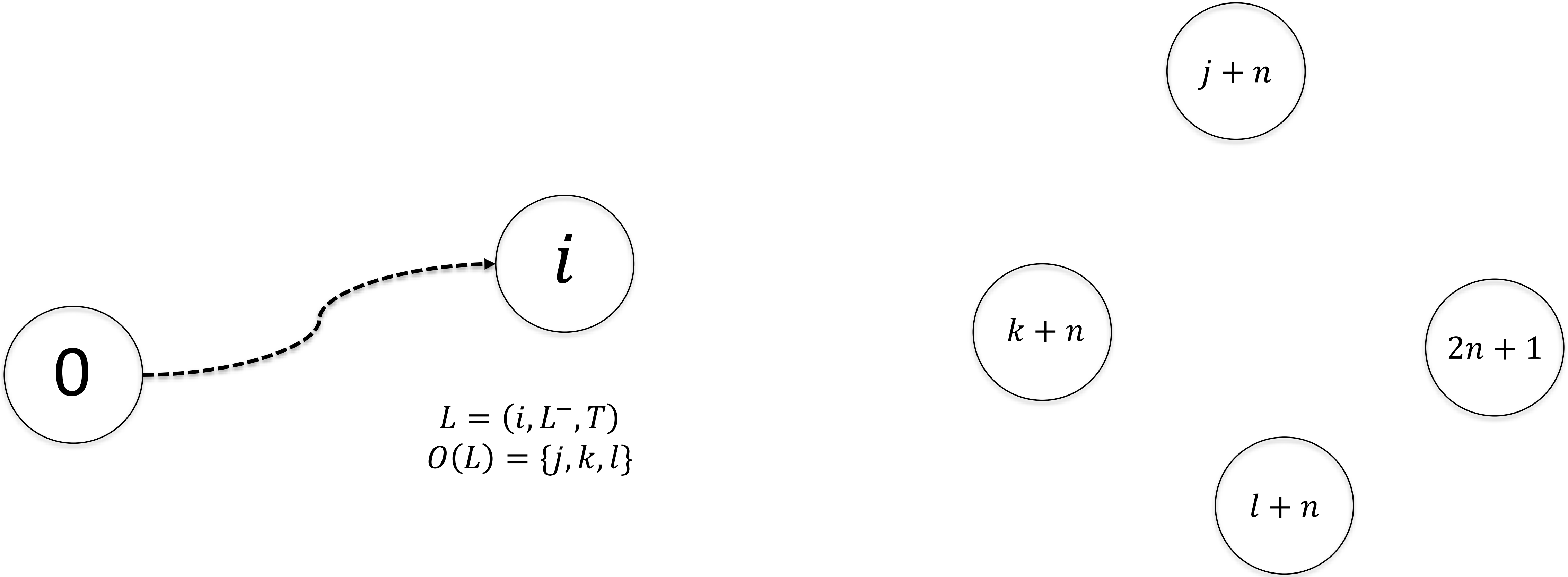
- Solve labeling alg. without «break»-resources
  - Only affects feasibility, not optimality.

- Three cases:
  - Finds no routes - > RMP is optimal
  - Finds at least one feasible route with positive reduced cost -> new CG iteration
  - Finds at least one route, but none are feasible -> solve full SP

Discarding Labels

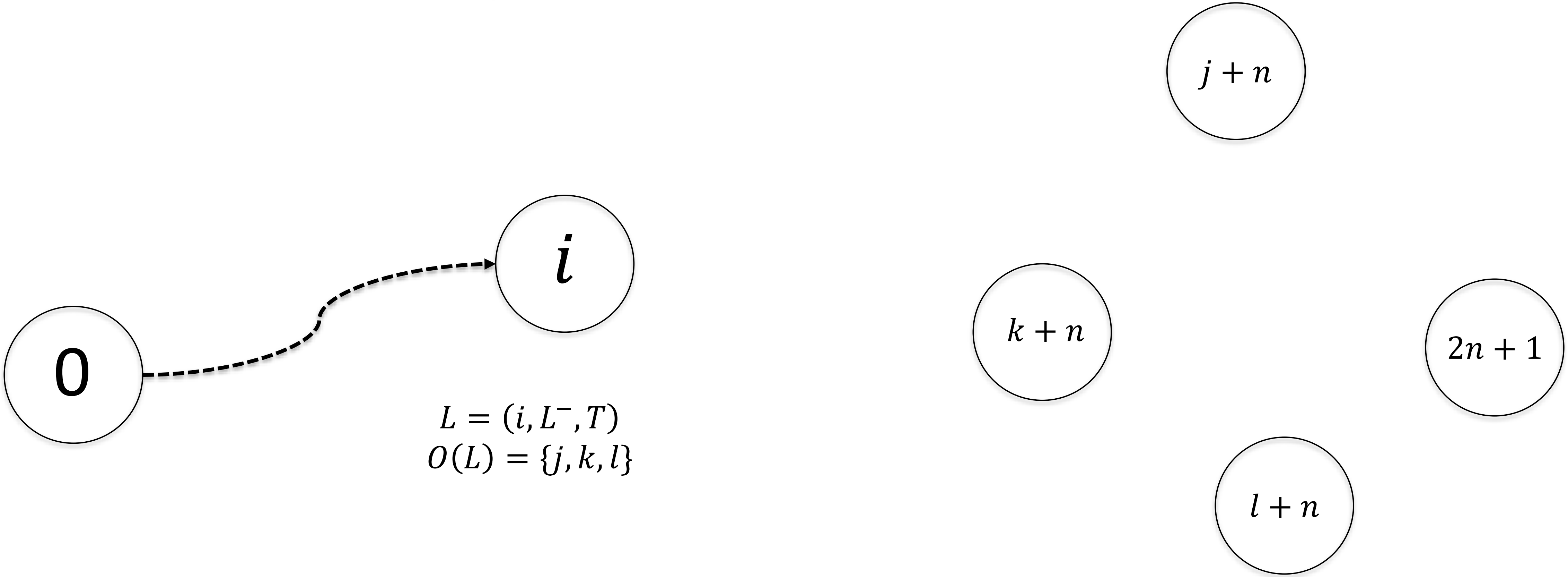


Discarding Labels



$$L = (i, L^{-}, T)$$
$$O(L) = \{j, k, l\}$$

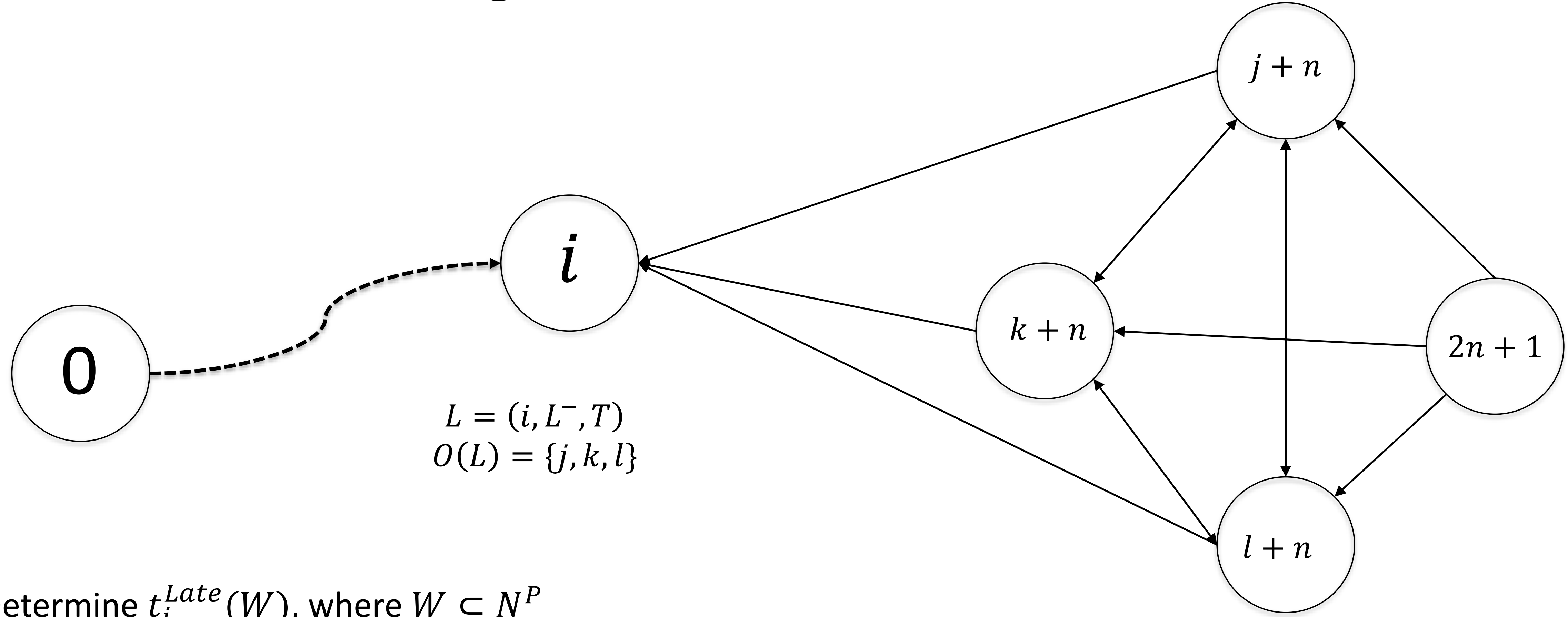
Discarding Labels



$$L = (i, L^{-}, T)$$
$$O(L) = \{j, k, l\}$$

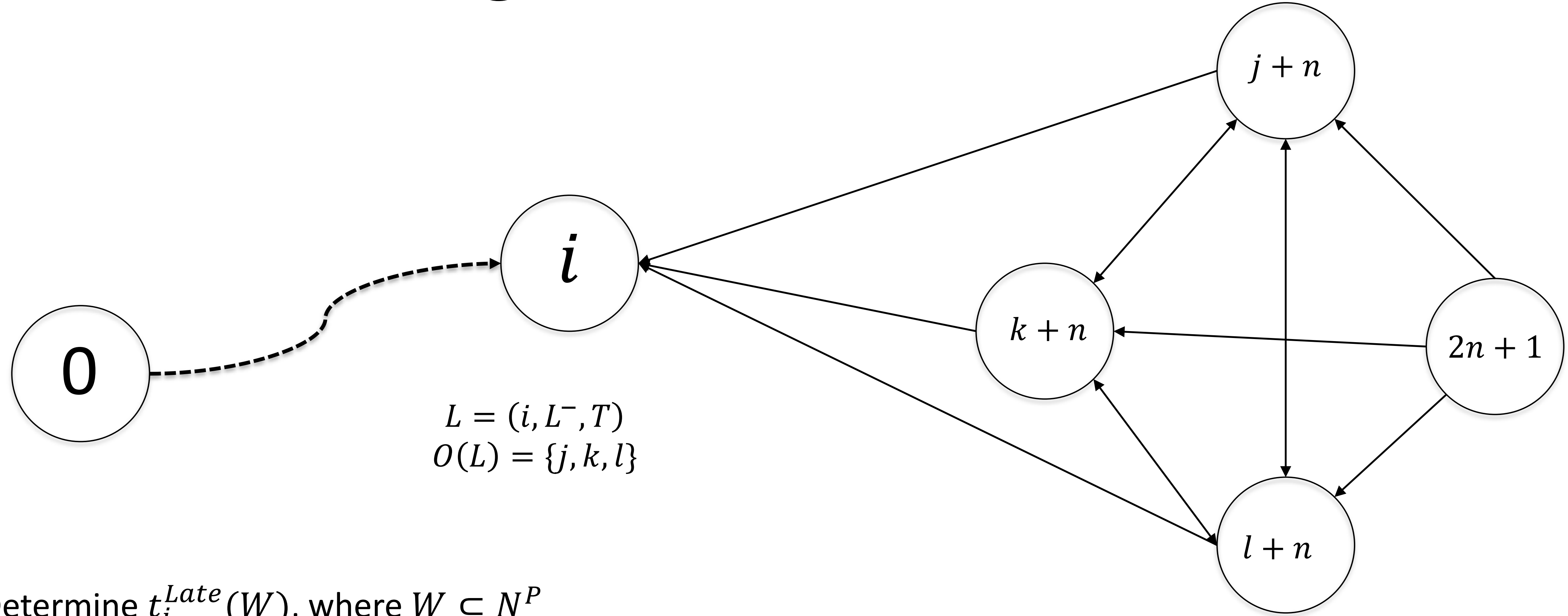
What is the latest time we can leave i, given that we have to visit all these nodes?

Discarding Labels



Determine $t_i^{Late}(W)$, where $W \subset N^P$

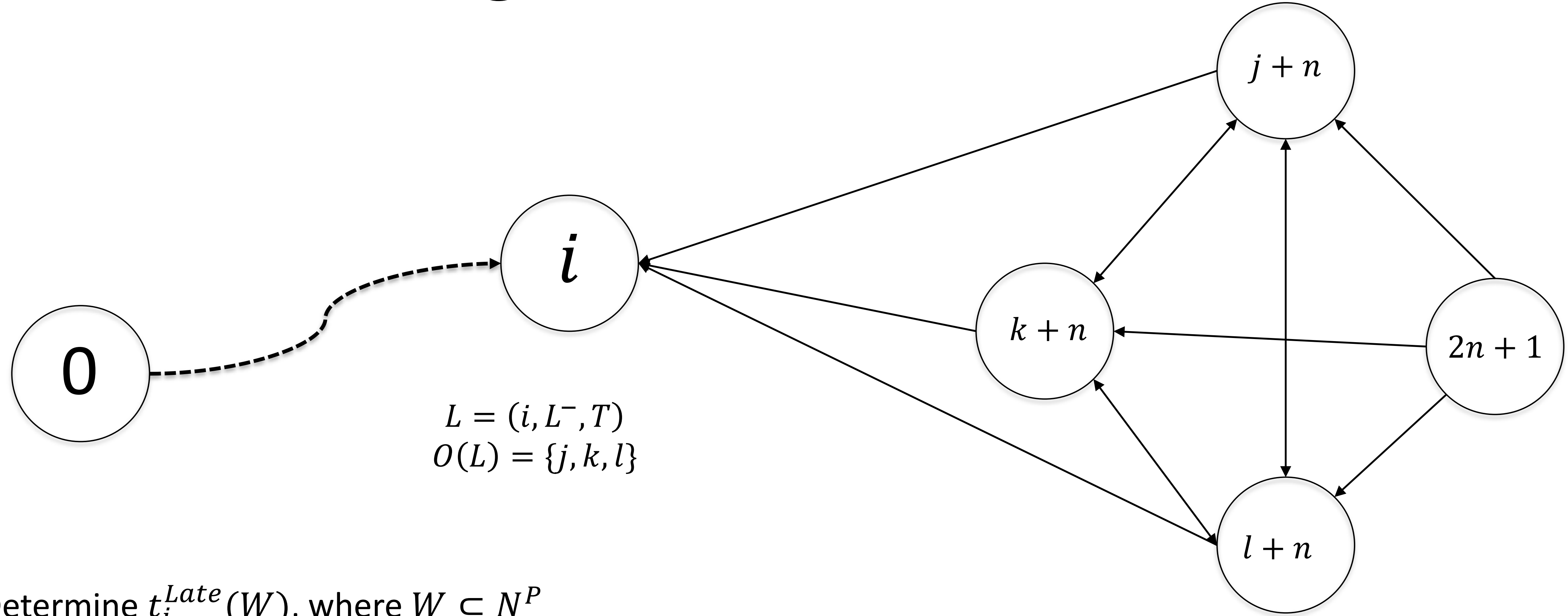
Discarding Labels



$L = (i, L^{-}, T)$
$O(L) = \{j, k, l\}$

Determine $t_i^{Late}(W)$, where $W \subset N^P$
This can be solved as a backward labeling, given one additional resource to ensure all nodes are visited

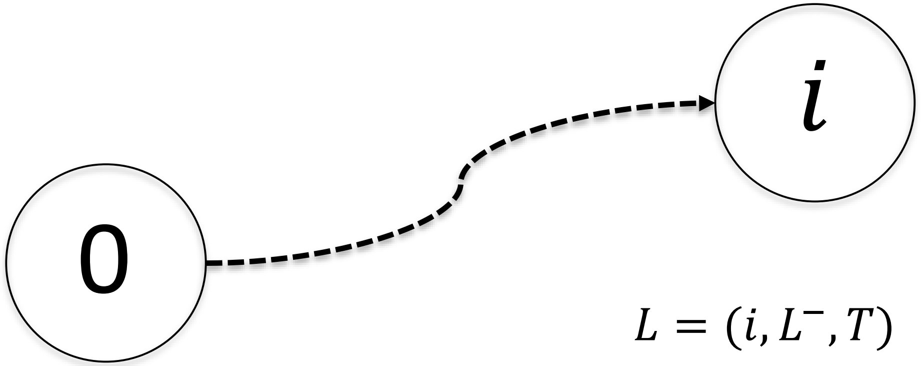
Discarding Labels



$L = (i, L^-, T)$
$O(L) = \{j, k, l\}$

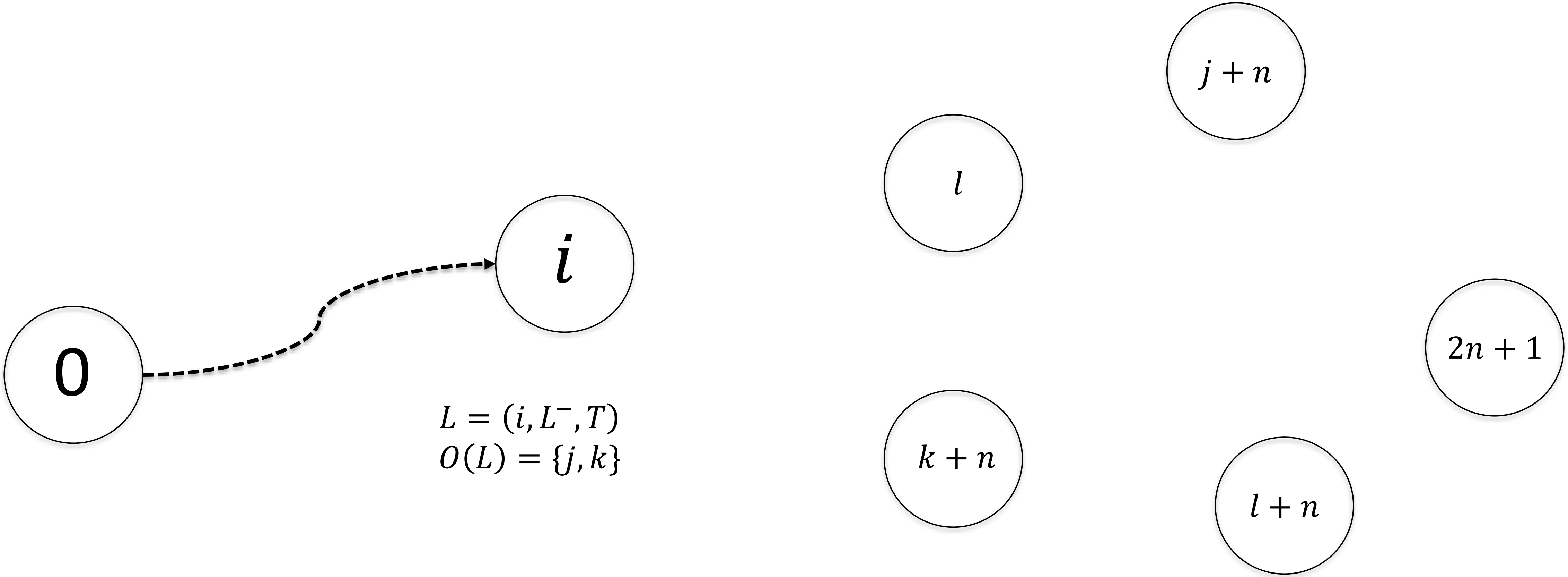
Determine $t_i^{Late}(W)$, where $W \subset N^P$
This can be solved as a backward labeling, given one additional resource to ensure all nodes are visited
If $t^{Time}(L) > t_i^{Late}(O(L))$ we may discard L

Increasing the unreachable set

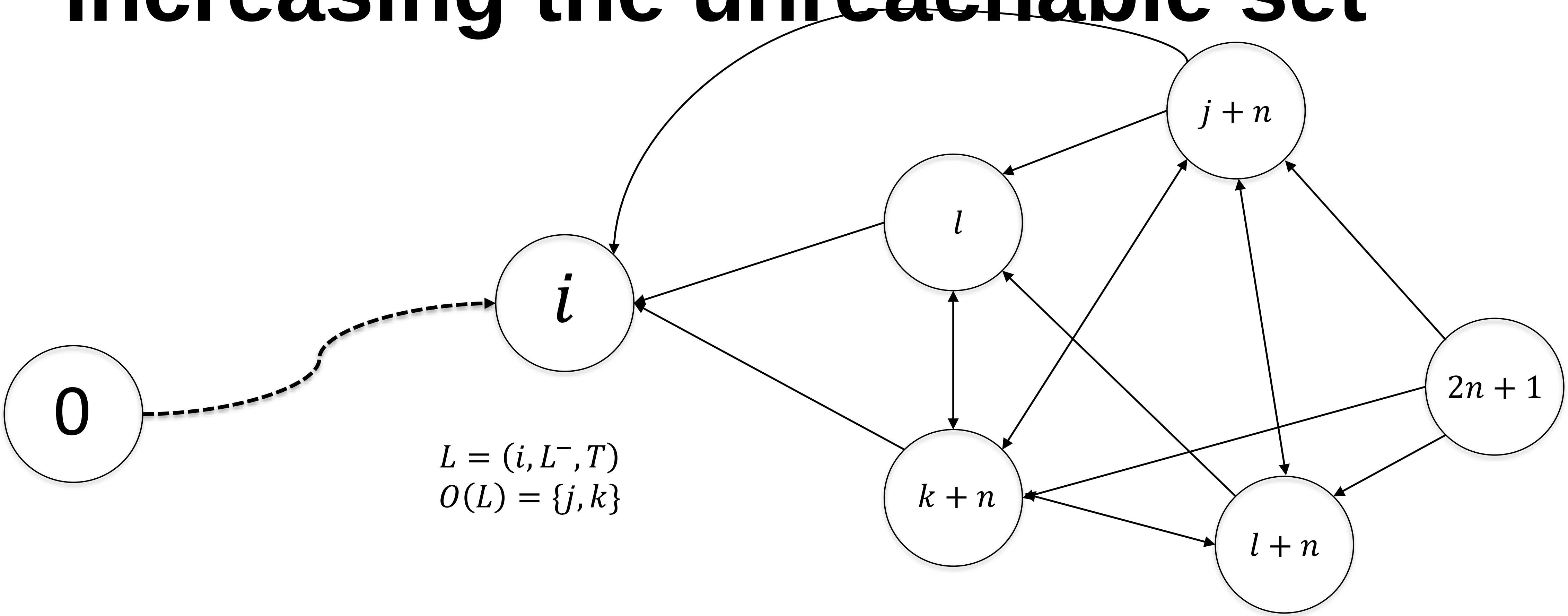


$$L = (i, L^{-}, T)$$

Increasing the unreachable set

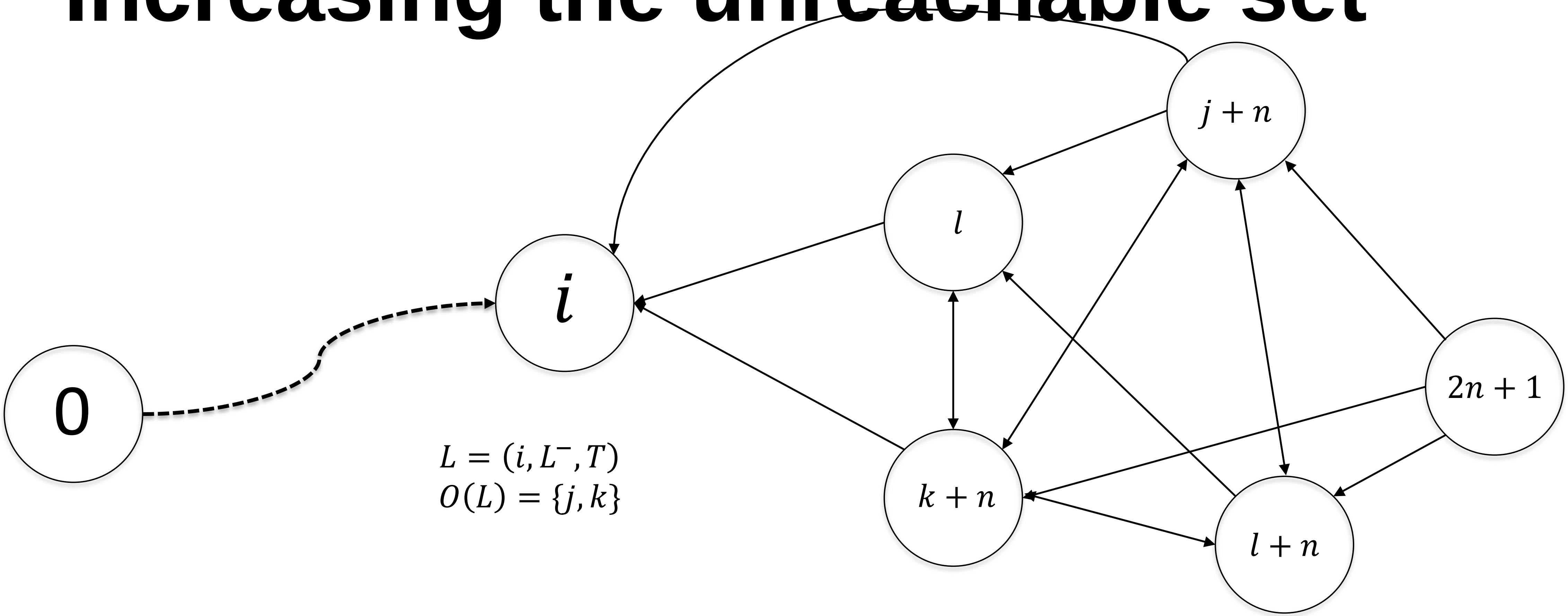


Increasing the unreachable set



$L = (i, L^{-}, T)$
$O(L) = \{j, k\}$

Increasing the unreachable set



Determine $t_i^{Late}(l, W)$, where $l \in N^P, W \subset N^P$

This can be solved as a backward labeling, ensuring all nodes are visited and presedence for $l, l+n$

$U(L) = U(L) \cup \{l \in N^P \mid t^{Time}(L) > t_i^{Late}(l, O(L))\}$

Preprocessing of t^{Late}

- $\forall i \in N, W \subset N^P, |W| \leq 3,$ calculate $t_i^{Late}(W)$
- $\forall i \in N, l \in N^P, W \subset N^P, |W| \leq 2,$ calculate $t_i^{Late}(l, W)$
- In both cases we omit the break resources

Preprocessing of t^{Late}

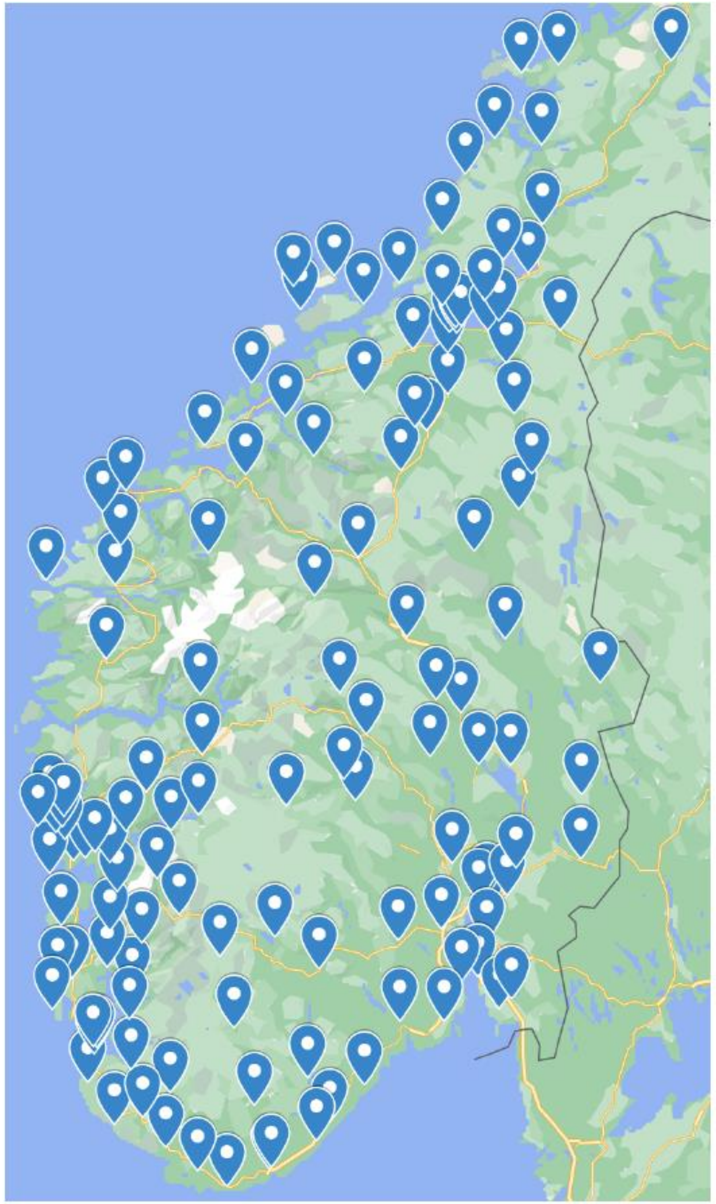
- $\forall i \in N, W \subset N^P, |W| \leq 3,$ calculate $t_i^{Late}(W)$
- $\forall i \in N, l \in N^P, W \subset N^P, |W| \leq 2,$ calculate $t_i^{Late}(l, W)$
- In both cases we omit the break resources

- Note that if
  - $\exists W \subseteq O(L), t^{Time}(L) > t_i^{Late}(W),$ we can discard L
  - $U(L) = U(L) \cup \{l \in N^P \middle| \exists W \subseteq O(L), t^{Time}(L) > t_i^{Late}(l, W)\}$

Computational Results

Test instances

- 132 locations from central and southern Norway
- Distances and times based on Google Maps
- Probability of drawing each location proportional to population
- Planning horizon of 144 hours
- Three time windows widths: 12-24, 24-48, 48-144 hours
- Two cargo sizes: 1-10, 10-20 (capacity of vehicle 30)
- # requests = 10, 15, 20, 25, 50, 75, 100, 150, 200
- # Vehicles = $\left\lfloor \frac{\#\,requests}{4} \right\rfloor, \left\lfloor \frac{\#\,requests}{5} \right\rfloor, \left\lfloor \frac{\#\,requests}{6} \right\rfloor$
- Four instances of each setting gives 648 instances total



Effect of the preprocessing

Table 3: The number of instances with a given number of requests where the BP method can find an optimal solution (# opt.) and a dual bound (# DB), and the average computing time (Avg. time) used, with and without the preprocessing techniques from Section 4.5.

	Without preprocess			With preprocess		
# requests	**# opt.**	**# DB**	**Avg. time**	**# opt.**	**# DB**	**Avg. time**
10	63	63	1218.79	66	68	631.99
15	46	47	3063.39	60	60	1217.78
20	28	29	4575.96	58	60	1679.74
25	23	24	4989.11	58	60	1801.29
50	6	6	6873.35	31	37	4357.43
Total	166	169	4144.12	273	285	1937.64

Effect of preprocessing

Table 4: The number of instances with a given time window width where the BP method can find an optimal solution (# opt.) and a dual bound (# DB), and the average computing time (Avg. time) used, with and without the preprocessing techniques from Section 4.5.

	Without preprocess			With preprocess		
Time Windows	**# opt.**	**# DB**	**Avg. time**	**# opt.**	**# DB**	**Avg. time**
12-24	81	81	2695.85	120	120	40.23
24-48	61	63	3775.15	103	109	1236.03
48-144	24	25	5961.36	50	56	4536.68

Effect of preprocessing on PDPTW

- Tested on the instances propoced by Røpke and Cordeau (2009)

	without preprocess	with preprocess
AA	637.7	591.9
BB	841.1	785.4
CC	3047.8	3038.9
DD	3124.0	3040.7

Summary

- Presented new problem (and acronym) to the research community
  - PDPTDS
- Preprocesing of time windows based on the open set can reduce computational time significantly
- Does this carry over to the bi-directional case?
- Instances, results and preprint of paper found:
  - http://axiomresearchproject.com/publications