Department of Economics and Business
Aarhus University

Ranking paths in stochastic time-dependent networks

Lars Relund Nielsen and Kim Allan Andersen

Department of Economics and Business, Aarhus University, Denmark (lars@relund.dk)

Daniele Pretolani

Department of Science and Methods of Engineering, University of Modena and Reggio Emilia, Italy

VeRoLog - June 18-20, 2012

Agenda

Stochastic time-dependent networks

Given a topological network $G = (N, A)$

- Departure and arrival times at nodes are integer.

Stochastic time-dependent networks

Given a topological network $G = (N, A)$

- Departure and arrival times at nodes are integer.
- Time-dependent - The travel time between two nodes depends on the leaving time from the tail node.

Stochastic time-dependent networks

Given a topological network $G = (N, A)$

- Departure and arrival times at nodes are integer.
- Time-dependent - The travel time between two nodes depends on the leaving time from the tail node.
- Stochastic - Travel time through an arc is not known in advance and represented by a discrete random variable.

Stochastic time-dependent networks

Given a topological network $G = (N, A)$

- Departure and arrival times at nodes are integer.
- Time-dependent - The travel time between two nodes depends on the leaving time from the tail node.
- Stochastic - Travel time through an arc is not known in advance and represented by a discrete random variable.
- No online information - Only information while travelling is arrival time.

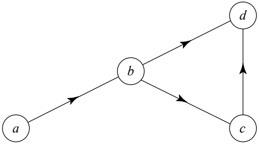
Stochastic time-dependent networks

Given a topological network $G = (N, A)$

- Departure and arrival times at nodes are integer.
- Time-dependent - The travel time between two nodes depends on the leaving time from the tail node.
- Stochastic - Travel time through an arc is not known in advance and represented by a discrete random variable.
- No online information - Only information while travelling is arrival time.
- No waiting allowed at nodes (arrival = departure).

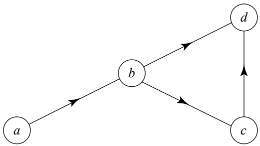
A hypergraph representation

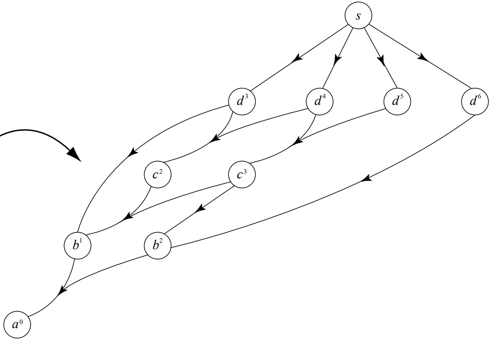
$(u,v),t$	$(a,b),0$	$(b,c),1$	$(b,c),2$	$(b,d),1$	$(b,d),2$	$(c,d),2$	$(c,d),3$
$I(u,v,t)$	$\{1,2\}$	$\{2,3\}$	$\{3\}$	$\{3\}$	$\{6\}$	$\{3,4\}$	$\{4,5\}$
p_{ijt}	$\{\frac{1}{2},\frac{1}{2}\}$	$\{\frac{1}{2},\frac{1}{2}\}$	$\{1\}$	$\{1\}$	$\{1\}$	$\{\frac{1}{2},\frac{1}{2}\}$	$\{\frac{1}{2},\frac{1}{2}\}$



A hypergraph representation

$(u,v),t$	$(a,b),0$	$(b,c),1$	$(b,c),2$	$(b,d),1$	$(b,d),2$	$(c,d),2$	$(c,d),3$
$I(u,v,t)$	$\{1,2\}$	$\{2,3\}$	$\{3\}$	$\{3\}$	$\{6\}$	$\{3,4\}$	$\{4,5\}$
p_{ijt}	$\{\frac{1}{2},\frac{1}{2}\}$	$\{\frac{1}{2},\frac{1}{2}\}$	$\{1\}$	$\{1\}$	$\{1\}$	$\{\frac{1}{2},\frac{1}{2}\}$	$\{\frac{1}{2},\frac{1}{2}\}$





Route choice in STD networks

A **routing routing strategy** between an origin *o* and a destination *d* when leaving the origin at time zero assigns to each node and possible leaving time and sucessor arc.

Route choice in STD networks

A **routing routing strategy** between an origin *o* and a destination *d* when leaving the origin at time zero assigns to each node and possible leaving time and sucessor arc.
Two ways of routing in an STD network

- Time-adaptive routing - The traveler may react on arrival time realizations (different sucessors allowed).

Route choice in STD networks

A **routing routing strategy** between an origin *o* and a destination *d* when leaving the origin at time zero assigns to each node and possible leaving time and sucessor arc.

Two ways of routing in an STD network

- Time-adaptive routing - The traveler may react on arrival time realizations (different sucessors allowed).
- A priori routing - A loopless path in *G* must be specified before travel begins (sucessors must be the same).

Route choice in STD networks

A **routing routing strategy** between an origin o and a destination d when leaving the origin at time zero assigns to each node and possible leaving time and sucessor arc.

Two ways of routing in an STD network

- Time-adaptive routing - The traveler may react on arrival time realizations (different sucessors allowed).
- A priori routing - A loopless path in G must be specified before travel begins (sucessors must be the same).

In this talk we consider a priori routing

Route selection criteria

We consider the following objectives:

- Minimizing the expected travel time (MET)

Under MEC we define the following costs:

- $c(u, v, t)$ – cost of leaving node u at time t along arc (u, v).
- $g(t)$ – penalty cost of arriving at node d at time t.

Route selection criteria

We consider the following objectives:

- Minimizing the expected travel time (MET)
- Minimizing the expected cost (MEC)

Under MEC we define the following costs:

- $c(u, v, t)$ – cost of leaving node u at time t along arc (u, v).
- $g(t)$ – penalty cost of arriving at node d at time t.

Route selection criteria

We consider the following objectives:

- Minimizing the expected travel time (MET)
- Minimizing the expected cost (MEC)

Under MEC we define the following costs:

- $c(u, v, t)$ – cost of leaving node u at time t along arc (u, v).
- $g(t)$ – penalty cost of arriving at node d at time t.

Route selection criteria

We consider the following objectives:

- Minimizing the expected travel time (MET)
- Minimizing the expected cost (MEC)

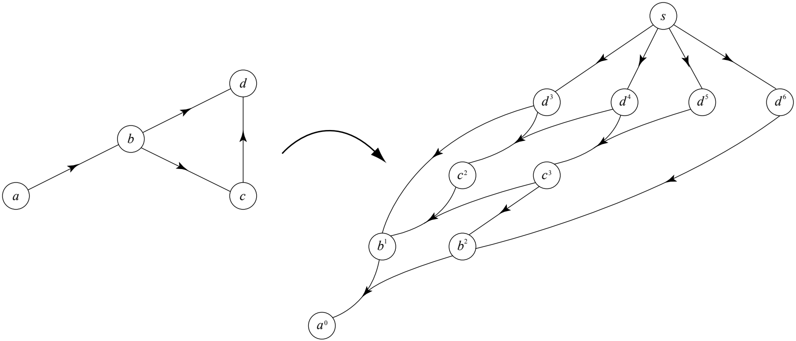
Under MEC we define the following costs:

- $c(u, v, t)$ – cost of leaving node u at time t along arc (u, v).
- $g(t)$ – penalty cost of arriving at node d at time t.

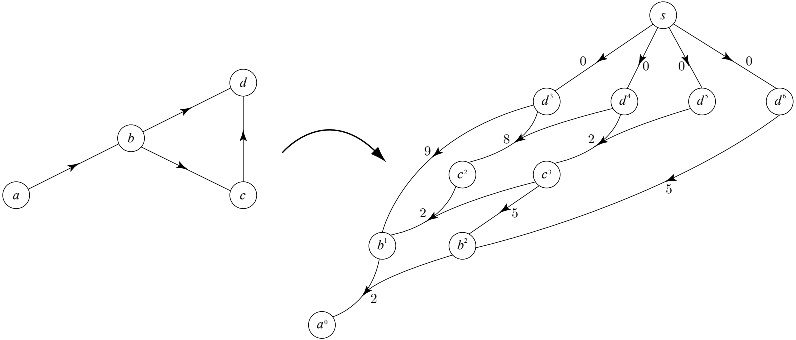
Known results

- Time-adaptive route choice – Finding the best strategy $\Leftrightarrow$ finding a minimum weight hyperpath.
- A priori route choice – Finding the shortest path is NP-hard.

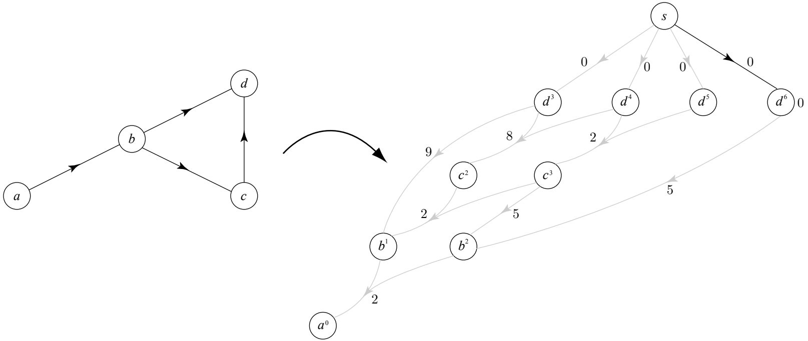
Example MEC (time-adaptive routing)



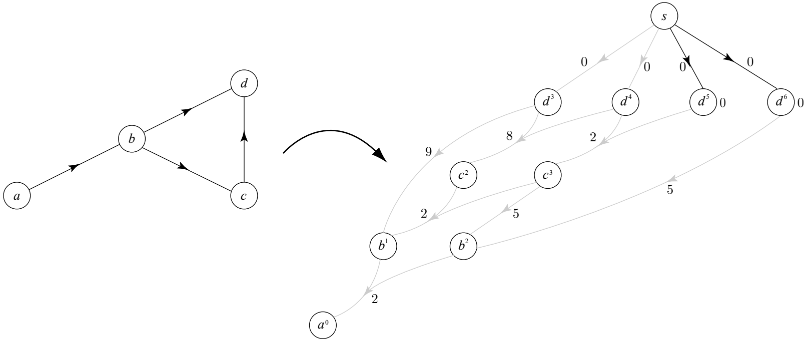
Example MEC (time-adaptive routing)



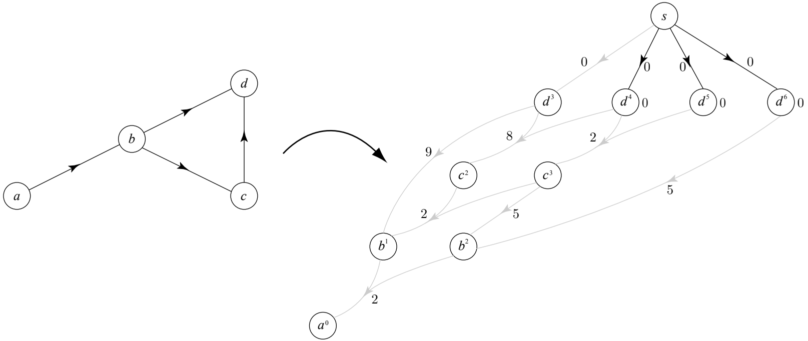
Example MEC (time-adaptive routing)



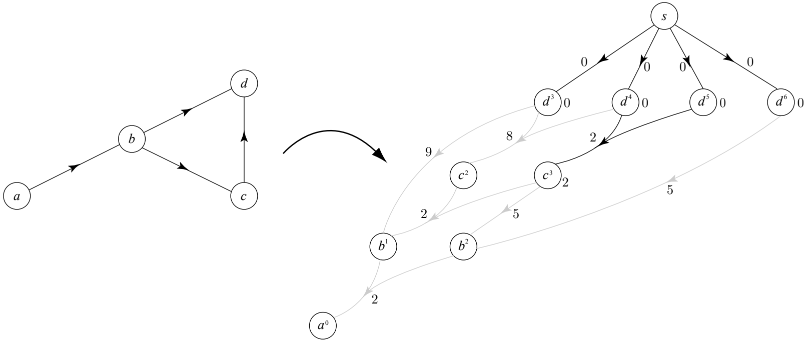
Example MEC (time-adaptive routing)



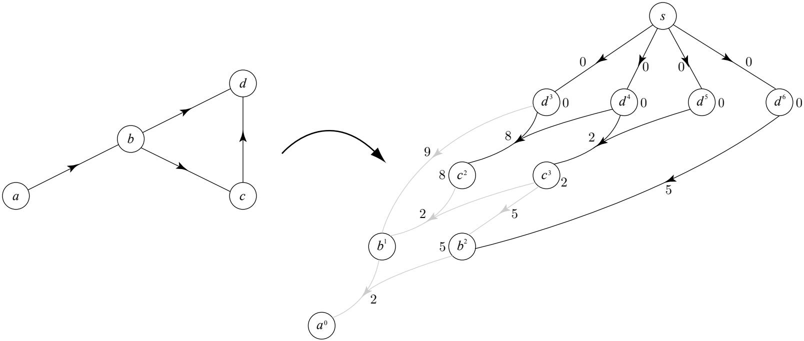
Example MEC (time-adaptive routing)



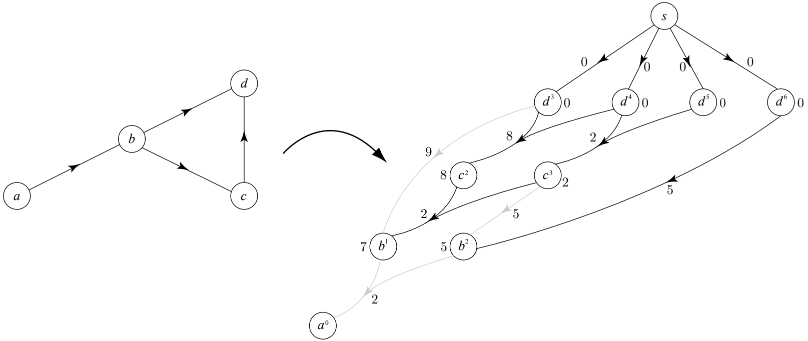
Example MEC (time-adaptive routing)



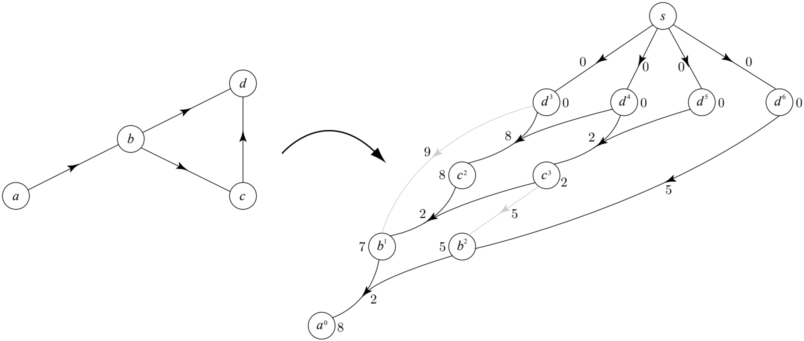
Example MEC (time-adaptive routing)



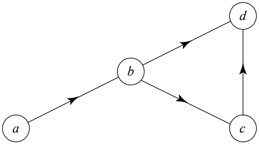
Example MEC (time-adaptive routing)

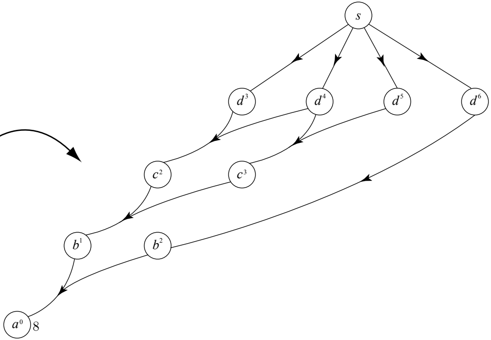


Example MEC (time-adaptive routing)



Example MEC (time-adaptive routing)





Agenda

Ranking scheme (deterministic case)

Let $\mathcal{P}$ denote the set of paths in G and p the shortest path.
Generic ranking scheme based on lower bounds:

1. Patition $\mathcal{P} \setminus \{p\}$ into q disjoint subsets $\mathcal{P}^1, \dots, \mathcal{P}^q$ based on p

Ranking scheme (deterministic case)

Let $\mathcal{P}$ denote the set of paths in G and p the shortest path.
Generic ranking scheme based on lower bounds:

1. Patition $\mathcal{P} \setminus \{p\}$ into q disjoint subsets $\mathcal{P}^1, \dots, \mathcal{P}^q$ based on p
2. For each set $\mathcal{P}^i$ find a lower bound w^i on the shortest path cost and add $(\mathcal{P}^i, w^i)$ to a sorted candidate set $\mathcal{C}$

Ranking scheme (deterministic case)

Let $\mathcal{P}$ denote the set of paths in G and p the shortest path.
Generic ranking scheme based on lower bounds:

1. Patition $\mathcal{P} \setminus \{p\}$ into q disjoint subsets $\mathcal{P}^1, \dots, \mathcal{P}^q$ based on p
2. For each set $\mathcal{P}^i$ find a lower bound w^i on the shortest path cost and add $(\mathcal{P}^i, w^i)$ to a sorted candidate set $\mathcal{C}$
3. Pick the best candidate $(\mathcal{P}^i, w^i)$ from $\mathcal{C}$ and find the shortest path $p^i \in \mathcal{P}^i$ with cost $\hat{w}^i$

Ranking scheme (deterministic case)

Let $\mathcal{P}$ denote the set of paths in G and p the shortest path.
Generic ranking scheme based on lower bounds:

1. Patition $\mathcal{P} \setminus \{p\}$ into q disjoint subsets $\mathcal{P}^1, \ldots, \mathcal{P}^q$ based on p
2. For each set $\mathcal{P}^i$ find a lower bound w^i on the shortest path cost and add $(\mathcal{P}^i, w^i)$ to a sorted candidate set $\mathcal{C}$
3. Pick the best candidate $(\mathcal{P}^i, w^i)$ from $\mathcal{C}$ and find the shortest path $p^i \in \mathcal{P}^i$ with cost $\hat{w}^i$
4. If $\exists k : w^k \leq \hat{w}^i, (\mathcal{P}^k, w^k) \in \mathcal{C}$ reinsert $(\mathcal{P}^i, \hat{w}^i)$ into $\mathcal{C}$ and goto step 3

Ranking scheme (deterministic case)

Let $\mathcal{P}$ denote the set of paths in G and p the shortest path.
Generic ranking scheme based on lower bounds:

1. Patition $\mathcal{P} \setminus \{p\}$ into q disjoint subsets $\mathcal{P}^1, \dots, \mathcal{P}^q$ based on p
2. For each set $\mathcal{P}^i$ find a lower bound w^i on the shortest path cost and add $(\mathcal{P}^i, w^i)$ to a sorted candidate set $\mathcal{C}$
3. Pick the best candidate $(\mathcal{P}^i, w^i)$ from $\mathcal{C}$ and find the shortest path $p^i \in \mathcal{P}^i$ with cost $\hat{w}^i$
4. If $\exists k : w^k \leq \hat{w}^i, (\mathcal{P}^k, w^k) \in \mathcal{C}$ reinsert $(\mathcal{P}^i, \hat{w}^i)$ into $\mathcal{C}$ and goto step 3
5. Output $(\hat{w}^i, p^i)$ and repeat using p^i and $\mathcal{P}^i$

Ranking paths in a STDN

The generic ranking scheme is hard to use in an STD network.

Ranking paths in a STDN

The generic ranking scheme is hard to use in an STD network.

Problem: finding the shortest path is *NP*-hard.

Ranking paths in a STDN

The generic ranking scheme is hard to use in an STD network.

Problem: finding the shortest path is *NP*-hard.

Idea: use time-adaptive routing as a relaxation of a priori routing.

- Finding the best routing strategy gives us a lower bound on the expected cost of the shortest path

Ranking paths in a STDN

The generic ranking scheme is hard to use in an STD network.

Problem: finding the shortest path is NP-hard.

Idea: use time-adaptive routing as a relaxation of a priori routing.

- Finding the best routing strategy gives us a lower bound on the expected cost of the shortest path
- The best routing strategy may be used to partition the solution space

Ranking scheme (STDN)

Let s denote the best strategy and p_s the corresponding path (`null` if s not is a path).

1. Patition $\mathcal{P} \setminus \{p_s\}$ into q disjoint subsets $\mathcal{P}^1, \ldots, \mathcal{P}^q$ based on a subpath of s

Ranking scheme (STDN)

Let s denote the best strategy and p_s the corresponding path (`null` if s not is a path).

1. Patition $\mathcal{P} \setminus \{p_s\}$ into q disjoint subsets $\mathcal{P}^1, \dots, \mathcal{P}^q$ based on a subpath of s
2. For each set $\mathcal{P}^i$ find a lower bound w^i on the shortest path cost and add $(\mathcal{P}^i, w^i)$ to a sorted candidate set $\mathcal{C}$

Ranking scheme (STDN)

Let s denote the best strategy and p_s the corresponding path (`null` if s not is a path).

1. Patition $\mathcal{P} \setminus \{p_s\}$ into q disjoint subsets $\mathcal{P}^1, \ldots, \mathcal{P}^q$ based on a subpath of s
2. For each set $\mathcal{P}^i$ find a lower bound w^i on the shortest path cost and add $(\mathcal{P}^i, w^i)$ to a sorted candidate set $\mathcal{C}$
3. Pick the best candidate $(\mathcal{P}^i, w^i)$ from $\mathcal{C}$ and find the best strategy $s^i \in \mathcal{P}^i$ with cost $\hat{w}^i$

Ranking scheme (STDN)

Let s denote the best strategy and p_s the corresponding path (`null` if s not is a path).

1. Patition $\mathcal{P} \setminus \{p_s\}$ into q disjoint subsets $\mathcal{P}^1, \dots, \mathcal{P}^q$ based on a subpath of s
2. For each set $\mathcal{P}^i$ find a lower bound w^i on the shortest path cost and add $(\mathcal{P}^i, w^i)$ to a sorted candidate set $\mathcal{C}$
3. Pick the best candidate $(\mathcal{P}^i, w^i)$ from $\mathcal{C}$ and find the best strategy $s^i \in \mathcal{P}^i$ with cost $\hat{w}^i$
4. If $\exists k : w^k \leq \hat{w}^i, (\mathcal{P}^k, w^k) \in \mathcal{C}$ reinsert $(\mathcal{P}^i, \hat{w}^i)$ into $\mathcal{C}$ and goto step 3

Ranking scheme (STDN)

Let s denote the best strategy and p_s the corresponding path (`null` if s not is a path).

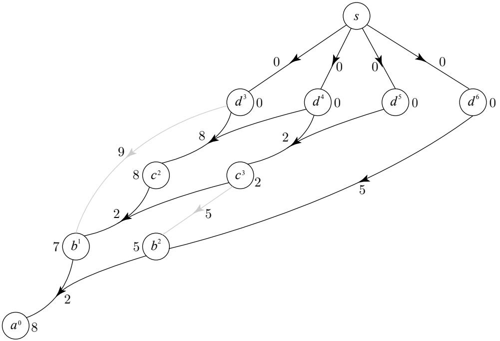
1. Patition $\mathcal{P} \setminus \{p_s\}$ into q disjoint subsets $\mathcal{P}^1, \ldots, \mathcal{P}^q$ based on a subpath of s
2. For each set $\mathcal{P}^i$ find a lower bound w^i on the shortest path cost and add $(\mathcal{P}^i, w^i)$ to a sorted candidate set $\mathcal{C}$
3. Pick the best candidate $(\mathcal{P}^i, w^i)$ from $\mathcal{C}$ and find the best strategy $s^i \in \mathcal{P}^i$ with cost $\hat{w}^i$
4. If $\exists k : w^k \leq \hat{w}^i, (\mathcal{P}^k, w^k) \in \mathcal{C}$ reinsert $(\mathcal{P}^i, \hat{w}^i)$ into $\mathcal{C}$ and goto step 3
5. If p_s^i is a path output $(\hat{w}^i, p_s^i)$

Ranking scheme (STDN)

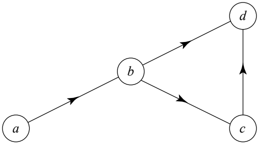
Let s denote the best strategy and p_s the corresponding path (`null` if s not is a path).

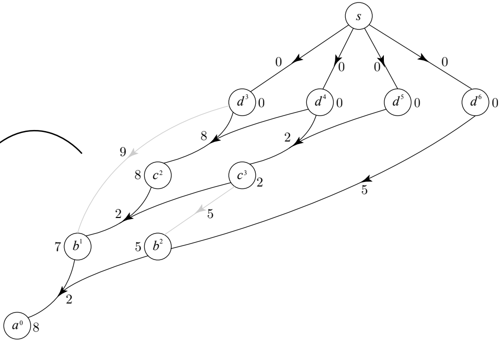
1. Patition $\mathcal{P} \setminus \{p_s\}$ into q disjoint subsets $\mathcal{P}^1, \ldots, \mathcal{P}^q$ based on a subpath of s
2. For each set $\mathcal{P}^i$ find a lower bound w^i on the shortest path cost and add $(\mathcal{P}^i, w^i)$ to a sorted candidate set $\mathcal{C}$
3. Pick the best candidate $(\mathcal{P}^i, w^i)$ from $\mathcal{C}$ and find the best strategy $s^i \in \mathcal{P}^i$ with cost $\hat{w}^i$
4. If $\exists k : w^k \leq \hat{w}^i, (\mathcal{P}^k, w^k) \in \mathcal{C}$ reinsert $(\mathcal{P}^i, \hat{w}^i)$ into $\mathcal{C}$ and goto step 3
5. If p_s^i is a path output $(\hat{w}^i, p_s^i)$
6. Repeat using s^i and $\mathcal{P}^i$

Optimal routing strategy

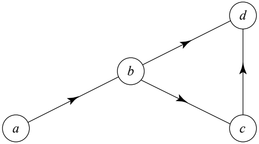


Optimal routing strategy

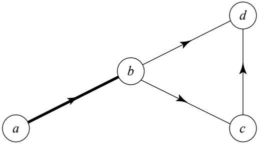




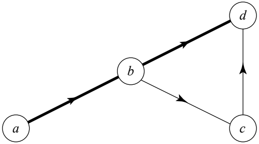
Choosing subpath



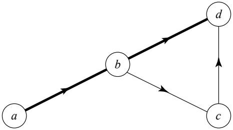
Choosing subpath



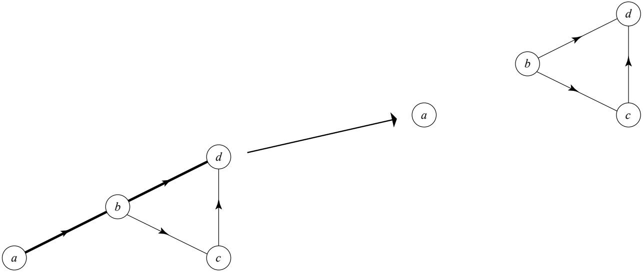
Choosing subpath



Disjoint subsets (subgraphs)

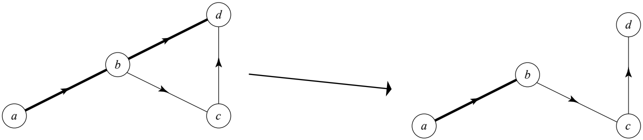


Disjoint subsets (subgraphs)



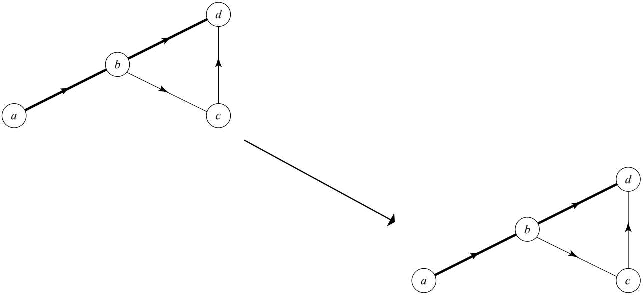
$\mathcal{P}^1$

Disjoint subsets (subgraphs)



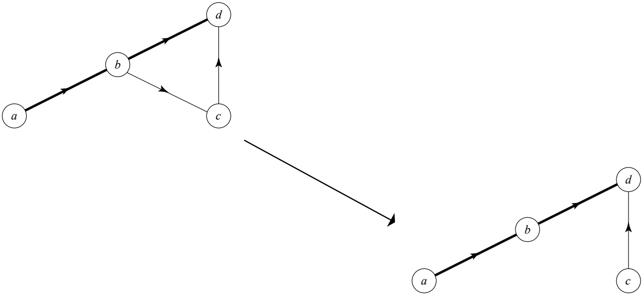
$\mathcal{P}^2$

Disjoint subsets (subgraphs)



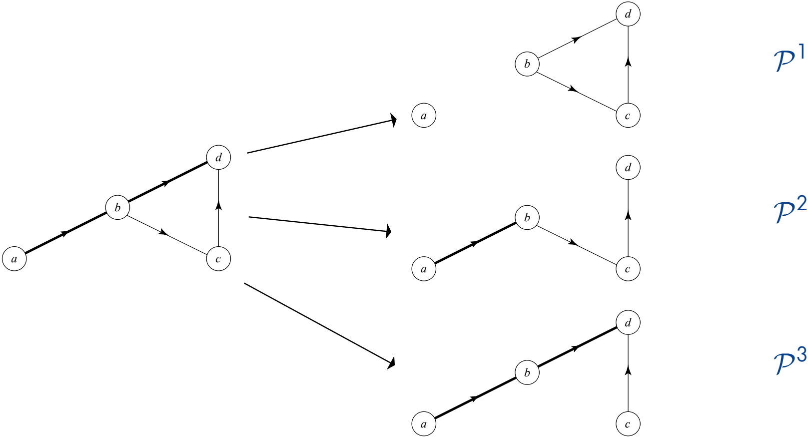
$\mathcal{P}^3$

Disjoint subsets (subgraphs)

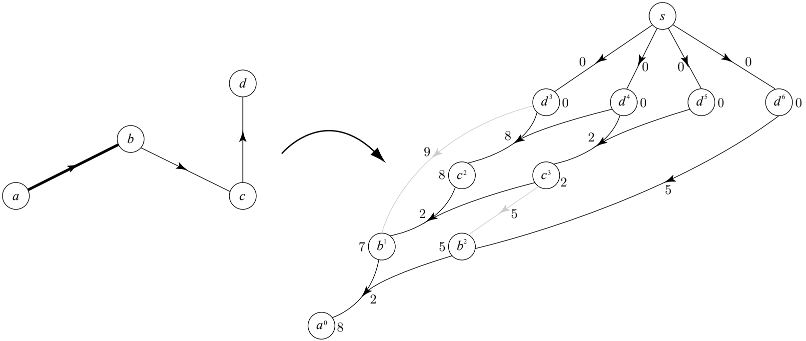


$\mathcal{P}^3$

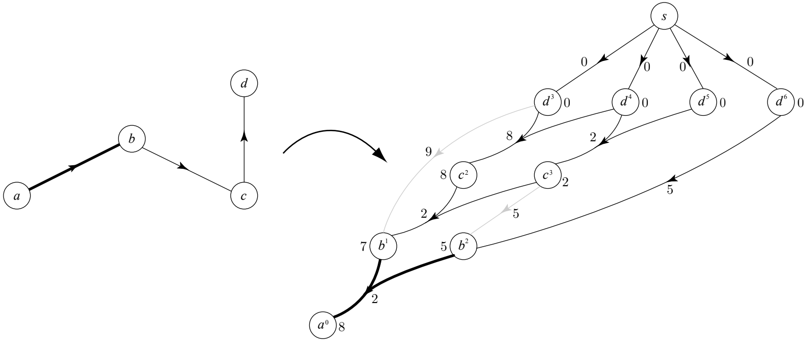
Disjoint subsets (subgraphs)



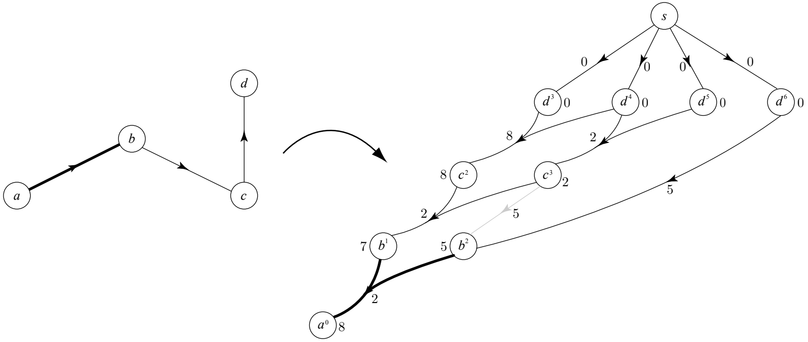
Calculate lower bounds $\mathcal{P}^2$



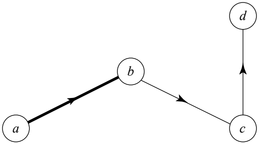
Calculate lower bounds $\mathcal{P}^2$

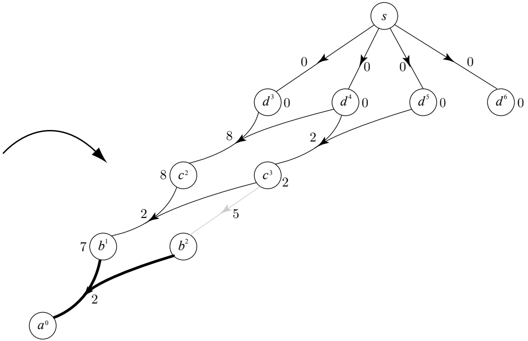


Calculate lower bounds $\mathcal{P}^2$

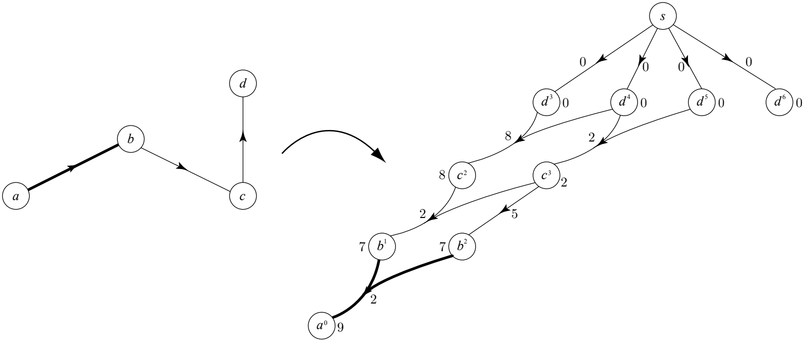


Calculate lower bounds $\mathcal{P}^2$

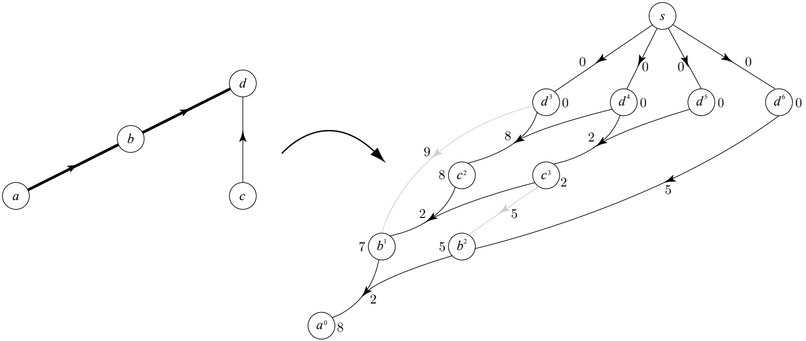




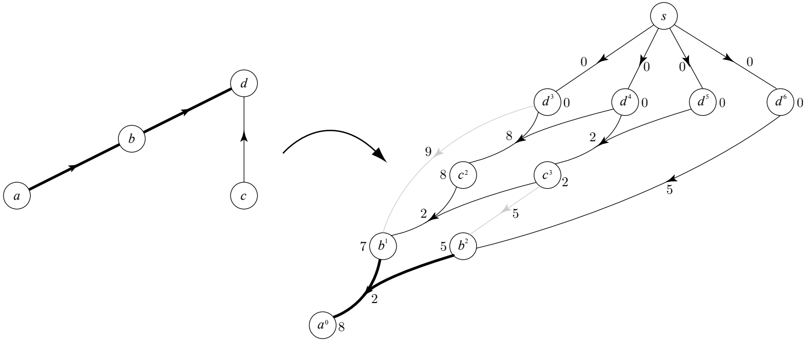
Calculate lower bounds $\mathcal{P}^2$



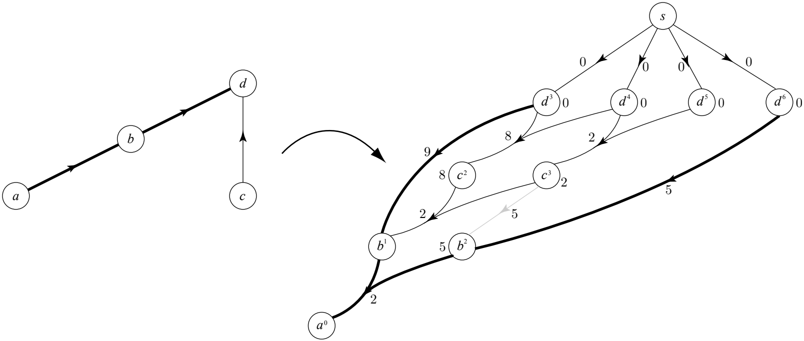
Calculate lower bounds $\mathcal{P}^3$



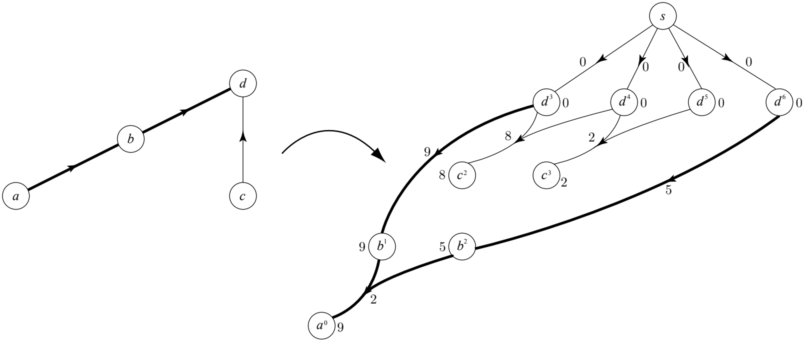
Calculate lower bounds $\mathcal{P}^3$



Calculate lower bounds $\mathcal{P}^3$



Calculate lower bounds $\mathcal{P}^3$



Agenda

Generator

- TEGP generator used (`www.research.relund.dk`).

Generator

- TEGP generator used (`www.research.relund.dk`).
- Cyclic time periods (e.g. a day)

Generator

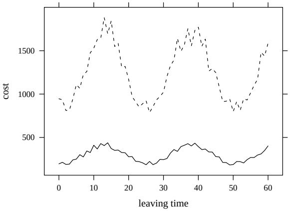
- TEGP generator used (`www.research.relund.dk`).
- Cyclic time periods (e.g. a day)
- In each cyclic period there are some peak periods.

Generator

- TEGP generator used (`www.research.relund.dk`).
- Cyclic time periods (e.g. a day)
- In each cyclic period there are some peak periods.
- Time unit 5 min, off-peak mean travel time [2, 6] (100% increase in peaks).

Generator

- TEGP generator used (`www.research.relund.dk`).
- Cyclic time periods (e.g. a day)
- In each cyclic period there are some peak periods.
- Time unit 5 min, off-peak mean travel time [2, 6] (100% increase in peaks).
- Off-peak costs [1, 1000] (100% increase in peaks) + random noise (10%).



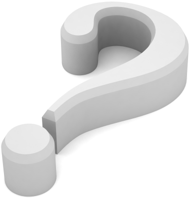
Test instances

	peak dependent costs			random costs		
Class	1	2	3	4	5	6
Grid size	5×10	10×10	20×10	5×10	10×10	20×10
n	2320	7573	21454	1497	3961	11856
m	7809	27278	79570	5056	14295	43991
H	118	156	237	75	101	155

Results ($K = 100$)

Class	ite_K	CPU_K	ite_1	CPU_1	$\delta(u) = 1$	$\delta(u) = 2$	$\delta(u) = 3$	$\delta(u) = 4$	inc	$inc_{S\text{-}PS}$
1	356	8	13	0.3	82	14	4	1	49	5
2	454	37	11	1.0	82	14	4	0	17	7
3	2359	875	133	53.5	85	13	2	0	10	5
4	1427	25	133	2.5	50	25	20	5	18	43
5	9942	564	2722	157.9	39	21	27	12	8	54
6	203479	34227	89519	15127.5	25	16	27	31	3	59

Questions



Contact: lars@relund.dk,
http://www.research.relund.dk/