On The (Robust) Shortest Path Problem

Based on: On The Shortest Path Problem
by Gang Yu and Jian Yang

Seminar Robust Optimization/Algorithms for Optimization under Uncertainty

Sommersemester 2024 *Anne Schwarz*

Kürzestester Weg zum Verständnis

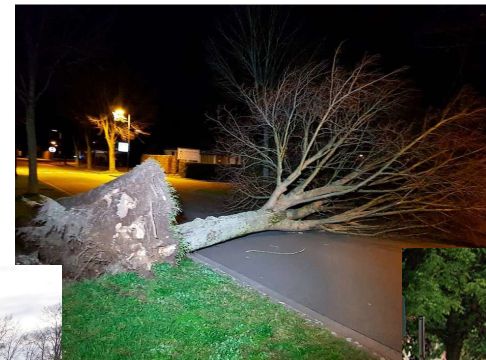
1. Modellierung Unsicherheit
    - Szenarien
2. Qualitätsmaß eines Weges
    - absolut robust und robust deviation shortest path
3. Komplexität und Lösungen
    - layered Graphen und allgemeine Graphen
    - feste und nicht feste Anzahl an Szenarien

Modellierung Unsicherheit: Idee

Makel der echten Welt

Modellierung Unsicherheit: Idee

Makel der echten Welt



Modellierung Unsicherheit: Idee

Makel der echten Welt

$\Longrightarrow$ Modellierung als Szenarios

- ein Szenario stellt eine mögliche Situation dar
- jedes Szenario tritt mit einer positiven Wahrscheinlichkeit auf
  (wir müssen diese aber nicht kennen)

Modellierung Unsicherheit: Formalisiert

Sei $G = (V, E)$ ein ungerichteter Graph.

Modellierung Unsicherheit: Formalisiert

Sei $G = (V, E)$ ein ungerichteter Graph.

Definition (Szenario)

Ein Szenario für G ist eine Funktion $r : E \to \mathbb{N}_0$, die jeder Kante des Graphen ein nichtnegatives Gewicht zuordnet
Wir schreiben auch c_e^r für $r(e), \forall e \in E$

Modellierung Unsicherheit: Formalisiert

Sei $G = (V, E)$ ein ungerichteter Graph.

Definition (Szenario)

Ein Szenario für G ist eine Funktion $r : E \to \mathbb{N}_0$, die jeder Kante des Graphen ein nichtnegatives Gewicht zuordnet
Wir schreiben auch c_e^r für $r(e), \forall e \in E$

Sei $G = (V, E)$ ein ungerichteter Graph, dann ist G unter Szenario r der ungerichtete, gewichtete Graph G mit Gewichtsfunktion r

Modellierung Unsicherheit: Formalisiert

Sei $G = (V, E)$ ein ungerichteter Graph.

Definition (Szenario)

Ein Szenario für G ist eine Funktion $r : E \to \mathbb{N}_0$, die jeder Kante des Graphen ein nichtnegatives Gewicht zuordnet
Wir schreiben auch c_e^r für $r(e)$, $\forall e \in E$

Sei $G = (V, E)$ ein ungerichteter Graph, dann ist G unter Szenario r der ungerichtete, gewichtete Graph G mit Gewichtsfunktion r

Definition (Unsicherheitsmenge)

Die Unsicherheitsmenge einer Kante $e \in E$ unter S (Menge an Szenarien) bezeichnen wir mit U_e, wobei $U_e := \{r(e) \mid r \in S\}$

Robuster kürzester Weg: Definition

Sei S eine Menge an Szenarien,

$G = (V, E)$ ein ungerichteter Graph mit $s, t \in V$

Idee: minmax

Robuster kürzester Weg: Definition

Sei S eine Menge an Szenarien,

$G = (V, E)$ ein ungerichteter Graph mit $s, t \in V$

Idee: minmax

Minimiere den maximalsten Fall?

Wähle die Lösungen, die im schlimmsten Fall, bestmöglich sind?

Robuster kürzester Weg: Definition

Sei S eine Menge an Szenarien,

$G = (V, E)$ ein ungerichteter Graph mit $s, t \in V$

Absolute Robust Shortest Path (ARSP)

- wähle s-t Pfad mit besten Wost-Case Kosten über alle Szenarien

Robuster kürzester Weg: Definition

Sei S eine Menge an Szenarien,

$G = (V, E)$ ein ungerichteter Graph mit $s, t \in V$

Absolute Robust Shortest Path (ARSP)

- wähle s-t Pfad mit besten Wost-Case Kosten über alle Szenarien

Robust Deviation Shortest Path (RDSP)

- wähle s-t Pfad mit kleinster Worst-Case Kostenabweichung von minimalen Kosten, über alle Szenarien

Robuster kürzester Weg: Definition

Sei S eine Menge an Szenarien,

$G = (V, E)$ ein ungerichteter Graph mit $s, t \in V$

Absolute Robust Shortest Path (ARSP)

- wähle s-t Pfad mit besten Wost-Case Kosten über alle Szenarien

Robust Deviation Shortest Path (RDSP)

- wähle s-t Pfad mit kleinster Worst-Case Kostenabweichung von minimalen Kosten, über alle Szenarien

Beobachtung. Für ARSP/RDSP muss nicht Wahrscheinlichkeitsverteilung der Szenarien betrachtet werden

Robuster kürzester Weg: Definition

$$(SP)_r \;\; z^r = \min \textstyle\sum_{(i,j)\in E} c_{ij}^r x_{ij}$$

subject to

$$\sum_{j\in V} x_{ij} - \sum_{k\in V} x_{ki} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$x_{ij} \in \{0, 1\} \;\; \forall \{i, j\} \in E \tag{2}$$

Robuster kürzester Weg: Definition

$$(SP)_r \; z^r = \min \textstyle\sum_{(i,j)\in E} c_{ij}^r x_{ij}$$

subject to

$$\sum_{j\in V} x_{ij} - \sum_{k\in V} x_{ki} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$x_{ij} \in \{0,1\} \;\; \forall \{i,j\} \in E \tag{2}$$

Absolute Robust Shortest Path (ARSP)

Ges. $z_A =$
$\min_x \max_{r\in S} \left\{ \sum_{(i,j)\in A} c_{ij}^r x_{ij} \right\}$

subject to (1) and (2)

Robuster kürzester Weg: Definition

$$(SP)_r \; z^r = \min \textstyle\sum_{(i,j)\in E} c_{ij}^r x_{ij}$$

subject to

$$\sum_{j\in V} x_{ij} - \sum_{k\in V} x_{ki} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

$$x_{ij} \in \{0,1\} \;\; \forall \{i,j\} \in E \qquad (2)$$

Absolute Robust Shortest Path (ARSP)

Ges. $z_A =$
$\min_x \max_{r\in S} \left\{ \sum_{(i,j)\in A} c_{ij}^r x_{ij} \right\}$

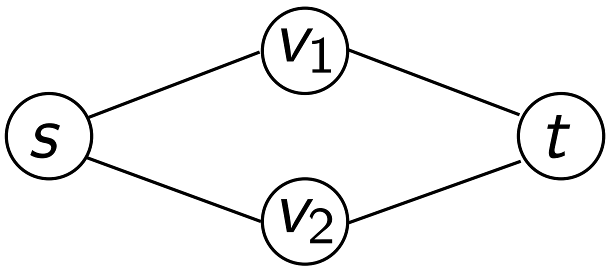
subject to (1) and (2)

Robust Deviation Shortest Path (RDSP)

Ges. $z_R =$
$\min_x \max_{r\in S} \left\{ \sum_{(i,j)\in A} c_{ij}^r x_{ij} - z^r \right\}$

subject to (1) and (2)

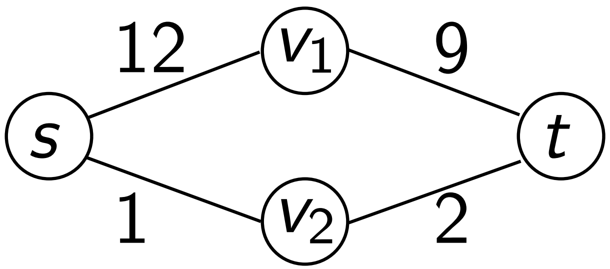
Robuster kürzester Weg: Beispiel



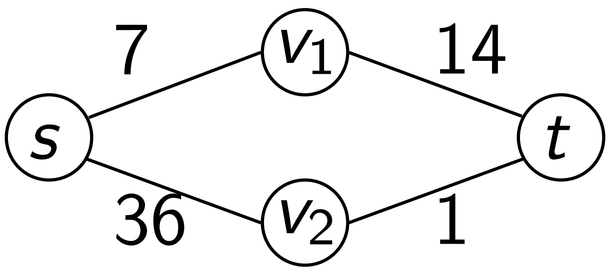
Robuster kürzester Weg: Beispiel

Sei $S = \{s_1, s_2\}$ ein Menge an Szenarien, wobei s_1, s_2 durch die Kantengewichte der folgenden Graphen beschrieben werden.

Szenario s_1:



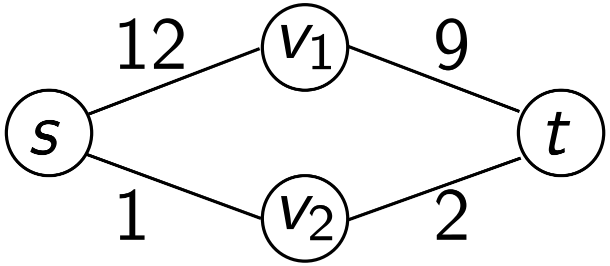
Szenario s_2:



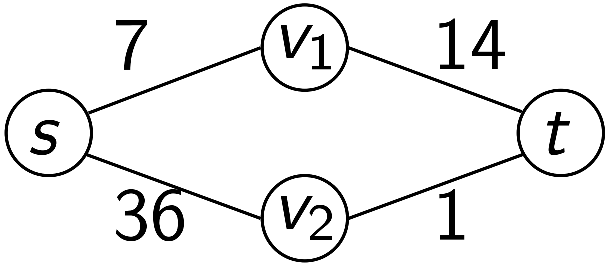
Robuster kürzester Weg: Beispiel

Sei $S = \{s_1, s_2\}$ ein Menge an Szenarien, wobei s_1, s_2 durch die Kantengewichte der folgenden Graphen beschrieben werden.

Szenario s_1:



Szenario s_2:



ARSP von G unter S?

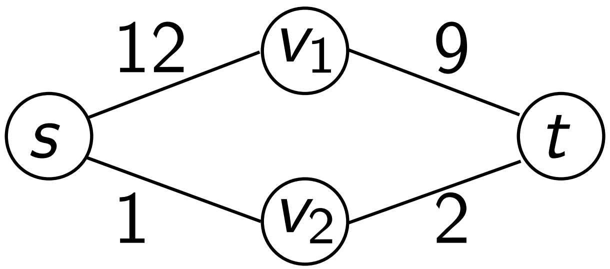
Erin.: (minimaler Worst-Case Pfad)

Ges. $z_A = \min_x \max_{r \in S} \left\{ \sum_{(i,j)\in A} c_{ij}^r x_{ij} \right\}$

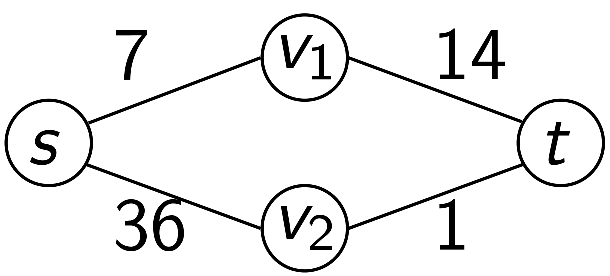
Robuster kürzester Weg: Beispiel

Sei $S = \{s_1, s_2\}$ ein Menge an Szenarien, wobei s_1, s_2 durch die Kantengewichte der folgenden Graphen beschrieben werden.

Szenario s_1:



Szenario s_2:



ARSP von G unter S?

Erin.: (minimaler Worst-Case Pfad)

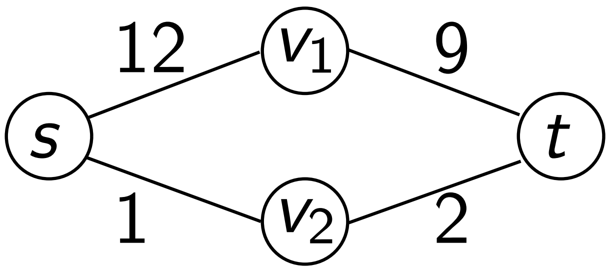
Ges. $z_A = \min_x \max_{r \in S} \left\{ \sum_{(i,j) \in A} c_{ij}^r x_{ij} \right\}$

$z_A = 21$

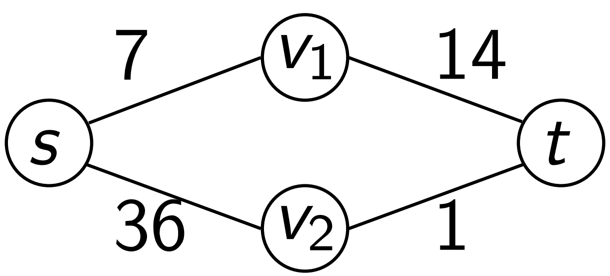
Robuster kürzester Weg: Beispiel

Sei $S = \{s_1, s_2\}$ ein Menge an Szenarien, wobei s_1, s_2 durch die Kantengewichte der folgenden Graphen beschrieben werden.

Szenario s_1:



Szenario s_2:



ARSP von G unter S?

Erin.: (minimaler Worst-Case Pfad)

Ges. $z_A = \min_x \max_{r \in S} \left\{ \sum_{(i,j) \in A} c_{ij}^r x_{ij} \right\}$

$$z_A = 21$$

RDSP von G unter S?

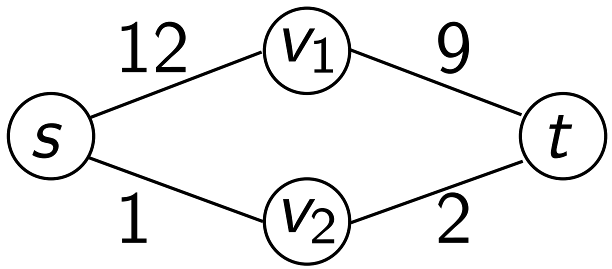
Erin.: (minimale Worst-Case Abweichung)

Ges. $z_R = \min_x \max_{r \in S} \left\{ \sum_{(i,j) \in A} c_{ij}^r x_{ij} - z^r \right\}$

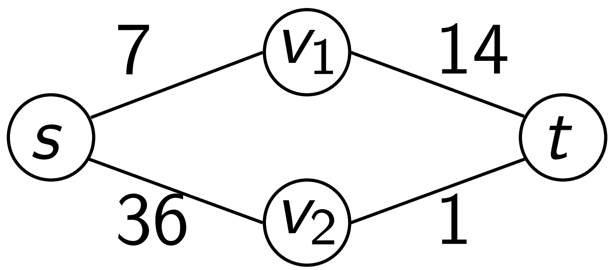
Robuster kürzester Weg: Beispiel

Sei $S = \{s_1, s_2\}$ ein Menge an Szenarien, wobei s_1, s_2 durch die Kantengewichte der folgenden Graphen beschrieben werden.

Szenario s_1:



Szenario s_2:



ARSP von G unter S?

Erin.: (minimaler Worst-Case Pfad)

Ges. $z_A = \min_x \max_{r \in S} \left\{ \sum_{(i,j) \in A} c_{ij}^r x_{ij} \right\}$

$$z_A = 21$$

RDSP von G unter S?

Erin.: (minimale Worst-Case Abweichung)

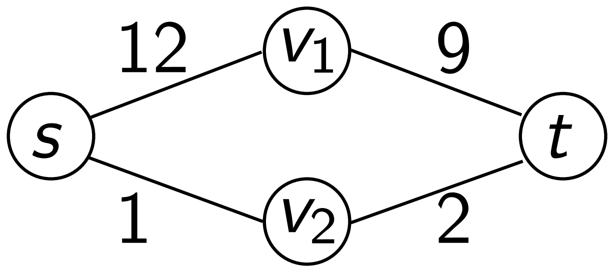
Ges. $z_R = \min_x \max_{r \in S} \left\{ \sum_{(i,j) \in A} c_{ij}^r x_{ij} - z^r \right\}$

$$z_R = 16$$

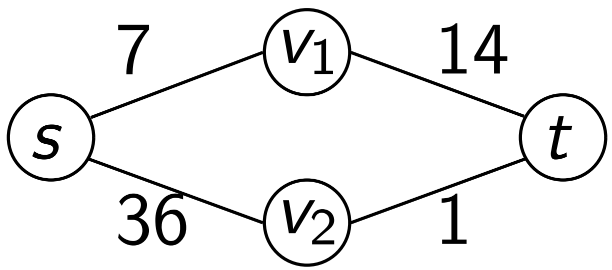
Robuster kürzester Weg: Beispiel

Sei $S = \{s_1, s_2\}$ ein Menge an Szenarien, wobei s_1, s_2 durch die Kantengewichte der folgenden Graphen beschrieben werden.

Szenario s_1:



Szenario s_2:



ARSP von G unter S?

Erin.: (minimaler Worst-Case Pfad)

Ges. $z_A = \min_x \max_{r \in S} \left\{ \sum_{(i,j) \in A} c_{ij}^r x_{ij} \right\}$

$$z_A = 21$$

RDSP von G unter S?

Erin.: (minimale Worst-Case Abweichung)

Ges. $z_R = \min_x \max_{r \in S} \left\{ \sum_{(i,j) \in A} c_{ij}^r x_{ij} - z^r \right\}$

$$z_R = 16$$

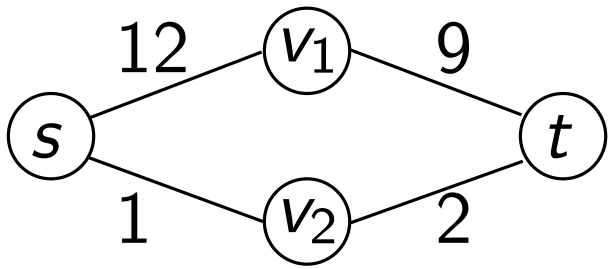
Welchen kürzesten Weg nehme ich?

Meine Oma feiert um 15:00 Geburtstag, es ist aber schon 14:39.
Wenn ich zu spät bin, bekomme ich kein Kuchen mehr :(

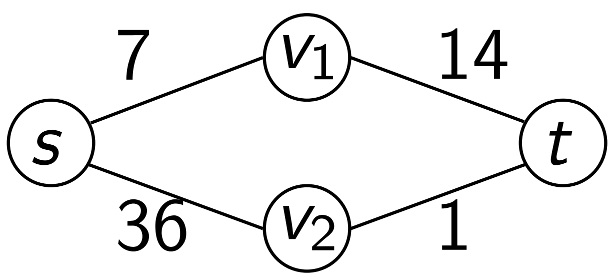
Robuster kürzester Weg: Beispiel

Sei $S = \{s_1, s_2\}$ ein Menge an Szenarien, wobei s_1, s_2 durch die Kantengewichte der folgenden Graphen beschrieben werden.

Szenario s_1:



Szenario s_2:



ARSP von G unter S?

Erin.: (minimaler Worst-Case Pfad)

Ges. $z_A = \min_x \max_{r \in S} \left\{ \sum_{(i,j) \in A} c_{ij}^r x_{ij} \right\}$

$$z_A = 21$$

RDSP von G unter S?

Erin.: (minimale Worst-Case Abweichung)

Ges. $z_R = \min_x \max_{r \in S} \left\{ \sum_{(i,j) \in A} c_{ij}^r x_{ij} - z^r \right\}$

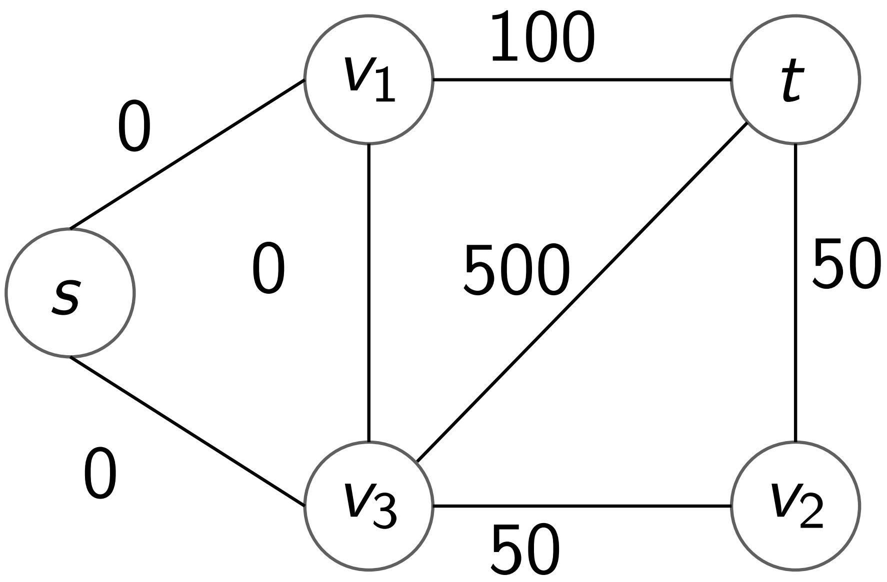
$$z_R = 16$$

Welchen kürzesten Weg nehme ich?

Ich möchte gerne möglichst zügig in dem Urlaub fahren.

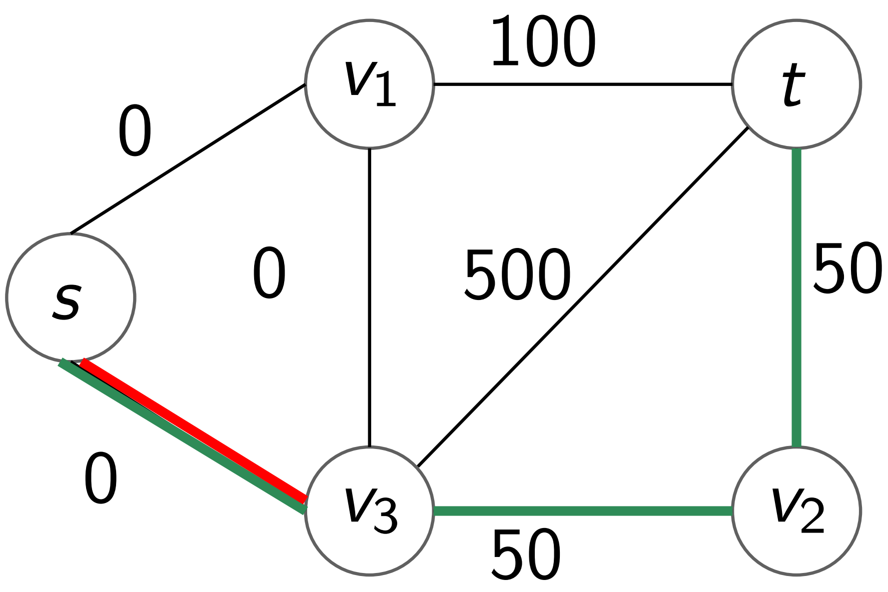
Versagen des Determinismus

ARSP und RDSP für Graphen mit nur einem Szenario?



Versagen des Determinismus

ARSP und RDSP für Graphen mit nur einem Szenario?



Dijkstra

Sei π ein kürzester s-t Weg, welcher v_3 enthält.

Dann ist π_{sv_3} kürzester $s - v_3$ Weg

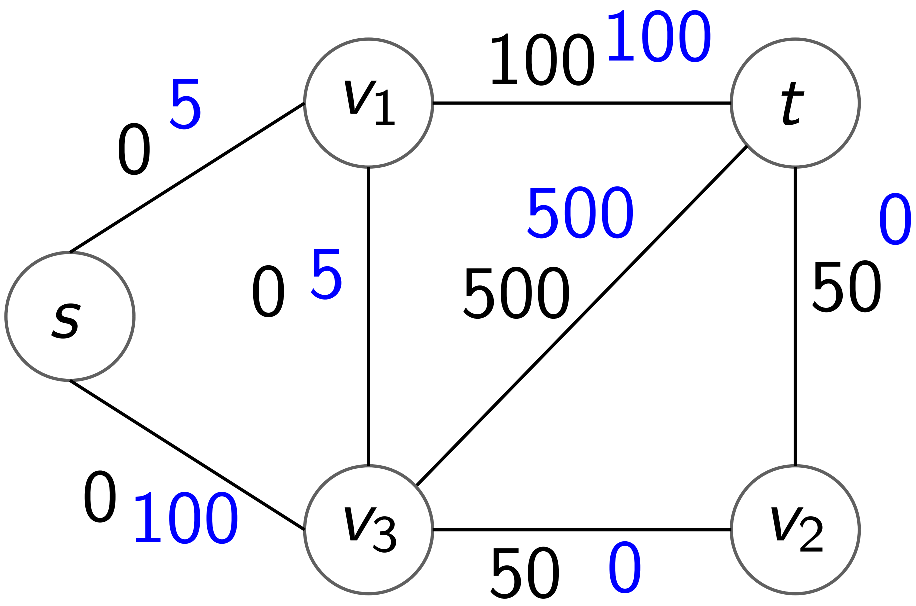
und $\pi_{v_3 t}$ kürzester $v_3 - t$ Weg

Versagen des Determinismus

ARSP und RDSP für Graphen mit nur einem Szenario?

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



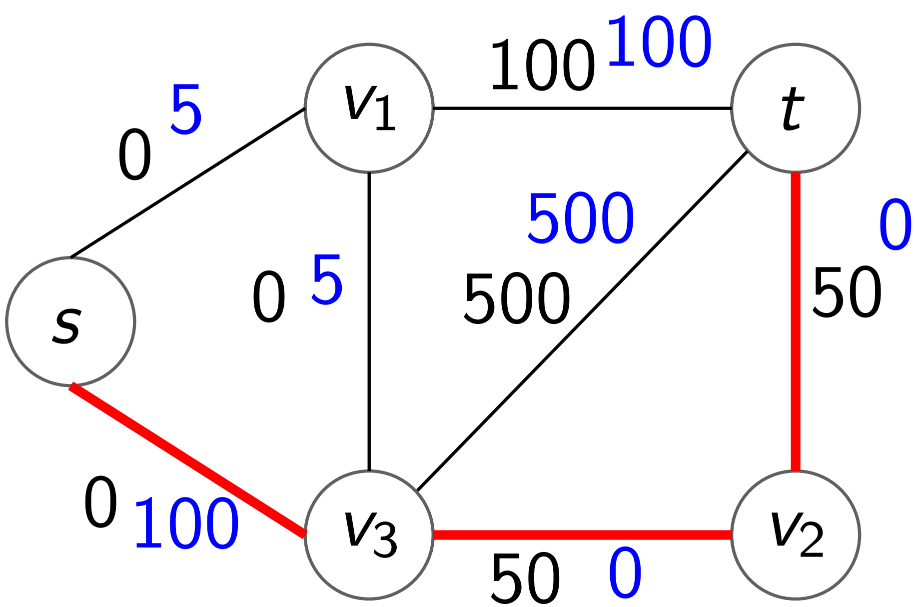
Dijkstra ?

Sei π ein absolut robuster kürzester $s-t$ Weg, welcher v_3 enthält.
Dann ist π_{sv_3} absolut robuster kürzester $s-v_3$ Weg

und π_{v_3t} absolut robuster kürzester v_3-t Weg?

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



absolut robuster kürzester s-t Weg

Dijkstra ?

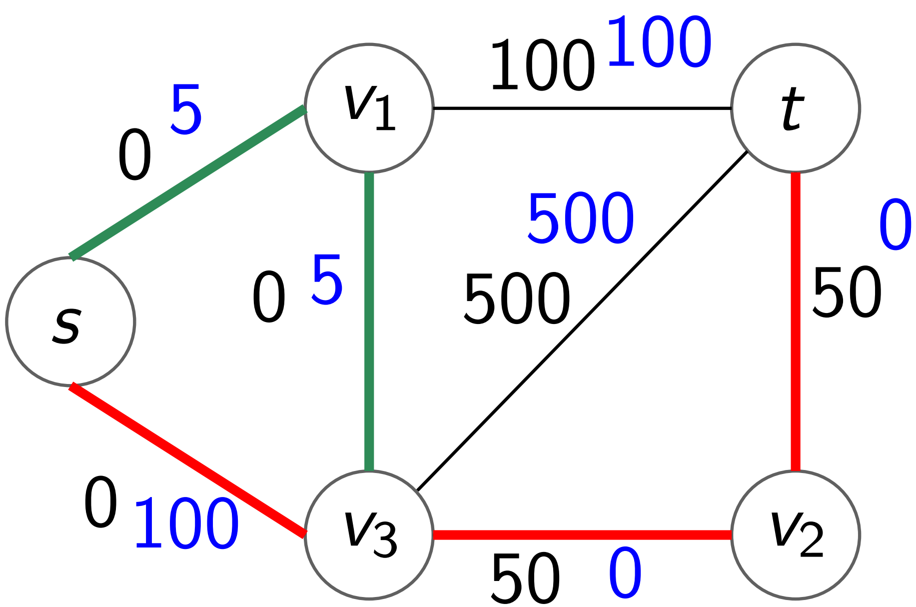
Sei π ein absolut robuster kürzester $s - t$ Weg, welcher v_3 enthält.

Dann ist π_{sv_3} absolut robuster kürzester $s - v_3$ Weg

und $\pi_{v_3 t}$ absolut robuster kürzester $v_3 - t$ Weg?

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



absolut robuster kürzester s-t Weg

absolut robuster kürzester s-v_3 Weg

Dijkstra ?

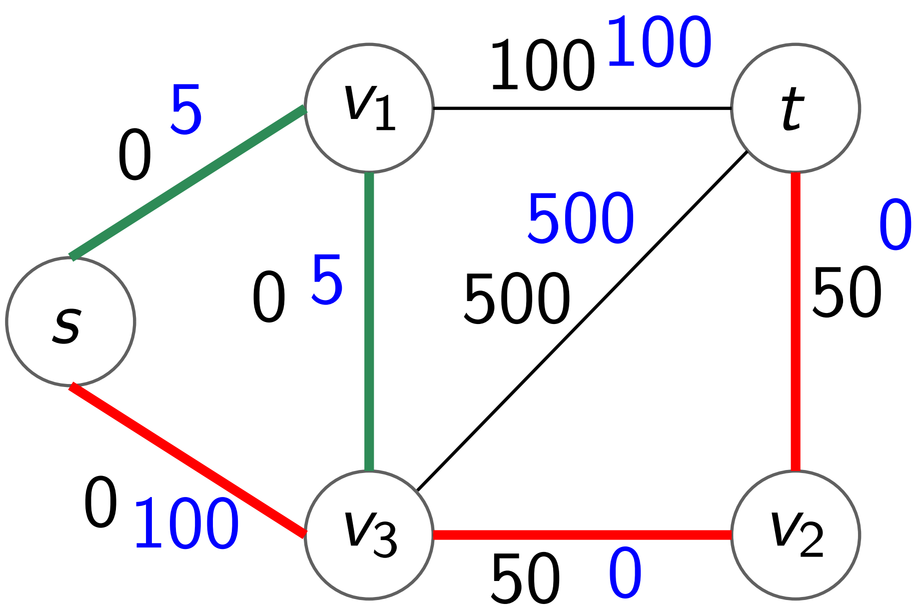
Sei π ein absolut robuster kürzester $s - t$ Weg, welcher v_3 enthält.

Dann ist π_{sv_3} absolut robuster kürzester $s - v_3$ Weg

und $\pi_{v_3 t}$ absolut robuster kürzester $v_3 - t$ Weg?

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



absolut robuster kürzester s-t Weg

absolut robuster kürzester s-v_3 Weg

Dijkstra ?

Sei π ein absolut robuster kürzester $s - t$ Weg, welcher v_3 enthält.

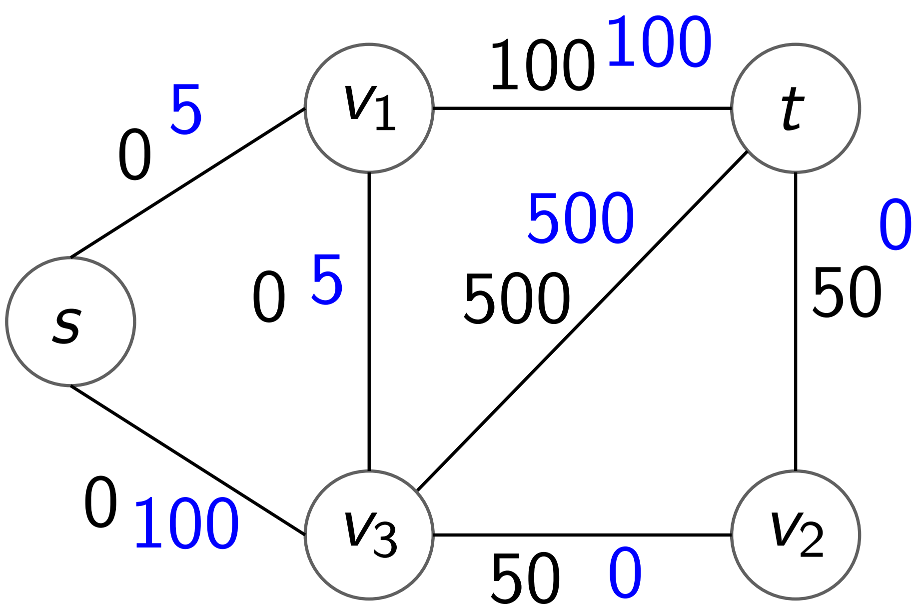
NEIN

Dann ist π_{sv_3} absolut robuster kürzester $s - v_3$ Weg

und $\pi_{v_3 t}$ absolut robuster kürzester $v_3 - t$ Weg?

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



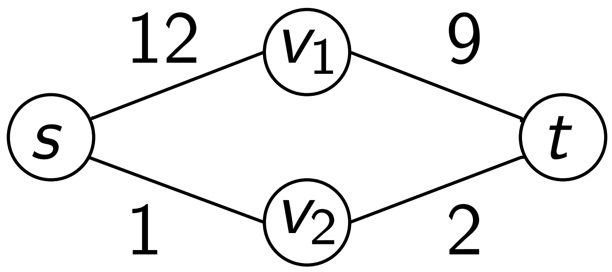
~~Dijkstra~~

Problem

- Dijkstra liefert kürzesten Pfad, keine Aussage über Worst-Case Laufzeit jedes Pfades
- Greedy Ansatz funktioniert nicht
- naiver Brute-Force Algorithmus nicht in polynomialzeit berechenbar

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



Für einfachere Graphen ...

Sei $G = (V, E)$ ein ungerichteter Graph.

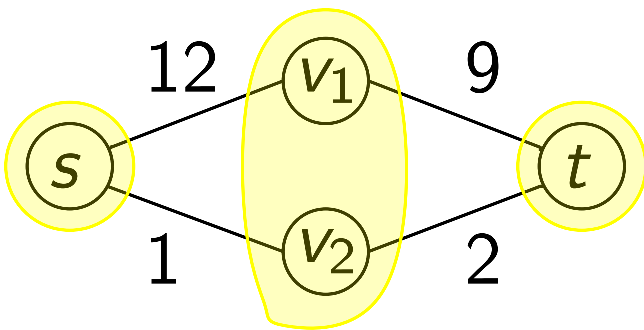
Definition(layered Graph)

G ist ein layered Graph, falls folgendes gilt

- V kann in disjunkte Menge $V = \{s\} \cup V_1 \cup V_2 \cup \cdots \cup V_m \cup \{t\}$ zerlegt werden
- $\forall \{i, j\} \in E : (i \in V_k \wedge j \in V_{k+1}$ für ein $k \in \{1, \ldots, m-1\}) \vee (i = s \wedge j \in V_1) \vee (i = t \wedge j \in V_m)$

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



Für einfachere Graphen ...

Sei $G = (V, E)$ ein ungerichteter Graph.

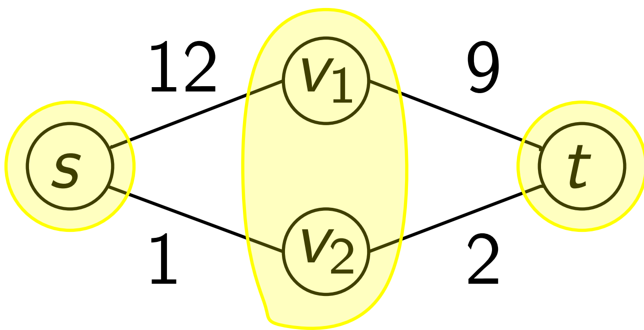
Definition(layered Graph)

G ist ein layered Graph, falls folgendes gilt

- V kann in disjunkte Menge $V = \{s\} \cup V_1 \cup V_2 \cup \cdots \cup V_m \cup \{t\}$ zerlegt werden
- $\forall \{i, j\} \in E : (i \in V_k \wedge j \in V_{k+1}$ für ein $k \in \{1, \ldots, m-1\}) \vee (i = s \wedge j \in V_1) \vee (i = t \wedge j \in V_m)$

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



Für einfachere Graphen ...

Sei $G = (V, E)$ ein ungerichteter Graph.

Definition(layered Graph)

G ist ein layered Graph, falls folgendes gilt

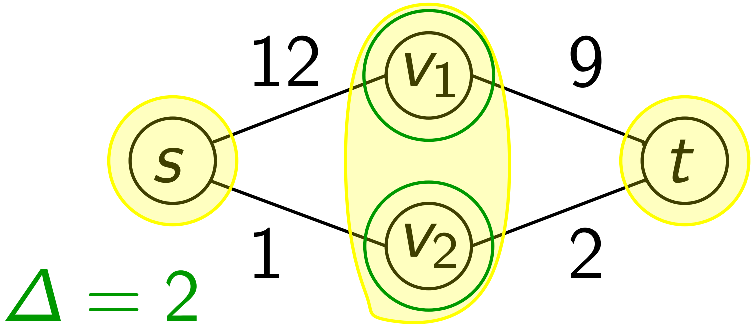
- V kann in disjunkte Menge $V = \{s\} \cup V_1 \cup V_2 \cup \cdots \cup V_m \cup \{t\}$ zerlegt werden
- $\forall \{i, j\} \in E : (i \in V_k \wedge j \in V_{k+1}$ für ein $k \in \{1, \ldots, m-1\}) \vee (i = s \wedge j \in V_1) \vee (i = t \wedge j \in V_m)$

Definition(Breite)

Ist G ein layered Graph, so bezeichnen wir mit Breite folgendes
$\Delta = \max\{|V_k| \mid k = 1, \ldots, m\}$

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



Definition(Breite)

Ist G ein layered Graph, so bezeichnen wir mit Breite folgendes
$\Delta = \max\{|V_k| \mid k = 1, \ldots, m\}$

Für einfachere Graphen ...

Sei $G = (V, E)$ ein ungerichteter Graph.

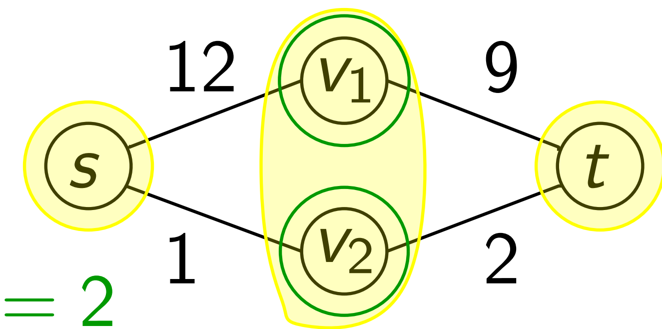
Definition(layered Graph)

G ist ein layered Graph, falls folgendes gilt

- V kann in disjunkte Menge $V = \{s\} \cup V_1 \cup V_2 \cup \cdots \cup V_m \cup \{t\}$ zerlegt werden
- $\forall \{i, j\} \in E : (i \in V_k \wedge j \in V_{k+1}$ für ein $k \in \{1, \ldots, m-1\}) \vee (i = s \wedge j \in V_1) \vee (i = t \wedge j \in V_m)$

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



$\Delta = 2$

Definition(Breite)

Ist G ein layered Graph, so bezeichnen wir mit Breite folgendes
$\Delta = \max\{|V_k| \mid k = 1, \ldots, m\}$

Für einfachere Graphen ...

Sei $G = (V, E)$ ein ungerichteter Graph.

Definition(layered Graph)

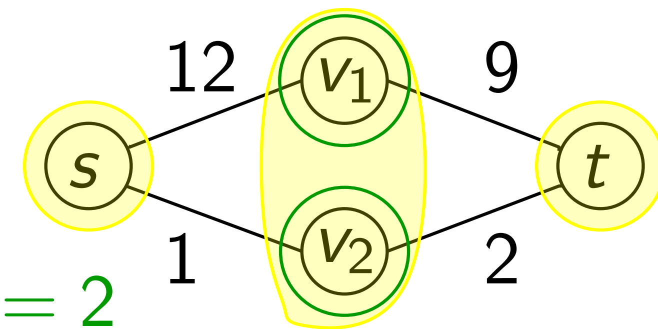
G ist ein layered Graph, falls folgendes gilt

- V kann in disjunkte Menge $V = \{s\} \cup V_1 \cup V_2 \cup \cdots \cup V_m \cup \{t\}$ zerlegt werden
- $\forall \{i, j\} \in E : (i \in V_k \wedge j \in V_{k+1}$ für ein $k \in \{1, \ldots, m-1\}) \vee (i = s \wedge j \in V_1) \vee (i = t \wedge j \in V_m)$

ARSP und RDSP für layered Graphen mit $\Delta < 1$?

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



$\Delta = 2$

Definition(Breite)

Ist G ein layered Graph, so bezeichnen wir mit Breite folgendes
$\Delta = \max\{|V_k| \mid k = 1, \ldots, m\}$

Für einfachere Graphen ...

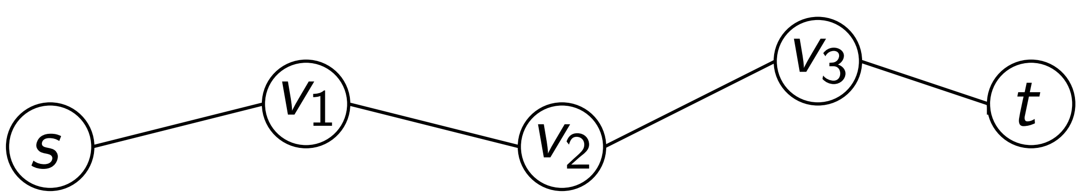
Sei $G = (V, E)$ ein ungerichteter Graph.

Definition(layered Graph)

G ist ein layered Graph, falls folgendes gilt

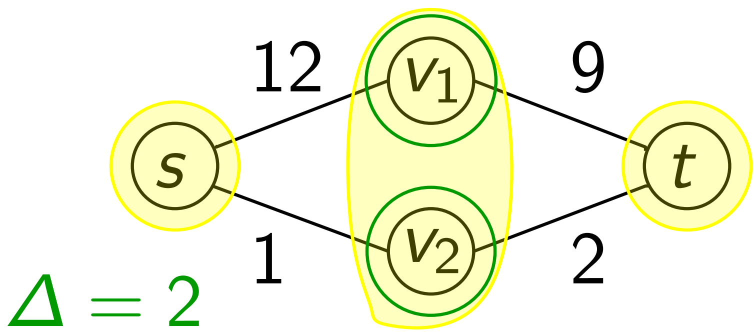
- V kann in disjunkte Menge $V = \{s\} \cup V_1 \cup V_2 \cup \cdots \cup V_m \cup \{t\}$ zerlegt werden
- $\forall \{i, j\} \in E : (i \in V_k \wedge j \in V_{k+1}$ für ein $k \in \{1, \ldots, m-1\}) \vee (i = s \wedge j \in V_1) \vee (i = t \wedge j \in V_m)$

ARSP und RDSP für layered Graphen mit $\Delta < 1$?



Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?



Definition(Breite)

Ist G ein layered Graph, so bezeichnen wir mit Breite folgendes
$\Delta = \max\{|V_k| \mid k = 1, \ldots, m\}$

Für einfachere Graphen ...

Sei $G = (V, E)$ ein ungerichteter Graph.

Definition(layered Graph)

G ist ein layered Graph, falls folgendes gilt

- V kann in disjunkte Menge $V = \{s\} \cup V_1 \cup V_2 \cup \cdots \cup V_m \cup \{t\}$ zerlegt werden
- $\forall \{i, j\} \in E : (i \in V_k \wedge j \in V_{k+1}$ für ein $k \in \{1, \ldots, m-1\}) \vee (i = s \wedge j \in V_1) \vee (i = t \wedge j \in V_m)$

ARSP und RDSP für layered Graphen mit $\Delta < 1$?

Lösung trivial, es gibt nur einen Weg

Versagen des Determinismus

ARSP und RDSP für Graphen mit mehr als einem Szenario?

Satz. ARSP/RDSP sind für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

Folgerung. ARSP/RDSP sind für Graphen mit zwei Szenarien NP-hart

Folgerung. Sei $k > 2$ fest
ARSP/RDSP sind für Graphen mit k Szenarien NP-hart

NP-härte von ARSP

Satz. ARSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

NP-härte von ARSP

Satz. ARSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

$$\begin{aligned}\text{ARSP}^2_E = \; & \{\langle G = (V, E), S = \{s_1, s_2\}, k\rangle \mid \\ & G \text{ ist ungerichteter Graph } \wedge S \text{ Menge an 2 Szenarien für } G \wedge \\ & \text{G besitzt einen Pfad mit Worst-Case Laufzeit über } S \text{ mit Länge } \leq k\}\end{aligned}$$

NP-härte von ARSP

Satz. ARSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

$$\begin{aligned}\text{ARSP}^2_E = \{ & \langle G = (V, E), S = \{s_1, s_2\}, k\rangle \mid \\ & G \text{ ist ungerichteter Graph } \wedge S \text{ Menge an 2 Szenarien für } G \wedge \\ & \text{G besitzt einen Pfad mit Worst-Case Laufzeit über } S \text{ mit Länge } \leq k\}\end{aligned}$$

Bemerkung.

Das Lösen des Optimierungsproblems ARSP für 2 Szenarien lässt sich effizient mit Hilfe von binärer Suche auf das Lösen des Entscheidungsproblems ARSP^2_E zurückführen.

NP-härte von ARSP

Satz. ARSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

Beweis. z.z. Für alle Probleme $A \in \mathbf{NP}$ gilt $A \leq \mathrm{ARSP}_E^2$

NP-härte von ARSP

Satz. ARSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

Beweis. Idee: polynomialzeit Reduktion von 2PART auf ARSP_E^2

$$2\mathrm{PART} \leq \mathrm{ARSP}_E^2$$

NP-härte von ARSP

Satz. ARSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

Beweis. **Idee:** polynomialzeit Reduktion von 2PART auf ARSP_E^2

$$\text{2PART} \leq \text{ARSP}_E^2$$

$$\text{2PART} = \{\langle a_1, \ldots, a_m \rangle \mid a_1, \ldots, a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1, \ldots m\} \text{ mit } \textstyle\sum_{i \in I'} a_i = \sum_{j \notin I'} a_j\}$$

NP-härte von ARSP

Satz. ARSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

Beweis. **Idee:** polynomialzeit Reduktion von 2PART auf ARSP_E^2

$$2\text{PART} \leq \text{ARSP}_E^2$$

$$\begin{aligned}\text{ARSP}_E^2 = \; & \{\langle G = (V, E), S = \{s_1, s_2\}, k\rangle \mid \\ & G \text{ ist ungerichteter Graph } \wedge S \text{ Menge an 2 Szenarien für } G \wedge \\ & \text{G besitzt einen Pfad mit Worst-Case Laufzeit über } S \text{ mit Länge } \leq k\}\end{aligned}$$

$$\begin{aligned}2\text{PART} = \; & \{\langle a_1, \ldots, a_m\rangle \mid \\ & a_1, \ldots, a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1, ...m\} \text{ mit } \textstyle\sum_{i \in I'} a_i = \sum_{j \notin I'} a_j\}\end{aligned}$$

NP-härte von ARSP

Satz. ARSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

Beweis. **Idee:** polynomialzeit Reduktion von 2PART auf ARSP_E^2

$$2\text{PART} \leq \text{ARSP}_E^2$$

$$\text{ARSP}_E^2 = \{\langle G = (V, E), S = \{s_1, s_2\}, k\rangle \mid$$
G ist ungerichteter Graph $\wedge$ S Menge an 2 Szenarien für G $\wedge$
G besitzt einen Pfad mit Worst-Case Laufzeit über S mit Länge $\leq k\}$

$$2\text{PART} = \{\langle a_1, \ldots, a_m\rangle \mid$$
$a_1, \ldots, a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1, ...m\}$ mit $\sum_{i \in I'} a_i = \sum_{j \notin I'} a_j\}$

Gesucht: totale, in polynomialzeit berechenbare Funktion f: $\mathbb{N} \to \mathbb{N}$, sodass gilt

$$\forall x \in \mathbb{N}(x \in 2\text{PART} \Leftrightarrow f(x) \in \text{ARSP}_E^2)$$

NP-härte von ARSP

$$\begin{aligned}
\text{ARSP}_E^2 = &\{\langle G = (V, E), S = \{s_1, s_2\}, k\rangle \mid \\
&G \text{ ist ungerichteter Graph } \wedge S \text{ Menge an 2 Szenarien für } G \wedge \\
&\text{G besitzt einen s-t Pfad mit Worst-Case Laufzeit über } S \text{ mit Länge } \leq k\}
\end{aligned}$$

$$\begin{aligned}
\text{2PART} = &\{\langle a_1, \ldots, a_m\rangle \mid \\
&a_1, \ldots, a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1, ...m\} \text{ mit } \textstyle\sum_{i \in I} a_i = \sum_{j \notin I'} a_j\}
\end{aligned}$$

NP-härte von ARSP

$$\begin{aligned}
\text{ARSP}_E^2 = &\{\langle G = (V,E), S = \{s_1, s_2\}, k\rangle \mid \\
&G \text{ ist ungerichteter Graph } \wedge S \text{ Menge an 2 Szenarien für } G \wedge \\
&\text{G besitzt einen s-t Pfad mit Worst-Case Laufzeit über } S \text{ mit Länge } \leq k\}
\end{aligned}$$

$$\begin{aligned}
\text{2PART} = &\{\langle a_1, \ldots, a_m\rangle \mid \\
&a_1, \ldots, a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1, ...m\} \text{ mit } \textstyle\sum_{i \in I} a_i = \sum_{j \notin I'} a_j\}
\end{aligned}$$

Sei

$$f(x) = \begin{cases} \langle G', S, k\rangle & \text{falls } x = \langle a_1, \ldots, a_m\rangle\text{: } a_1, \ldots, a_m, m \in \mathbb{N} \\ \langle(\emptyset, \emptyset), \emptyset, 1\rangle & \sum_{i \in \{1,\ldots,m\}} a_i \text{ ungerade und sonst} \end{cases}$$

wobei $G' = (V', E')$ und $S = \{1, 2\}$ wie folgt definiert sind und $k = \frac{1}{2}\sum_{j \in I} a_j$

NP-härte von ARSP

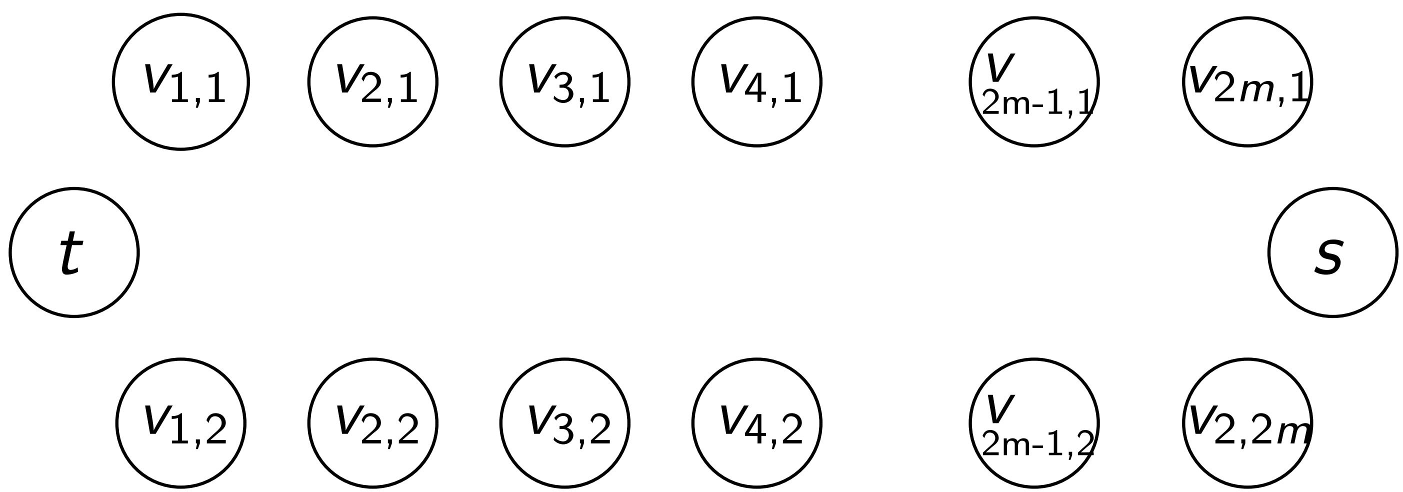
$$\begin{aligned}\text{ARSP}_E^2 = \{\langle G = (V,E), S = \{s_1, s_2\}, k\rangle \mid \\ G \text{ ist ungerichteter Graph } \wedge S \text{ Menge an 2 Szenarien für } G \wedge \\ \text{G besitzt einen s-t Pfad mit Worst-Case Laufzeit über } S \text{ mit Länge } \leq k\}\end{aligned}$$

$$\begin{aligned}\text{2PART} = \{\langle a_1, \ldots, a_m\rangle \mid \\ a_1, \ldots, a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1, \ldots m\} \text{ mit } \textstyle\sum_{i \in I} a_i = \sum_{j \notin I'} a_j\}\end{aligned}$$

Sei

$$f(x) = \begin{cases} \langle G', S, k\rangle & \text{falls } x = \langle a_1, \ldots, a_m\rangle \colon a_1, \ldots, a_m, m \in \mathbb{N} \\ \langle(\emptyset, \emptyset), \emptyset, 1\rangle & \sum_{i \in \{1,\ldots,m\}} a_i \text{ ungerade und sonst} \end{cases}$$

wobei $G' = (V', E')$ und $S = \{1, 2\}$ wie folgt definiert sind und $k = \frac{1}{2} \sum_{j \in I} a_j$



NP-härte von ARSP

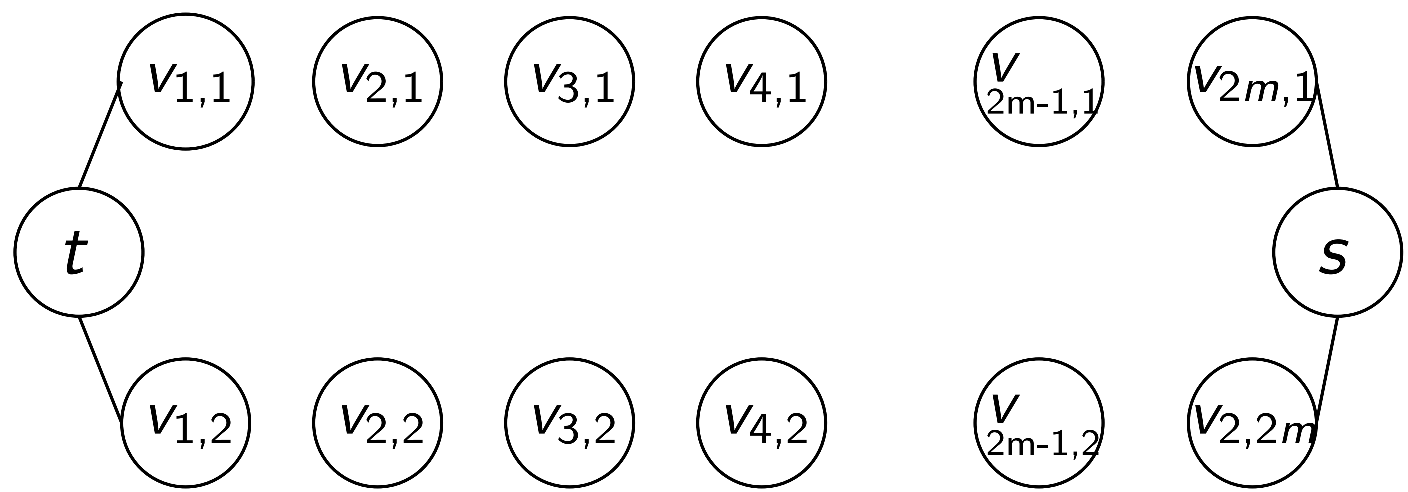
$$\begin{aligned}\text{ARSP}^2_E = \{ \langle G = (V,E), S = \{s_1, s_2\}, k \rangle \mid\ & G \text{ ist ungerichteter Graph } \wedge S \text{ Menge an 2 Szenarien für } G \wedge \\ & \text{G besitzt einen s-t Pfad mit Worst-Case Laufzeit über } S \text{ mit Länge } \leq k \}\end{aligned}$$

$$\begin{aligned}\text{2PART} = \{ \langle a_1, \ldots, a_m \rangle \mid\ & a_1, \ldots, a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1, ...m\} \text{ mit } \textstyle\sum_{i \in I} a_i = \sum_{j \notin I'} a_j \}\end{aligned}$$

Sei

$$f(x) = \begin{cases} \langle G', S, k \rangle & \text{falls } x = \langle a_1, \ldots, a_m \rangle : a_1, \ldots, a_m, m \in \mathbb{N} \\ \langle (\emptyset, \emptyset), \emptyset, 1 \rangle & \sum_{i \in \{1,\ldots,m\}} a_i \text{ ungerade und sonst} \end{cases}$$

wobei $G' = (V', E')$ und $S = \{1, 2\}$ wie folgt definiert sind und $k = \frac{1}{2} \sum_{j \in I} a_j$



NP-härte von ARSP

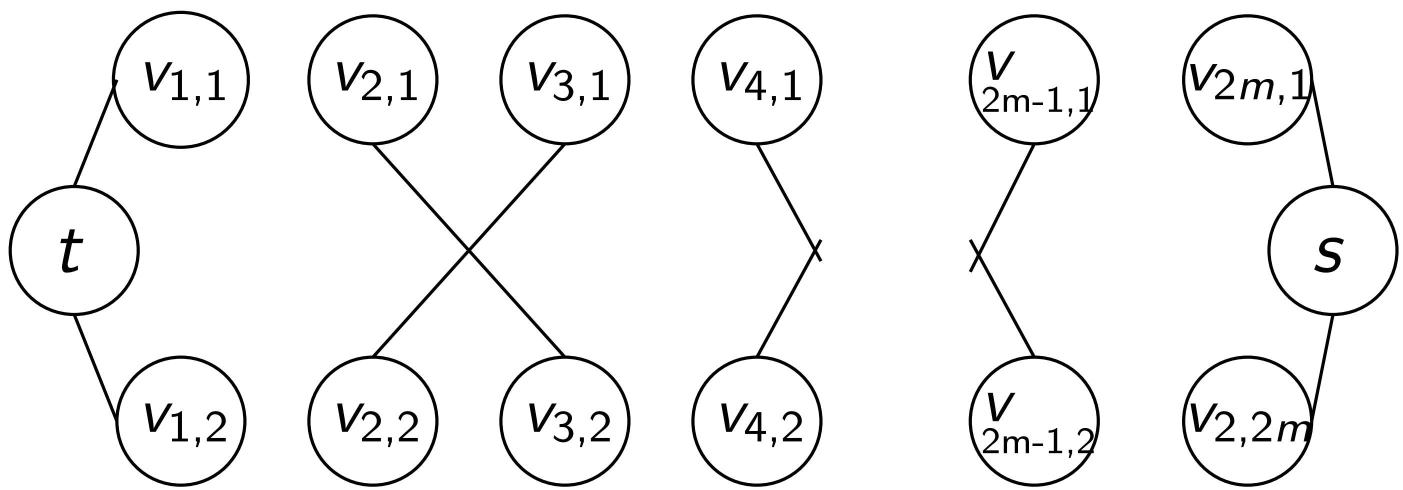
$\text{ARSP}^2_E = \{\langle G = (V, E), S = \{s_1, s_2\}, k\rangle \mid$
G ist ungerichteter Graph $\wedge$ S Menge an 2 Szenarien für G $\wedge$
G besitzt einen s-t Pfad mit Worst-Case Laufzeit über S mit Länge $\leq k\}$

$\text{2PART} = \{\langle a_1, \ldots, a_m\rangle \mid$
$a_1, \ldots, a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1, ...m\}$ mit $\sum_{i \in I} a_i = \sum_{j \notin I'} a_j\}$

Sei

$$f(x) = \begin{cases} \langle G', S, k\rangle & \text{falls } x = \langle a_1, \ldots, a_m\rangle\text{: } a_1, \ldots, a_m, m \in \mathbb{N} \\ \langle(\emptyset, \emptyset), \emptyset, 1\rangle & \sum_{i \in \{1,\ldots,m\}} a_i \text{ ungerade und sonst} \end{cases}$$

wobei $G' = (V', E')$ und $S = \{1, 2\}$ wie folgt definiert sind und $k = \frac{1}{2}\sum_{j \in I} a_j$



NP-härte von ARSP

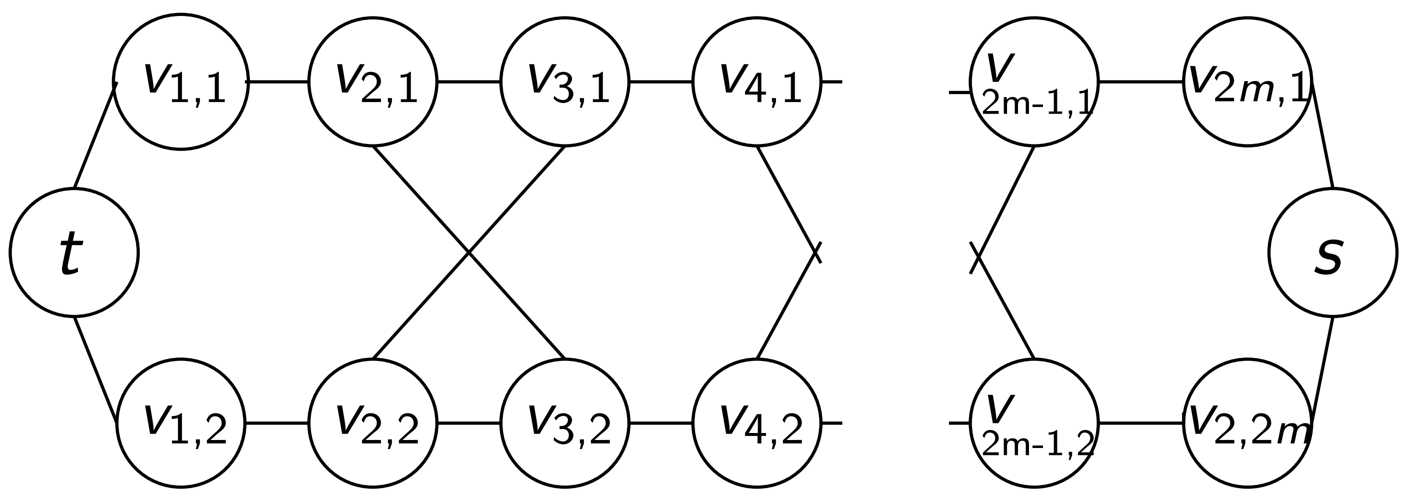
$\text{ARSP}_E^2 = \{\langle G = (V,E), S = \{s_1, s_2\}, k\rangle \mid$
G ist ungerichteter Graph $\wedge$ S Menge an 2 Szenarien für G $\wedge$
G besitzt einen s-t Pfad mit Worst-Case Laufzeit über S mit Länge $\leq k\}$

$\text{2PART} = \{\langle a_1, \ldots, a_m\rangle \mid$
$a_1, \ldots, a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1, ...m\}$ mit $\sum_{i \in I} a_i = \sum_{j \notin I'} a_j\}$

Sei

$$f(x) = \begin{cases} \langle G', S, k\rangle & \text{falls } x = \langle a_1, \ldots, a_m\rangle\text{: } a_1, \ldots, a_m, m \in \mathbb{N} \\ \langle (\emptyset, \emptyset), \emptyset, 1\rangle & \sum_{i \in \{1,\ldots,m\}} a_i \text{ ungerade und sonst} \end{cases}$$

wobei $G' = (V', E')$ und $S = \{1, 2\}$ wie folgt definiert sind und $k = \frac{1}{2} \sum_{j \in I} a_j$



NP-härte von ARSP

$$\begin{aligned}\text{ARSP}^2_E = \{ \langle G = (V,E), S = \{s_1, s_2\}, k \rangle \mid & \\ & G \text{ ist ungerichteter Graph } \wedge S \text{ Menge an 2 Szenarien für } G \wedge \\ & \text{G besitzt einen s-t Pfad mit Worst-Case Laufzeit über } S \text{ mit Länge } \leq k \}\end{aligned}$$

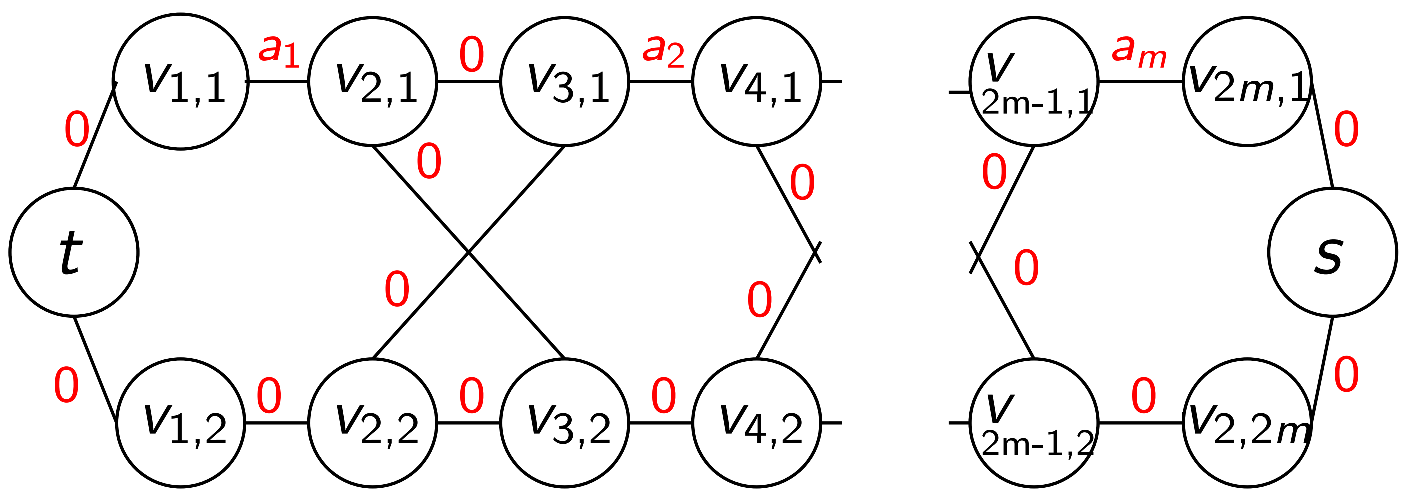
$$\begin{aligned}\text{2PART} = \{ \langle a_1, \ldots, a_m \rangle \mid & \\ & a_1, \ldots, a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1, \ldots m\} \text{ mit } \textstyle\sum_{i \in I} a_i = \sum_{j \notin I'} a_j \}\end{aligned}$$

Sei

$$f(x) = \begin{cases} \langle G', S, k \rangle & \text{falls } x = \langle a_1, \ldots, a_m \rangle\text{: } a_1, \ldots, a_m, m \in \mathbb{N} \\ \langle (\emptyset, \emptyset), \emptyset, 1 \rangle & \sum_{i \in \{1, \ldots, m\}} a_i \text{ ungerade und sonst} \end{cases}$$

wobei $G' = (V', E')$ und $S = \{1, 2\}$ wie folgt definiert sind und $k = \frac{1}{2} \sum_{j \in I} a_j$

$$c^1_{v_{2k-1,i}, v_{2k,i}} = \begin{cases} a_k & \text{falls } i = 1 \\ 0 & \text{falls } i = 2 \end{cases}$$



NP-härte von ARSP

$$\begin{aligned}\mathrm{ARSP}^2_E = \{\langle G=(V,E), S=\{s_1,s_2\}, k\rangle \mid\ & G \text{ ist ungerichteter Graph } \wedge S \text{ Menge an 2 Szenarien für } G \wedge \\ & \text{G besitzt einen s-t Pfad mit Worst-Case Laufzeit über } S \text{ mit Länge } \leq k\}\end{aligned}$$

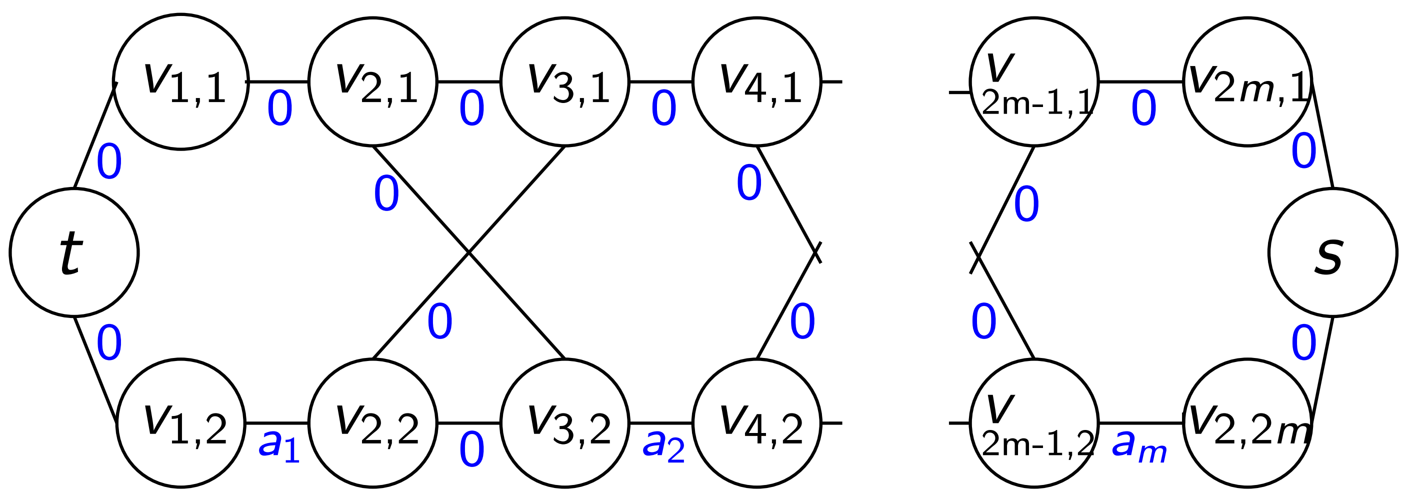
$$\begin{aligned}\mathrm{2PART} = \{\langle a_1,\ldots,a_m\rangle \mid\ & a_1,\ldots,a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1,\ldots m\} \text{ mit } \textstyle\sum_{i\in I} a_i = \sum_{j\notin I'} a_j\}\end{aligned}$$

Sei

$$f(x) = \begin{cases} \langle G', S, k\rangle & \text{falls } x = \langle a_1,\ldots,a_m\rangle\text{: } a_1,\ldots,a_m, m \in \mathbb{N} \\ \langle(\emptyset,\emptyset),\emptyset,1\rangle & \sum_{i\in\{1,\ldots,m\}} a_i \text{ ungerade und sonst}\end{cases}$$

wobei $G' = (V', E')$ und $S = \{1,2\}$ wie folgt definiert sind und $k = \frac{1}{2}\sum_{j\in I} a_j$

$$c^2_{v_{2k-1,i},v_{2k,i}} = \begin{cases} a_k & \text{falls } i = 1 \\ 0 & \text{falls } i = 2\end{cases}$$



NP-härte von ARSP

$$\begin{aligned}\mathrm{ARSP}_E^2 = \{ \langle G = (V,E), S = \{s_1, s_2\}, k \rangle \mid & \\ & G \text{ ist ungerichteter Graph } \wedge S \text{ Menge an 2 Szenarien für } G \wedge \\ & \text{G besitzt einen s-t Pfad mit Worst-Case Laufzeit über } S \text{ mit Länge } \leq k \}\end{aligned}$$

$$\begin{aligned}\mathrm{2PART} = \{ \langle a_1, \ldots, a_m \rangle \mid & \\ & a_1, \ldots, a_m, m \in \mathbb{N} \wedge \exists I' \subseteq \{1, \ldots m\} \text{ mit } \textstyle\sum_{i \in I} a_i = \sum_{j \notin I'} a_j \}\end{aligned}$$

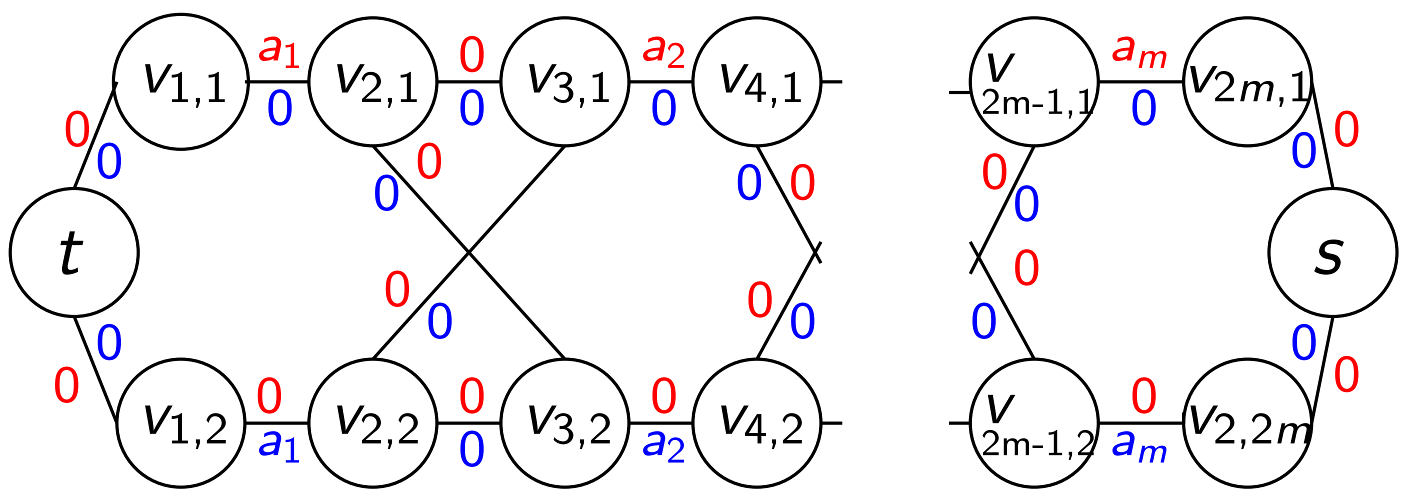
Sei

$$f(x) = \begin{cases} \langle G', S, k \rangle & \text{falls } x = \langle a_1, \ldots, a_m \rangle\text{: } a_1, \ldots, a_m, m \in \mathbb{N} \\ \langle (\emptyset, \emptyset), \emptyset, 1 \rangle & \sum_{i \in \{1, \ldots, m\}} a_i \text{ ungerade und sonst} \end{cases}$$

wobei $G' = (V', E')$ und $S = \{1, 2\}$ wie folgt definiert sind und $k = \frac{1}{2} \sum_{j \in I} a_j$

$$c^1_{v_{2k-1,i}, v_{2k,i}} = \begin{cases} a_k & \text{falls } i = 1 \\ 0 & \text{falls } i = 2 \end{cases}$$

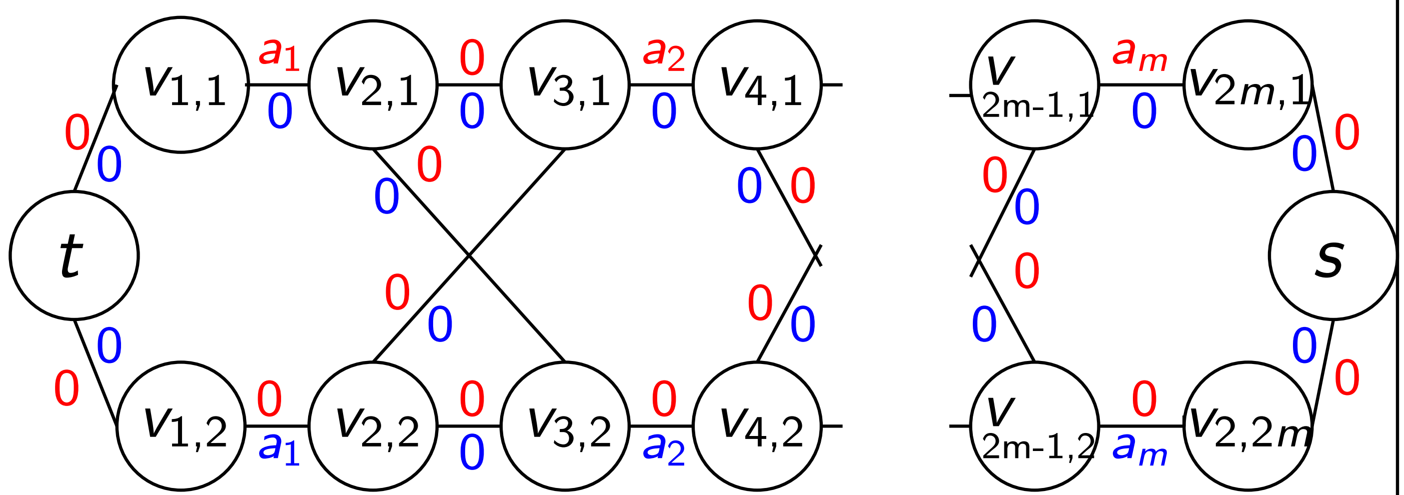
$$c^2_{v_{2k-1,i}, v_{2k,i}} = \begin{cases} a_k & \text{falls } i = 1 \\ 0 & \text{falls } i = 2 \end{cases}$$



NP-härte von ARSP

Beobachtung. Sei π ein s-t Pfad

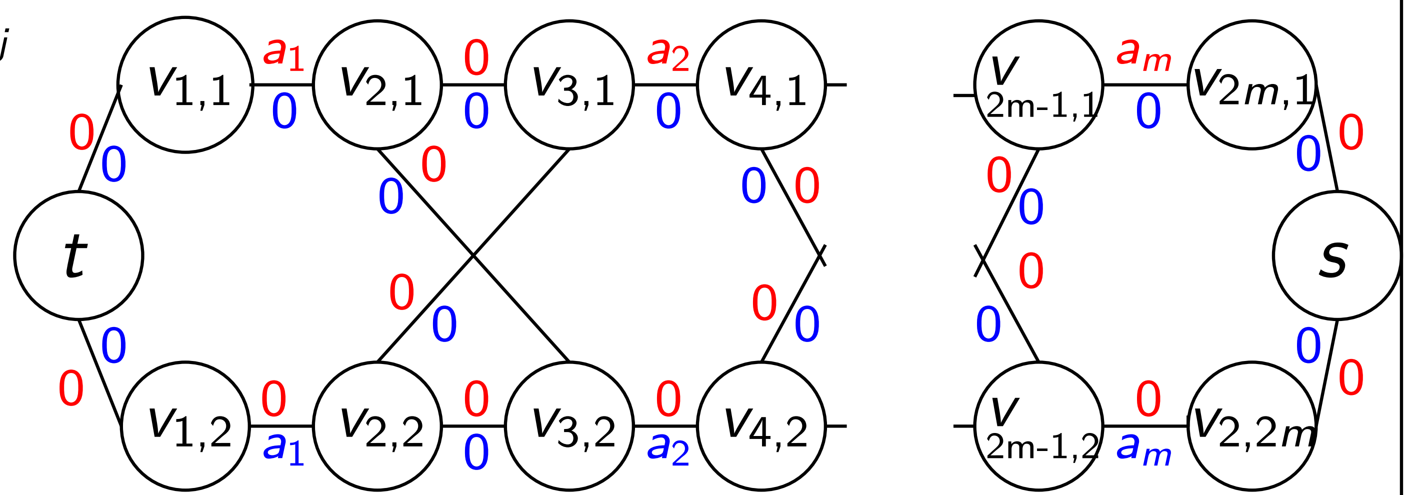
$c^1(\pi) + c^2(\pi) =?$



NP-härte von ARSP

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

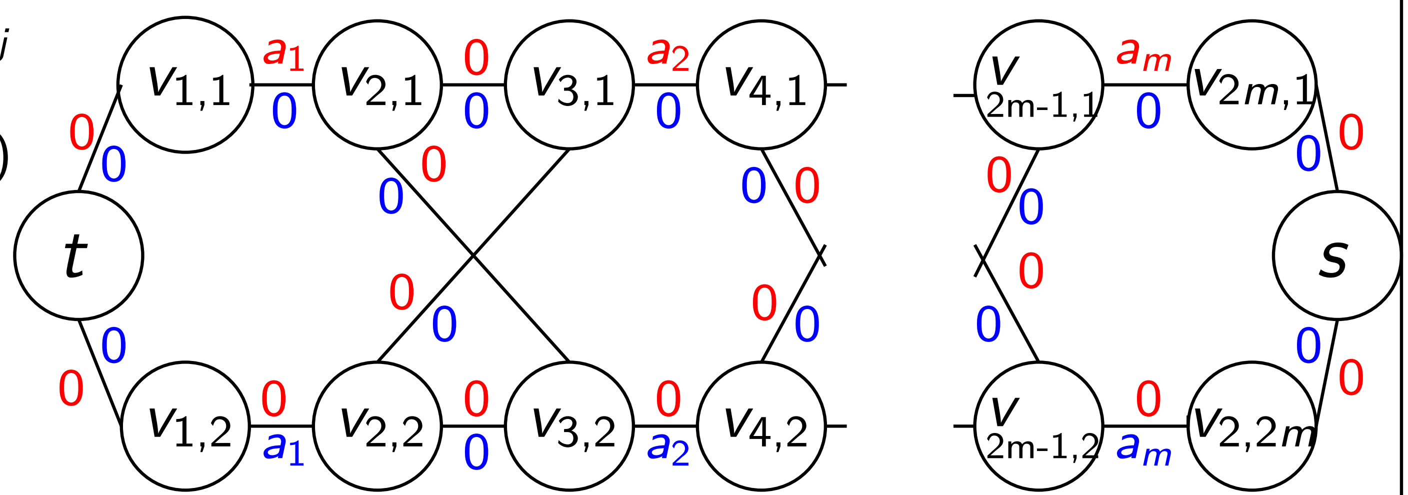


NP-härte von ARSP

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

$I' = \{w \mid (v_{2w-1,1}, v_{2w,1})$ wird von π durchquert$\}$



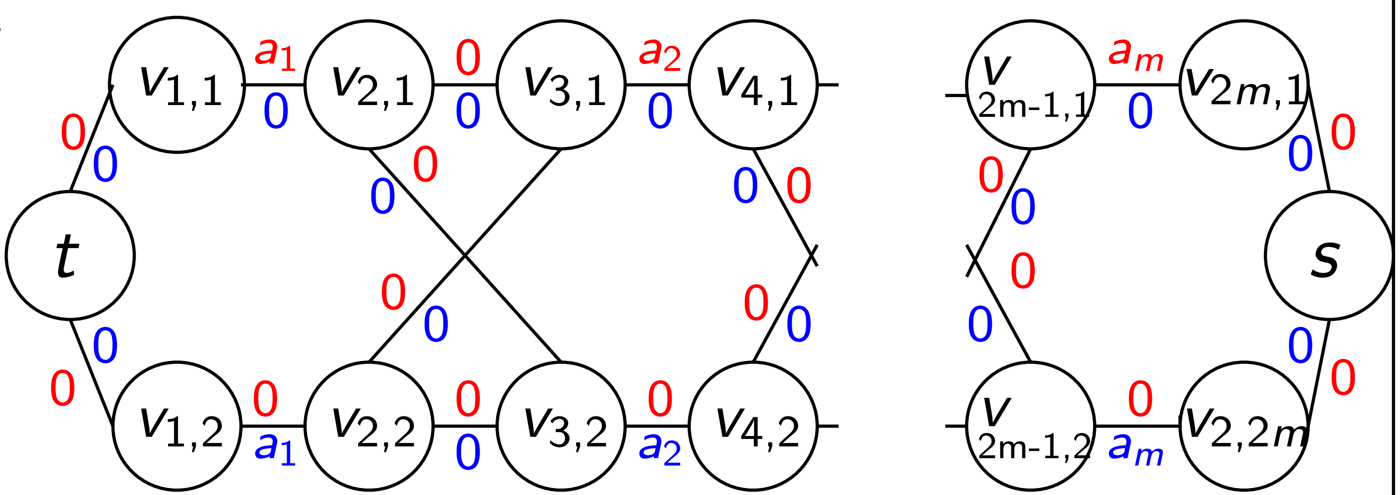
NP-härte von ARSP

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

$I' = \{w \mid (v_{2w-1,1}, v_{2w,1})$ wird von π durchquert$\}$

$\{1, \ldots, m\} - I'$
$= \{w \mid (v_{2w-1,2}, v_{2w,2})$ wird von π durchquert$\}$



NP-härte von ARSP

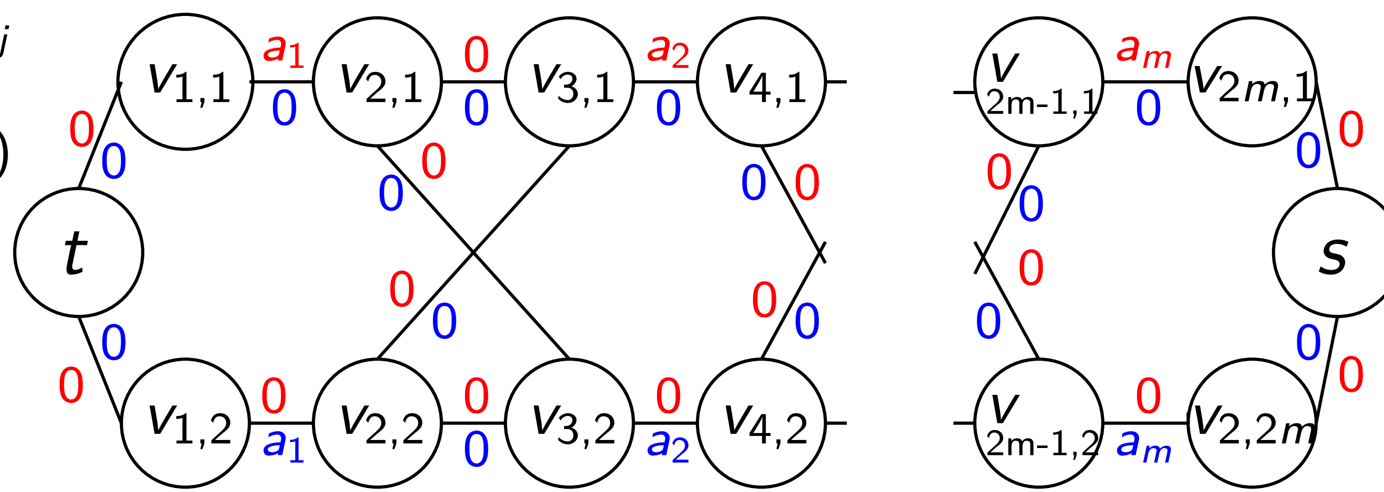
n.z.z $\forall x \in \mathbb{N}(x \in 2\text{PART} \Leftrightarrow f(x) \in \text{ARSP}_E^2)$

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

$I' = \{w \mid (v_{2w-1,1}, v_{2w,1})$ wird von π durchquert$\}$

$\{1, \ldots, m\} - I'$
$= \{w \mid (v_{2w-1,2}, v_{2w,2})$ wird von π durchquert$\}$



NP-härte von ARSP

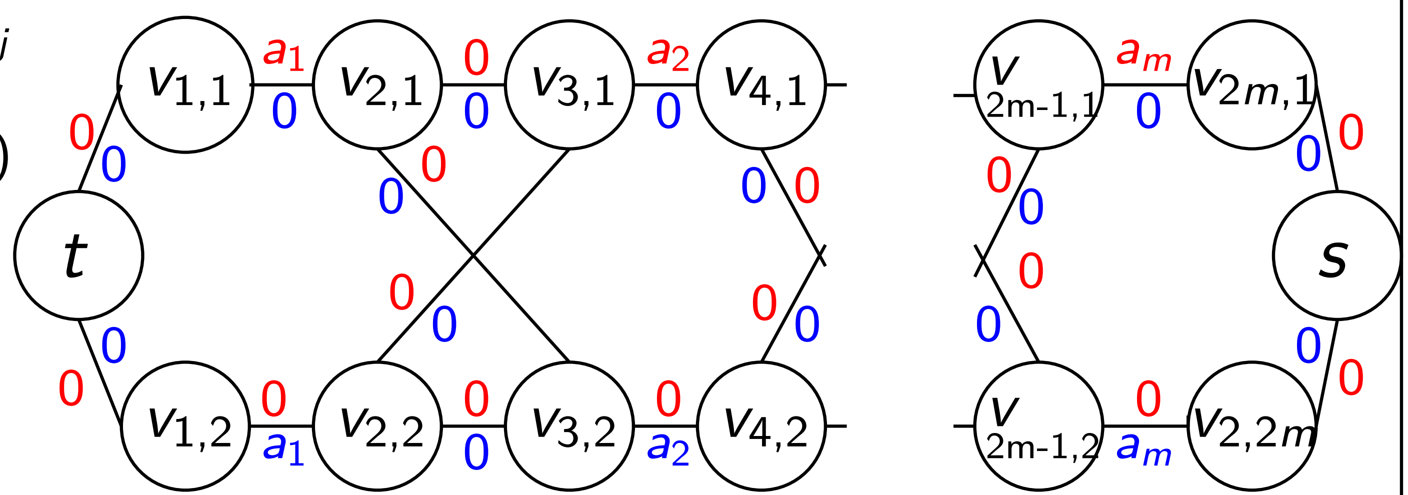
n.z.z $\forall x \in \mathbb{N}(x \in 2\text{PART} \Leftrightarrow f(x) \in \text{ARSP}^2_E)$
$x \in 2\text{PART}$

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

$I' = \{w \mid (v_{2w-1,1}, v_{2w,1})$ wird von π durchquert$\}$

$\{1, \ldots, m\} - I'$
$= \{w \mid (v_{2w-1,2}, v_{2w,2})$ wird von π durchquert$\}$



NP-härte von ARSP

n.z.z $\forall x \in \mathbb{N}(x \in 2\text{PART} \Leftrightarrow f(x) \in \text{ARSP}_E^2)$

$x \in 2\text{PART}$

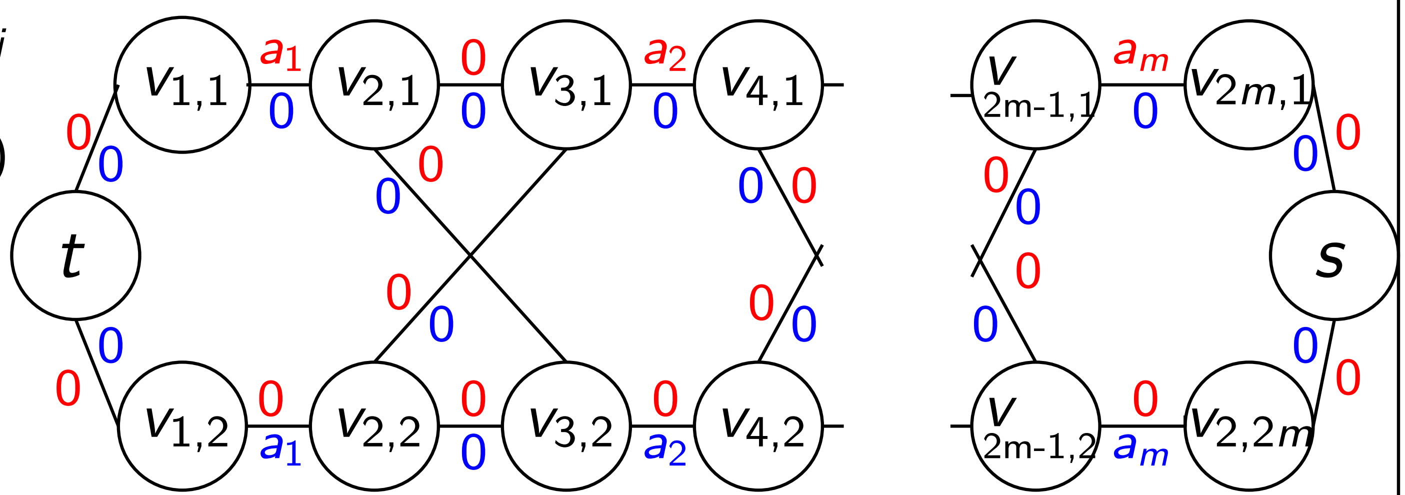
$\Rightarrow \exists I' \subseteq \{1, \ldots, m\}$ mit
$\sum_{i \in I'} a_i = \sum_{j \notin I'} a_j$

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

$I' = \{w \mid (v_{2w-1,1}, v_{2w,1})$ wird von π durchquert$\}$

$\{1, \ldots, m\} - I'$
$= \{w \mid (v_{2w-1,2}, v_{2w,2})$ wird von π durchquert$\}$



NP-härte von ARSP

n.z.z $\forall x \in \mathbb{N}(x \in \text{2PART} \Leftrightarrow f(x) \in \text{ARSP}_E^2)$

$x \in \text{2PART}$

$\Rightarrow \exists I' \subseteq \{1, \ldots, m\}$ mit
$\sum_{i \in I'} a_i = \sum_{j \notin I'} a_j$

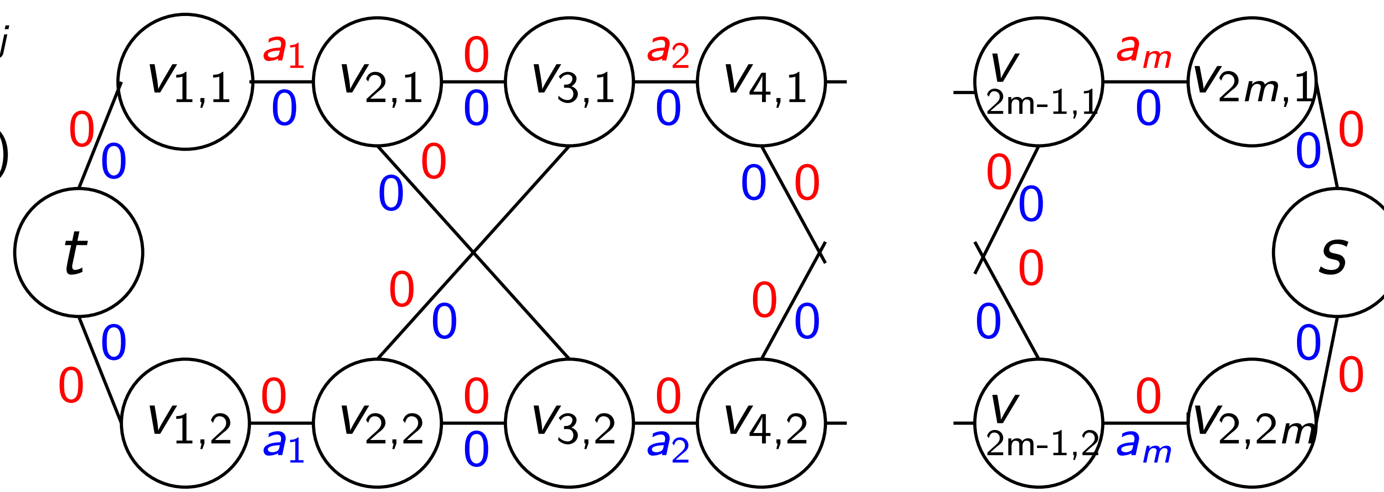
$\Rightarrow \exists$ s-t Pfad π mit
$c^1(\pi) = c^2(\pi) = \frac{1}{2} \sum_{j \in I} a_j$

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

$I' = \{w \mid (v_{2w-1,1}, v_{2w,1})$ wird von π durchquert$\}$

$\{1, \ldots, m\} - I'$
$= \{w \mid (v_{2w-1,2}, v_{2w,2})$ wird von π durchquert$\}$



NP-härte von ARSP

n.z.z $\forall x \in \mathbb{N}(x \in \text{2PART} \Leftrightarrow f(x) \in \text{ARSP}_E^2)$

$x \in$ 2PART

$\Rightarrow \exists I' \subseteq \{1, \ldots, m\}$ mit $\sum_{i \in I'} a_i = \sum_{j \notin I'} a_j$

$\Rightarrow \exists$ s-t Pfad π mit $c^1(\pi) = c^2(\pi) = \frac{1}{2} \sum_{j \in I} a_j$

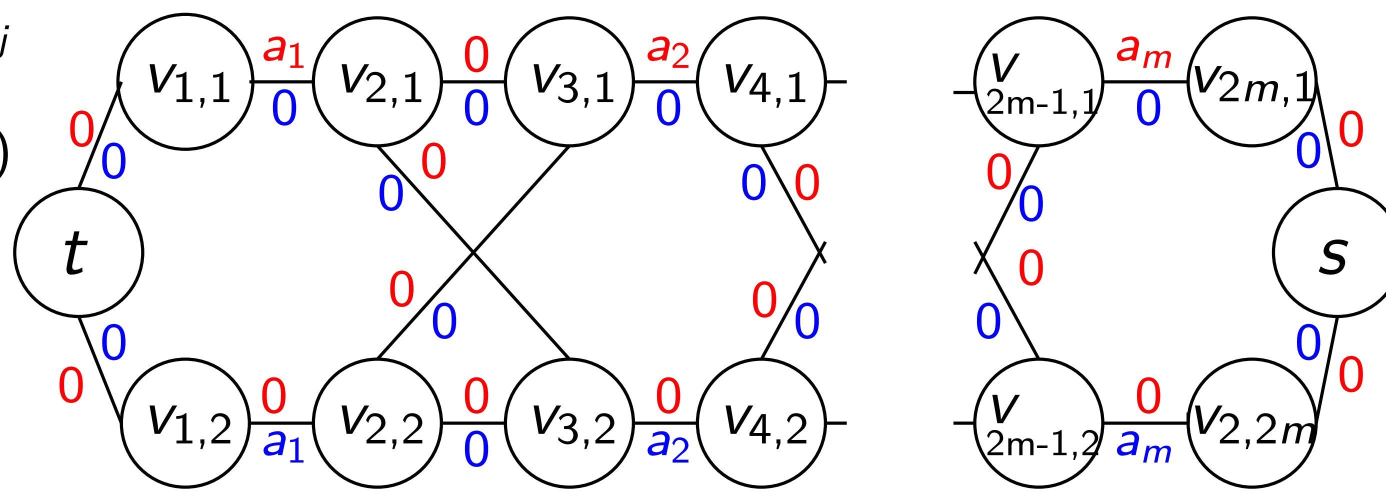
$\Rightarrow f(x) \in \text{ARSP}_E^2$

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

$I' = \{w \mid (v_{2w-1,1}, v_{2w,1})$ wird von π durchquert$\}$

$\{1, \ldots, m\} - I'$
$= \{w \mid (v_{2w-1,2}, v_{2w,2})$ wird von π durchquert$\}$



NP-härte von ARSP

n.z.z $\forall x \in \mathbb{N}(x \in 2\text{PART} \Leftrightarrow f(x) \in \text{ARSP}_E^2)$

$x \in 2\text{PART}$

$\Rightarrow \exists I' \subseteq \{1, \ldots, m\}$ mit $\sum_{i \in I'} a_i = \sum_{j \notin I'} a_j$

$\Rightarrow \exists$ s-t Pfad π mit $c^1(\pi) = c^2(\pi) = \frac{1}{2} \sum_{j \in I} a_j$

$\Rightarrow f(x) \in \text{ARSP}_E^2$

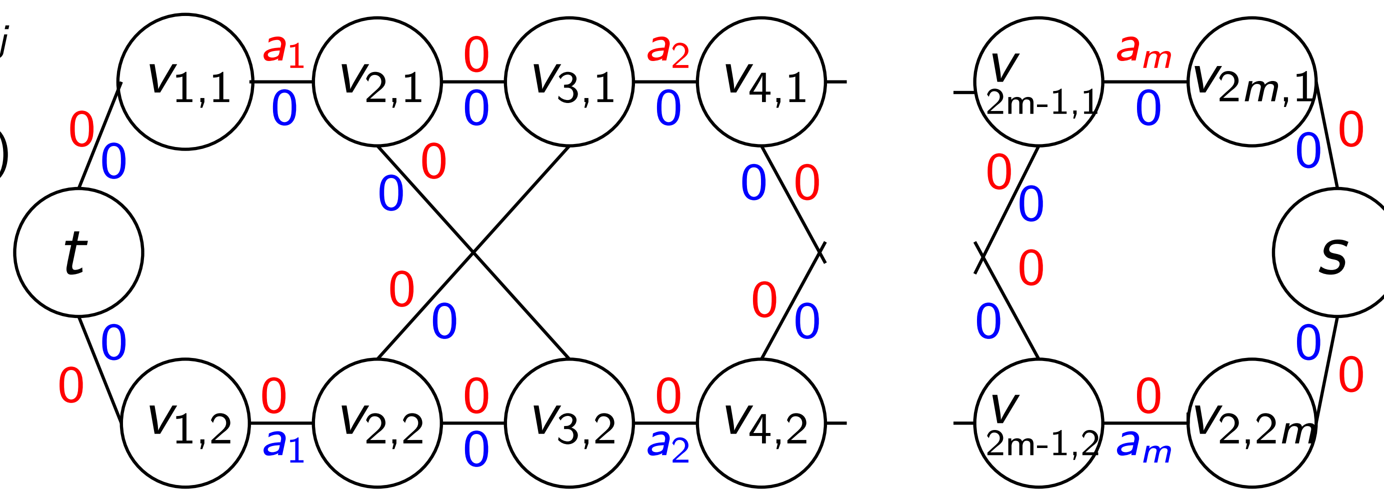
$x \notin 2\text{PART}$

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

$I' = \{w \mid (v_{2w-1,1}, v_{2w,1})$ wird von π durchquert$\}$

$\{1, \ldots, m\} - I'$
$= \{w \mid (v_{2w-1,2}, v_{2w,2})$ wird von π durchquert$\}$



NP-härte von ARSP

n.z.z $\forall x \in \mathbb{N}(x \in 2\text{PART} \Leftrightarrow f(x) \in \text{ARSP}^2_E)$

$x \in 2\text{PART}$

$\Rightarrow \exists I' \subseteq \{1, \ldots, m\}$ mit
$\sum_{i \in I'} a_i = \sum_{j \notin I'} a_j$

$\Rightarrow \exists$ s-t Pfad π mit
$c^1(\pi) = c^2(\pi) = \frac{1}{2} \sum_{j \in I} a_j$

$\Rightarrow f(x) \in \text{ARSP}^2_E$

$x \notin 2\text{PART}$

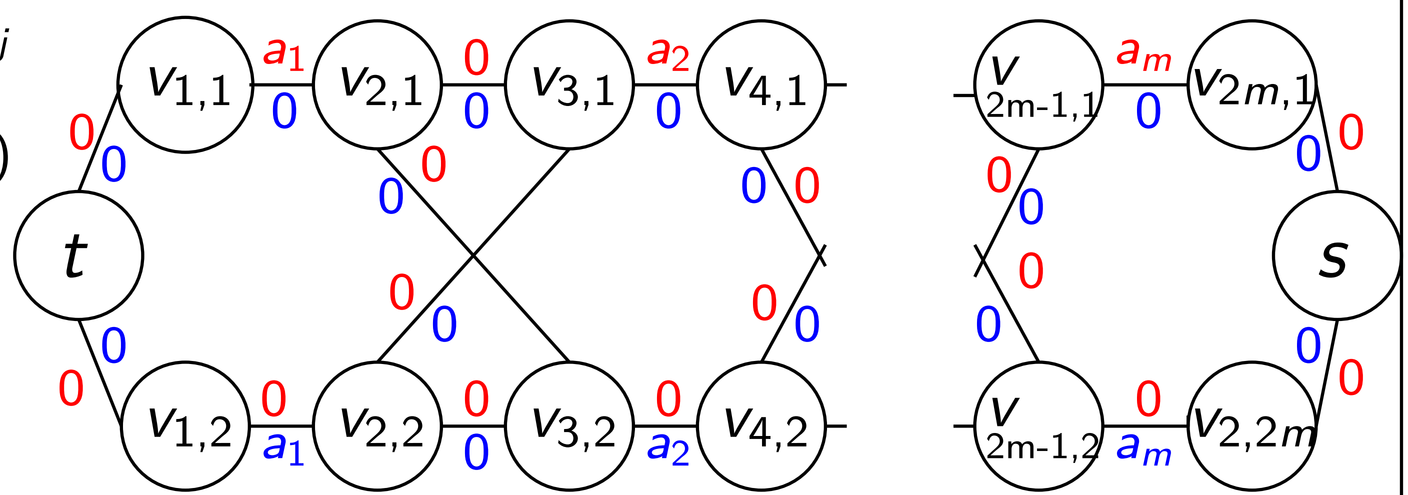
$\Rightarrow \forall I' \subseteq \{1, \ldots, m\}$ gilt
$\sum_{i \in I'} a_i \neq \sum_{j \notin I'} a_j$

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

$I' = \{w \mid (v_{2w-1,1}, v_{2w,1})$ wird von π durchquert$\}$

$\{1, \ldots, m\} - I'$
$= \{w \mid (v_{2w-1,2}, v_{2w,2})$ wird von π durchquert$\}$



NP-härte von ARSP

n.z.z $\forall x \in \mathbb{N}(x \in \text{2PART} \Leftrightarrow f(x) \in \text{ARSP}^2_E)$

$x \in$ 2PART

$\Rightarrow \exists I' \subseteq \{1, \ldots, m\}$ mit $\sum_{i \in I'} a_i = \sum_{j \notin I'} a_j$

$\Rightarrow \exists$ s-t Pfad π mit $c^1(\pi) = c^2(\pi) = \frac{1}{2} \sum_{j \in I} a_j$

$\Rightarrow f(x) \in \text{ARSP}^2_E$

$x \notin$ 2PART

$\Rightarrow \forall I' \subseteq \{1, \ldots, m\}$ gilt $\sum_{i \in I'} a_i \neq \sum_{j \notin I'} a_j$

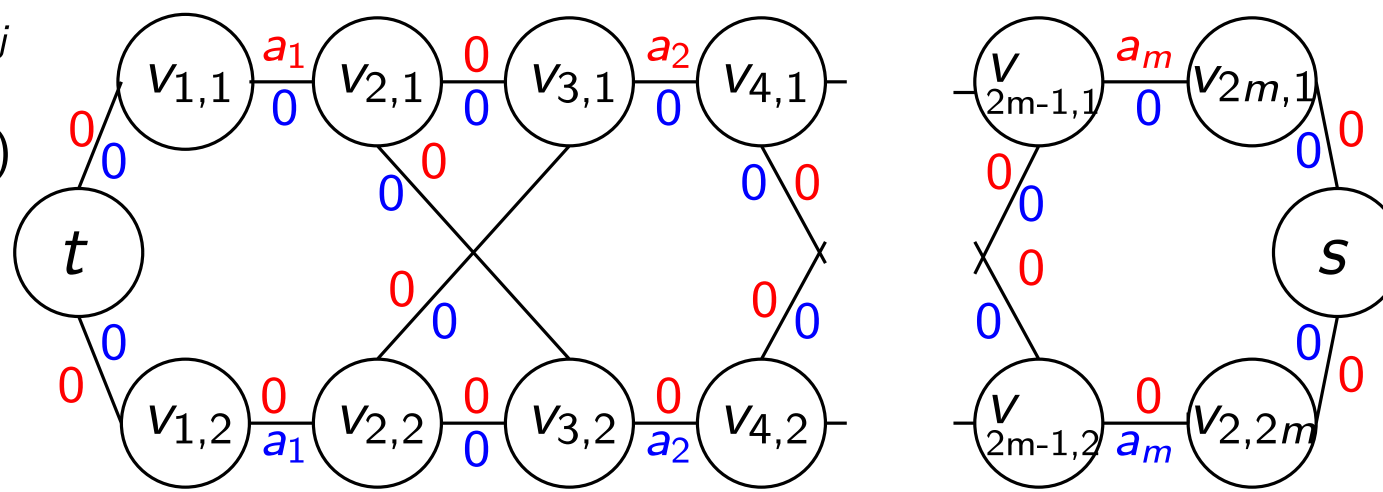
$\Rightarrow \forall$ s-t- Pfade π gilt $(c^1(\pi) \geq \frac{1}{2} \sum_{j \in I} a_j) \vee (c^2(\pi) \geq \frac{1}{2} \sum_{j \in I} a_j)$

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

$I' = \{w \mid (v_{2w-1,1}, v_{2w,1})$ wird von π durchquert$\}$

$\{1, \ldots, m\} - I'$
$= \{w \mid (v_{2w-1,2}, v_{2w,2})$ wird von π durchquert$\}$



NP-härte von ARSP

n.z.z $\forall x \in \mathbb{N}(x \in 2\text{PART} \Leftrightarrow f(x) \in \text{ARSP}_E^2)$

$x \in 2\text{PART}$

$\Rightarrow \exists I' \subseteq \{1, \ldots, m\}$ mit $\sum_{i \in I'} a_i = \sum_{j \notin I'} a_j$

$\Rightarrow \exists$ s-t Pfad π mit $c^1(\pi) = c^2(\pi) = \frac{1}{2} \sum_{j \in I} a_j$

$\Rightarrow f(x) \in \text{ARSP}_E^2$

$x \notin 2\text{PART}$

$\Rightarrow \forall I' \subseteq \{1, \ldots, m\}$ gilt $\sum_{i \in I'} a_i \neq \sum_{j \notin I'} a_j$

$\Rightarrow \forall$ s-t- Pfade π gilt $(c^1(\pi) \geq \frac{1}{2} \sum_{j \in I} a_j) \vee (c^2(\pi) \geq \frac{1}{2} \sum_{j \in I} a_j)$

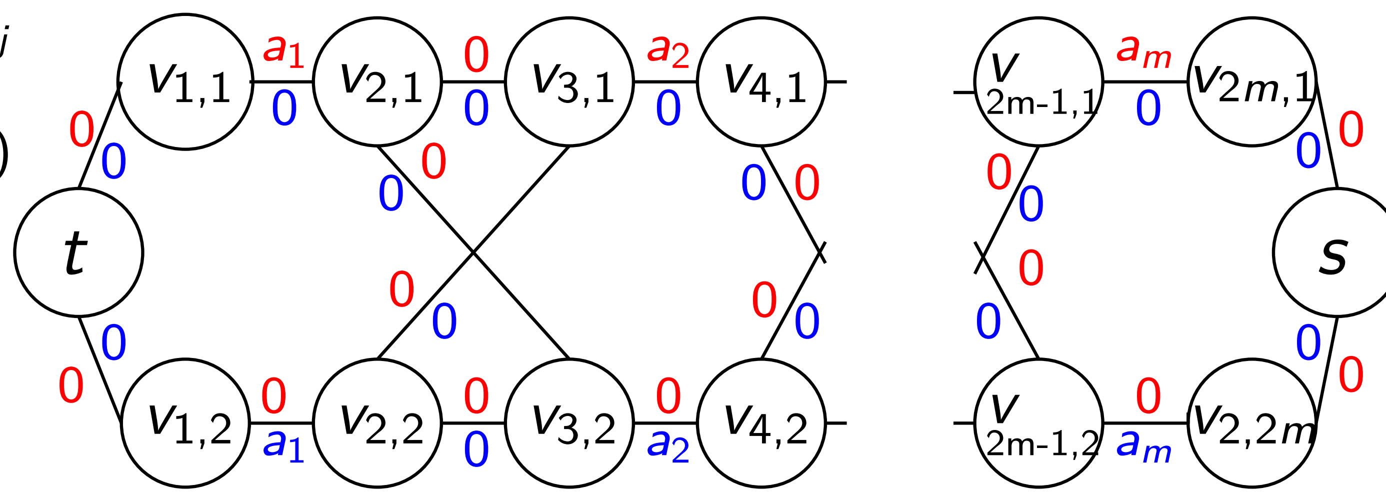
$\Rightarrow f(x) \notin \text{ARSP}_E^2$

Beobachtung. Sei π ein s-t Pfad

$c^1(\pi) + c^2(\pi) = \sum_{j \in I} a_j$

$I' = \{w \mid (v_{2w-1,1}, v_{2w,1})$ wird von π durchquert$\}$

$\{1, \ldots, m\} - I'$
$= \{w \mid (v_{2w-1,2}, v_{2w,2})$ wird von π durchquert$\}$



NP-härte von RDSP

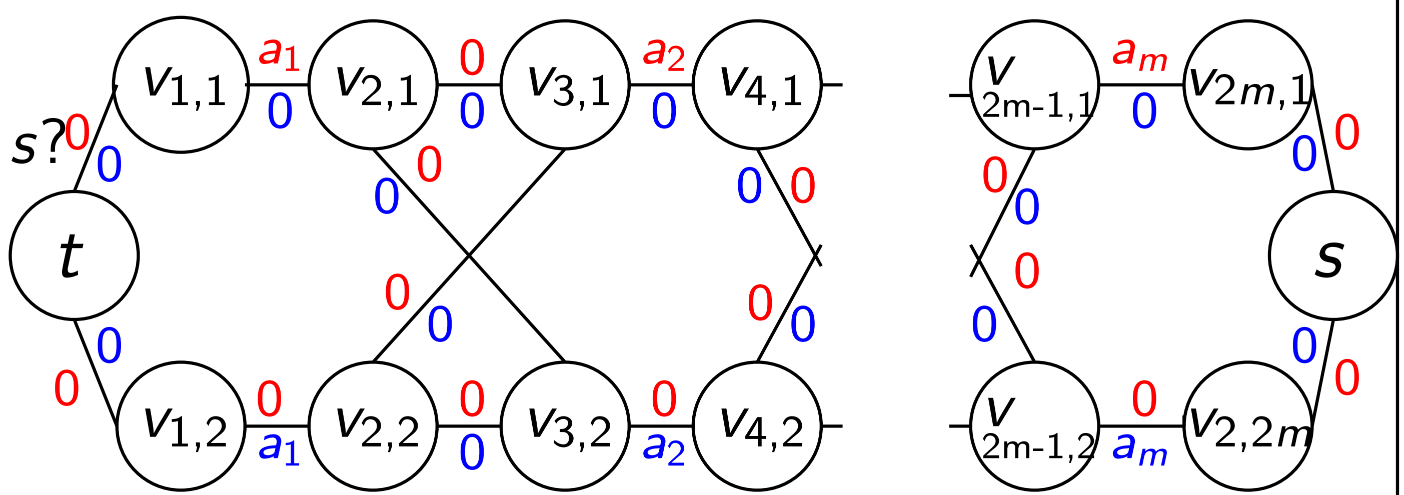
Satz. RDSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

NP-härte von RDSP

Satz. RDSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

Beobachtung. Sei π ein s-t Pfad

Länge eines kürzesten Weges unter Szenario s?



NP-härte von RDSP

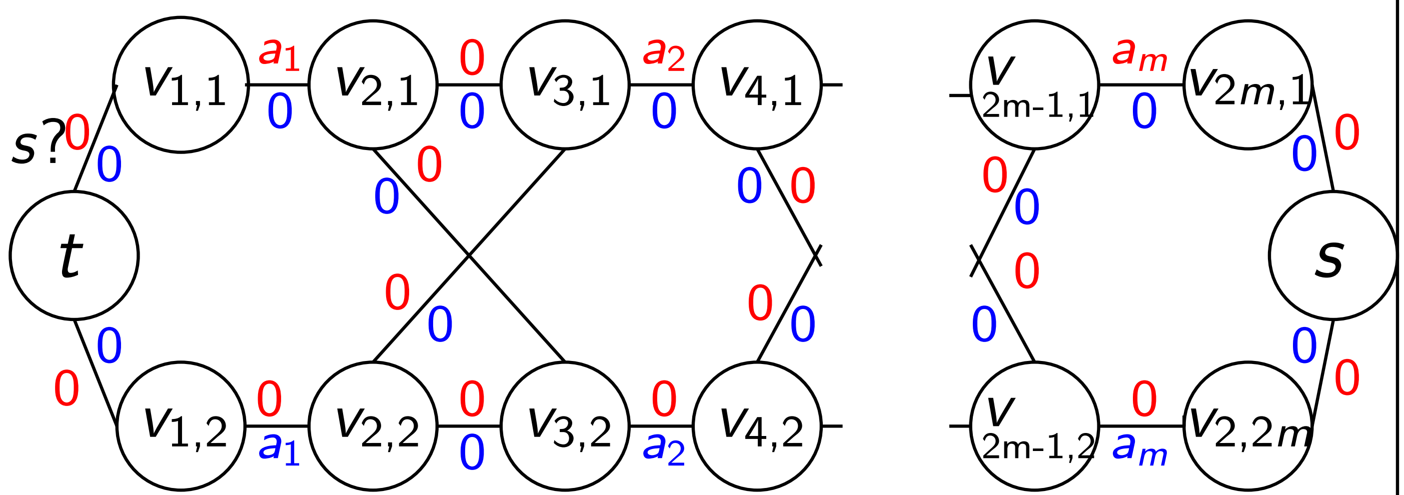
Satz. RDSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

Für jedes Szenario besitzt der kürzeste Pfad Länge 0, also ARSP $=$ RDSP für Konstruktion
Funktioniert wie bei ARSP

Beobachtung. Sei π ein s-t Pfad

Länge eines kürzesten Weges unter Szenario s?

- 0



NP-härte von RDSP

Satz. RDSP ist für zwei Szenarien und layered Graphen mit Breite 2 NP-hart

Beweis.

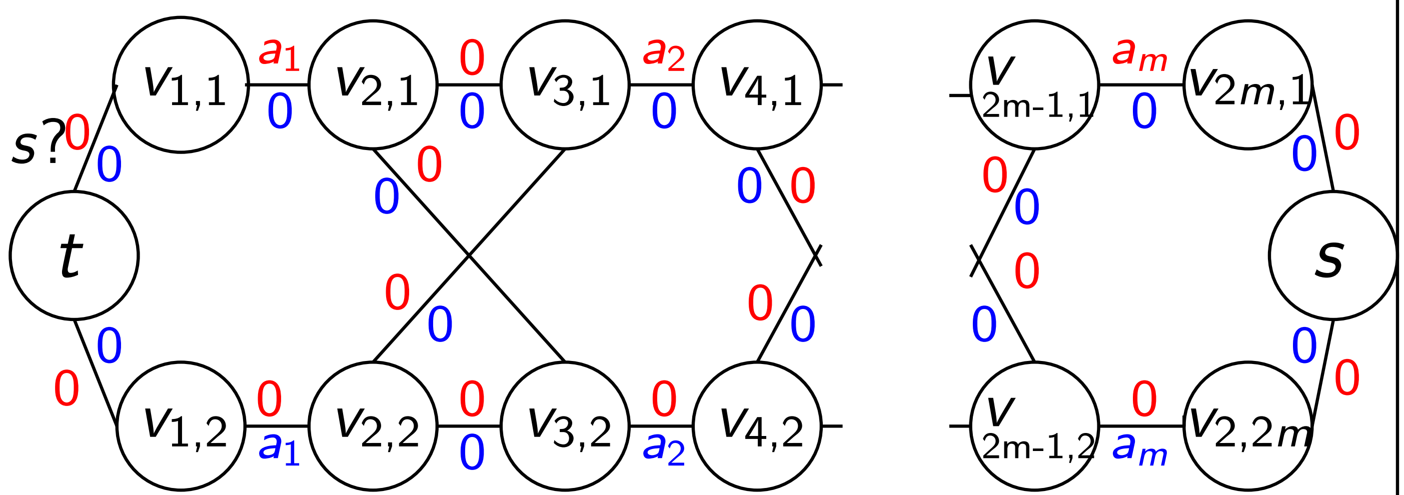
Für jedes Szenario besitzt der kürzeste Pfad Länge 0, also ARSP = RDSP für Konstruktion

Funktioniert wie bei ARSP

Beobachtung. Sei π ein s-t Pfad

Länge eines kürzesten Weges unter Szenario s?

- 0



Hoffnung für ARSP/RDSP

Fakt. 2PART ist in pseudopolynomialzeit berechenbar, mit Hilfe von dynamischer Programmierung

Hoffnung für ARSP/RDSP

Fakt. 2PART ist in pseudopolynomialzeit berechenbar, mit Hilfe von dynamischer Programmierung

Pseudopolynomialzeit. Länge der Eingabe wird in unärer Kodierung gemessen
Laufzeit ist abhängig vom numerischen Wert der Eingabe

Hoffnung für ARSP/RDSP

Fakt. 2PART ist in pseudopolynomialzeit berechenbar, mit Hilfe von dynamischer Programmierung

Pseudopolynomialzeit. Länge der Eingabe wird in unärer Kodierung gemessen
Laufzeit ist abhängig vom numerischen Wert der Eingabe

Hoffnung. ARSP/RDSP ist in pseudopolynomialzeit berechenbar

- Ja, unter fester Anzahl Szenarios
- Nein, für beliebige Anzahl Szenarios

Pseudopolynomialzeit für ARSP

Satz. ARSP ist für feste Anzahl an Szenarien in pseudopolynomialzeit berechenbar.

für allgemeine Graphen

Pseudopolynomialzeit für ARSP

Satz. ARSP ist für feste Anzahl an Szenarien in pseudopolynomialzeit berechenbar.

Idee. Dynamische Programmierung

Pseudopolynomialzeit für ARSP

Satz. ARSP ist für feste Anzahl an Szenarien in pseudopolynomialzeit berechenbar.

Idee. Dynamische Programmierung

- Ein kürzester s-t Pfad besitzt höchstens Länge $|V| - 1$
- Länge eines s-t Pfades unter einem Szenarios r ist beschränkt durch $P_r =$ Summer der $|V| - 1$ größten Kantengewichte unter r

Pseudopolynomialzeit für ARSP

Satz. ARSP ist für feste Anzahl an Szenarien in pseudopolynomialzeit berechenbar.

Idee. Dynamische Programmierung

- Ein kürzester s-t Pfad besitzt höchstens Länge $|V| - 1$
- Länge eines s-t Pfades unter einem Szenarios r ist beschränkt durch $P_r =$ Summer der $|V| - 1$ größten Kantengewichte unter r

Struktur einer optimale Lösung

Pseudopolynomialzeit für ARSP

Satz. ARSP ist für feste Anzahl an Szenarien in pseudopolynomialzeit berechenbar.

Idee. Dynamische Programmierung

- Ein kürzester s-t Pfad besitzt höchstens Länge $|V| - 1$
- Länge eines s-t Pfades unter einem Szenarios r ist beschränkt durch $P_r =$ Summer der $|V| - 1$ größten Kantengewichte unter r

Struktur einer optimale Lösung

Für alle Knoten $v_i \in V$ berechne mininmale maximale Kosten eines s-t Weges für den gilt

- Von s nach v_i werden Kosten $\alpha_1, \ldots, \alpha_{|S|}$ benötigt
- von v nach t werden höchstens $\tau \in \{1, \ldots, |V| - 1\}$ Kanten benötigt.

Pseudopolynomialzeit für ARSP

Satz. ARSP ist für feste Anzahl an Szenarien in pseudopolynomialzeit berechenbar.

Kennen wir Werte für $\tau - 1$ für alle $v \in V$, so können wir Werte für τ berechnen.

Struktur einer optimale Lösung

Für alle Knoten $v_i \in V$ berechne mininmale maximale Kosten eines s-t Weges für den gilt

- Von s nach v_i werden Kosten $\alpha_1, \ldots, \alpha_{|S|}$ benötigt
- von v nach t werden höchstens $\tau \in \{1, \ldots, |V| - 1\}$ Kanten benötigt.

Pseudopolynomialzeit für ARSP

Satz. ARSP ist für feste Anzahl an Szenarien in pseudopolynomialzeit berechenbar.

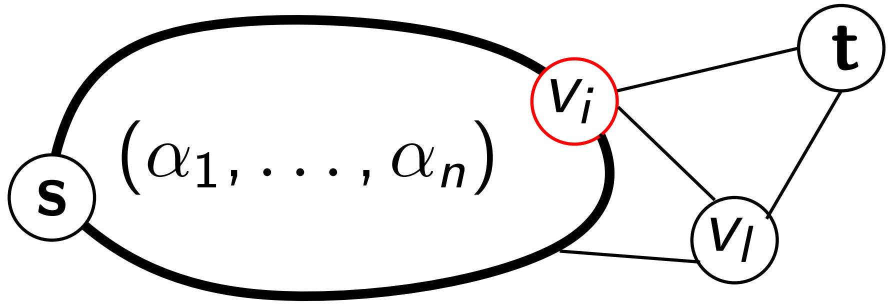
berechne für alle möglichen Werte für $\alpha_1, \ldots, \alpha_{|S|}$

Struktur einer optimale Lösung

Für alle Knoten $v_i \in V$ berechne mininmale maximale Kosten eines s-t Weges für den gilt

- Von s nach v_i werden Kosten $\alpha_1, \ldots, \alpha_{|S|}$ benötigt
- von v nach t werden höchstens $\tau \in \{1, \ldots, |V| - 1\}$ Kanten benötigt.

Pseudopolynomialzeit für ARSP

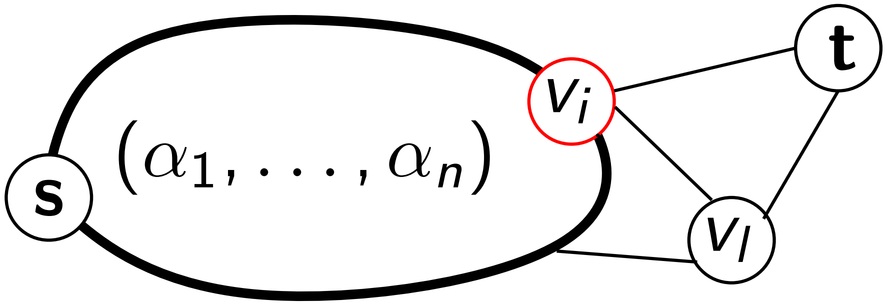


Struktur einer optimale Lösung

Für alle Knoten $v_i \in V$ berechne mininmale maximale Kosten eines s-t Weges für den gilt

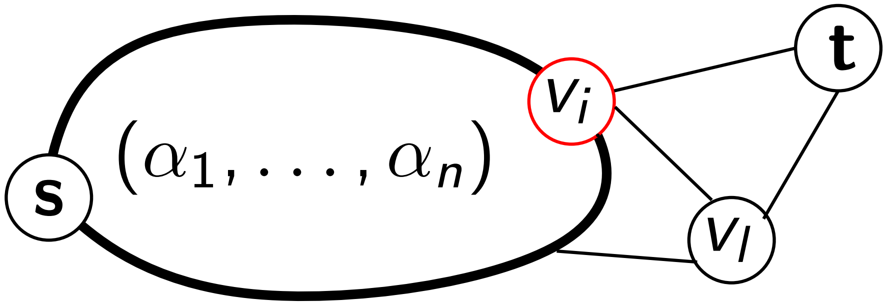
- Von s nach v_i werden Kosten $\alpha_1, \ldots, \alpha_{|S|}$ benötigt
- von v nach t werden höchstens $\tau \in \{1, \ldots, |V| - 1\}$ Kanten benötigt.

Pseudopolynomialzeit für ARSP



Wert einer optimalen Lösung rekursiv definieren

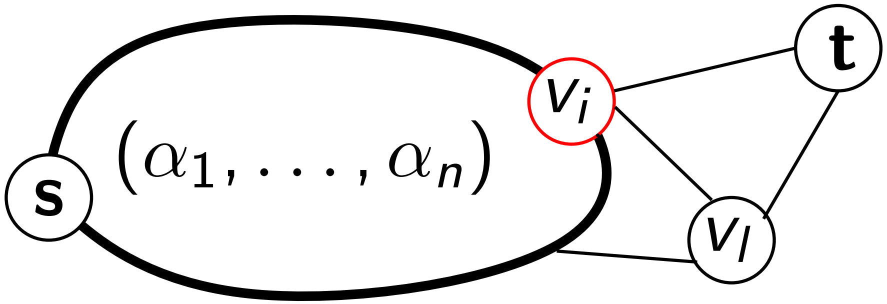
Pseudopolynomialzeit für ARSP



Wert einer optimalen Lösung rekursiv definieren

$$f^1_{v_i}(\alpha_1, \dots, \alpha_{|S|}) = \begin{cases} \max_{r \in S}\{\alpha_r + c^r_{vt}\} & \text{falls } t \in adj[v_i]\} \\ \max_{r \in S}\{\alpha_r\} & \text{falls } t = v_i\} \\ \infty & \text{sonst} \end{cases}$$

Pseudopolynomialzeit für ARSP



Wert einer optimalen Lösung rekursiv definieren

$$f_{v_i}^1(\alpha_1, \ldots, \alpha_{|S|}) = \begin{cases} \max_{r \in S}\{\alpha_r + c_{vt}^r\} & \text{falls } t \in adj[v_i]\} \\ \max_{r \in S}\{\alpha_r\} & \text{falls } t = v_i\} \\ \infty & \text{sonst} \end{cases}$$

$$f_{v_i}^\tau(\alpha_1, \ldots, \alpha_{|S|}) = \min_{v_j \in adj[v_i]}\{f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_{|S|} + c_{ij}^{|S|}), f_{v_i}^{\tau-1}(\alpha_1, \ldots, \alpha_{|S|})\}$$

für $\tau = 2, \ldots, |V| - 1$

Pseudopolynomialzeit für ARSP

Wert einer optimalen Lösung berechnen (bottom-up)

Pseudopolynomialzeit für ARSP

Wert einer optimalen Lösung berechnen (bottom-up)

RobustARSP$(G = (V, E), S = \{1, \dots, n\})$

Pseudopolynomialzeit für ARSP

Wert einer optimalen Lösung berechnen (bottom-up)

numerische
Beschränkung
berechnen

RobustARSP$(G = (V, E), S = \{1, \dots, n\})$

// Initialisierung

ber. P_r für $r \in \{1, \dots, n\}$, ber. $P_{\max} = \max_{r \in S}\{P_r\}$

Pseudopolynomialzeit für ARSP

Wert einer optimalen Lösung berechnen (bottom-up)

numerische Beschränkung berechnen

Laufzeit?

RobustARSP$(G = (V, E), S = \{1, \dots, n\})$

// Initialisierung

ber. P_r für $r \in \{1, \dots, n\}$, ber. $P_{\max} = \max_{r \in S}\{P_r\}$

Pseudopolynomialzeit für ARSP

Wert einer optimalen Lösung berechnen (bottom-up)

numerische Beschränkung berechnen

Laufzeit?
$\mathcal{O}(|E||S|)$

RobustARSP$(G = (V, E), S = \{1, \ldots, n\})$

// Initialisierung
ber. P_r für $r \in \{1, \ldots, n\}$, ber. $P_{\max} = \max_{r \in S}\{P_r\}$

Pseudopolynomialzeit für ARSP

Wert einer optimalen Lösung berechnen (bottom-up)

$$f^1_{v_i}(\alpha_1, \ldots, \alpha_{|S|}) = \begin{cases} \max_{r \in S}\{\alpha_r + c^r_{vt}\} & \text{falls } t \in adj[v_i]\} \\ \max_{r \in S}\{\alpha_r\} & \text{falls } t = v_i\} \\ \infty & \text{sonst} \end{cases}$$

RobustARSP$(G = (V, E), S = \{1, \ldots, n\})$

// Initialisierung
ber. P_r für $r \in \{1, \ldots, n\}$, ber. $P_{\max} = \max_{r \in S}\{P_r\}$
// Basis
for $\alpha_1 = 0, \ldots, P_1, \ldots,$ for $\alpha_n = 0, \ldots, P_n$
for $v_i \in V$ if $v_i \in adj[t]$: $f^1_{v_i}(\alpha_1, \ldots, \alpha_n) = \max_{r \in S}\{\alpha_r + c_{v_i t}\}$
else if $v = t$: $f^1_{v_i}(\alpha_1, \ldots, \alpha_n) = \max_{r \in S}\{\alpha_r\}$
else: $f^1_{v_i}(\alpha_1, \ldots, \alpha_n) = \infty$

Pseudopolynomialzeit für ARSP

Wert einer optimalen Lösung berechnen (bottom-up)

$$f^1_{v_i}(\alpha_1, \dots, \alpha_{|S|}) = \begin{cases} \max_{r \in S}\{\alpha_r + c^r_{vt}\} & \text{falls } t \in adj[v_i]\} \\ \max_{r \in S}\{\alpha_r\} & \text{falls } t = v_i\} \\ \infty & \text{sonst} \end{cases}$$

Laufzeit?

RobustARSP$(G = (V, E), S = \{1, \dots, n\})$

// Initialisierung

ber. P_r für $r \in \{1, \dots, n\}$, ber. $P_{\max} = \max_{r \in S}\{P_r\}$

// Basis

for $\alpha_1 = 0, \dots, P_1, \dots,$ for $\alpha_n = 0, \dots, P_n$

for $v_i \in V$ if $v_i \in adj[t]$: $f^1_{v_i}(\alpha_1, \dots, \alpha_n) = \max_{r \in S}\{\alpha_r + c_{v_i t}\}$

else if $v = t$: $f^1_{v_i}(\alpha_1, \dots, \alpha_n) = \max_{r \in S}\{\alpha_r\}$

else: $f^1_{v_i}(\alpha_1, \dots, \alpha_n) = \infty$

Pseudopolynomialzeit für ARSP

Wert einer optimalen Lösung berechnen (bottom-up)

$$f^1_{v_i}(\alpha_1, \ldots, \alpha_{|S|}) = \begin{cases} \max_{r \in S}\{\alpha_r + c^r_{vt}\} & \text{falls } t \in adj[v_i]\} \\ \max_{r \in S}\{\alpha_r\} & \text{falls } t = v_i\} \\ \infty & \text{sonst} \end{cases}$$

Laufzeit?

$\mathcal{O}(|V| P_{\max}^{|S|})$

RobustARSP$(G = (V, E), S = \{1, \ldots, n\})$

// Initialisierung

ber. P_r für $r \in \{1, \ldots, n\}$, ber. $P_{\max} = \max_{r \in S}\{P_r\}$

// Basis

for $\alpha_1 = 0, \ldots, P_1, \ldots,$ for $\alpha_n = 0, \ldots, P_n$

for $v_i \in V$ if $v_i \in adj[t]$: $f^1_{v_i}(\alpha_1, \ldots, \alpha_n) = \max_{r \in S}\{\alpha_r + c_{v_i t}\}$

else if $v = t$: $f^1_{v_i}(\alpha_1, \ldots, \alpha_n) = \max_{r \in S}\{\alpha_r\}$

else: $f^1_{v_i}(\alpha_1, \ldots, \alpha_n) = \infty$

Pseudopolynomialzeit für ARSP

Pseudopolynomialzeit für ARSP

// Rekursion

for $\tau = 2, \ldots, |V| - 1$, for $v_i \in V$

for $\alpha_1 = 0, \ldots, P^1, \ldots,$ for $\alpha_n = 0, \ldots, P^n$

$f_v^\tau(\alpha_1, \ldots, \alpha_n) = f_v^{\tau-1}(\alpha_1, \ldots, \alpha_n)$

for $v_j \in adj[v_j]$

if $\alpha_r + c_{ij}^r \leq P_r$ für $r \in S$

if $f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_n + c_{ij}^n) < f_{v_i}^\tau(\alpha_1, \ldots, \alpha_n)$

then $f_{v_i}^\tau(\alpha_1, \ldots, \alpha_n) = f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_n + c_{ij}^n)$

$$f_{v_i}^\tau(\alpha_1, \ldots, \alpha_{|S|}) = \min_{v_j \in adj[v_i]}\{f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_{|S|} + c_{ij}^{|S|}),$$
$$f_{v_i}^{\tau-1}(\alpha_1, \ldots, \alpha_{|S|})\}$$
$$\text{für } \tau = 2, \ldots, |V| - 1$$

Pseudopolynomialzeit für ARSP

// Rekursion

for $\tau = 2, \dots, |V| - 1$, for $v_i \in V$

for $\alpha_1 = 0, \dots, P^1, \dots,$ for $\alpha_n = 0, \dots, P^n$

$f_v^\tau(\alpha_1, \dots, \alpha_n) = f_v^{\tau-1}(\alpha_1, \dots, \alpha_n)$

for $v_j \in adj[v_i]$

if $\alpha_r + c_{ij}^r \leq P_r$ für $r \in S$

if $f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \dots, \alpha_n + c_{ij}^n) < f_{v_i}^\tau(\alpha_1, \dots, \alpha_n)$

then $f_{v_i}^\tau(\alpha_1, \dots, \alpha_n) = f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \dots, \alpha_n + c_{ij}^n)$

$$f_{v_i}^\tau(\alpha_1, \dots, \alpha_{|S|}) = \min_{v_j \in adj[v_i]}\{f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \dots, \alpha_{|S|} + c_{ij}^{|S|}), f_{v_i}^{\tau-1}(\alpha_1, \dots, \alpha_{|S|})\}$$

für $\tau = 2, \dots, |V| - 1$

Laufzeit?

$\mathcal{O}(|V|^3 P_{\max}^{|S|})$

Pseudopolynomialzeit für ARSP

// Rekursion

for $\tau = 2, \ldots, |V| - 1$, for $v_i \in V$

for $\alpha_1 = 0, \ldots, P^1, \ldots,$ for $\alpha_n = 0, \ldots, P^n$

$f_v^\tau(\alpha_1, \ldots, \alpha_n) = f_v^{\tau-1}(\alpha_1, \ldots, \alpha_n)$

for $v_j \in adj[v_i]$

if $\alpha_r + c_{ij}^r \leq P_r$ für $r \in S$

if $f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_n + c_{ij}^n) < f_{v_i}^\tau(\alpha_1, \ldots, \alpha_n)$

then $f_{v_i}^\tau(\alpha_1, \ldots, \alpha_n) = f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_n + c_{ij}^n)$

Laufzeit?

$\mathcal{O}(|V|^3 P_{\max}^{|S|})$

insgesamt: $\mathcal{O}(|V|^3 P_{\max}^{|S|})$

für layered Graphen ex. ein Algorithmus mit Laufzeit $\mathcal{O}(|E| L_{\max}^{|S|})$

Pseudopolynomialzeit für ARSP

// Rekursion

for $\tau = 2, \ldots, |V| - 1$, for $v_i \in V$

for $\alpha_1 = 0, \ldots, P^1, \ldots,$ for $\alpha_n = 0, \ldots, P^n$

$f_v^{\tau}(\alpha_1, \ldots, \alpha_n) = f_v^{\tau-1}(\alpha_1, \ldots, \alpha_n)$

for $v_j \in adj[v_j]$

if $\alpha_r + c_{ij}^r \leq P_r$ für $r \in S$

if $f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_n + c_{ij}^n) < f_{v_i}^{\tau}(\alpha_1, \ldots, \alpha_n)$

then $f_{v_i}^{\tau}(\alpha_1, \ldots, \alpha_n) = f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_n + c_{ij}^n)$

Optimale Lösung berechnen

Pseudopolynomialzeit für ARSP

// Rekursion

for $\tau = 2, \ldots, |V| - 1$, for $v_i \in V$

for $\alpha_1 = 0, \ldots, P^1, \ldots$, for $\alpha_n = 0, \ldots, P^n$

$f_v^{\tau}(\alpha_1, \ldots, \alpha_n) = f_v^{\tau-1}(\alpha_1, \ldots, \alpha_n)$

for $v_j \in adj[v_i]$

if $\alpha_r + c_{ij}^r \leq P_r$ für $r \in S$

if $f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_n + c_{ij}^n) < f_{v_i}^{\tau}(\alpha_1, \ldots, \alpha_n)$

then $f_{v_i}^{\tau}(\alpha_1, \ldots, \alpha_n) = f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_n + c_{ij}^n)$

// Ausgabe

return $f_s^{|V|-1}(0, \ldots, 0)$

Optimale Lösung berechnen

Pseudopolynomialzeit für ARSP

// Rekursion

for $\tau = 2, \ldots, |V| - 1$, for $v_i \in V$

for $\alpha_1 = 0, \ldots, P^1, \ldots,$ for $\alpha_n = 0, \ldots, P^n$

$f_v^\tau(\alpha_1, \ldots, \alpha_n) = f_v^{\tau-1}(\alpha_1, \ldots, \alpha_n)$

for $v_j \in adj[v_i]$

if $\alpha_r + c_{ij}^r \leq P_r$ für $r \in S$

if $f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_n + c_{ij}^n) < f_{v_i}^\tau(\alpha_1, \ldots, \alpha_n)$

then $f_{v_i}^\tau(\alpha_1, \ldots, \alpha_n) = f_{v_j}^{\tau-1}(\alpha_1 + c_{ij}^1, \ldots, \alpha_n + c_{ij}^n)$

// Ausgabe

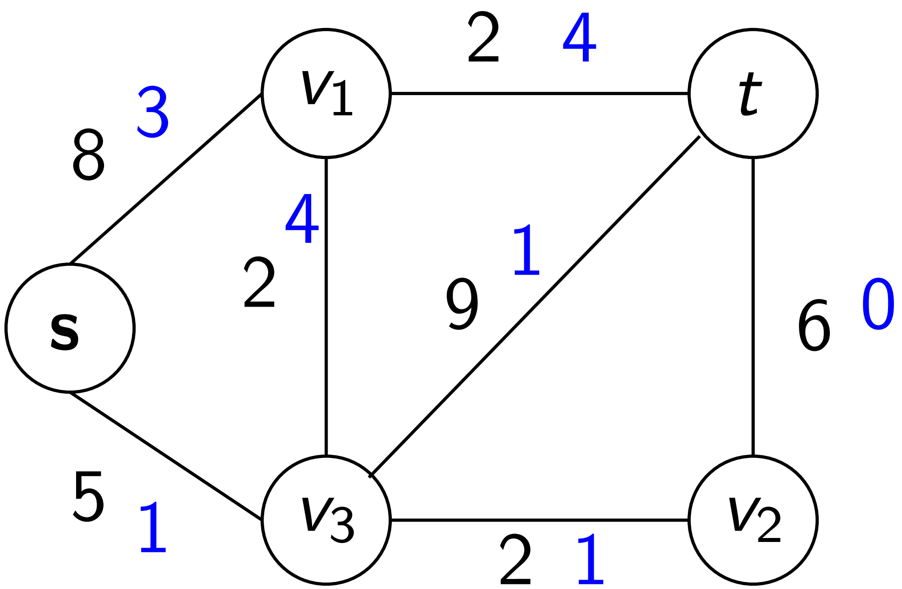
return $f_s^{|V|-1}(0, \ldots, 0)$

Optimale Lösung berechnen

"Vorgänger" merken

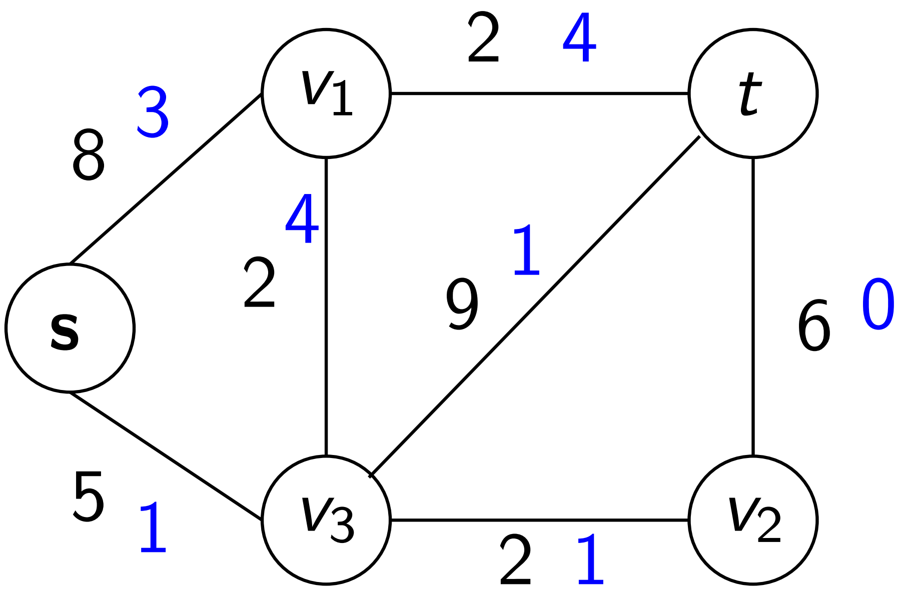
Pseudopolynomialzeit für ARSP

Beispiel



Pseudopolynomialzeit für ARSP

Beispiel



$|V| - 1 =$

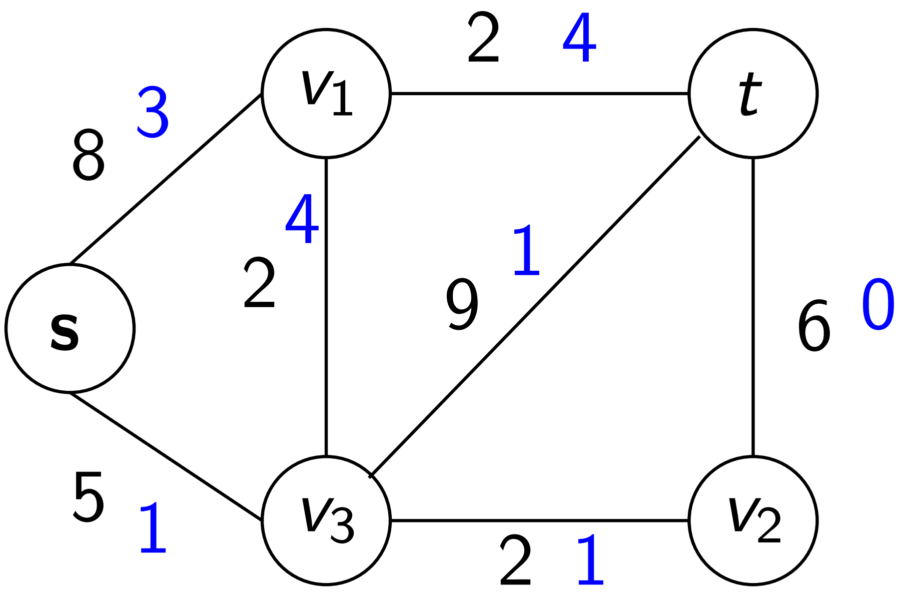
$P_1 =$

$P_2 =$

$P_{\max} =$

Pseudopolynomialzeit für ARSP

Beispiel



$|V| - 1 = 4$

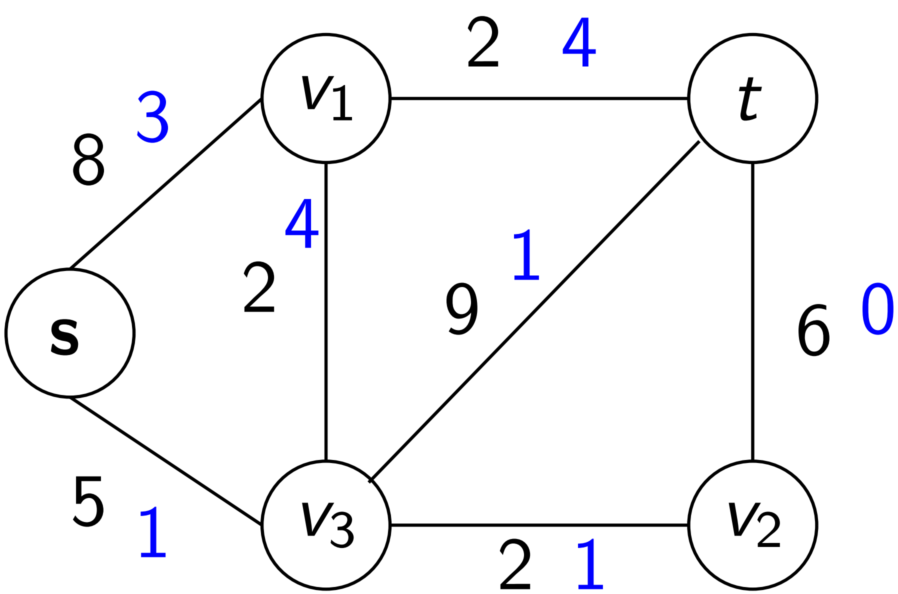
$P_1 = 28$

$P_2 = 12$

$P_{\max} = 28$

Pseudopolynomialzeit für ARSP

Beispiel



$$|V| - 1 = 4$$

$$P_1 = 28$$

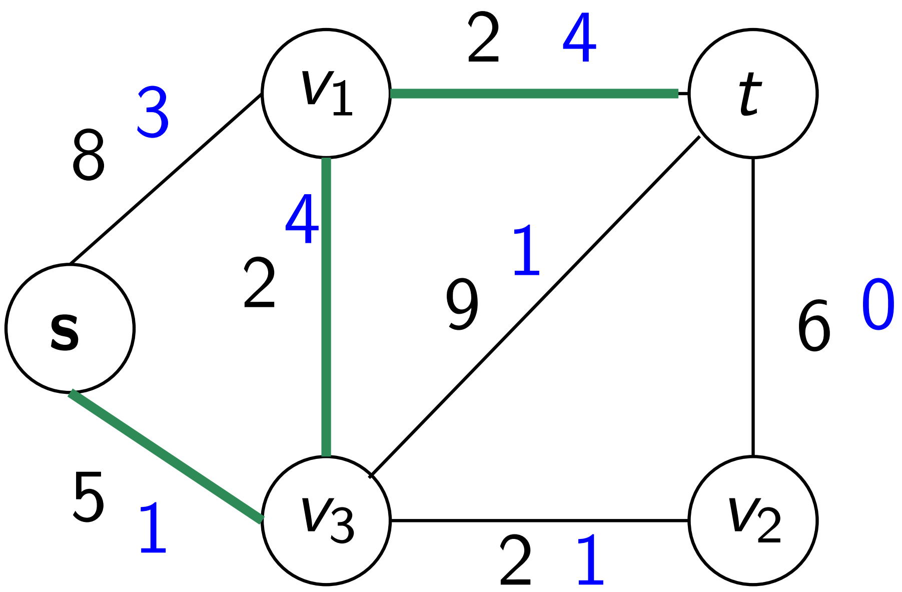
$$P_2 = 12$$

$$P_{\max} = 28$$

$$5 \cdot 4 \cdot 28 \cdot 12 = 6720 \text{ Werte}$$

Pseudopolynomialzeit für ARSP

Beispiel



$$|V| - 1 = 4$$

$$P_1 = 28$$

$$P_2 = 12$$

$$P_{\max} = 28$$

Pseudopolynomialzeit für RDSP

Satz. RDSP ist für feste Anzahl an Szenarien in pseudopolynomialzeit berechenbar.

Pseudopolynomialzeit für RDSP

Satz. RDSP ist für feste Anzahl an Szenarien in pseudopolynomialzeit berechenbar.

Idee. Dynamische Programmierung

- wie bei ARSP mit Veränderungen

  es wird minimale maximale Abweichung von optimalen Kosten unter einem Szenario gesucht

Pseudopolynomialzeit für RDSP

Satz. RDSP ist für feste Anzahl an Szenarien in pseudopolynomialzeit berechenbar.

Idee. Dynamische Programmierung

- wie bei ARSP mit Veränderungen

  es wird minimale maximale Abweichung von optimalen Kosten unter einem Szenario gesucht

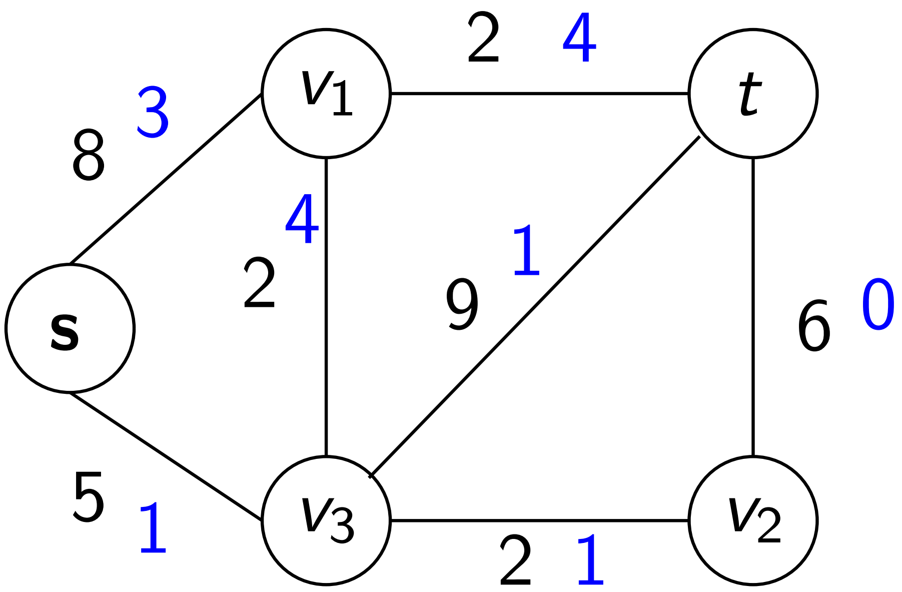
  berechne kürzesten Pfad z^r unter jedem Szenario $r \in S$

  berechne Teilinstanzen für $\alpha_i = -z^i, \dots, P_i - z^i$

  gib $f_s^{|V|-1}(0, \dots, 0)$ aus

Pseudopolynomialzeit für RDSP

Beispiel



$|V| - 1 = 4$

$P_1 = 28$

$P_2 = 12$

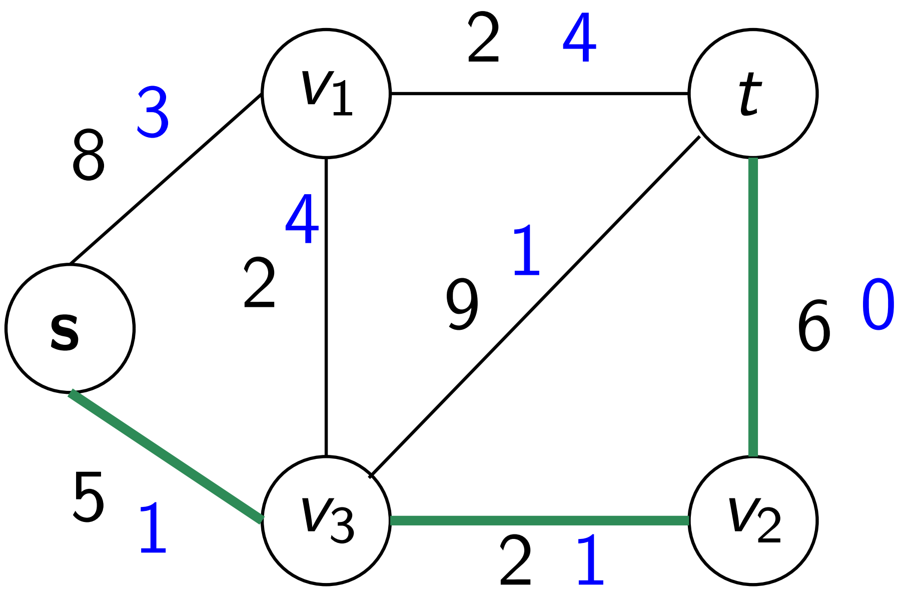
$P_{\max} = 28$

$z^1 = 2$

$z^2 = 9$

Pseudopolynomialzeit für RDSP

Beispiel



$$|V| - 1 = 4$$

$$P_1 = 28$$

$$P_2 = 12$$

$$P_{\max} = 28$$

$$z^1 = 2$$

$$z^2 = 9$$

… und für beliebig viele Szenarien

Pseudopolynomialzeitalgo: $\mathcal{O}(|V|^3 (G_{\max})^{|S|})$

… und für beliebig viele Szenarien

Pseudopolynomialzeitalgo: $\mathcal{O}(|V|^3(G_{\max})^{|S|})$!

- Ist $|S|$ abhängig von der Eingabe, so ist dies keine polynomiale Laufzeit mehr

… und für beliebig viele Szenarien

Pseudopolynomialzeitalgo: $\mathcal{O}(|V|^3(G_{\max})^{|S|})$

- Ist $|S|$ abhängig von der Eingabe, so ist dies keine polynomiale Laufzeit mehr

Ein NP-vollständiges (hartes) Problem A ist …

… und für beliebig viele Szenarien

Pseudopolynomialzeitalgo: $\mathcal{O}(|V|^3(G_{\max})^{|S|})$

- Ist $|S|$ abhängig von der Eingabe, so ist dies keine polynomiale Laufzeit mehr

Ein NP-vollständiges (hartes) Problem A ist …

weakly NP-vollständig(hart)

strongly NP-vollständig(hart)

... und für beliebig viele Szenarien

Pseudopolynomialzeitalgo: $\mathcal{O}(|V|^3(G_{\max})^{|S|})$

- Ist $|S|$ abhängig von der Eingabe, so ist dies keine polynomiale Laufzeit mehr

Ein NP-vollständiges (hartes) Problem A ist ...

weakly NP-vollständig(hart)

falls A unär kodiert in P liegt

strongly NP-vollständig(hart)

... und für beliebig viele Szenarien

Pseudopolynomialzeitalgo: $\mathcal{O}(|V|^3(G_{\max})^{|S|})$

!

- Ist $|S|$ abhängig von der Eingabe, so ist dies keine polynomiale Laufzeit mehr

Ein NP-vollständiges (hartes) Problem A ist ...

weakly NP-vollständig(hart)	strongly NP-vollständig(hart)
falls A unär kodiert in P liegt	falls A unär kodiert NP-vollständig(hart) ist

… und für beliebig viele Szenarien

Satz. ARSP/RDSP ist für beliebige Anzahl Szenarien NP-hart

Ein NP-vollständiges (hartes) Problem A ist …

weakly NP-vollständig(hart)	strongly NP-vollständig(hart)
falls A unär kodiert in P liegt	falls A unär kodiert NP-vollständig(hart) ist

... und für beliebig viele Szenarien

Satz. ARSP/RDSP ist für beliebige Anzahl Szenarien NP-hart

Fakt. 3PART ist strongly NP-hart

$$\begin{aligned}3\text{PART} = \ & \{\langle a_1, \ldots, a_{3m}, B\rangle \mid \\ & a_1, \ldots, a_m, m, B \in \mathbb{N} \wedge \forall i \in \{1, \ldots, 3m\} : \tfrac{B}{4} < a_i < \tfrac{B}{2} \wedge \\ & \exists \text{disjunkte } I_1, \ldots, I_m \subseteq \{1, ...m\} \text{ mit } \forall j \in \{1, \ldots, m\} \textstyle\sum_{k \in I_j} a_k = B\}\end{aligned}$$

Ein NP-vollständiges (hartes) Problem A ist ...

weakly NP-vollständig(hart)	strongly NP-vollständig(hart)
falls A unär kodiert in P liegt	falls A unär kodiert NP-vollständig(hart) ist

Heuristik für ARSP/RDSP

Hoffnung Heuristiken

- gesucht sind Faustregeln

- gute Lösungen schnell erzeugen

Surrogation-Heuristik

Heuristik für ARSP/RDSP

Surrogation Heuristik für ARSP Sei $G = (V, E)$ ein ungerichteter Graph und S eine Menge an Szenarien

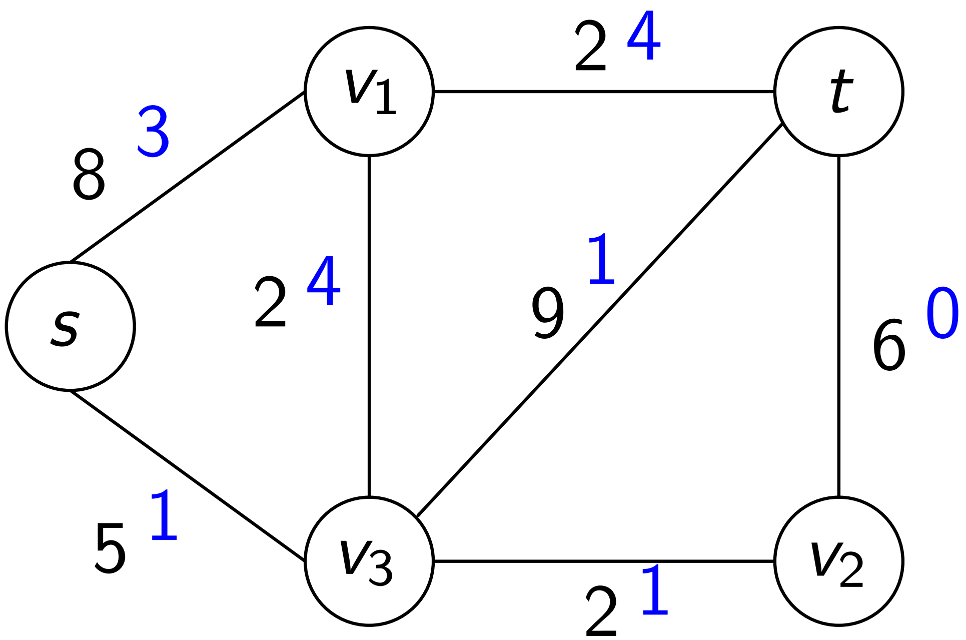
Idee:

1. Initialisierung
   neue Kantengewichte: $\forall e \in E : \overline{c}_e = \frac{1}{|S|} \sum_{r \in S} c_e^r$
2. Lösen einer Instanz: **Dijkstra**
   Sei π der gelieferte Pfad mit Surrogationswert z_L
3. Ausgabe einer Lösung: Gib π aus mit heuristischen Pfadkosten
   $z_A^H = \max_{r \in S} c_r(\pi)$

Heuristik für ARSP/RDSP

Surrogation Heuristik für ARSP

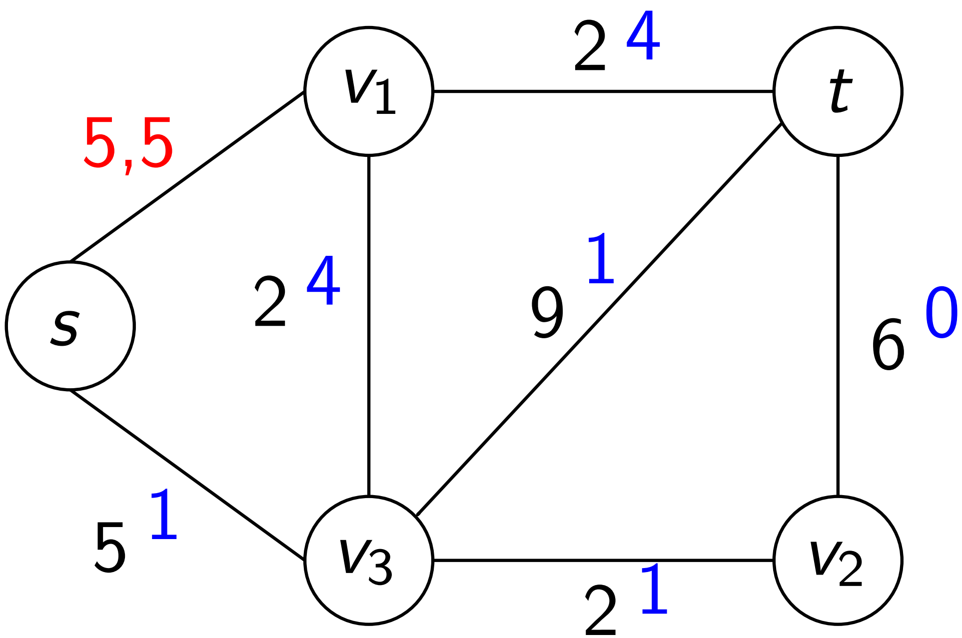
Beispiel.



Heuristik für ARSP/RDSP

Surrogation Heuristik für ARSP

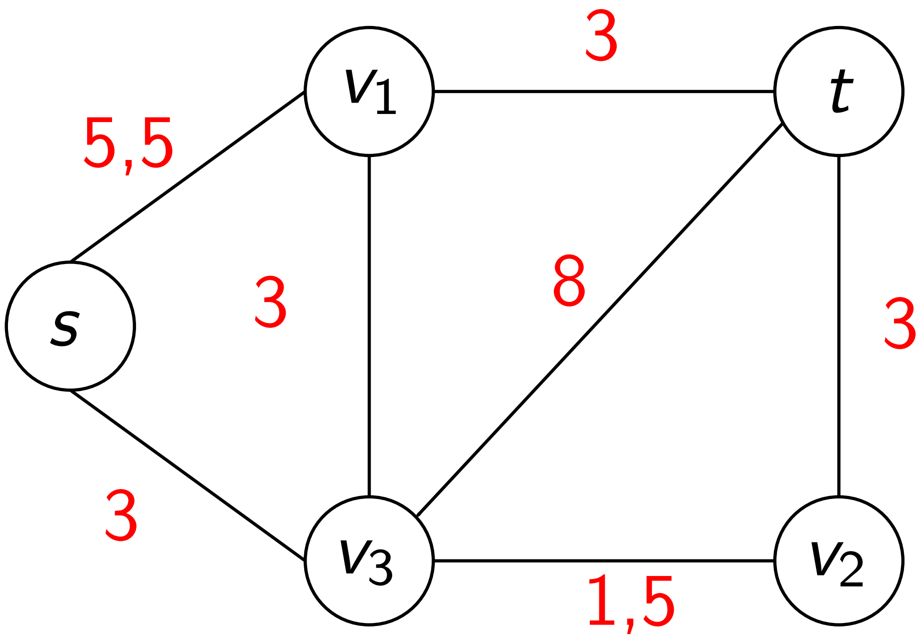
Beispiel.



Heuristik für ARSP/RDSP

Surrogation Heuristik für ARSP

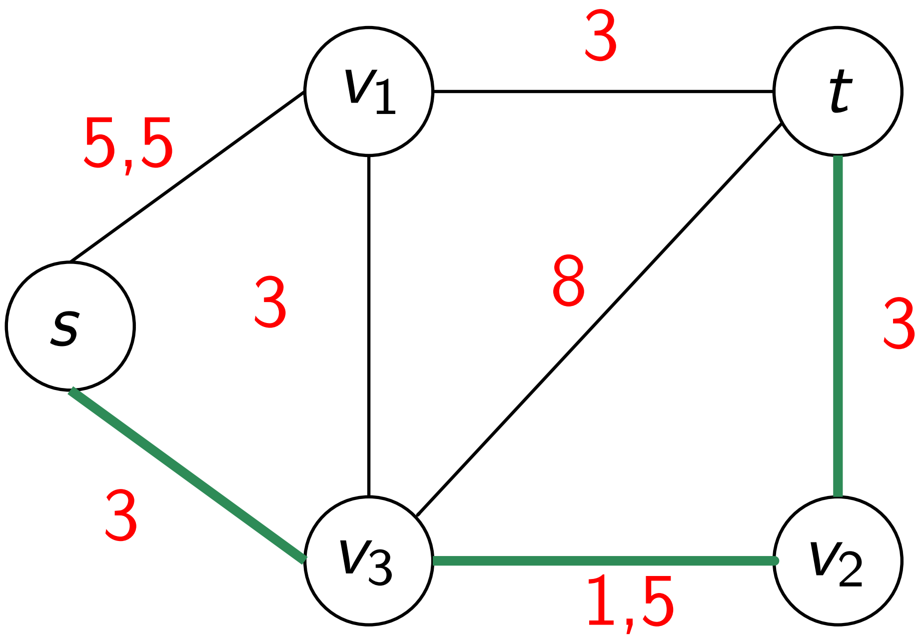
Beispiel.



Heuristik für ARSP/RDSP

Surrogation Heuristik für ARSP

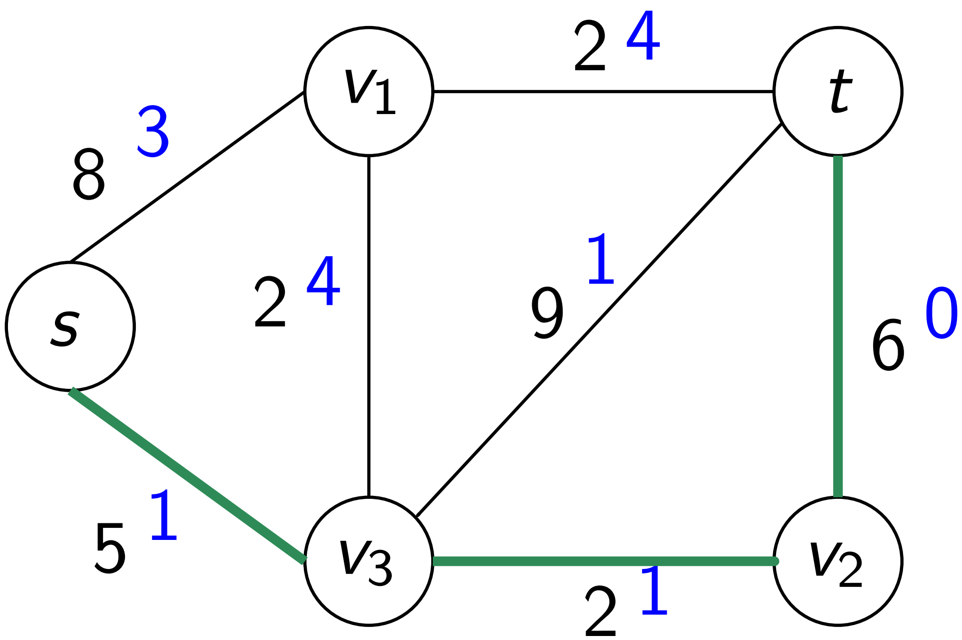
Beispiel.



Heuristik für ARSP/RDSP

Surrogation Heuristik für ARSP

Beispiel.



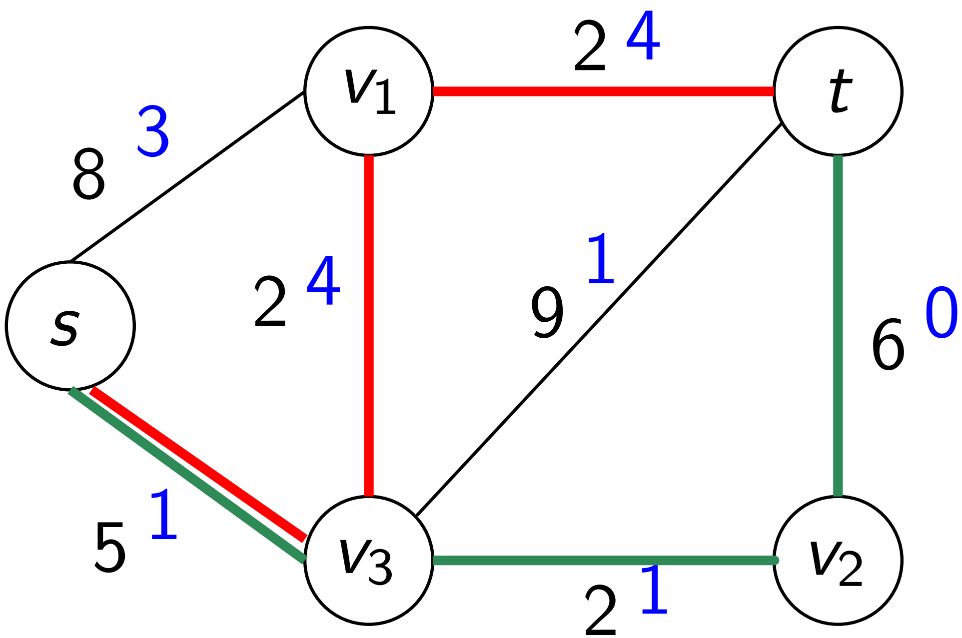
Heuristik ARSP

$z_A^H = 13$
$\pi^H = (s, v_3, v_2, t)$

Heuristik für ARSP/RDSP

Surrogation Heuristik für ARSP

Beispiel.



Heuristik ARSP

$z_A^H = 13$

$\pi^H = (s, v_3, v_2, t)$

ARSP

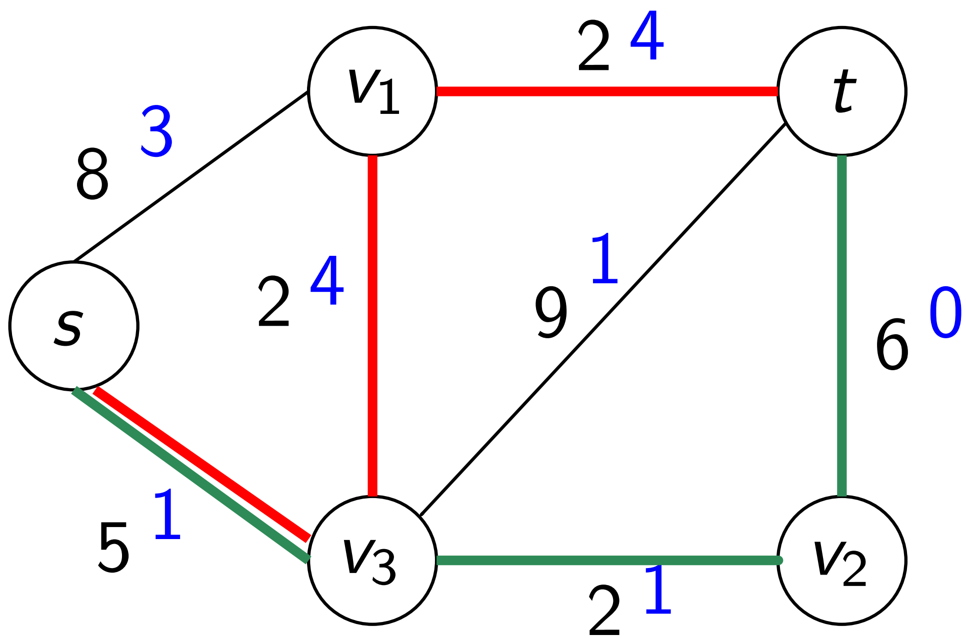
$z_A = 9$

$\pi = (s, v_3, t)$

Heuristik für ARSP/RDSP

Surrogation Heuristik für ARSP

Beispiel.



Wie gut ist Heuristik?

$$\frac{z_A^H}{z_A} \leq \frac{\theta |S|}{\theta + |S| - 1}$$

mit $\theta = \frac{\max_{r \in S} r(\pi)}{\min_{r \in S} r(\pi)}$

Heuristik ARSP

$z_A^H = 13$

$\pi^H = (s, v_3, v_2, t)$

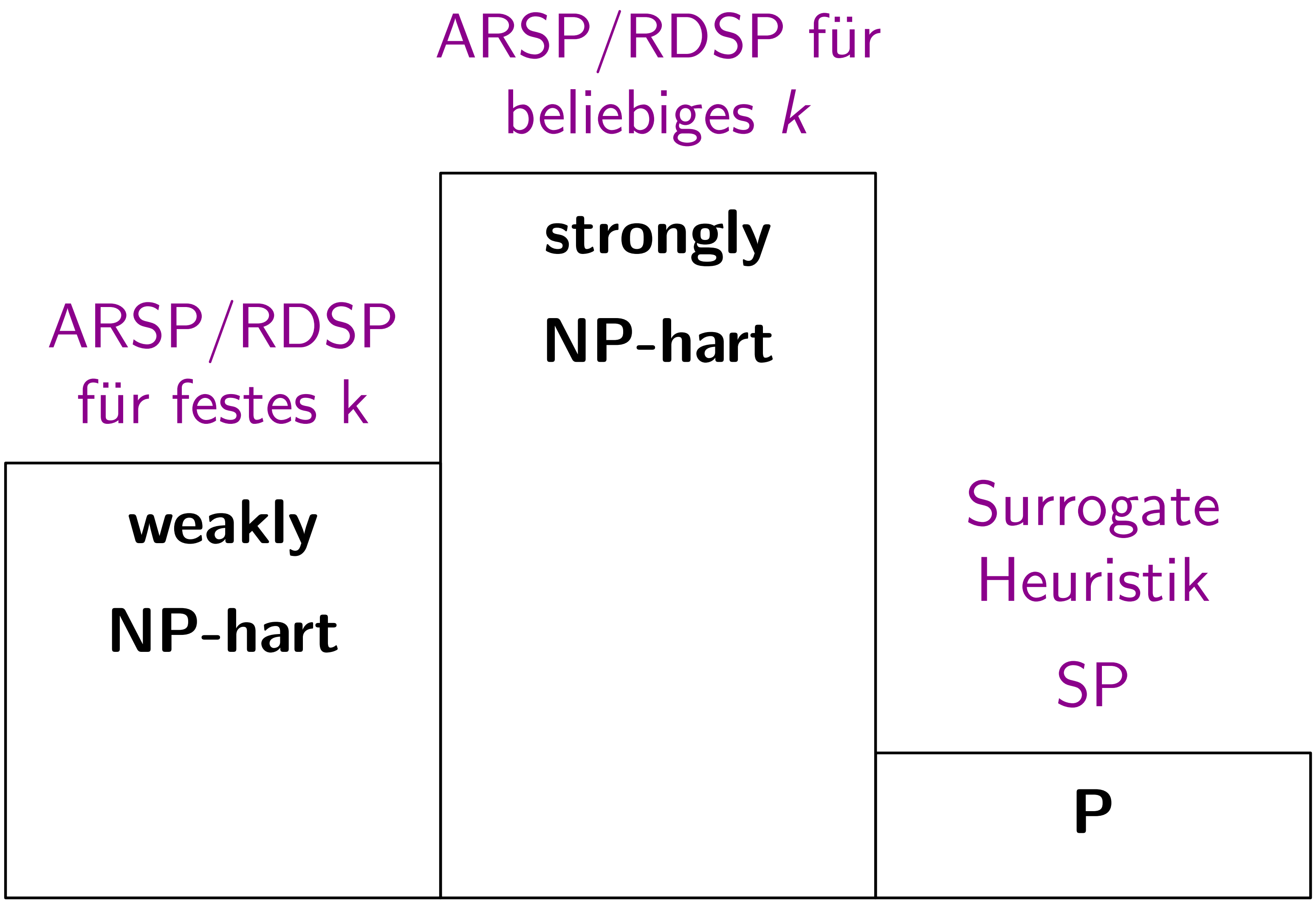
ARSP

$z_A = 9$

$\pi = (s, v_3, t)$

Fazit

Unsicherheit erschwert das Problem
sei k die Anzahl der Szenarien



Vielen Dank fürs Zuhören!

Quellen

- On the robust shortest path problem
  Gang Yu and Jian Yang
- "Strong" NP-Completeness Results: Motivation, Examples, and Implications
  M.R. Garey and D.S. Johnson

Grundlagen aus den Vorlesungen:

- Algorithmen und Datenstrukturen - Prof. Wolff
- Theoretische Informatik - Prof. Claßer