Finding the K Shortest (Hyper-)Paths in a Hypergraph

(aka Synthesis Planning)

DM840 - 2022 - Week 45

Department of Mathematics and Computer Science
University of Southern Denmark

November 08, 2022

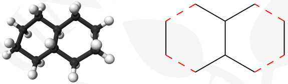
Synthesis Planning

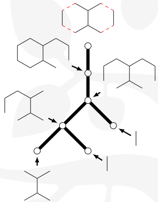
Retrosynthetic method[1]:

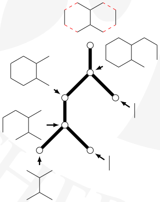
- Bondset
  - Stage 1: Choose bondset
  - **Stage 2: Choose plan for fixing bonds in bondset**
- Types of reactions:
  - **Construction: affixations & cyclizations**
  - Functionalizations

  Plan = sequencing of bond fixations = **unary-binary tree**
- Cost measure for plans
  - External Path Length (EPL)
  - Total Weight of Starting Materials (W)

Decaline







[1] Systematic Synthesis Design. 6. Yield Analysis and Convergency, Hendrickson, James B., 1977

Synthesis Planning

Stage 2: Choose plan for bondset

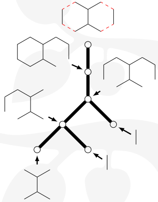
One bondset represents many plans.

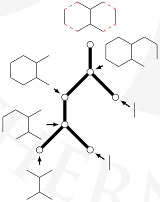
Questions:

- How do we choose the best? Known[2].
- Is one answer enough?

Here: $\boxed{\text{How to compute the } K \text{ best plans}}$

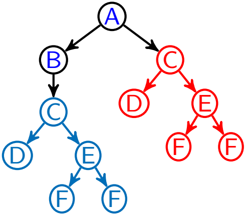
Decaline



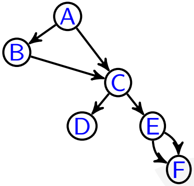


[2] Computational complexity of synthetic chemistry – Basic facts, Smith, Warren D, 1997

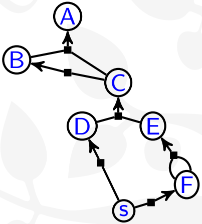
Synthesis Plans as Hypergraphs



Unary-Binary tree



DAG



Hypergraph

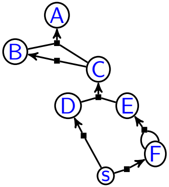
(Actually, the hypergraph above is a hyper**path**)

Hyperpath in Hypergraph[3]

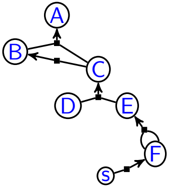
Definition

A *hyperpath* $\pi_{st} = (V_\pi, E_\pi)$ from a source vertex s to a target vertex t is a subhypergraph of H with the following properties: E_π can be ordered in a sequence $\langle e_1, e_2, ..., e_q \rangle$ such that

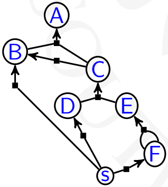
1. $T(e_i) \subseteq \{s\} \cup \{H(e_1), H(e_2), ..., H(e_{i-1})\}$ for all i
2. $t = H(e_q)$
3. Every $v \in V_\pi \setminus \{t\}$ has at least one outgoing hyperarc in E_π
4. Every $v \in V_\pi \setminus \{s\}$ has exactly one ingoing hyperarc in E_π



Hyperpath



Not hyperpath



Not hyperpath

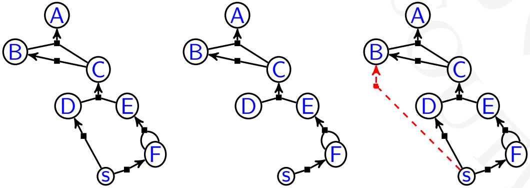
[3] variation of Ausiello, G., Franciosa, P., Frigioni, D., TCS (2001)

Hyperpath in Hypergraph[3]

Definition

A *hyperpath* $\pi_{st} = (V_\pi, E_\pi)$ from a source vertex s to a target vertex t is a subhypergraph of H with the following properties: E_π can be ordered in a sequence $\langle e_1, e_2, ..., e_q \rangle$ such that

1. $T(e_i) \subseteq \{s\} \cup \{H(e_1), H(e_2), ..., H(e_{i-1})\}$ for all i
2. $t = H(e_q)$
3. Every $v \in V_\pi \setminus \{t\}$ has at least one outgoing hyperarc in E_π
4. Every $v \in V_\pi \setminus \{s\}$ has exactly one ingoing hyperarc in E_π



Hyperpath Not hyperpath Not hyperpath

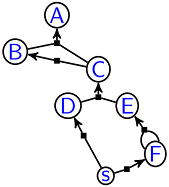
[3]variation of Ausiello, G., Franciosa, P., Frigioni, D., TCS (2001)

Hyperpath in Hypergraph[3]

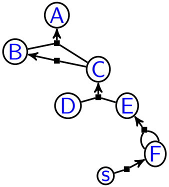
Definition

A *hyperpath* $\pi_{st} = (V_\pi, E_\pi)$ from a source vertex s to a target vertex t is a subhypergraph of H with the following properties: E_π can be ordered in a sequence $\langle e_1, e_2, ..., e_q \rangle$ such that

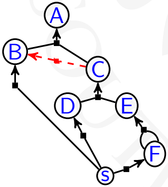
1. $T(e_i) \subseteq \{s\} \cup \{H(e_1), H(e_2), ..., H(e_{i-1})\}$ for all i
2. $t = H(e_q)$
3. Every $v \in V_\pi \setminus \{t\}$ has at least one outgoing hyperarc in E_π
4. Every $v \in V_\pi \setminus \{s\}$ has exactly one ingoing hyperarc in E_π



Hyperpath



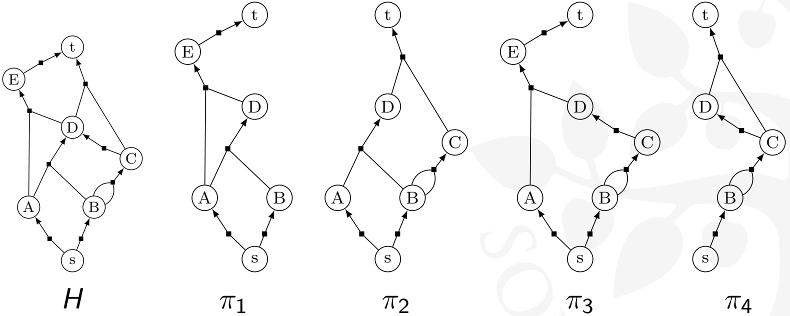
Not hyperpath



Not hyperpath

[3] variation of Ausiello, G., Franciosa, P., Frigioni, D., TCS (2001)

Hyperpaths in Hypergraph, Example



Hypergraph of Synthesis Plans (HoSP)

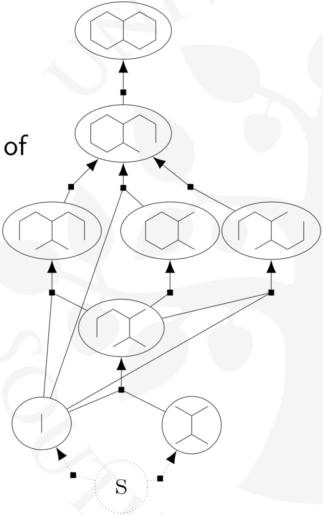
Definition

R: Finite set of reactions

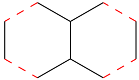
S: Set of starting materials

Let E_R be the representation of R as a set of hyperarcs. Let V_R be the set of vertices appearing in the heads and tails of the hyperarcs in E_R. The *hypergraph of synthesis plans* (*HoSP*) is the hypergraph

$$H = (V_R \cup \{s\}, E_R \cup E_s)$$



Decaline

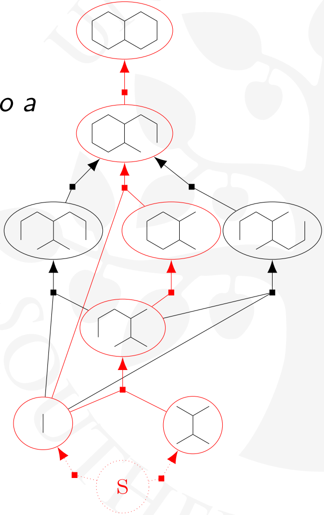


Properties of the HoSP

Lemma

Let H be a HoSP. Then any hyperpath π_{sv} from s to any other vertex v corresponds to a synthesis plan for v.

HoSP is acyclic $\Rightarrow$ topological sorting of vertices exist.



“Cost Functions” – External Path Length

$$\begin{aligned} \pi \quad & st\text{-hyperpath.} \\ \mathcal{S} \quad & \text{set potential starting materials.} \\ i \in \mathcal{S} \cap \pi \quad & \text{starting material of } \pi. \\ P_{it} = (i, e_1, v_1, e_2, v_2, ..., e_{|P_{it}|}, t) \quad & \text{simple } it\text{-path th in } \pi. \end{aligned}$$

$$\mathrm{EPL}_\pi = \sum_i \sum_{P_{it} \in \pi} |P_{it}|, \tag{1}$$

Cost Functions - Total Weight of Starting Materials

π	*st*-hyperpath.
$\mathcal{S}$	set potential starting materials.
$i \in \mathcal{S} \cap \pi$	starting material of π.
$P_{it} = (i, e_1, v_1, e_2, v_2, ..., e_{\lvert P_{it} \rvert}, t)$	simple *it*-path in π.
$r_{v,e}$	retro yield

$$\mathrm{W}_\pi = \sum_i \sum_{P_{it} \in \pi} \prod_{j=0}^{\lvert P_{it} \rvert} r_{v_j, e_{j+1}} \tag{2}$$

UNIVERSITYOFSOUTHERNDENMARK.DK

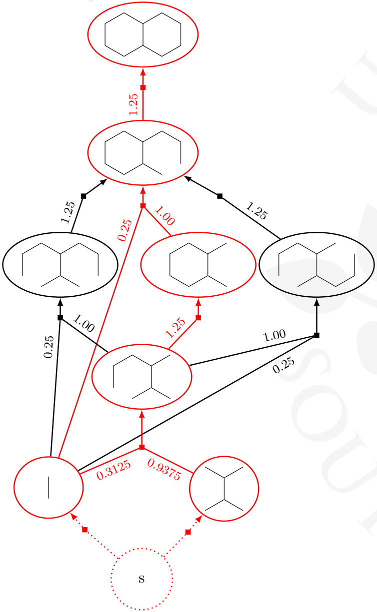
Cost Functions - Total Weight of Starting Materials

$$\begin{array}{rl} \pi & st\text{-hyperpath.} \\ \mathcal{S} & \text{set potential starting materials.} \\ i \in \mathcal{S} \cap \pi & \text{starting material of } \pi. \\ P_{it} = (i, e_1, v_1, e_2, v_2, ..., e_{|P_{it}|}, t) & \text{simple } it\text{-path in } \pi. \\ r_{v,e} & \text{retro yield} \end{array}$$

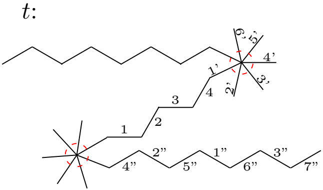
$$W_\pi = \sum_{i} \sum_{P_{it} \in \pi} \prod_{j=0}^{|P_{it}|} r_{v_j, e_{j+1}} \tag{2}$$

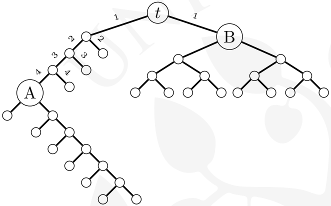
$$W(u) = \begin{cases} 0 & \text{if } u = s \\ 1 & \text{if } u \in \mathcal{S} \\ \sum\limits_{v \in T(p(u))} r_{v,p(u)} W(v) & \text{otherwise} \end{cases} \tag{3}$$

Cost Functions : Example



The Problem with EPL

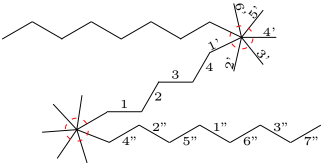


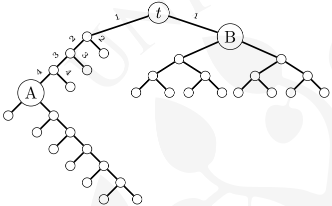


EPL=96

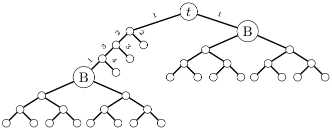
The Problem with EPL

t:

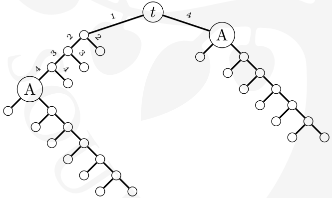




EPL=96



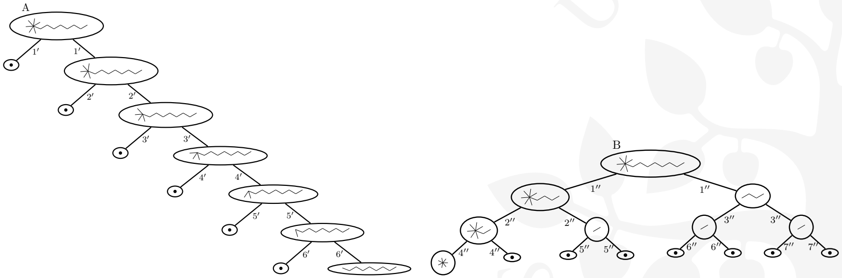
EPL=97



EPL=98

EPL is not *additive*

The Problem with EPL



Yens Algorithm [4]

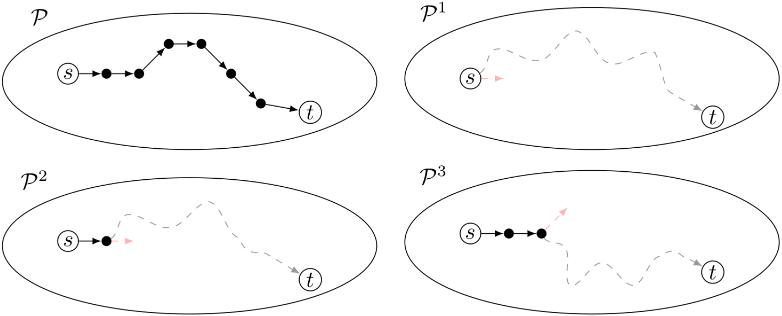
K shortest simple paths from s to t in a (standard) directed graph

[4] Finding the k Shortest Loopless Paths in a Network, Yen, Jin Y, 1971

Yens Algorithm [4]

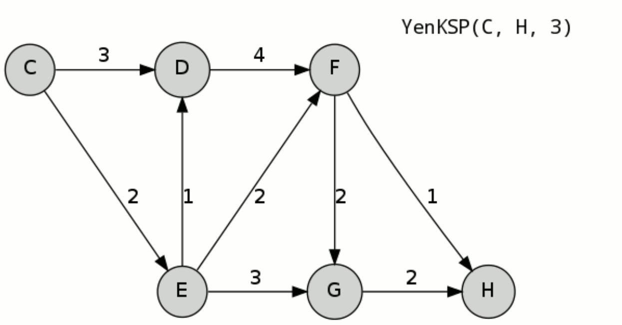
K shortest simple paths from s to t in a (standard) directed graph

$\mathcal{P} = \{P | P \text{ is a path from } s \text{ to } t\}$
$P' \in \mathcal{P}$ is the best.
Partition $\mathcal{P} \setminus P'$ into $|P'|$ disjoint sets, creating $|P'|$ shortest path problems.

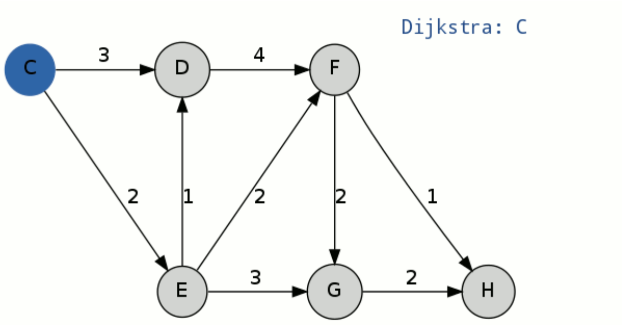


[4] Finding the k Shortest Loopless Paths in a Network, Yen, Jin Y, 1971

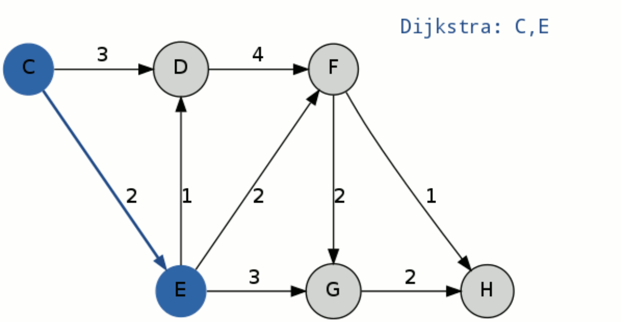
Yens Algorithm



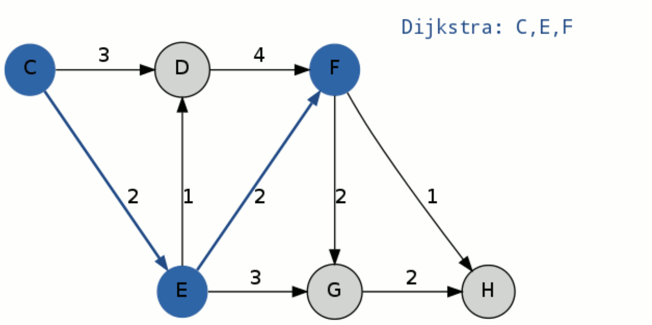
Yens Algorithm



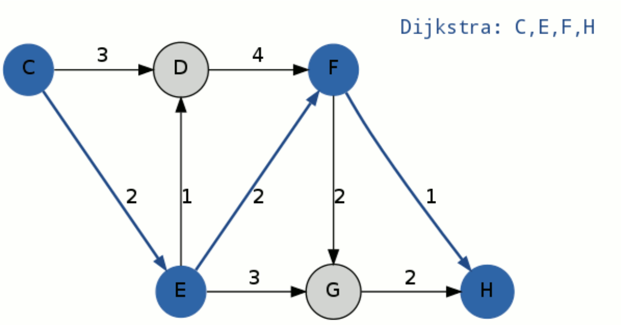
Yens Algorithm



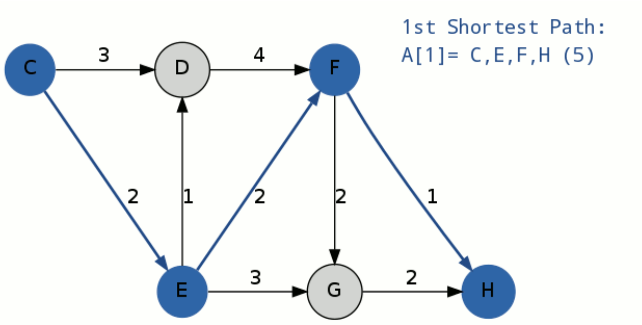
Yens Algorithm



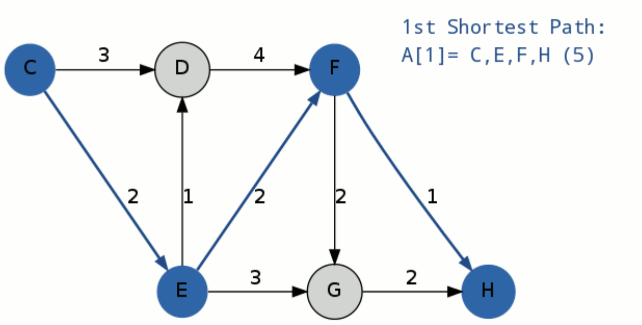
Yens Algorithm



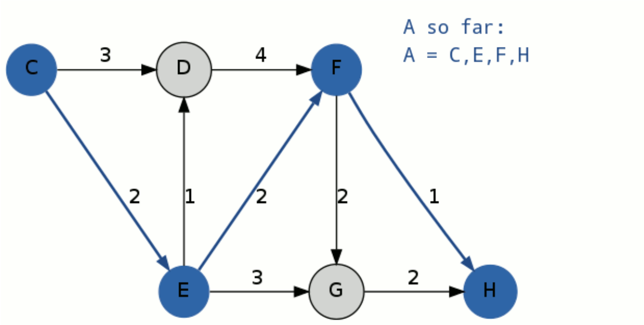
Yens Algorithm



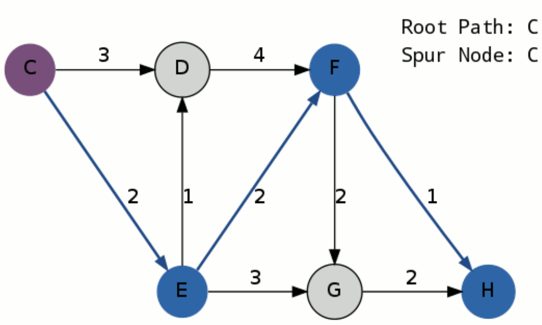
Yens Algorithm



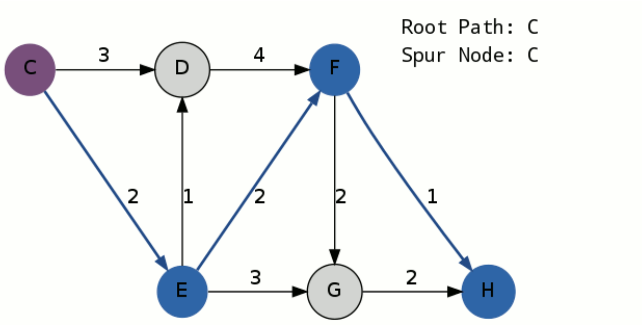
Yens Algorithm



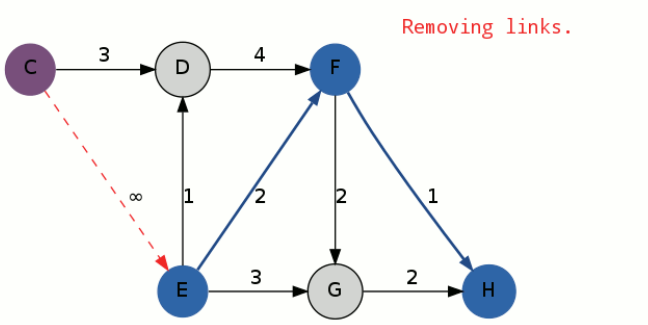
Yens Algorithm



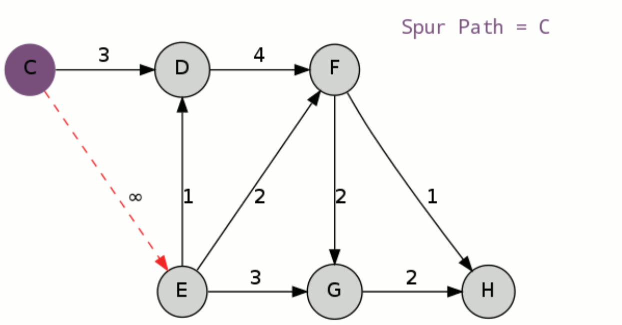
Yens Algorithm



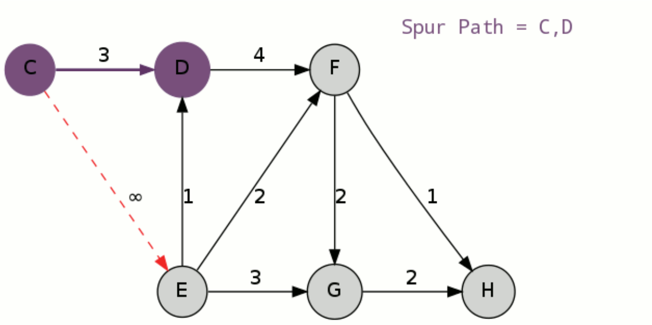
Yens Algorithm



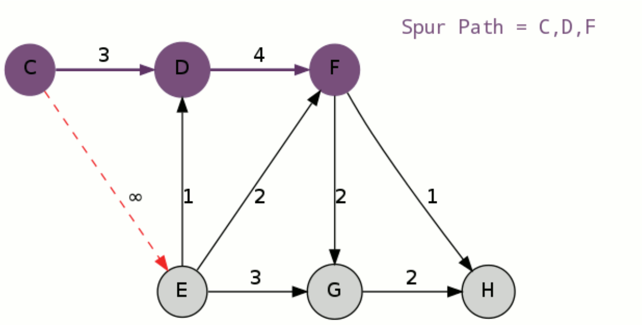
Yens Algorithm



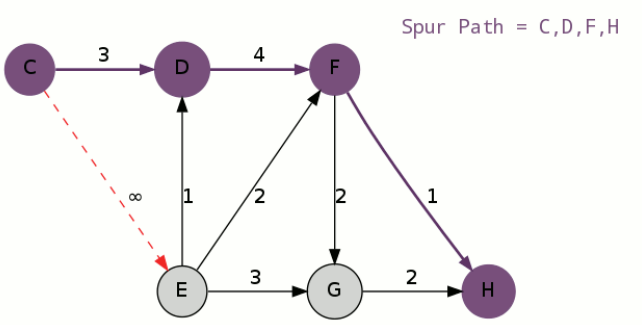
Yens Algorithm



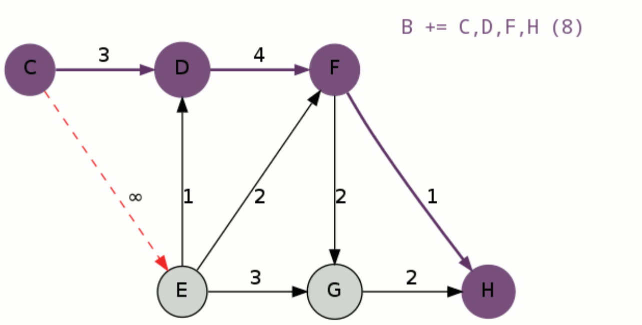
Yens Algorithm



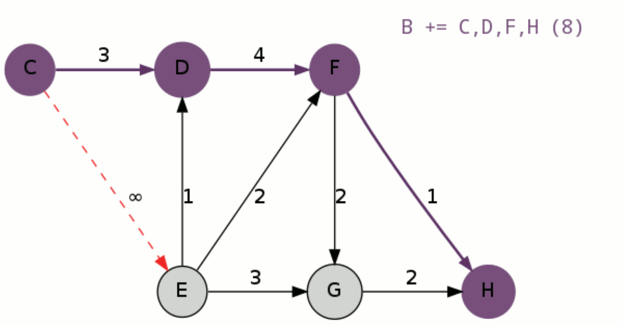
Yens Algorithm



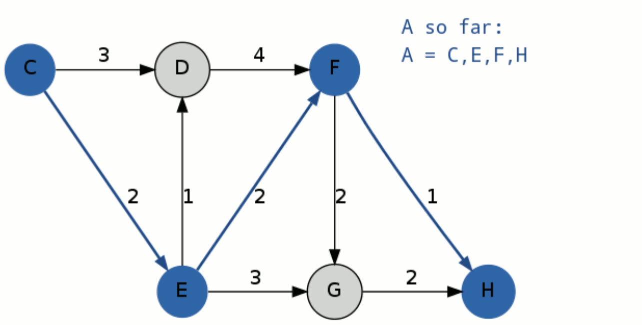
Yens Algorithm



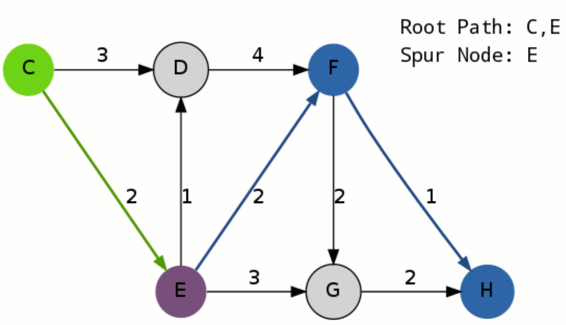
Yens Algorithm



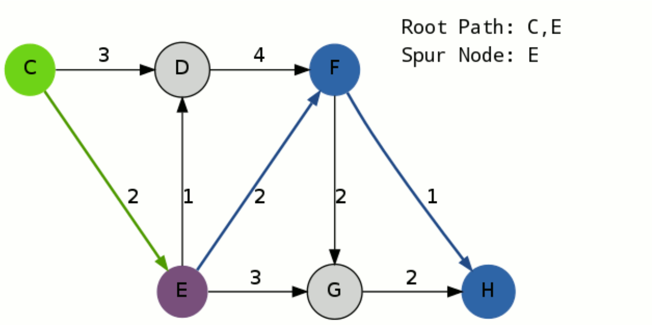
Yens Algorithm



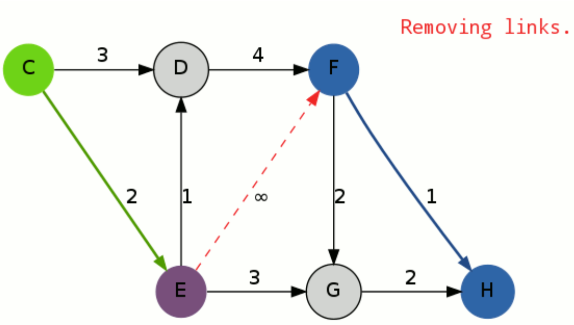
Yens Algorithm



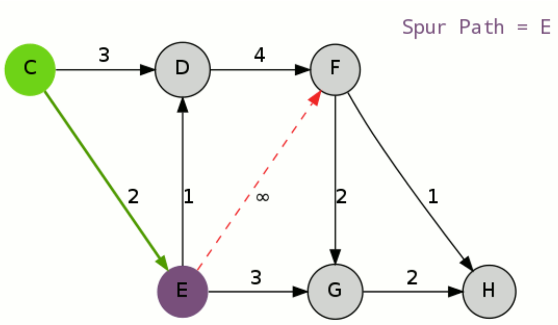
Yens Algorithm



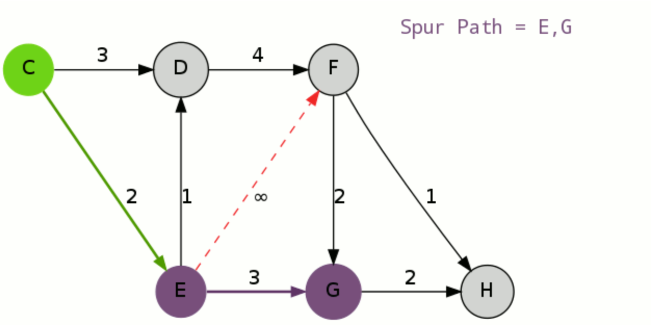
Yens Algorithm



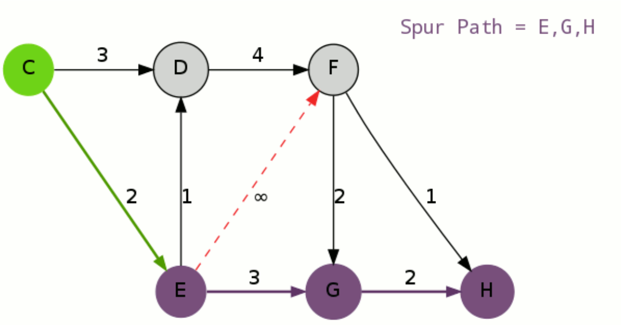
Yens Algorithm



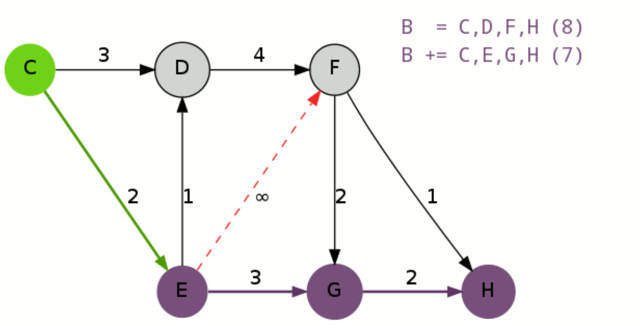
Yens Algorithm



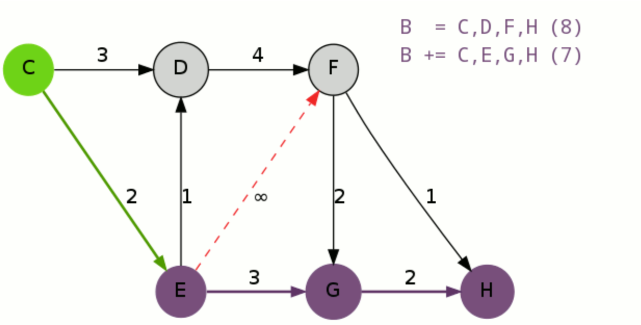
Yens Algorithm



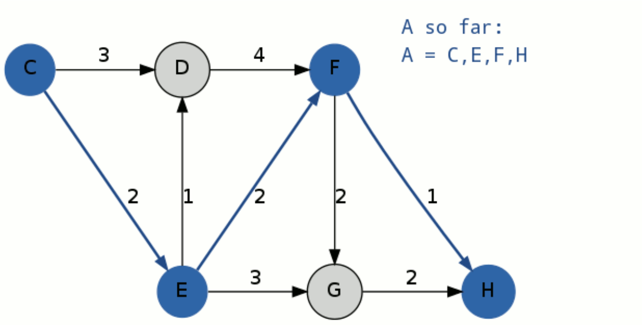
Yens Algorithm



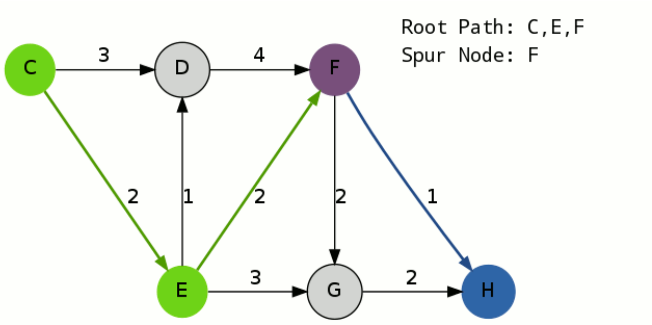
Yens Algorithm



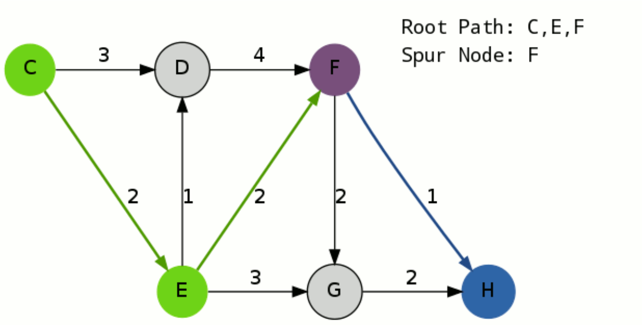
Yens Algorithm



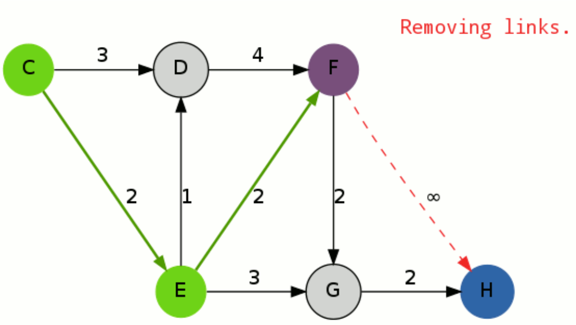
Yens Algorithm



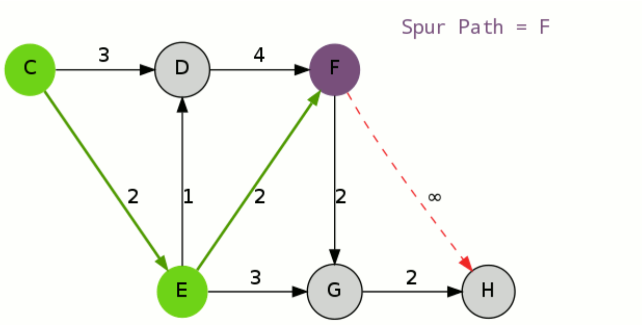
Yens Algorithm



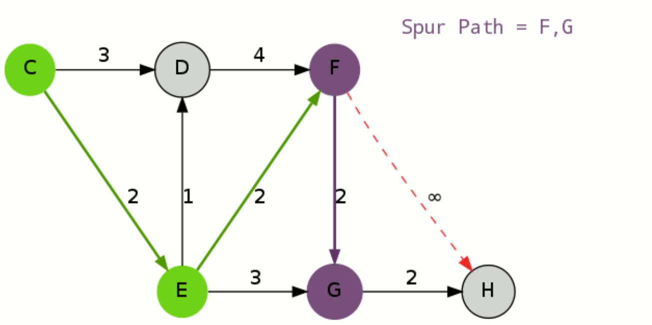
Yens Algorithm



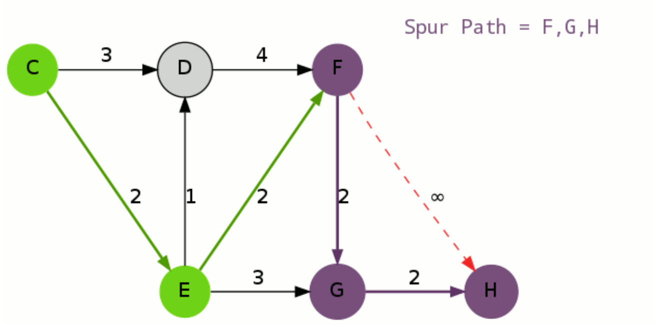
Yens Algorithm



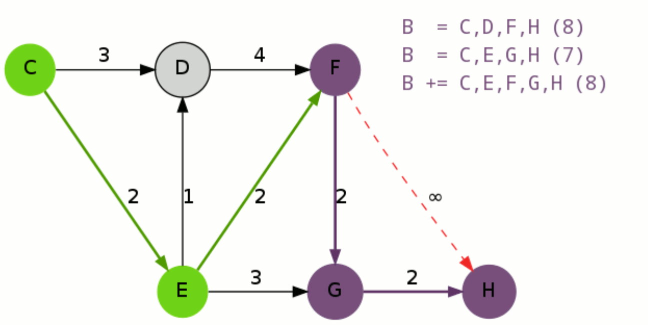
Yens Algorithm



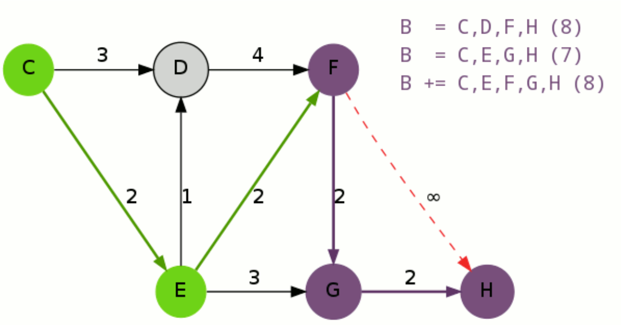
Yens Algorithm



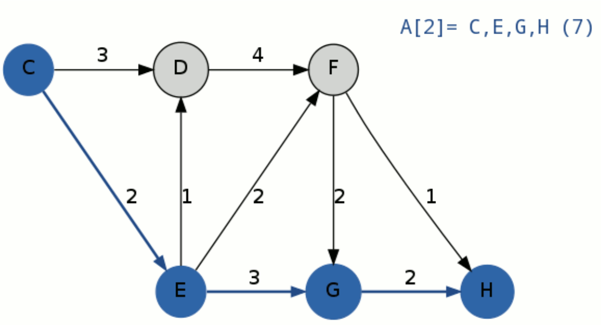
Yens Algorithm



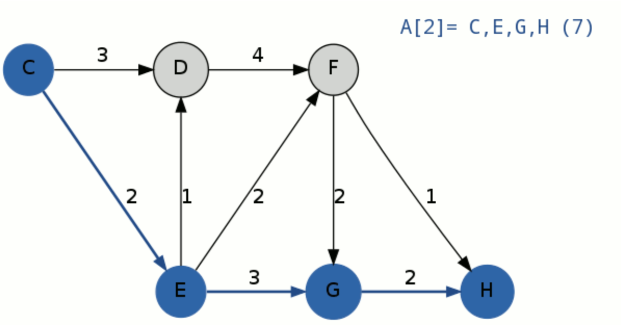
Yens Algorithm



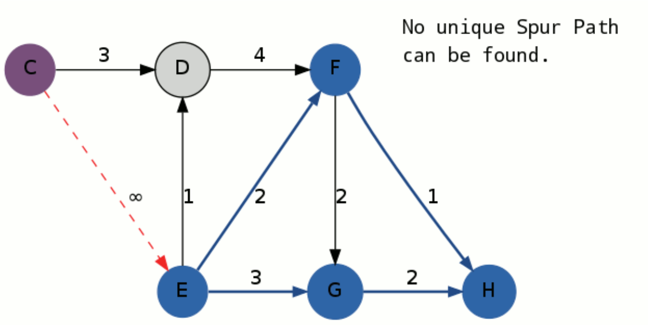
Yens Algorithm



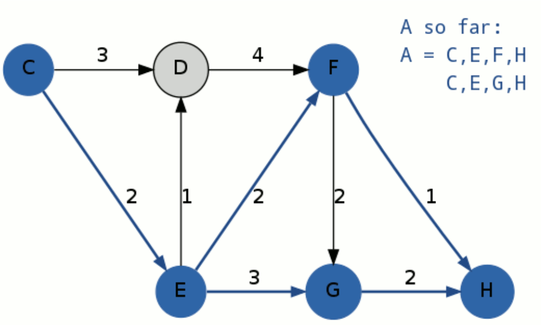
Yens Algorithm



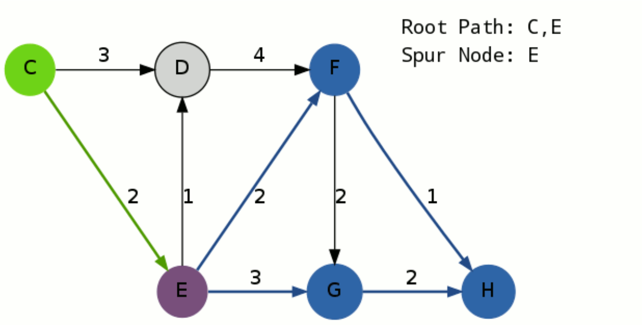
Yens Algorithm



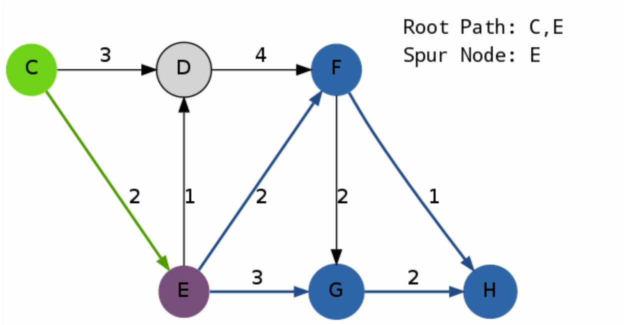
Yens Algorithm



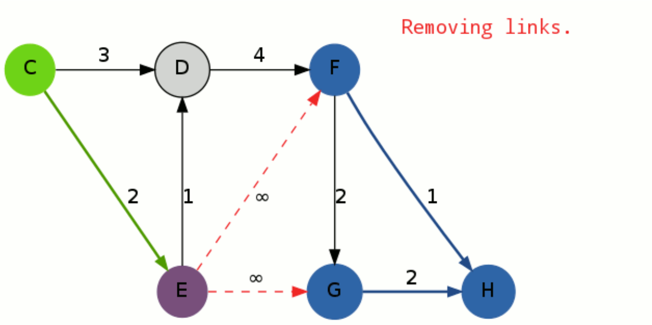
Yens Algorithm



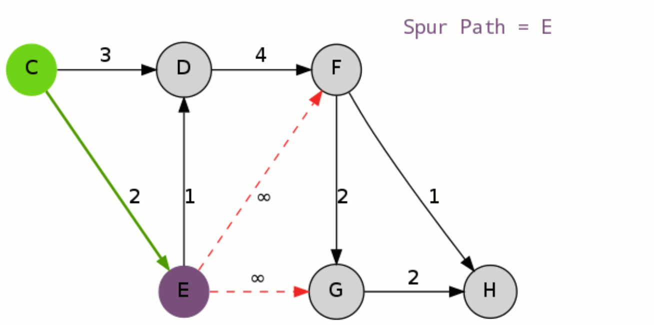
Yens Algorithm



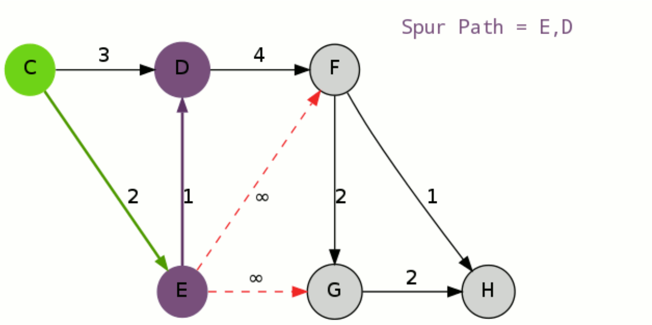
Yens Algorithm



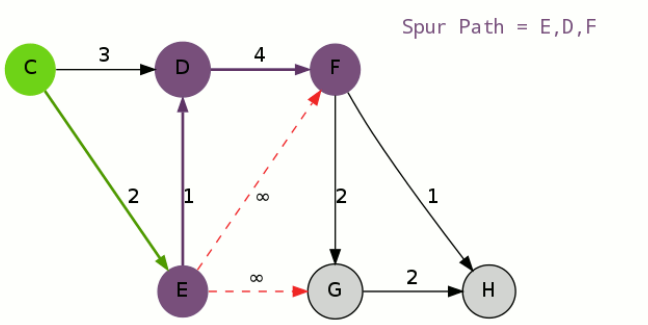
Yens Algorithm



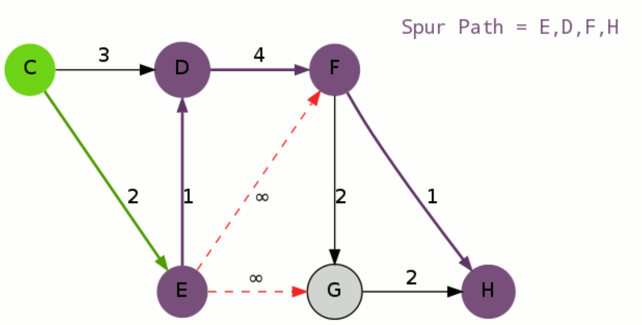
Yens Algorithm



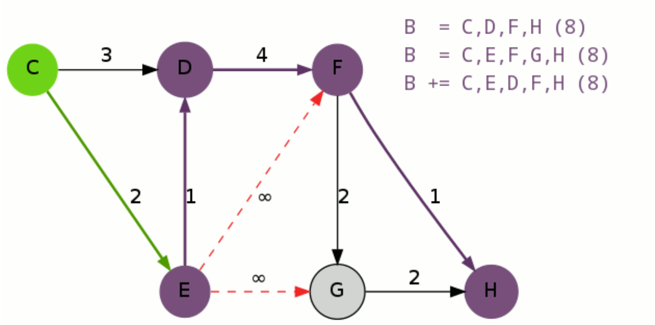
Yens Algorithm



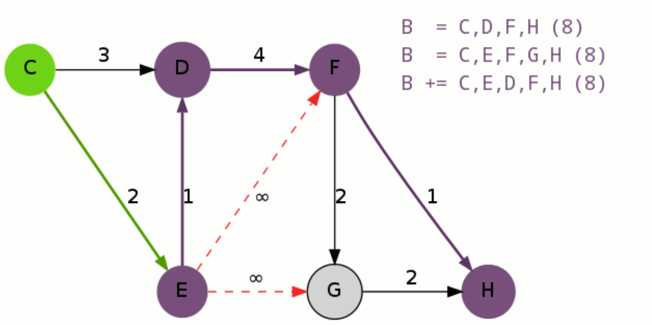
Yens Algorithm



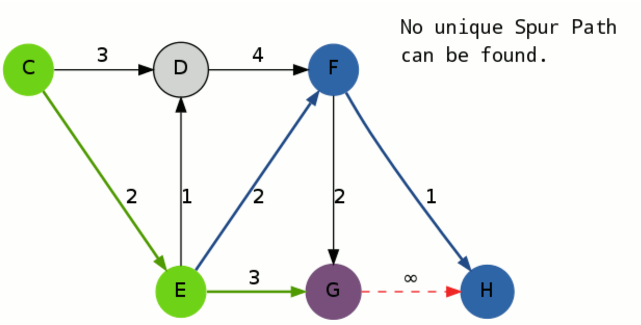
Yens Algorithm



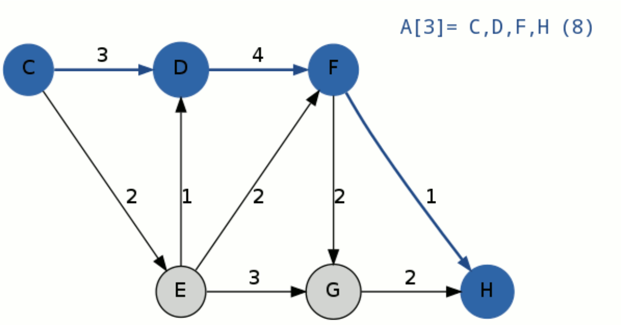
Yens Algorithm



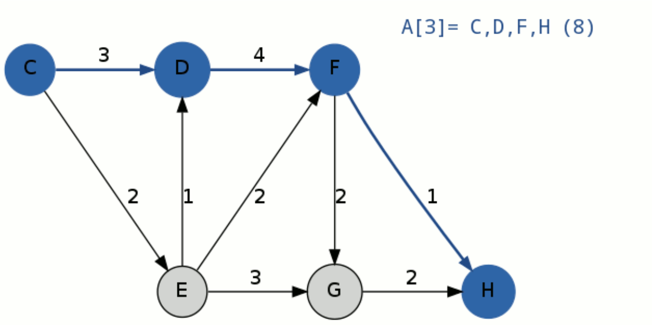
Yens Algorithm



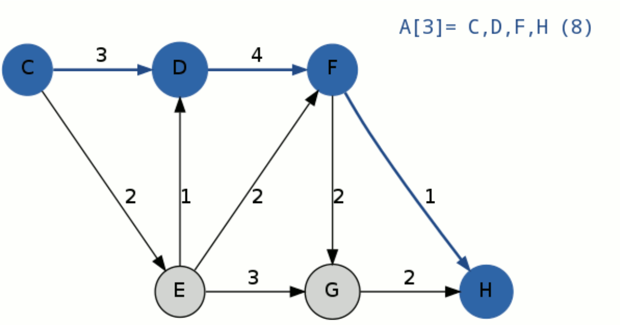
Yens Algorithm



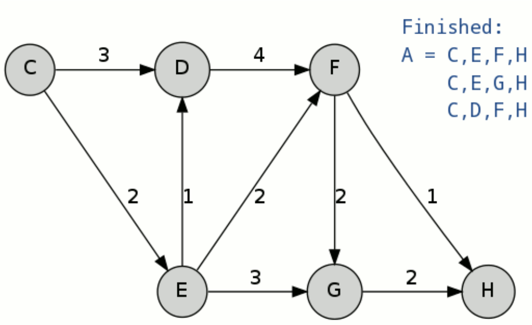
Yens Algorithm



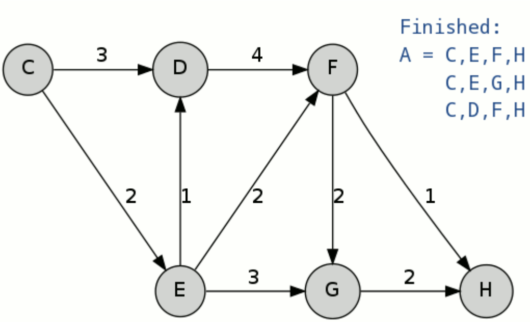
Yens Algorithm



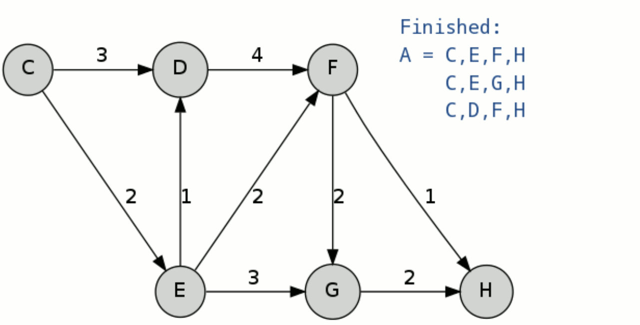
Yens Algorithm



Yens Algorithm



Yens Algorithm



Yens Algorithm

- Implementation details omitted
- Runtime : dominated by $K \cdot |V|$ Dijkstra calls
  (i.e., in any of the K iterations/partitionings max "length of current path" many Dijkstras (i.e., max. $|V|$ many Dijkstras))

K Shortest Hyperpaths Algorithm[5]

Setup:	$H = (V, E)$	directed hypergraph
	$s, t \in V$	s is hyperconnected to t.
	$\mathcal{P} =$	$\{\pi \mid \pi$ is a hyperpath from s to $t\}$
	$\pi_{st} \in \mathcal{P}$	is the shortest.

π_{st} :

$$\left.\begin{array}{l} \text{Topological ordering} \\ \quad (s, u_1, ..., u_{q-2}, u_{q-1}, u_t) \\ \text{Predecessor function} \\ \quad p : V_\pi \to E_\pi \end{array}\right\} \Rightarrow (p(u_1), p(u_2), ..., p(u_{q-1}), p(t))$$

[5] Finding the K shortest hyperpaths, Nielsen et. al., 2005

K Shortest Hyperpaths

Setup:
$$\begin{aligned} \mathcal{P} = &\quad \{\pi | \pi \text{ is a hyperpath from } s \text{ to } t\} \\ \pi_{st} \in \mathcal{P} &\quad \text{is the shortest.} \\ E_{\pi_{st}} = &\quad (p(u_1), p(u_2), ..., p(u_{q-1}), p(t)) \end{aligned}$$

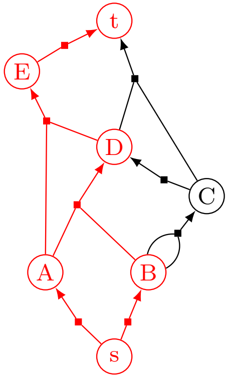
Partition $\mathcal{P} \setminus \{\pi_{st}\}$ to $\mathcal{P}^i, 1 \leq i \leq q$ s.t.:

$$\pi \in \mathcal{P}^i$$
$$\Updownarrow$$
$$E_\pi = (e_1, e_2, ..., e_{m-1}, e_m, p(u_{i+1}), ..., p(u_{q-2}), p(u_{q-1}), p(t)),$$
$$e_m \neq p(u_i)$$

K Shortest Hyperpaths

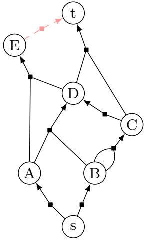
1st iteration

$L = \{(H, \pi_1)\}$

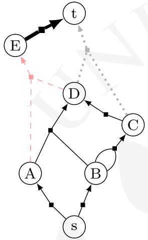


$H : \{\pi_1, \pi_2, \pi_3, \pi_4\}$

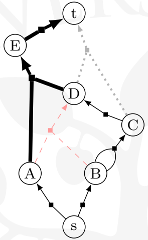
π_1 is red



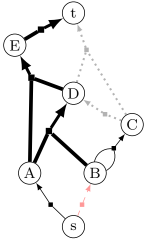
$H^5 : \{\pi_2, \pi_4\}$



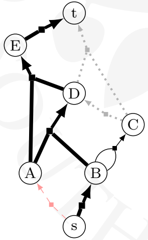
$H^4 : \emptyset$



$H^3 : \{\pi_3\}$



$H^2 : \emptyset$

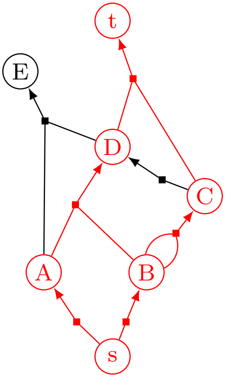


$H^1 : \emptyset$

K Shortest Hyperpaths

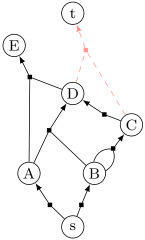
2nd iteration

$L = \{(H^5, \pi_2), (H^3, \pi_3)\}$

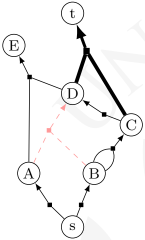


H^5 : $\{\pi_2, \pi_4\}$

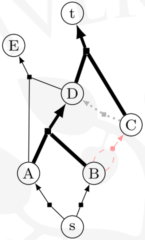
π_2 is red



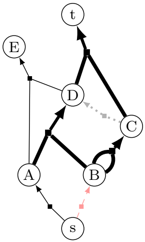
H^{55} : $\emptyset$



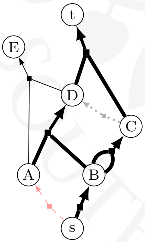
H^{54} : $\{\pi_4\}$



H^{53} : $\emptyset$



H^{52} : $\emptyset$



H^{51} : $\emptyset$

Backwards Branching

$\text{Back-Branch}(\widetilde{H}, \widetilde{\pi})$

$\mathcal{B} = \emptyset$
for $i = 1$ **to** q
    Let $\widetilde{H}^i$ be a new hypergraph
    $\widetilde{H}^i.V = \widetilde{H}.V$
    // Remove hyperarc from of $\widetilde{H}$
    $\widetilde{H}^i.E = \widetilde{H}.E \setminus \{\widetilde{\pi}.p(u_i)\}$
    // Fix back tree
    for $j = i + 1$ **to** q
        $\widetilde{H}^i.\mathrm{BS}(u_j) = \{\widetilde{\pi}.p(u_j)\}$
    $\mathcal{B} = \mathcal{B} \cup \{\widetilde{H}^i\}$
return $\mathcal{B}$

Running time: $O(|V|(|V| + |E|))$

K Shortest Hyperpaths

$\text{K-SYNTHESIS}(H, s, t, K)$

Let L be a new priority queue
$\pi = \text{SHORTESTPATH}(H, s, t)$
$\text{INSERT}\,(L, (H, \pi))$
for $k = 1$ **to** K
    if $L = \emptyset$
        break
    $(H', \pi') = \text{EXTRACT-MIN}(L)$
    output π'
    if $k == K$
        break
    for each H^i in $\text{BACK-BRANCH}(H', \pi')$
        $\pi^i = \text{SHORTESTPATH}(H^i, s, t)$
        if $W(\pi^i) < \infty$
            $\text{INSERT}\,\left(L, (H^i, \pi^i)\right)$

Running time: $O(K|V|(|V| + |E|) + K \lg K)$