Tradeoffs for Routing Reconfiguration in WDM Networks

Nathann Cohen
David Coudert
Dorian Mazauric
Napoleão Nepomuceno
Nicolas Nisse

MASCOTTE, INRIA, I3S, CNRS, UNS, Sophia Antipolis, France

ANR ALADDIN meeting, Bordeaux, November 26-27th 2009

Routing in WDM Networks

Physical Network, Links provide several wavelengths

multi-graph $G = (V, E)$

an edge $(u, v) \Leftrightarrow$ one wavelength on the link (u, v)

Routing of a set of requests/connections

set of requests $\mathcal{R} \subseteq 2^{V \times V}$

routing: for each request (u, v),

a path from u to v and 1 wavelength.

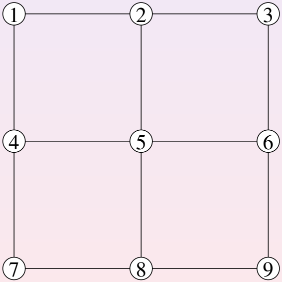
Problem: due to dynamicity of traffic, failures

how to maintain an efficient routing?

What happens in "real" world

Variation of traffic + dynamicity induced by failures
⇒ Online processes to route all requests: e.g., greedy routing

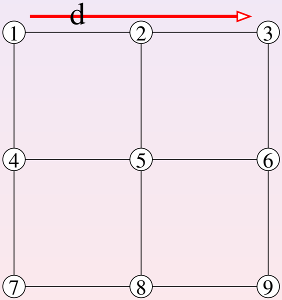
Example of a grid network with directed symmetric links



What happens in "real" world

Variation of traffic + dynamicity induced by failures
⇒ Online processes to route all requests: e.g., greedy routing

Example of a grid network with directed symmetric links

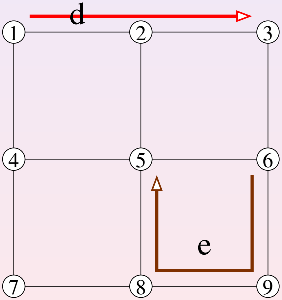


Request d : $1 \rightarrow 3$

What happens in "real" world

Variation of traffic + dynamicity induced by failures
⇒ Online processes to route all requests: e.g., greedy routing

Example of a grid network with directed symmetric links



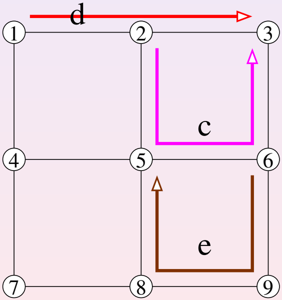
Request d : 1 → 3
Request e : 6 → 5

What happens in "real" world

Variation of traffic + dynamicity induced by failures

$\Rightarrow$ Online processes to route all requests: e.g., greedy routing

Example of a grid network with directed symmetric links

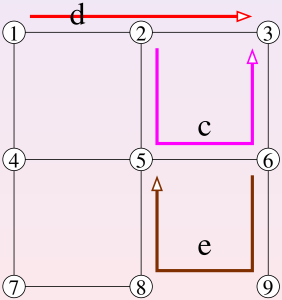


Request d : $1 \longrightarrow 3$
Request e : $6 \longrightarrow 5$
Request c : $2 \longrightarrow 3$

What happens in "real" world

Variation of traffic + dynamicity induced by failures
$\Rightarrow$ Online processes to route all requests: e.g., greedy routing

Example of a grid network with directed symmetric links

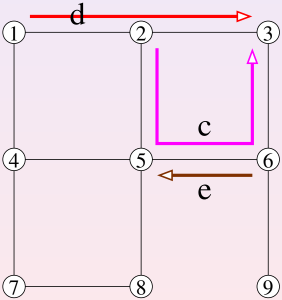


Request d : $1 \rightarrow 3$
Request e : $6 \rightarrow 5$
Request c : $2 \rightarrow 3$
Failure of link $\{8, 9\}$

What happens in "real" world

Variation of traffic + dynamicity induced by failures
⇒ Online processes to route all requests: e.g., greedy routing

Example of a grid network with directed symmetric links

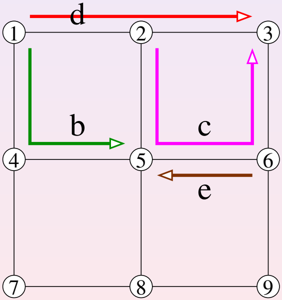


Request d : $1 \longrightarrow 3$
Request e : $6 \longrightarrow 5$
Request c : $2 \longrightarrow 3$
Rerouting of request e

What happens in "real" world

Variation of traffic + dynamicity induced by failures
⇒ Online processes to route all requests: e.g., greedy routing

Example of a grid network with directed symmetric links



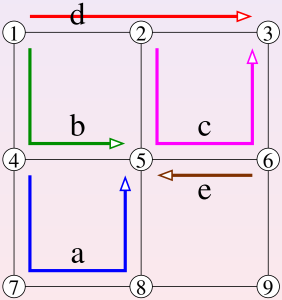
Request d : $1 \longrightarrow 3$
Request e : $6 \longrightarrow 5$
Request c : $2 \longrightarrow 3$
Request b : $1 \longrightarrow 5$
New link $\{8, 9\}$

What happens in "real" world

Variation of traffic + dynamicity induced by failures

$\Rightarrow$ Online processes to route all requests: e.g., greedy routing

Example of a grid network with directed symmetric links

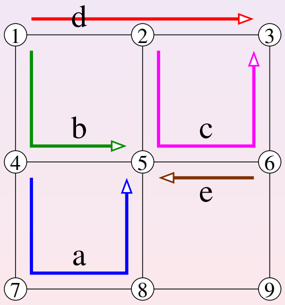


Request d : $1 \longrightarrow 3$
Request e : $6 \longrightarrow 5$
Request c : $2 \longrightarrow 3$
Request b : $1 \longrightarrow 5$
Request a : $4 \longrightarrow 5$

What happens in "real" world

Variation of traffic + dynamicity induced by failures
$\Rightarrow$ Online processes to route all requests: e.g., greedy routing

Example of a grid network with directed symmetric links



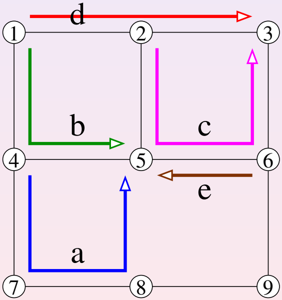
Leads to a poor usage of ressources

Sometimes greedy routing is impossible
even if several requests are allowed to be moved

What happens in "real" world

Variation of traffic + dynamicity induced by failures
$\Rightarrow$ Online processes to route all requests: e.g., greedy routing

Example of a grid network with directed symmetric links



Leads to a poor usage of ressources

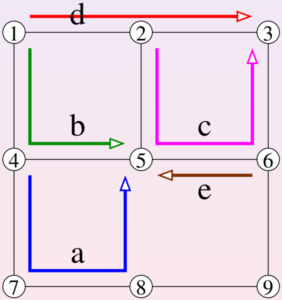
Sometimes greedy routing is impossible
even if several requests are allowed to be moved

If $\{5, 8\}$ fails:
Move-to-Vacant impossible

What happens in "real" world

Variation of traffic + dynamicity induced by failures
⇒ Online processes to route all requests: e.g., greedy routing

Example of a grid network with directed symmetric links



2 questions arise:

1. Compute new routing
2. Switch from initial routing to final one

We focus on 2

Two ways of switching one request

Make-before-break:

Establish new path before switching the connection

$\Longrightarrow$ Destination resources must be available

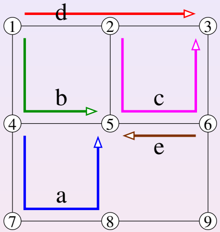
Break-before-make:

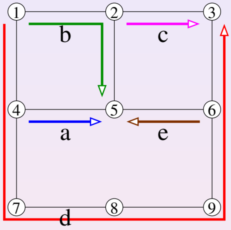
Break connection before establishing the new path

$\Longrightarrow$ Traffic stopped $=$ interruption

The Routing Reconfiguration Problem

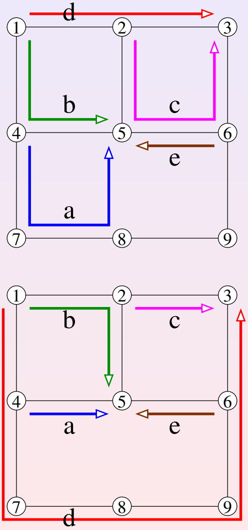
How to go from the initial routing (left) to the final one (right)?

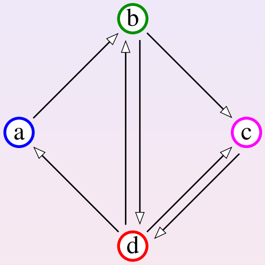




Inputs:	Set of connection requests + current & new routing
Output:	Scheduling for switching connection requests from current to new routes
Constraint:	A connection is switched **only once**
ObjectiveS	Number of Interruptions (detailled later)

Dependency digraph

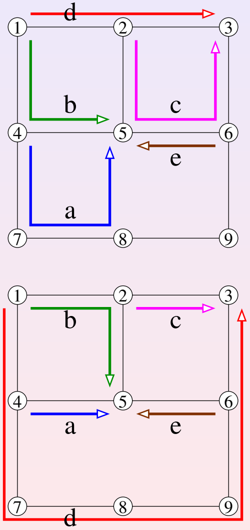


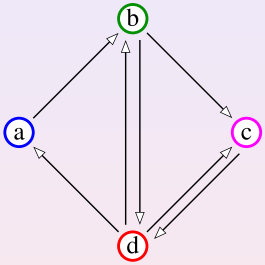


$u \longrightarrow v$
if u needs ressources of v
if v must be rerouted/interrupted before u

b needs ressources used by d and c

Dependency digraph

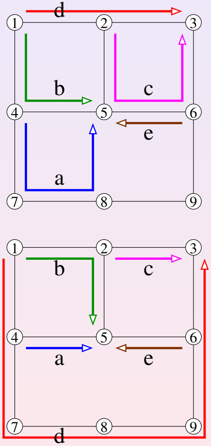


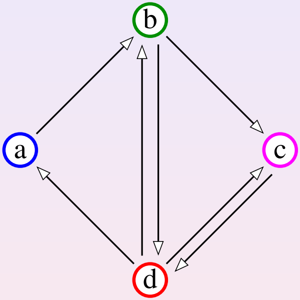


Dependancy Digraph

- one vertex per connection with different routes in $\mathcal{I}$ and $\mathcal{F}$
- arc from u to v if ressources needed by u in $\mathcal{F}$ are used by v in $\mathcal{I}$

A game on dependency digraph

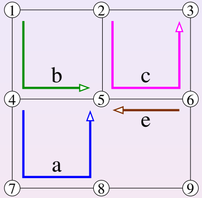


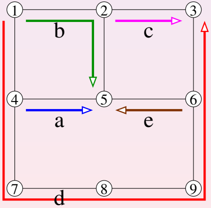


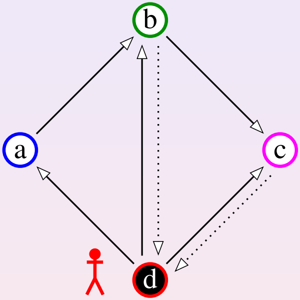
cyclic dependancies

$\Rightarrow$ Interruption required

A game on dependency digraph

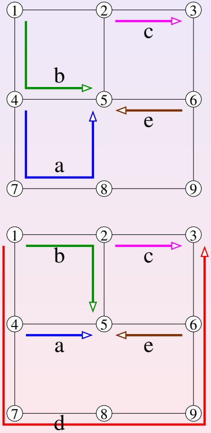


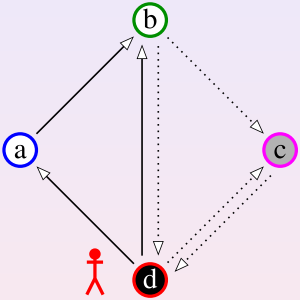




put an agent on node *d*
break request *d*

A game on dependency digraph

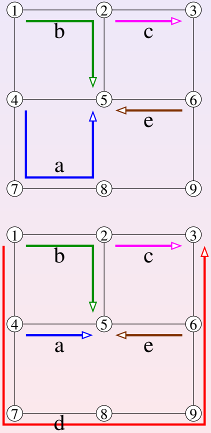


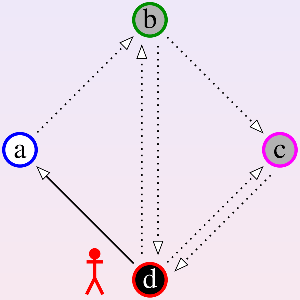


process node c

reroute request c

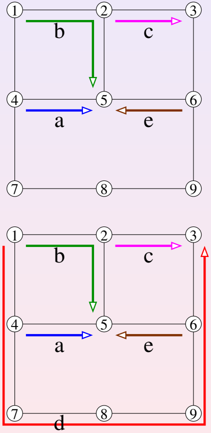
A game on dependency digraph

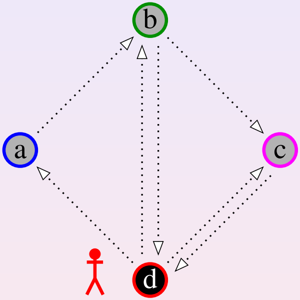




process node b
reroute request b

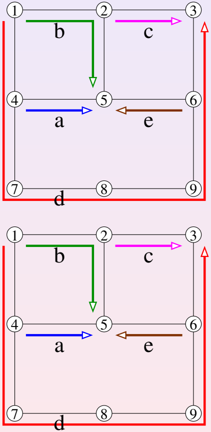
A game on dependency digraph

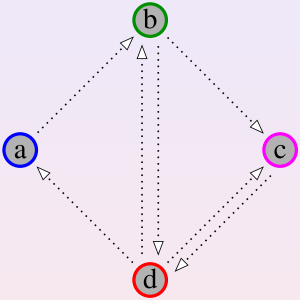




process node *a*
reroute request *a*

A game on dependency digraph

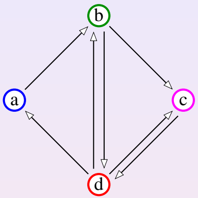


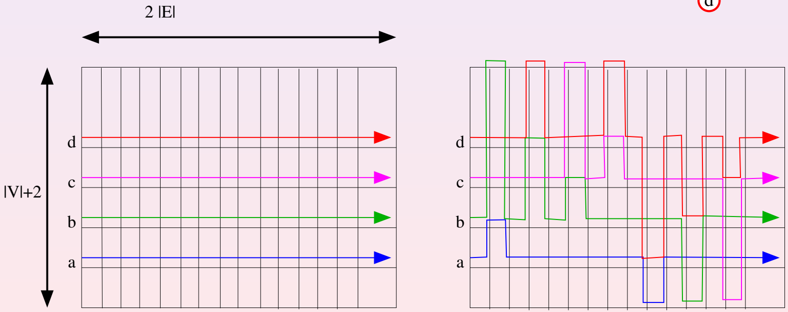


process node d
and remove agent
route request d

From now on: problem on digraphs

Any directed graph is a dependency digraph

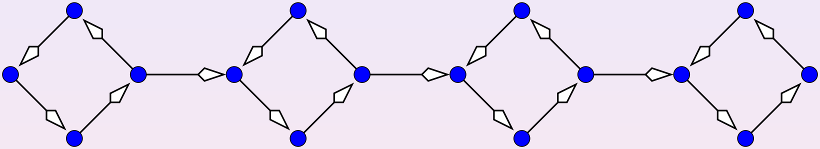




Two possible objectives

Minimize overall number of interrupted requests

Minimum Feedback Vertex Set (MFVS), here $N/4$

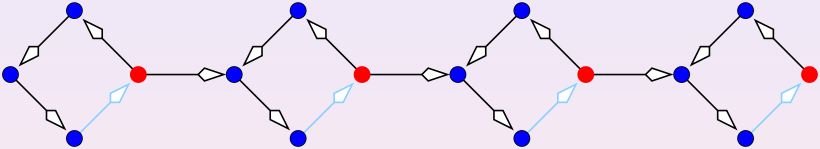


Remarks: MFVS is NP-complete and non APX in digraphs
2-approx in undirected (directed symmetric) graphs

Two possible objectives

Minimize overall number of interrupted requests

Minimum Feedback Vertex Set (MFVS), here $N/4$

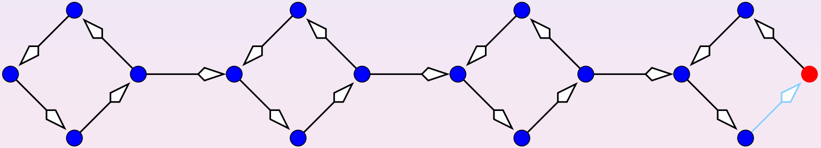


Remarks: MFVS is NP-complete and non APX in digraphs
2-approx in undirected (directed symmetric) graphs

Two possible objectives

Minimize overall number of interrupted requests

Minimum Feedback Vertex Set (MFVS), here $N/4$



Minimize number of simultaneous interrupted requests

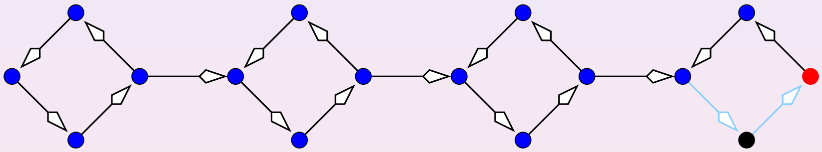
Process Number, $pn =$ smallest number of requests that have to be **simultaneously** interrupted.

Here, $pn = 1 \Rightarrow$ Gap with MFVS up to $N/2$

Two possible objectives

Minimize overall number of interrupted requests

Minimum Feedback Vertex Set (MFVS), here $N/4$



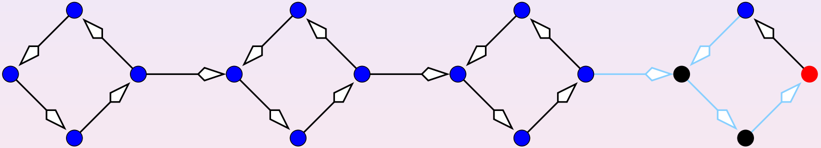
Minimize number of simultaneous interrupted requests

Process Number, $pn =$ smallest number of requests that have to be **simultaneously** interrupted.
Here, $pn = 1 \Rightarrow$ Gap with MFVS up to $N/2$

Two possible objectives

Minimize overall number of interrupted requests

Minimum Feedback Vertex Set (MFVS), here $N/4$



Minimize number of simultaneous interrupted requests

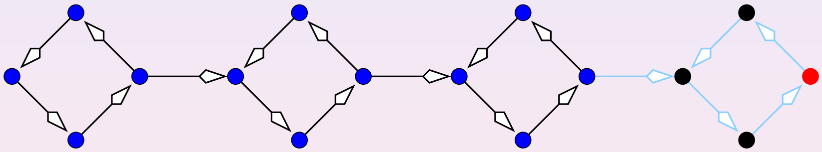
Process Number, $pn =$ smallest number of requests that have to be **simultaneously** interrupted.

Here, $pn = 1 \Rightarrow$ Gap with MFVS up to $N/2$

Two possible objectives

Minimize overall number of interrupted requests

Minimum Feedback Vertex Set (MFVS), here $N/4$



Minimize number of simultaneous interrupted requests

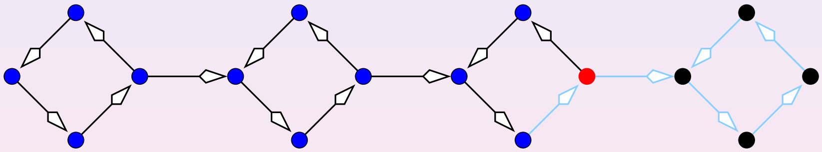
Process Number, $pn =$ smallest number of requests that have to be **simultaneously** interrupted.

Here, $pn = 1 \Rightarrow$ Gap with MFVS up to $N/2$

Two possible objectives

Minimize overall number of interrupted requests

Minimum Feedback Vertex Set (MFVS), here $N/4$



Minimize number of simultaneous interrupted requests

Process Number, pn = smallest number of requests that have to be **simultaneously** interrupted.
Here, $pn = 1 \Rightarrow$ Gap with MFVS up to $N/2$

Routing Reconfiguration, Process number

Game with Agents on the Dependency digraph D

Sequence of three basic operations,...

1. Place a searcher at a node = **interrupt the request**;
2. Process a node if all its out-neighbors are either processed or occupied by an agent = **(Re)route a connection when final resources are available**;

   A processed node is removed from the dependency digraph.
3. Remove an agent from a node, after having processed it.

...that must result in processing all nodes

Process number $pn(D) = \min p \mid D$ can be processed with p agents

Remark: In undirected graphs or symmetric digraphs:

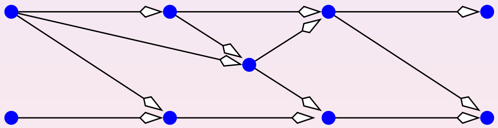
Graph Searching game when a fugitive is captured when surrounded

Example: DAG

Only one operation is used

1. Place a searcher at a node = interrupt the request;
2. Process a node if all its out-neighbors are either processed or occupied by an agent = **(Re)route a connection when final resources are available**;
3. Remove an agent from a node, after having processed it.

DAG



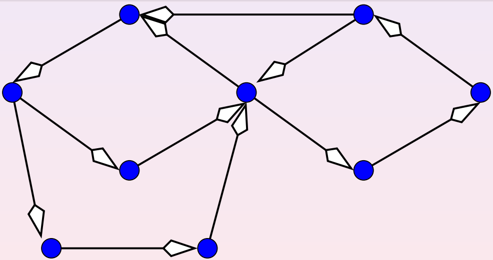
Theorem

$pn(D) = 0$ iff D is a DAG

Digraphs with process number 1

One agent is used

1. Place a searcher at a node = **interrupt the request**;
2. Process a node if all its out-neighbors are either processed or occupied by an agent = **(Re)route a connection when final resources are available**;
3. Remove an agent from a node, after having processed it.



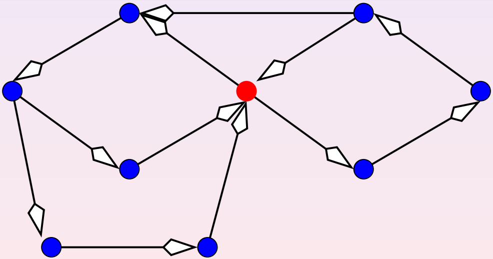
Theorem

$pn(D) = 1 \Leftrightarrow \forall SCC$, MFVS$(SCC) = 1$ $\qquad O(N + M)$

Digraphs with process number 1

One agent is used

1. Place a searcher at a node = **interrupt the request**;
2. Process a node if all its out-neighbors are either processed or occupied by an agent = **(Re)route a connection when final resources are available**;
3. Remove an agent from a node, after having processed it.



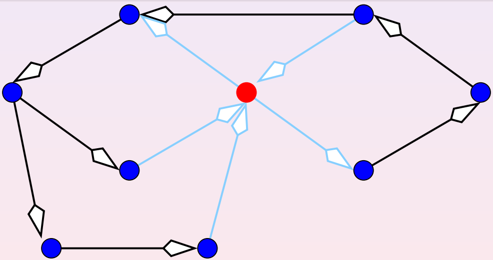
Theorem

$pn(D) = 1 \Leftrightarrow \forall SCC,\ \mathrm{MFVS}(SCC) = 1 \qquad O(N+M)$

Digraphs with process number 1

One agent is used

1. Place a searcher at a node = **interrupt the request**;
2. Process a node if all its out-neighbors are either processed or occupied by an agent = **(Re)route a connection when final resources are available**;
3. Remove an agent from a node, after having processed it.



Theorem

$pn(D) = 1 \Leftrightarrow \forall SCC,\ \mathrm{MFVS}(SCC) = 1$ $\quad O(N + M)$

Process number versus Other Parameters

a parameter of directed (and undirected) graphs

vs, vertex separation

in undirected graph or symetric digraph: vs = pathwidth

$vs(G) = pw(G)$ **Kinnersley [IPL 92]**

Theorem **(Coudert & Sereni, 2007)**

$vs(D) \leq pn(D) \leq vs(D) + 1$

Complexity: NP-Complete, Not APX

- Characterization of digraphs with process number 0, 1, 2
  (Coudert & Sereni, 2007)

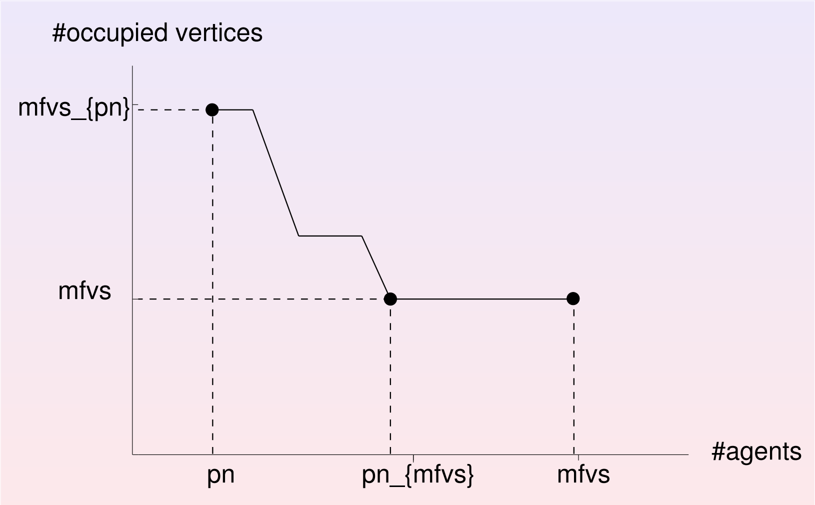
State of the Art

- distributed $O(n \log n)$-time exact algorithm in trees
  (Coudert, Huc, Mazauric [DISC 08])

- generalized Model handling priority connections
  connections that cannot be interrupted
  heuristic using random walk
  (Coudert, Huc, Mazauric, Nisse, Sereni [ONDM 09])
  heuristic using LP **(Solano [Globecom 09])**

- generalized Model allowing bandwidth sharing
  deciding whether reconfiguration may be done without interruption:
  NP-complete **(Coudert, Mazauric, Nisse [AGT 09])**

Tradeoff: total/ max simultaneous interruptions



Complexity

- Smallest number of agents such that the number of occupied vertices is minimum $= pn_{mfvs}(D)$
- $\mu = \frac{pn_{mfvs}(D)}{pn(D)}$

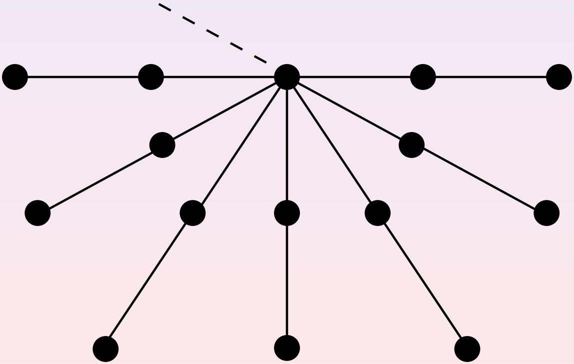
- Smallest total number of occupied vertices such that the number of agents is minimum $= mfvs_{pn}(D)$
- $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$

Theorem

The problems of determining $pn_{mfvs}(D)$, $mfvs_{pn}(D)$, μ, and λ are NP-Complete and not APX.

$\#$ agents for minimizing $\#$ occupied vertices

$\exists$ digraphs with arbitrary large ratio: $\mu = \frac{pn_{mfvs}(D)}{pn(D)}$.



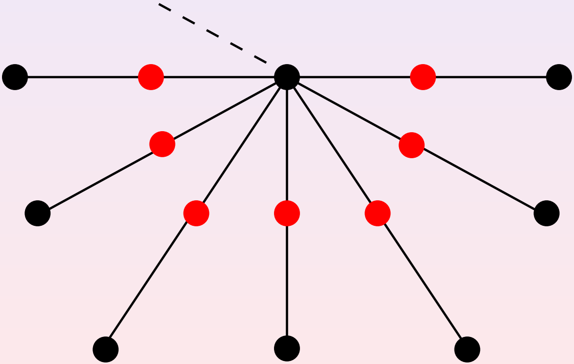
$mfvs(D) = n$

$pn(D) = 2$

$pn_{mfvs}(D) = n$

$\#$ agents for minimizing $\#$ occupied vertices

$\exists$ digraphs with arbitrary large ratio: $\mu = \frac{pn_{mfvs}(D)}{pn(D)}$.



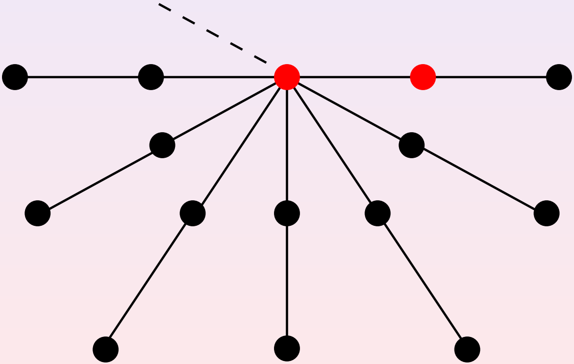
$$mfvs(D) = n$$

$$pn(D) = 2$$

$$pn_{mfvs}(D) = n$$

$\#$ agents for minimizing $\#$ occupied vertices

$\exists$ digraphs with arbitrary large ratio: $\mu = \frac{pn_{mfvs}(D)}{pn(D)}$.



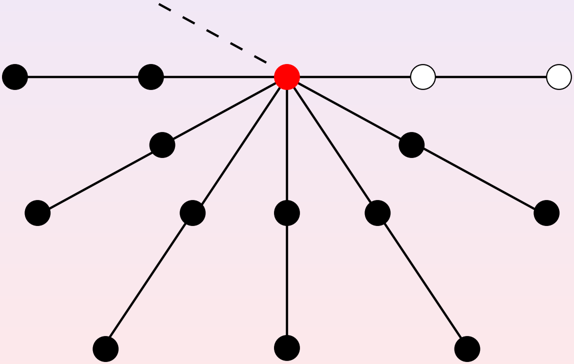
$$mfvs(D) = n$$

$$pn(D) = 2$$

$$pn_{mfvs}(D) = n$$

agents for minimizing # occupied vertices

$\exists$ digraphs with arbitrary large ratio: $\mu = \frac{pn_{mfvs}(D)}{pn(D)}$.



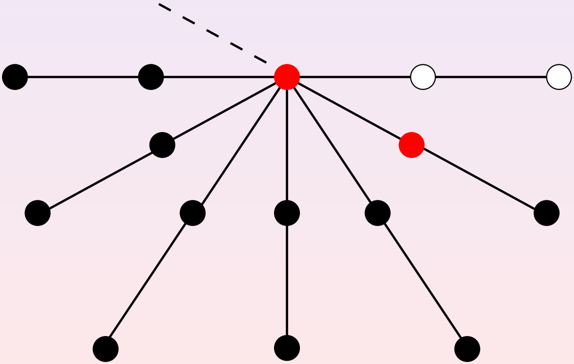
$mfvs(D) = n$

$pn(D) = 2$

$pn_{mfvs}(D) = n$

$\#$ agents for minimizing $\#$ occupied vertices

$\exists$ digraphs with arbitrary large ratio: $\mu = \frac{pn_{mfvs}(D)}{pn(D)}$.



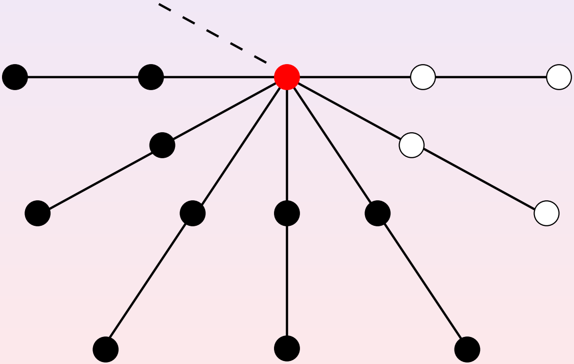
$mfvs(D) = n$

$pn(D) = 2$

$pn_{mfvs}(D) = n$

$\#$ agents for minimizing $\#$ occupied vertices

$\exists$ digraphs with arbitrary large ratio: $\mu = \frac{pn_{mfvs}(D)}{pn(D)}$.



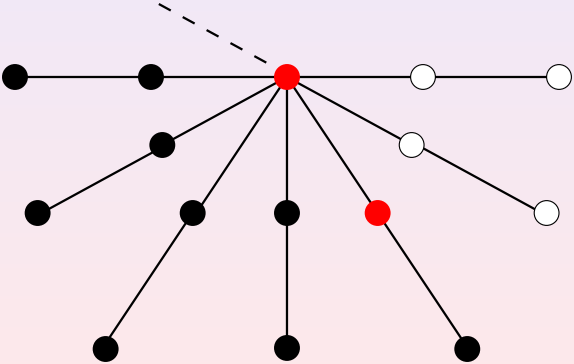
$mfvs(D) = n$

$pn(D) = 2$

$pn_{mfvs}(D) = n$

$\#$ agents for minimizing $\#$ occupied vertices

$\exists$ digraphs with arbitrary large ratio: $\mu = \frac{pn_{mfvs}(D)}{pn(D)}$.



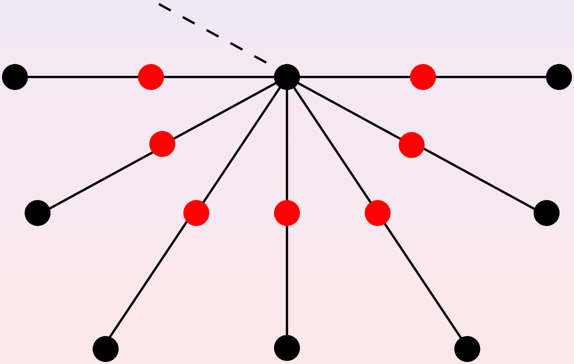
$mfvs(D) = n$

$pn(D) = 2$

$pn_{mfvs}(D) = n$

agents for minimizing # occupied vertices

$\exists$ digraphs with arbitrary large ratio: $\mu = \frac{pn_{mfvs}(D)}{pn(D)}$.



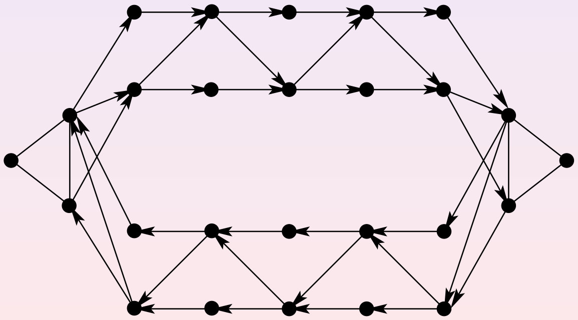
$mfvs(D) = n$

$pn(D) = 2$

$pn_{mfvs}(D) = n$

occupied vertices by the minimum # agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



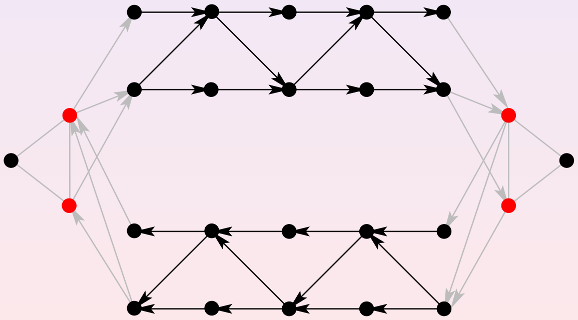
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

occupied vertices by the minimum # agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



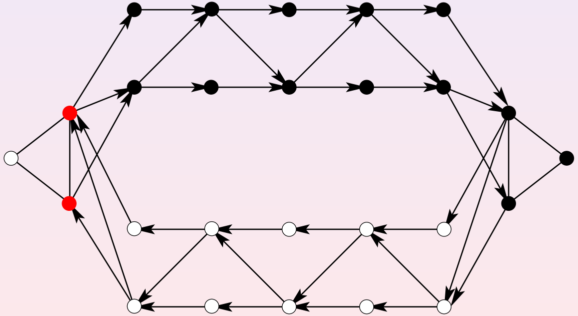
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

$\#$ occupied vertices by the minimum $\#$ agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



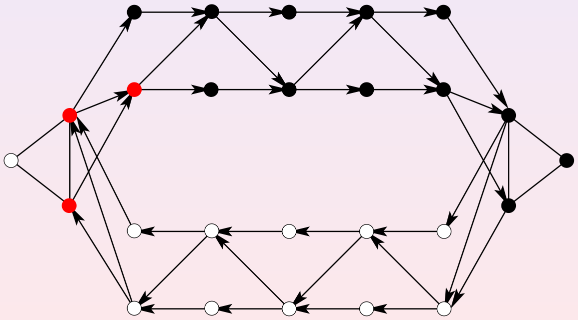
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

occupied vertices by the minimum # agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



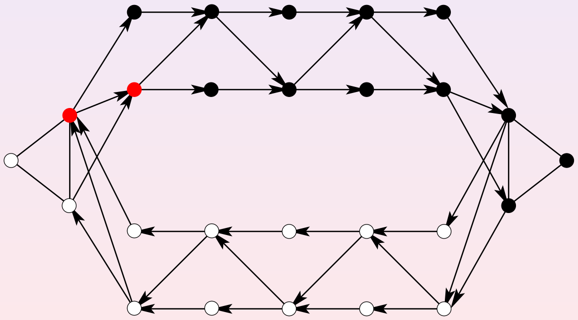
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

$\#$ occupied vertices by the minimum $\#$ agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



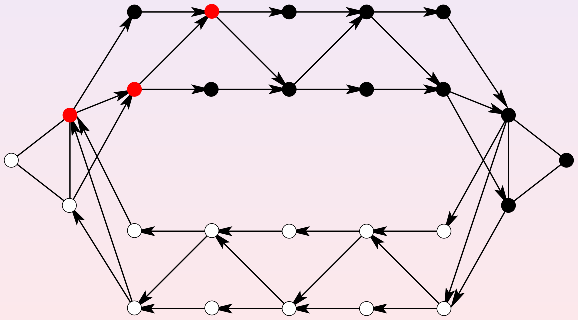
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

occupied vertices by the minimum # agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



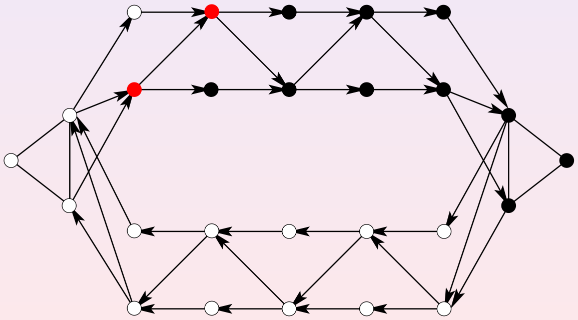
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

occupied vertices by the minimum # agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



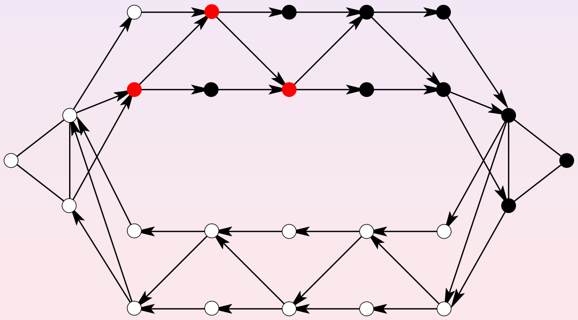
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

occupied vertices by the minimum # agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



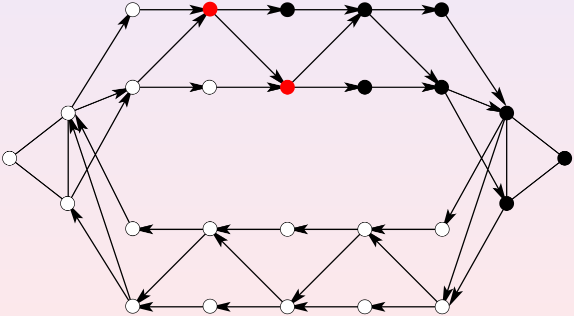
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

occupied vertices by the minimum # agents

∃ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



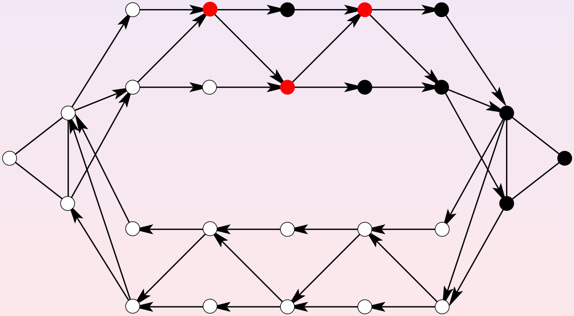
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

occupied vertices by the minimum # agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



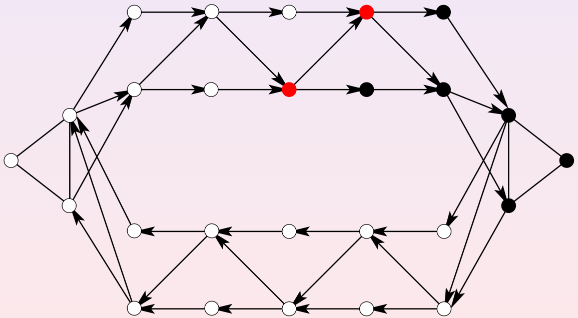
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

occupied vertices by the minimum # agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



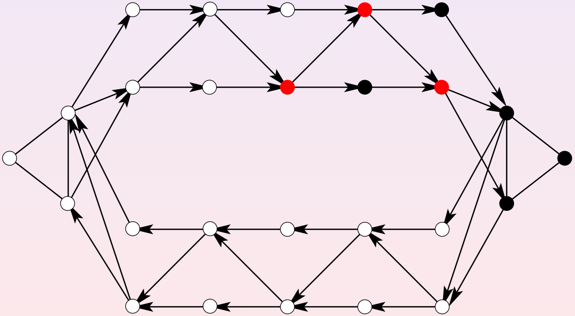
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

$\#$ occupied vertices by the minimum $\#$ agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



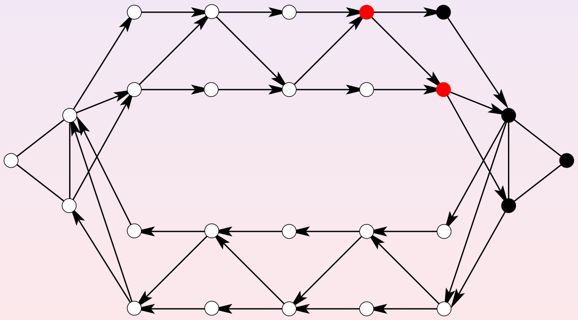
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

$\#$ occupied vertices by the minimum $\#$ agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



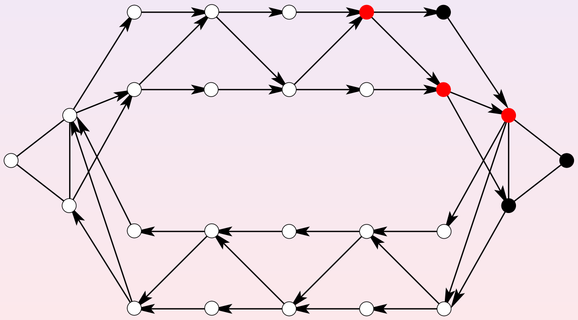
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

$\#$ occupied vertices by the minimum $\#$ agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



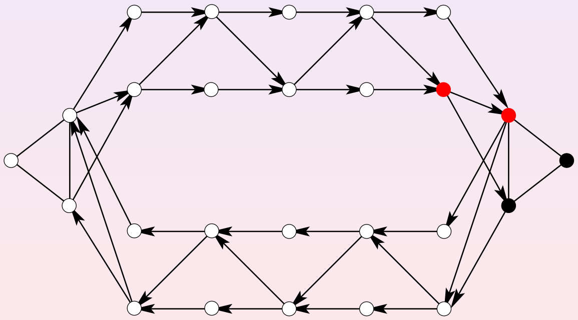
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

occupied vertices by the minimum # agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



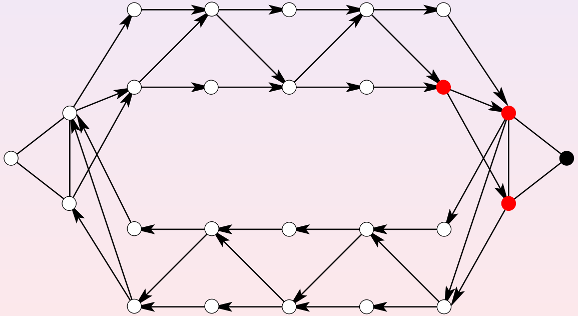
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

$\#$ occupied vertices by the minimum $\#$ agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



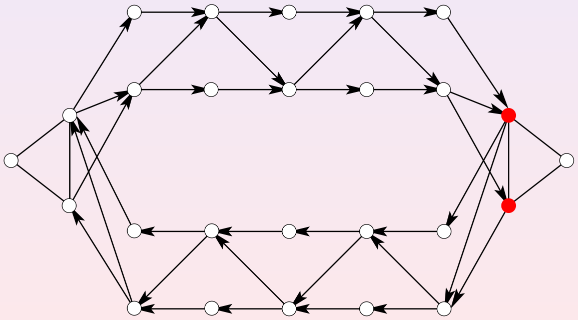
$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

occupied vertices by the minimum # agents

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.

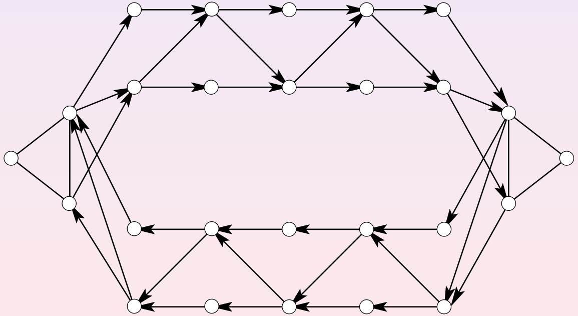


$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

$\exists$ digraphs with arbitrary large ratio: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)}$.



$mfvs(D) = 4$

$pn(D) = 3$

$mfvs_{pn}(D) = n + 4$

$\#$ occupied vertices by the minimum $\#$ agents

Directed graphs with BOUNDED Process Number:
$\lambda =$ occupied vertices / mfvs UNBOUNDED

What if G is undirected ??

Let G be a symmetric directed/undirected graph,
$\lambda = \frac{mfvs_{pn}(G)}{mfvs(G)} \leq pn(G)$

$\#$ occupied vertices by the minimum $\#$ agents

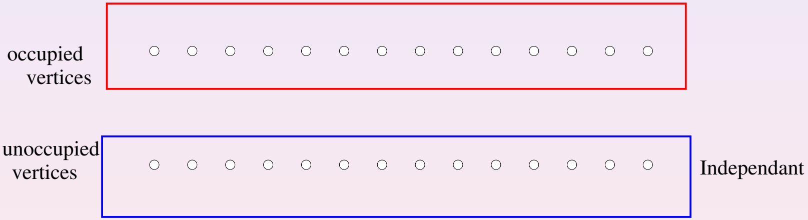
Directed graphs with BOUNDED Process Number:
$\lambda =$ occupied vertices / mfvs UNBOUNDED

What if G is undirected ??

Let G be a symmetric directed/undirected graph,
$\lambda = \frac{mfvs_{pn}(G)}{mfvs(G)} \leq pn(G)$

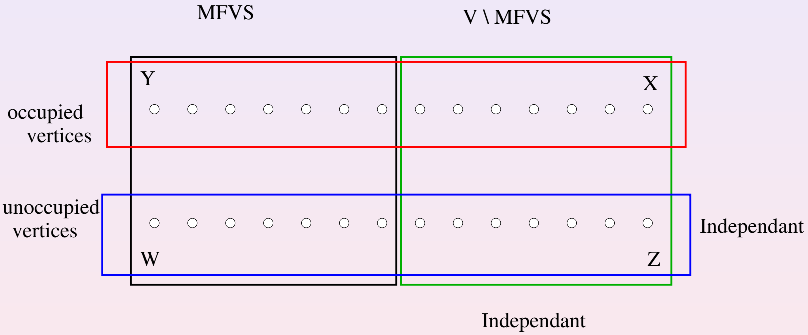
occupied vertices by the minimum # agents

Consider a MFVS of G. S using $pn(G)$ agents and occupying $mfvs_{pn}(G)$ vertices, such that occupies the minimum number of vertices in MFVS



$\#$ occupied vertices by the minimum $\#$ agents

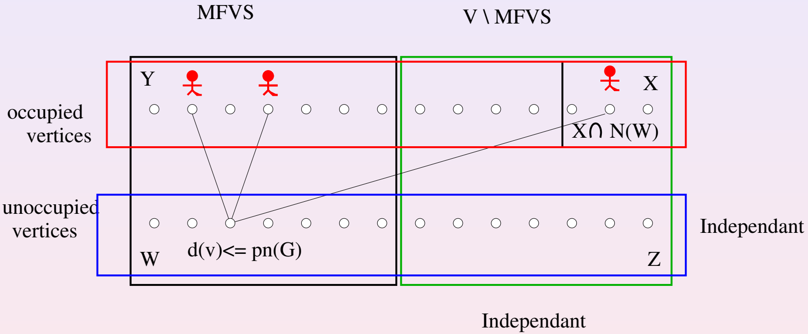
Consider a MFVS of G. S using $pn(G)$ agents and occupying $mfvs_{pn}(G)$ vertices, such that occupies the minimum number of vertices in MFVS



$$\lambda = \frac{mfvs_{pn}(G)}{mfvs(G)} = \frac{Y+X}{Y+W}$$

occupied vertices by the minimum # agents

Consider a MFVS of G. S using $pn(G)$ agents and occupying $mfvs_{pn}(G)$ vertices, such that occupies the minimum number of vertices in MFVS

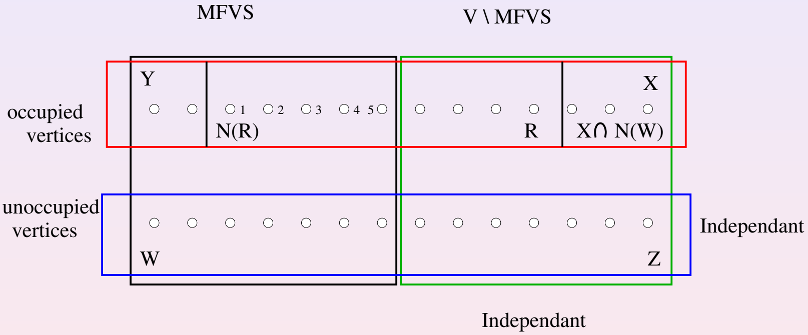


$$\lambda = \frac{mfvs_{pn}(G)}{mfvs(G)} = \frac{|Y|+|X|}{|Y|+|W|}$$

$$|X| = |X \cap N(W)| + |R| \leq |W|.pn(G) + |R|$$

$\#$ occupied vertices by the minimum $\#$ agents

Consider a MFVS of G. S using $pn(G)$ agents and occupying $mfvs_{pn}(G)$ vertices, such that occupies the minimum number of vertices in MFVS

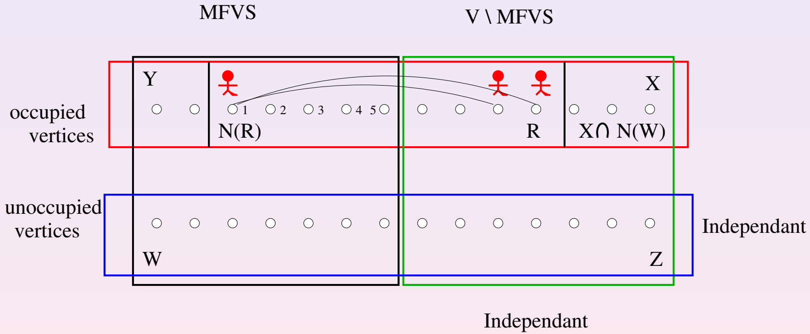


$$\lambda = \frac{mfvs_{pn}(G)}{mfvs(G)} = \frac{|Y|+|X|}{|Y|+|W|} \leq \frac{|Y|+|W|.pn(G)+|R|}{|Y|+|W|}$$

$N(R) = \{v_1, \cdots, v_r\} \subseteq Y$: ordering in which agents are removed

occupied vertices by the minimum # agents

Consider a MFVS of G. S using $pn(G)$ agents and occupying $mfvs_{pn}(G)$ vertices, such that occupies the minimum number of vertices in MFVS

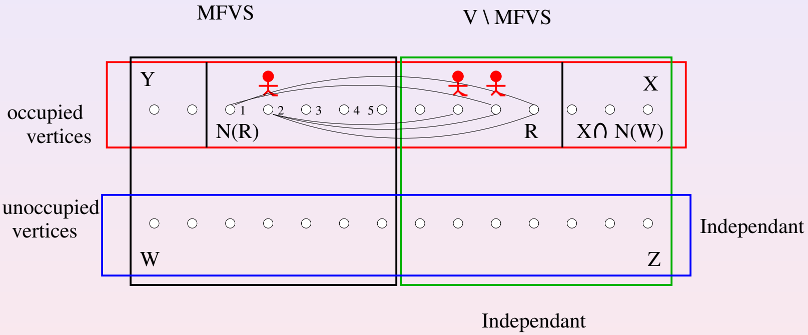


$$\lambda = \frac{mfvs_{pn}(G)}{mfvs(G)} = \frac{|Y|+|X|}{|Y|+|W|} \leq \frac{|Y|+|W|.pn(G)+|R|}{|Y|+|W|}$$

$$|N(v_1)| \leq pn(G) - 1$$

occupied vertices by the minimum # agents

Consider a MFVS of G. S using $pn(G)$ agents and occupying $mfvs_{pn}(G)$ vertices, such that occupies the minimum number of vertices in MFVS

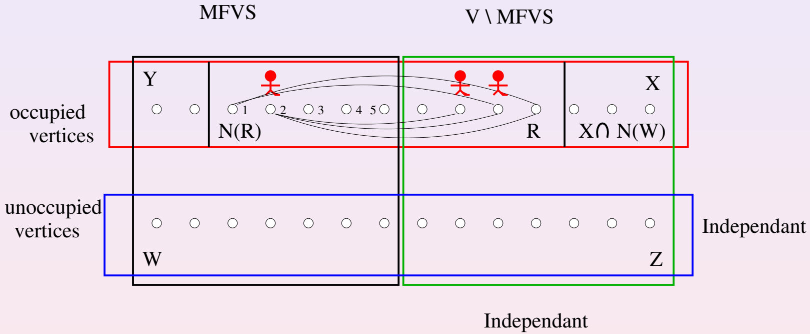


$$\lambda = \frac{mfvs_{pn}(G)}{mfvs(G)} = \frac{|Y|+|X|}{|Y|+|W|} \leq \frac{|Y|+|W|.pn(G)+|R|}{|Y|+|W|}$$

$|N(v_2) \setminus N(v_1)| \leq pn(G) - 1$, $|N(v_i) \setminus \bigcup_{j<i} N(v_j)| \leq pn(G) - 1$

occupied vertices by the minimum # agents

Consider a MFVS of G. S using $pn(G)$ agents and occupying $mfvs_{pn}(G)$ vertices, such that occupies the minimum number of vertices in MFVS

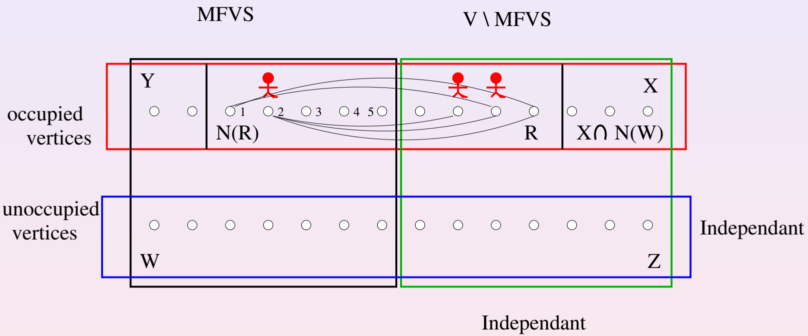


$$\lambda = \frac{mfvs_{pn}(G)}{mfvs(G)} = \frac{|Y|+|X|}{|Y|+|W|} \leq \frac{|Y|+|W|.pn(G)+|R|}{|Y|+|W|}$$

so $|R| \leq |N(R)|(pn(G)-1) \leq |Y|(pn(G)-1)$

occupied vertices by the minimum # agents

Consider a MFVS of G. S using $pn(G)$ agents and occupying $mfvs_{pn}(G)$ vertices, such that occupies the minimum number of vertices in MFVS

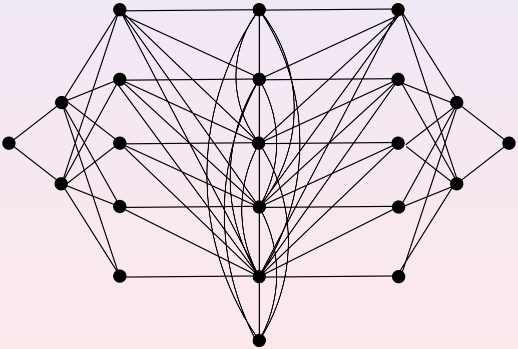


$$\lambda = \frac{mfvs_{pn}(G)}{mfvs(G)} = \frac{|Y|+|X|}{|Y|+|W|} \leq \frac{|Y|+|W|.pn(G)+|R|}{|Y|+|W|}$$

$$\lambda \leq \frac{|Y|+|W|.pn(G)+|Y|(pn(G)-1)}{|Y|+|W|} = pn(G)$$

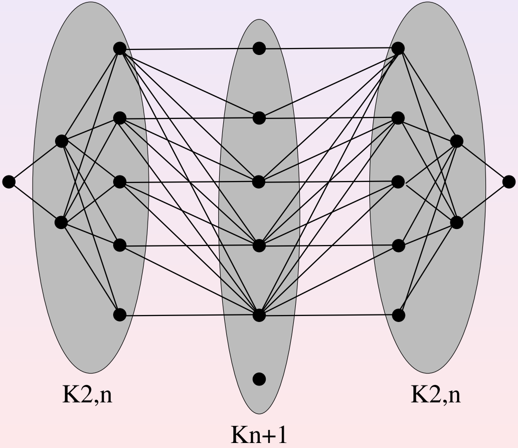
occupied vertices by the minimum # agents

$\forall \epsilon$, $\exists$ symmetric digraphs D: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)} > 3 - \epsilon$.



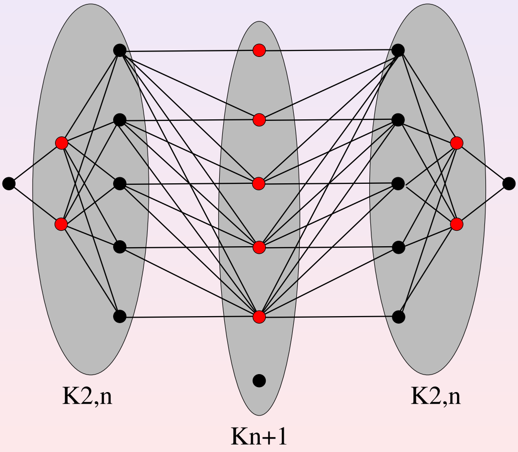
$\#$ occupied vertices by the minimum $\#$ agents

$\forall \epsilon$, $\exists$ symmetric digraphs D: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)} > 3 - \epsilon$.



$\#$ occupied vertices by the minimum $\#$ agents

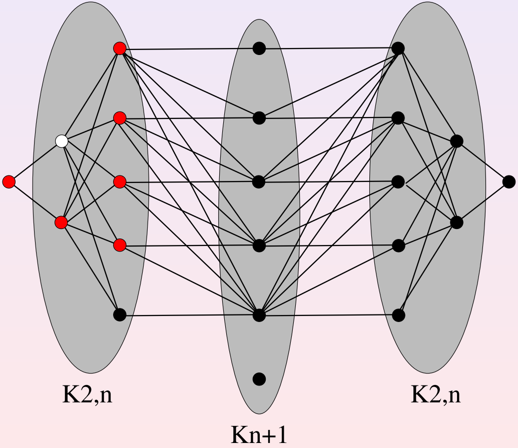
$\forall\epsilon$, $\exists$ symmetric digraphs D: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)} > 3 - \epsilon.$



$mfvs(D) = n + 4$

$\#$ occupied vertices by the minimum $\#$ agents

$\forall\epsilon$, $\exists$ symmetric digraphs D: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)} > 3 - \epsilon$.

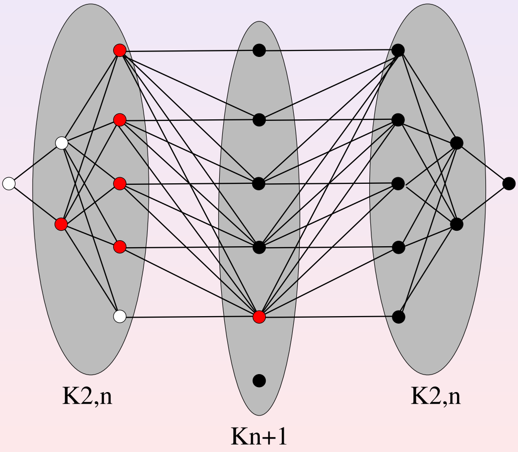


$mfvs(D) = n + 4$

$pn(D) = n + 1$

occupied vertices by the minimum # agents

$\forall \epsilon$, $\exists$ symmetric digraphs D: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)} > 3 - \epsilon$.

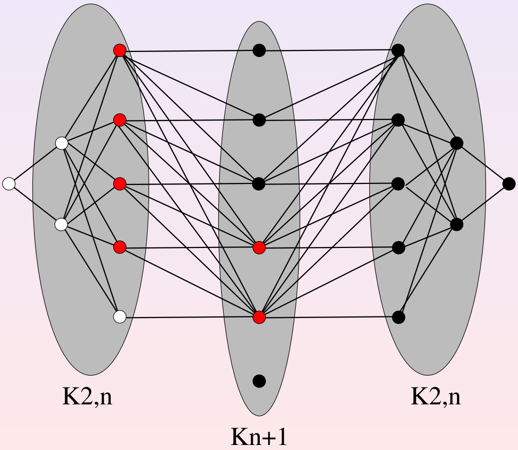


$mfvs(D) = n + 4$

$pn(D) = n + 1$

$\#$ occupied vertices by the minimum $\#$ agents

$\forall \epsilon$, $\exists$ symmetric digraphs D: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)} > 3 - \epsilon$.

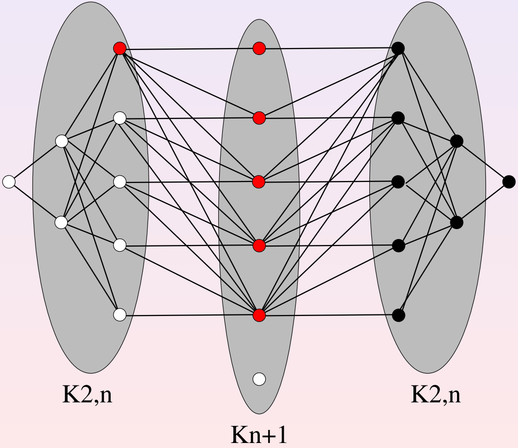


$mfvs(D) = n + 4$

$pn(D) = n + 1$

occupied vertices by the minimum # agents

$\forall \epsilon$, $\exists$ symmetric digraphs D: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)} > 3 - \epsilon$.

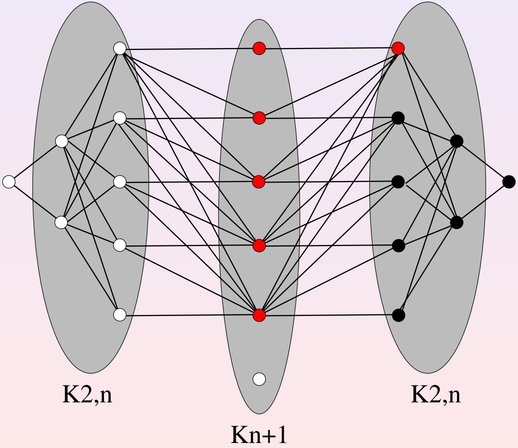


$mfvs(D) = n + 4$

$pn(D) = n + 1$

$\#$ occupied vertices by the minimum $\#$ agents

$$\forall \epsilon,\ \exists \text{ symmetric digraphs } D\text{: } \lambda = \frac{mfvs_{pn}(D)}{mfvs(D)} > 3 - \epsilon.$$

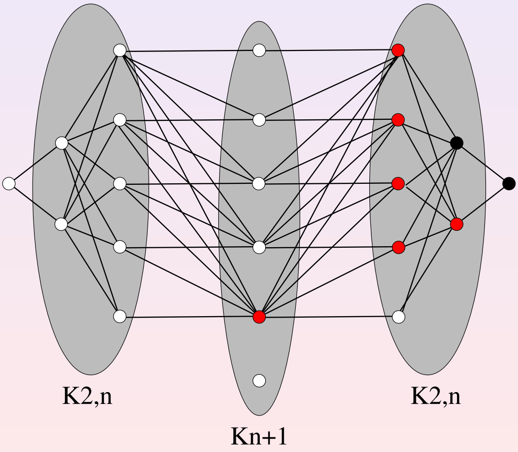


$mfvs(D) = n + 4$

$pn(D) = n + 1$

occupied vertices by the minimum # agents

$\forall \epsilon$, $\exists$ symmetric digraphs D: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)} > 3 - \epsilon$.

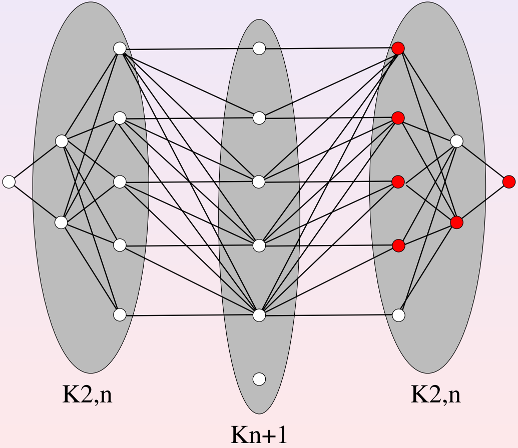


$mfvs(D) = n + 4$

$pn(D) = n + 1$

$\#$ occupied vertices by the minimum $\#$ agents

$\forall \epsilon$, $\exists$ symmetric digraphs D: $\lambda = \frac{mfvs_{pn}(D)}{mfvs(D)} > 3 - \epsilon$.



$mfvs(D) = n + 4$

$pn(D) = n + 1$

$mfvs_{pn}(D) = 3n + 2$

Some open questions

A lot of "bad" news... No tradeoff ?

Can we restrict the class of dependancy digraphs ?

No... even if the physical network is a directed path...

Conjecture

Let G be a symmetric directed/undirected graph,
$\lambda = \frac{mfvs_{pn}(G)}{mfvs(G)} \leq 3$

- Approximation and Heuristic algorithms for these parameters

Link between random walks and separators of graphs ?

Some open questions

A lot of "bad" news... No tradeoff ?

Can we restrict the class of dependancy digraphs ?

No... even if the physical network is a directed path...

Conjecture

Let G be a symmetric directed/undirected graph,
$\lambda = \frac{mfvs_{pn}(G)}{mfvs(G)} \leq 3$

- Approximation and Heuristic algorithms for these parameters

Link between random walks and separators of graphs ?

Merci