KU LEUVEN kulak

A definition of scale-independent communities & a flow-free vertex cut algorithm

Alexander Braekevelt, Pieter Leyman & Patrick De Causmaecker

Department of Computer Science, KU Leuven, Belgium

What is the subject?

We collect a lot of data...

We collect a lot of data...

- social networks

Image source: Paul Butler, facebook intern, 14 december 2010

We collect a lot of data...

- social networks
- street maps

Image source: ©2018 Google, Inst. Geogr. Nacional

We collect a lot of data...

- social networks
- street maps
- chemical interactions

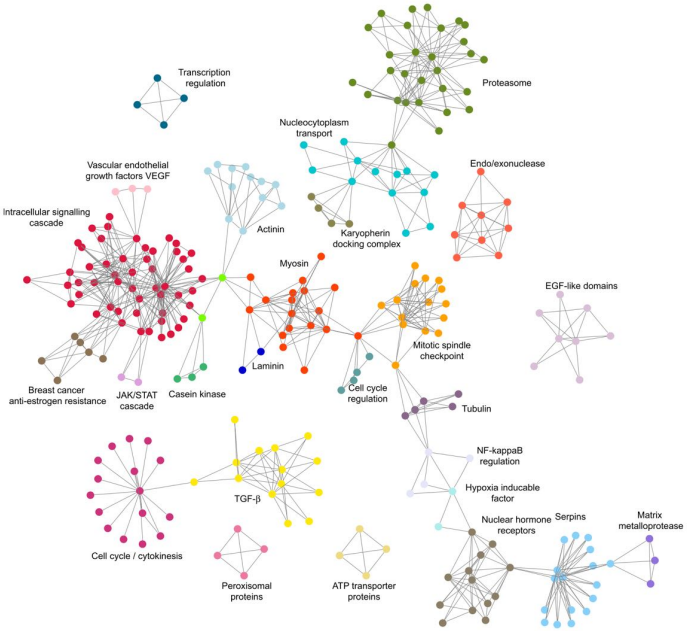


Image source: S. Fortunato, Community detection in graphs

We collect a lot of data...

- social networks
- street maps
- chemical interactions
- animal behaviour

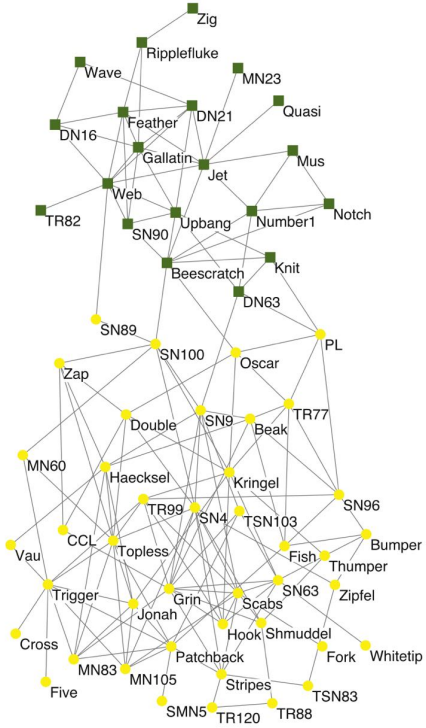


Image source: S. Fortunato, Community detection in graphs

We collect a lot of data...

- social networks
- street maps
- chemical interactions
- animal behaviour
- word associations

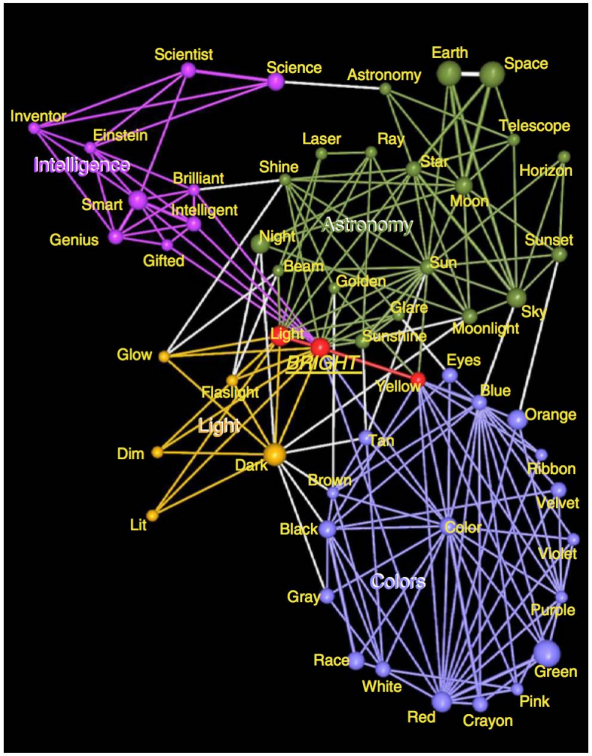


Image source: S. Fortunato, Community detection in graphs

We collect a lot of data...

- social networks
- street maps
- chemical interactions
- animal behaviour
- word associations
- and many more ...

We collect a lot of data...

- social networks
- street maps
- chemical interactions
- animal behaviour
- word associations
- and many more ...

... which we want to model.

We collect a lot of data...

- social networks → social groups
- street maps
- chemical interactions
- animal behaviour
- word associations
- and many more ...

... which we want to model.

We collect a lot of data...

- social networks → social groups
- street maps → cities, villages
- chemical interactions
- animal behaviour
- word associations
- and many more ...

... which we want to model.

We collect a lot of data...

- social networks → social groups
- street maps → cities, villages
- chemical interactions
- animal behaviour
- word associations
- and many more ...

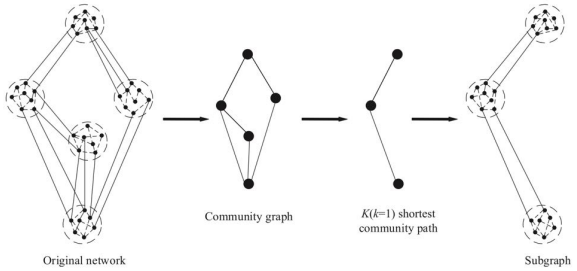


Image source: M. Gong et al., 2016

... which we want to model.

We collect a lot of data...

- social networks → social groups
- street maps → cities, villages
- chemical interactions → interacting molecules
- animal behaviour
- word associations
- and many more ...

... which we want to model.

We collect a lot of data...

- social networks → social groups
- street maps → cities, villages
- chemical interactions → interacting molecules
- animal behaviour → communities
- word associations
- and many more ...

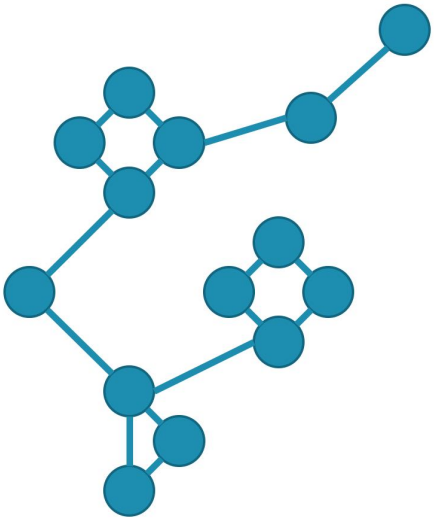
... which we want to model.

We collect a lot of data...

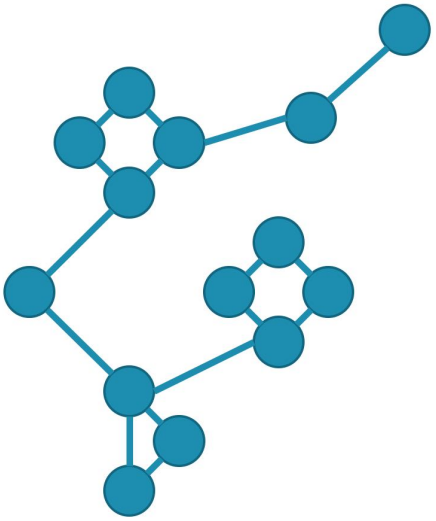
- social networks → social groups
- street maps → cities, villages
- chemical interactions → interacting molecules
- animal behaviour → communities
- word associations → concepts
- and many more ...

... which we want to model.

Problem of “community detection”

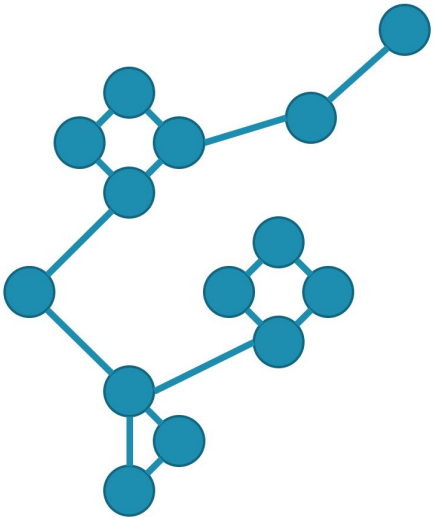


Problem of “community detection”



The data (input)

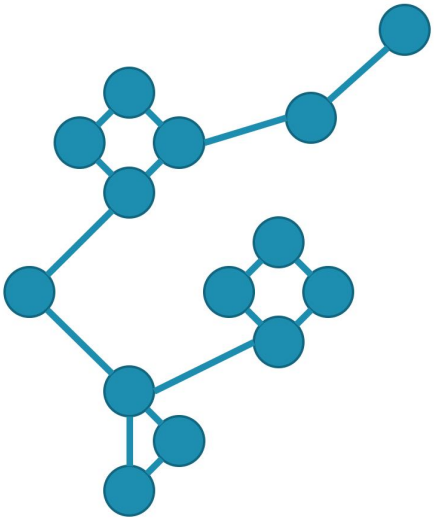
Problem of “community detection”



The data (input)

- graph = network = data

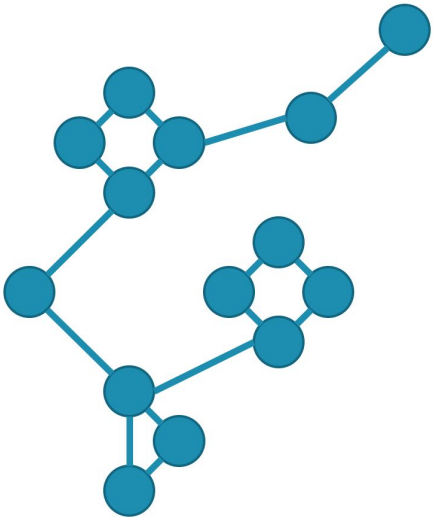
Problem of “community detection”



The data (input)

- graph = network = data
- vertex = node = object

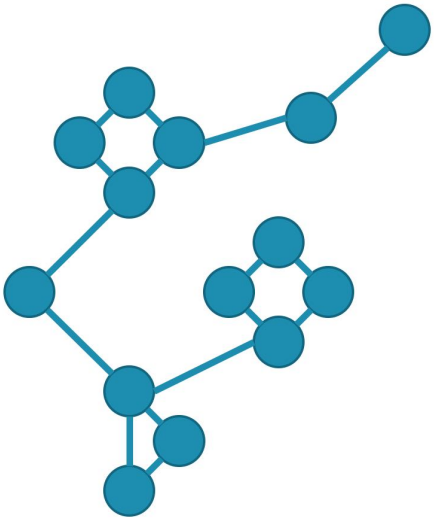
Problem of “community detection”



The data (input)

- graph = network = data
- vertex = node = object
- edge = arc = relation

Problem of “community detection”

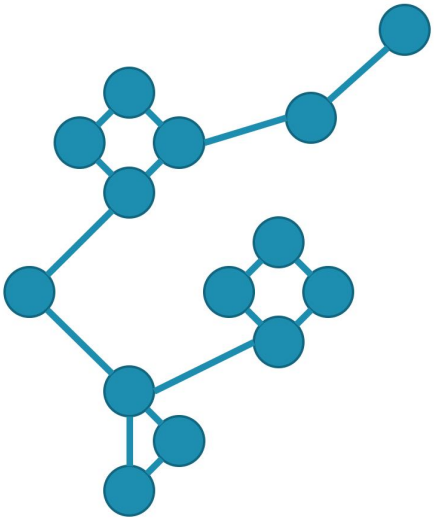


The data (input)

- graph = network = data
- vertex = node = object
- edge = arc = relation

- undirected

Problem of “community detection”

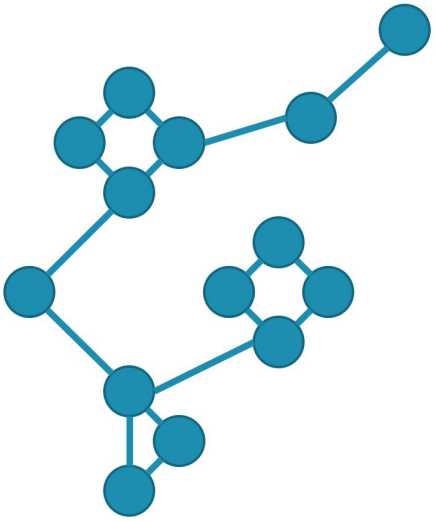


The data (input)

- graph = network = data
- vertex = node = object
- edge = arc = relation

- undirected
- sparse

Problem of “community detection”

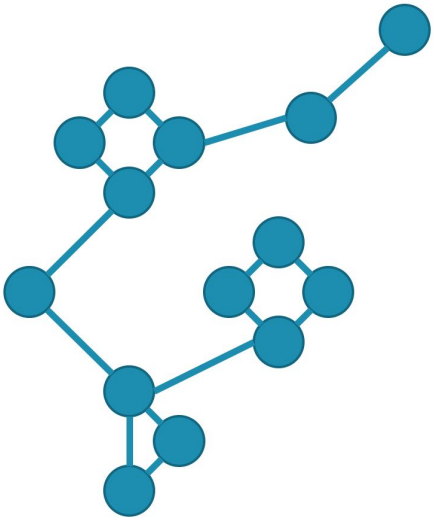


The data (input)

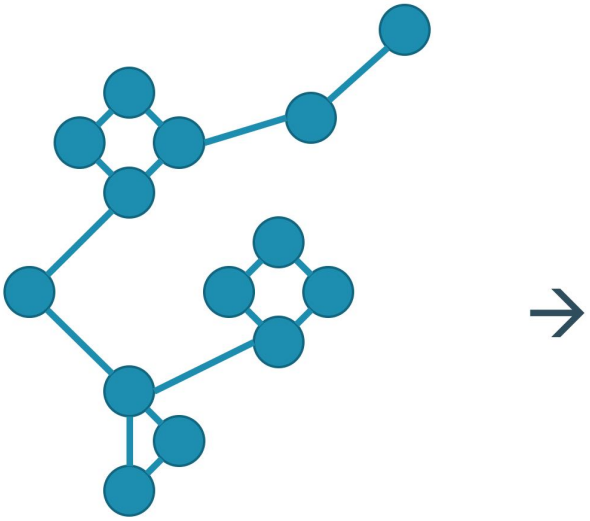
- graph = network = data
- vertex = node = object
- edge = arc = relation

- undirected
- sparse

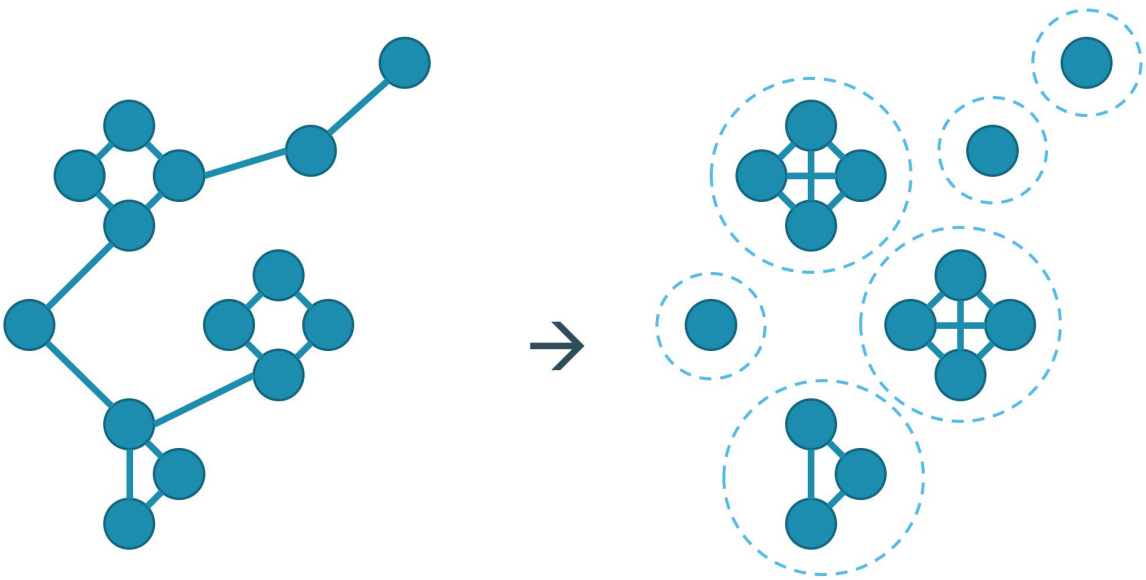
Problem of “community detection”



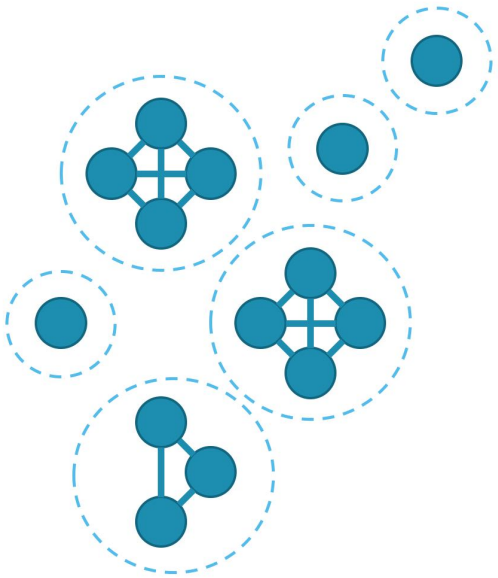
Problem of “community detection”



Problem of “community detection”

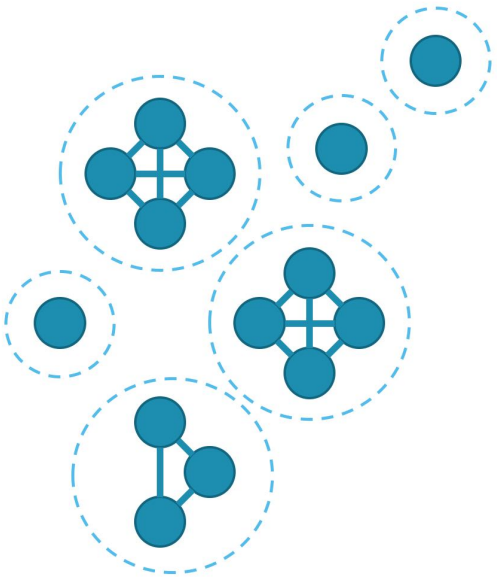


Problem of “community detection”



Problem of “community detection”

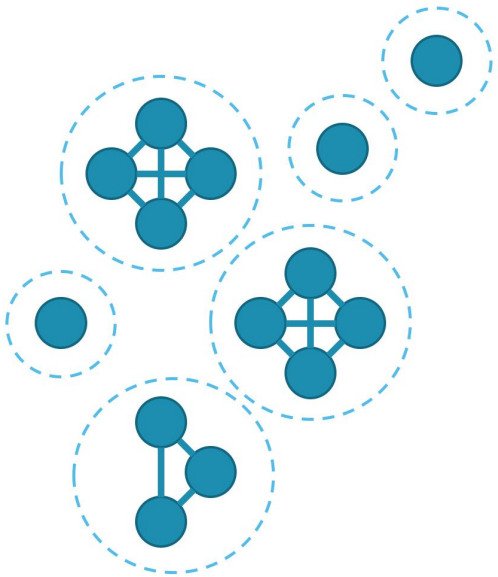
The result (output)



Problem of “community detection”

The result (output)

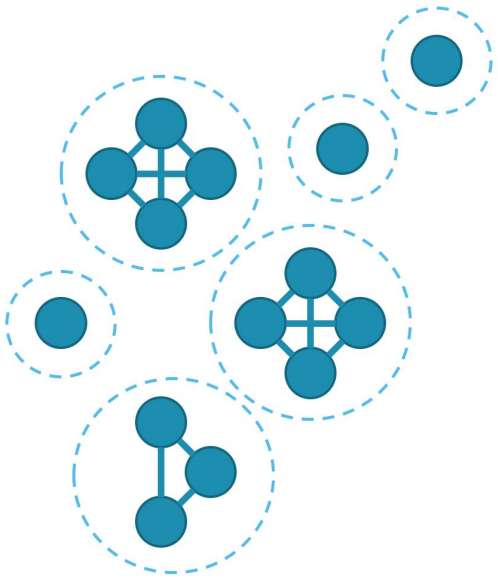
- “communities”



Problem of “community detection”

The result (output)

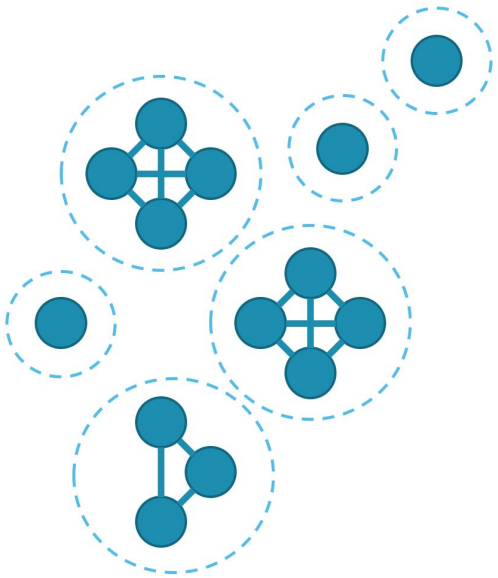
- “communities”
- equivalence relation



Problem of “community detection”

The result (output)

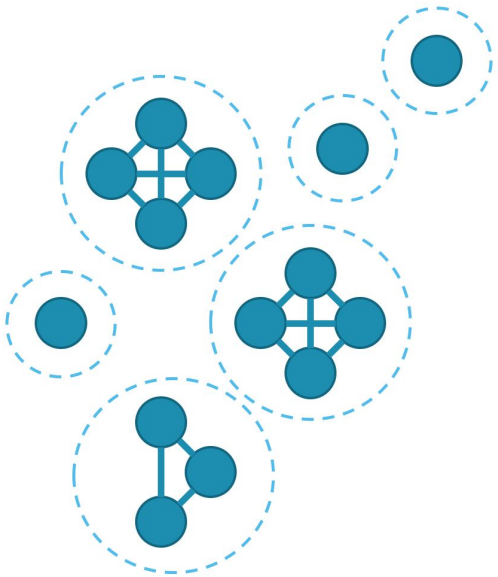
- “communities”
- equivalence relation
- sets of vertices



Problem of “community detection”

The result (output)

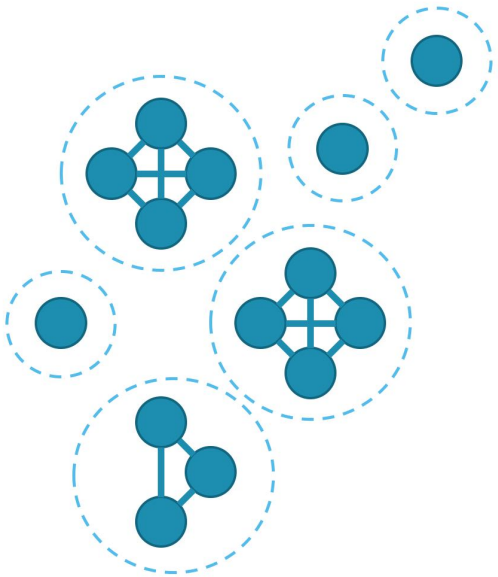
- “communities”
- equivalence relation
- sets of vertices
- possibly overlapping



Problem of “community detection”

The result (output)

- “communities”
- equivalence relation
- sets of vertices
- possibly overlapping



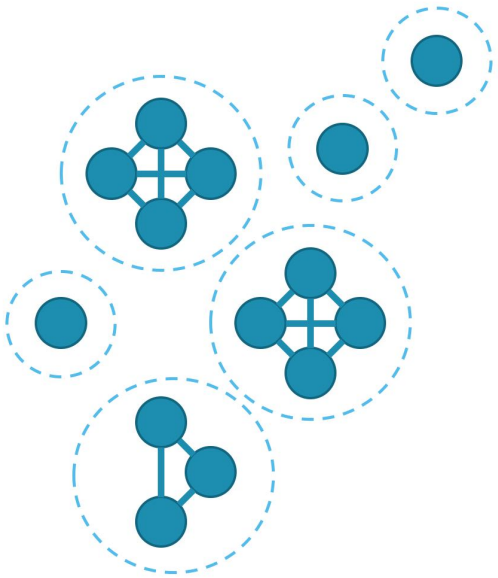
“strongly connected individuals with few outgoing connections”

Problem of “community detection”

The result (output)

- “communities”
- equivalence relation
- sets of vertices
- possibly overlapping

“strongly connected individuals with few outgoing connections”



What are the research questions?

Research question 1

What is a good definition for a community such that it's properties correspond to the intuitive concept of "strongly connected individuals with few outgoing connections"?

Research question 1

What is a good definition for a community such that it's properties correspond to the intuitive concept of "strongly connected individuals with few outgoing connections"?

Research question 1

What is a good definition for a community such that it's properties correspond to the intuitive concept of "strongly connected individuals with few outgoing connections"?

Research question 2

How does an algorithm for finding this definition scale with respect to the data size?

Research question 1

What is a good definition for a community such that it's properties correspond to the intuitive concept of "strongly connected individuals with few outgoing connections"?

Research question 2

How does an algorithm for finding this definition scale with respect to the data size?

Research question 1:

definition of community

definition of community

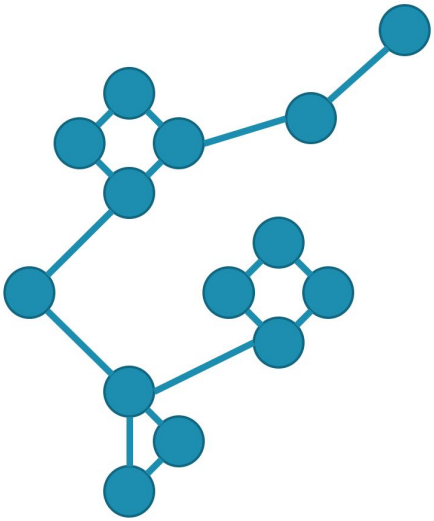
=

measurement of connectedness

definition of community

=

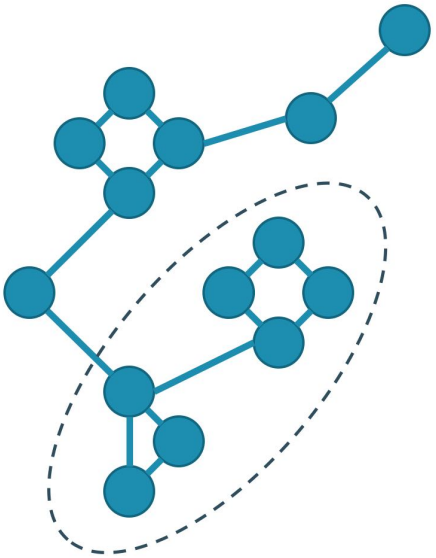
measurement of connectedness



definition of community

=

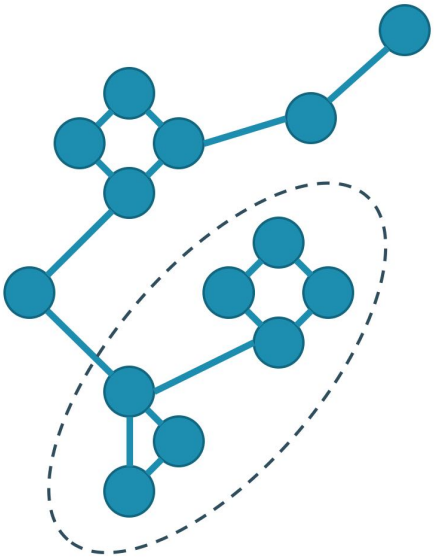
measurement of connectedness



definition of community

=

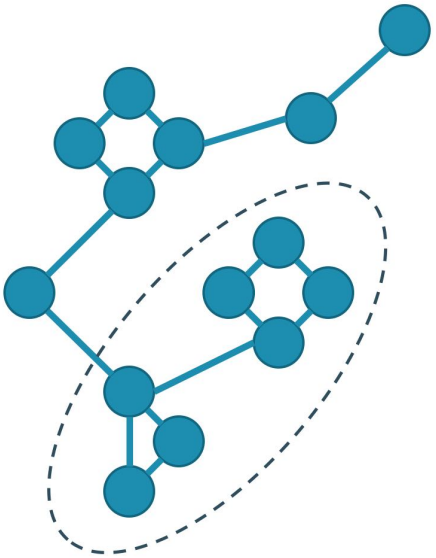
measurement of connectedness



definition of community

=

measurement of connectedness



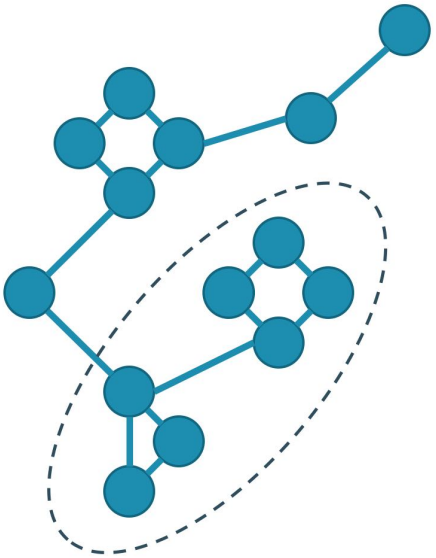
→

7
6
5
4
3

definition of community

=

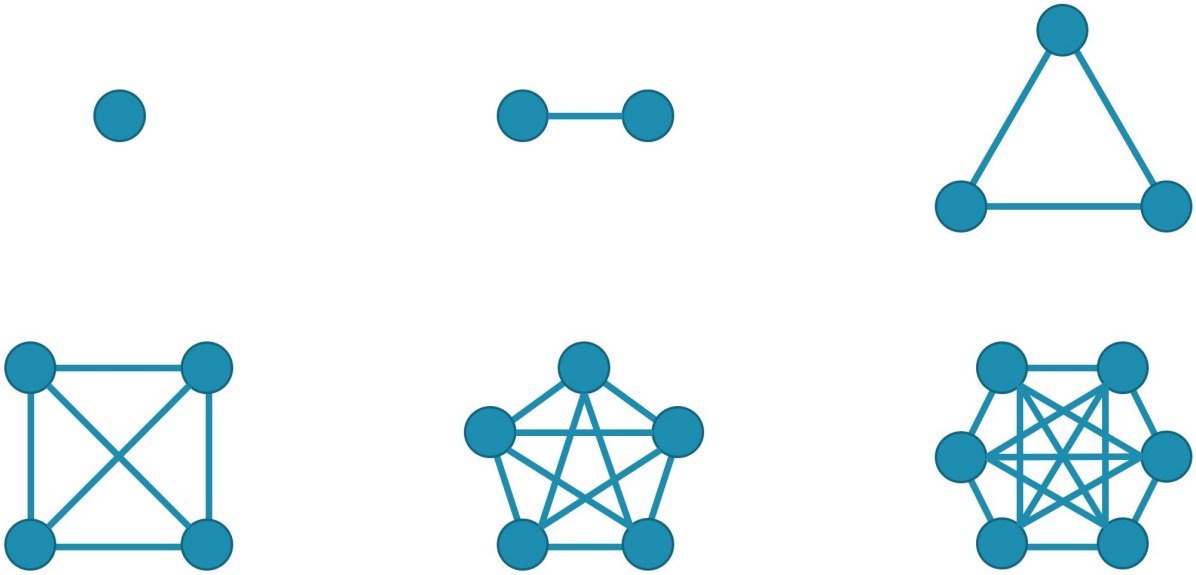
measurement of connectedness



7
6
5
4
3

The perfect community: clique

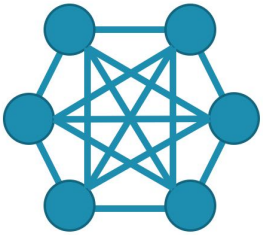
The perfect community: clique



The perfect community: clique

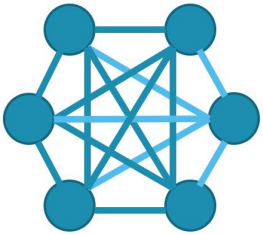
The perfect community: clique

Distance
Diameter
Domination
Degree
Edge connectivity
Vertex connectivity
Edge density



The perfect community: clique

- Distance
- Diameter
- Domination
- **Degree**
- Edge connectivity
- Vertex connectivity
- Edge density

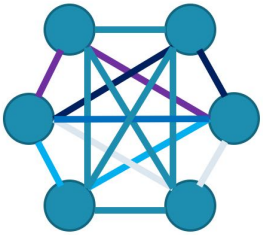


$$size - 1 = 5$$

The **degree** is
smallest number of neighbors of a vertex.

The perfect community: clique

- Distance
- Diameter
- Domination
- Degree
- **Edge connectivity**
- Vertex connectivity
- Edge density

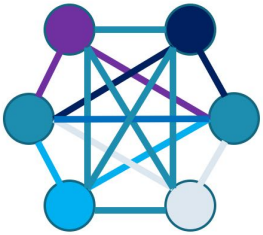


$size - 1 = 5$

The **edge connectivity** is
smallest number of edge-independent paths
between every pair of vertices.

The perfect community: clique

- Distance
- Diameter
- Domination
- Degree
- Edge connectivity
- **Vertex connectivity**
- Edge density

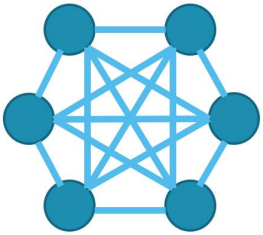


$size - 1 = 5$

The **vertex connectivity** is
smallest number of vertex-independent paths
between every pair of vertices.

The perfect community: clique

- Distance
- Diameter
- Domination
- Degree
- Edge connectivity
- Vertex connectivity
- **Edge density**



$$size * (size - 1) = 30$$

The **edge density** is number of edges.

The perfect community: clique

Distance
Diameter
Domination
Degree
Edge connectivity
Vertex connectivity
Edge density

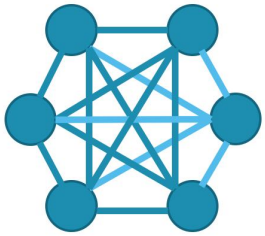
Clique relaxation framework

	ensure lower bound		relax upper bound	
	absolute	**relative**	**absolute**	**relative**
Distance	s-clique	def. 3	def. 8	def. 3
Diameter	s-club	def. 4	def. 9	def. 4
Domination	s-plex	def. 5	def. 10	def. 5
Degree	k-core	λ in (λ,γ)-quasi-clique	s-plex	λ in (λ,γ)-quasi-clique
Edge connectivity	def. 1	def. 6	def. 11	def. 6
Vertex connectivity	k-block	def. 7	s-bundle	def. 7
Edge density	def. 2	γ-quasi-clique	s-defective-clique	γ-quasi-clique

Property	Definition	Vertex connectivity k	
Distance	s-clique	$k \geq 0$ *	absolute lower bound
Diameter	s-club	$k \geq 1$ *	
Domination	s-plex	$k \geq n - 2x + 2$ *	
Degree	k-core	$k \geq 2x + 2 - n$ *	
Edge connectivity	def. 1	$k \geq 2x + 2 - n$	
Vertex connectivity	k-block	$k \geq x$ *	
Edge density	def. 2	$k \geq \lceil x\,n / 2 - (n-1)\,(n-2) / 2 \rceil$ *	
Distance	def. 3	$k \geq 0$ *	relative bound
Diameter	def. 4	$k \geq 1$ *	
Domination	def. 5	$k \geq n - 2x\,(n-1) + 2$ *	
Degree	λ in (λ,γ)-quasi-clique	$k \geq 2x\,(n-1) + 2 - n$ *	
Edge connectivity	def. 6	$k \geq 2x\,(n-1) + 2 - n$	
Vertex connectivity	def. 7	$k \geq x\,(n-1)$ *	
Edge density	γ-quasi-clique	$k \geq \lceil x\,n\,(n-1) / 2 - (n-1)\,(n-2) / 2 \rceil$ *	
Distance	def. 8	$k \geq 0$ *	absolute upper bound
Diameter	def. 9	$k \geq 1$ *	
Domination	def. 10	$k \geq n - 2\,(n-x) + 2$ *	
Degree	s-plex	$k \geq 2\,(n-x) + 2 - n$ *	
Edge connectivity	def. 11	$k \geq 2(n-x) + 2 - n$	
Vertex connectivity	s-bundle	$k \geq n - x$ *	
Edge density	s-defective-clique	$k \geq \lceil (n-x)\,n / 2 - (n-1)\,(n-2) / 2 \rceil$ *	
	clique (= fully connected)	$\boldsymbol{k = n - 1}$	

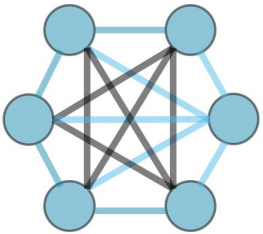
Whitney’s theorem

Whitney's theorem

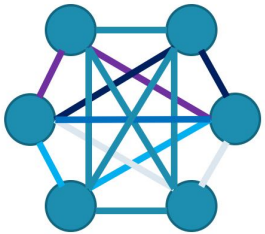


Degree connectivity

Whitney's theorem



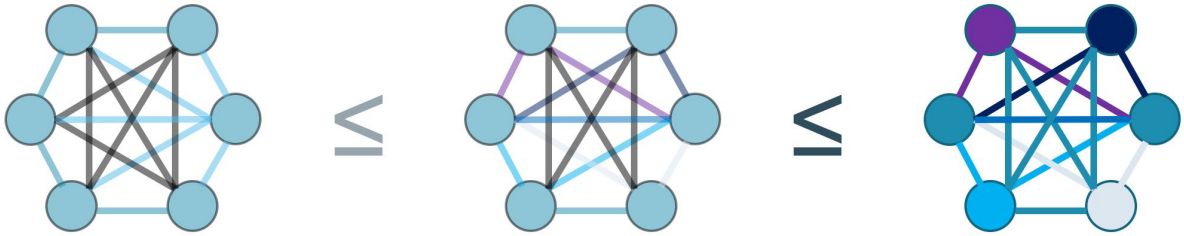
≤



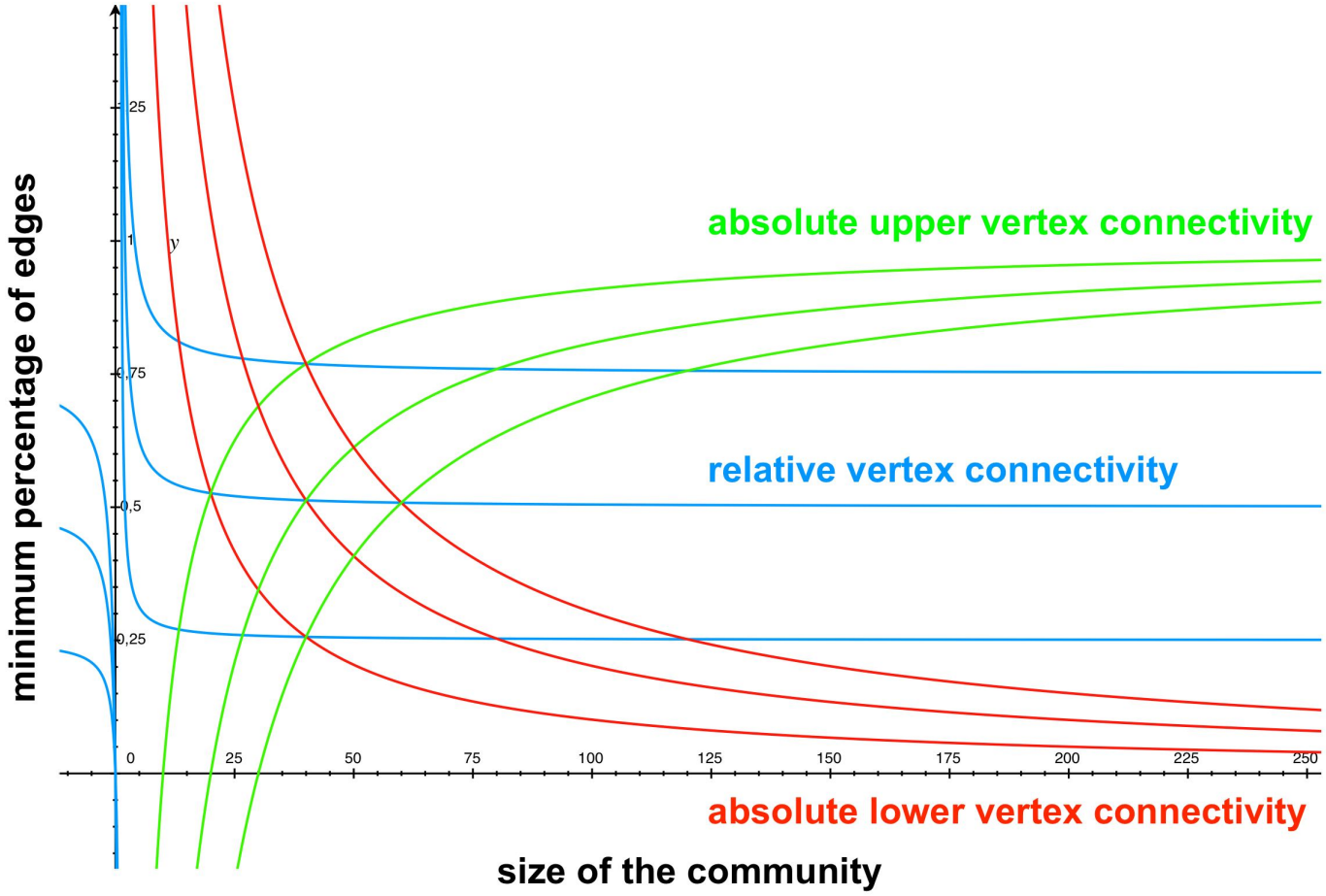
Degree connectivity

Edge connectivity

Whitney's theorem



preferred definition


minimum percentage of edges
absolute upper vertex connectivity
relative vertex connectivity
absolute lower vertex connectivity
size of the community
y
25
1
0,75
0,5
0,25
0
25
50
75
100
125
150
175
200
225
250

Guarantied properties

Distance	$dG(v1, v2) \leq \lfloor (n - 2) / x + 1 \rfloor$	**k-block**
Diameter	$diam(S) \leq \lfloor (n - 2) / x + 1 \rfloor$	
Domination	$\lvert D \rvert \leq n - x$	
Degree	$\delta(S) \geq x$	
Edge connectivity	$\lambda(S) \geq x$	
Vertex connectivity	$\kappa(S) \geq x$	
Edge density	$\rho(S) \geq x / (n - 1)$	
Distance	$dG(v1, v2) \leq \lfloor (n - 2) / (x (n - 1)) + 1 \rfloor$	**def. 7**
Diameter	$diam(S) \leq \lfloor (n - 2) / (x (n - 1)) + 1 \rfloor$	
Domination	$\lvert D \rvert \leq n - x (n - 1)$	
Degree	$\delta(S) \geq x (n - 1)$	
Edge connectivity	$\lambda(S) \geq x (n - 1)$	
Vertex connectivity	$\kappa(S) \geq x (n - 1)$	
Edge density	$\rho(S) \geq x$	
Distance	$dG(v1, v2) \leq \lfloor (n - 2) / (n - x) + 1 \rfloor$	**s-bundle**
Diameter	$diam(S) \leq \lfloor (n - 2) / (n - x) + 1 \rfloor$	
Domination	$\lvert D \rvert \leq x$	
Degree	$\delta(S) \geq n - x$	
Edge connectivity	$\lambda(S) \geq n - x$	
Vertex connectivity	$\kappa(S) \geq n - x$	
Edge density	$\rho(S) \geq (n - x) / (n - 1)$	

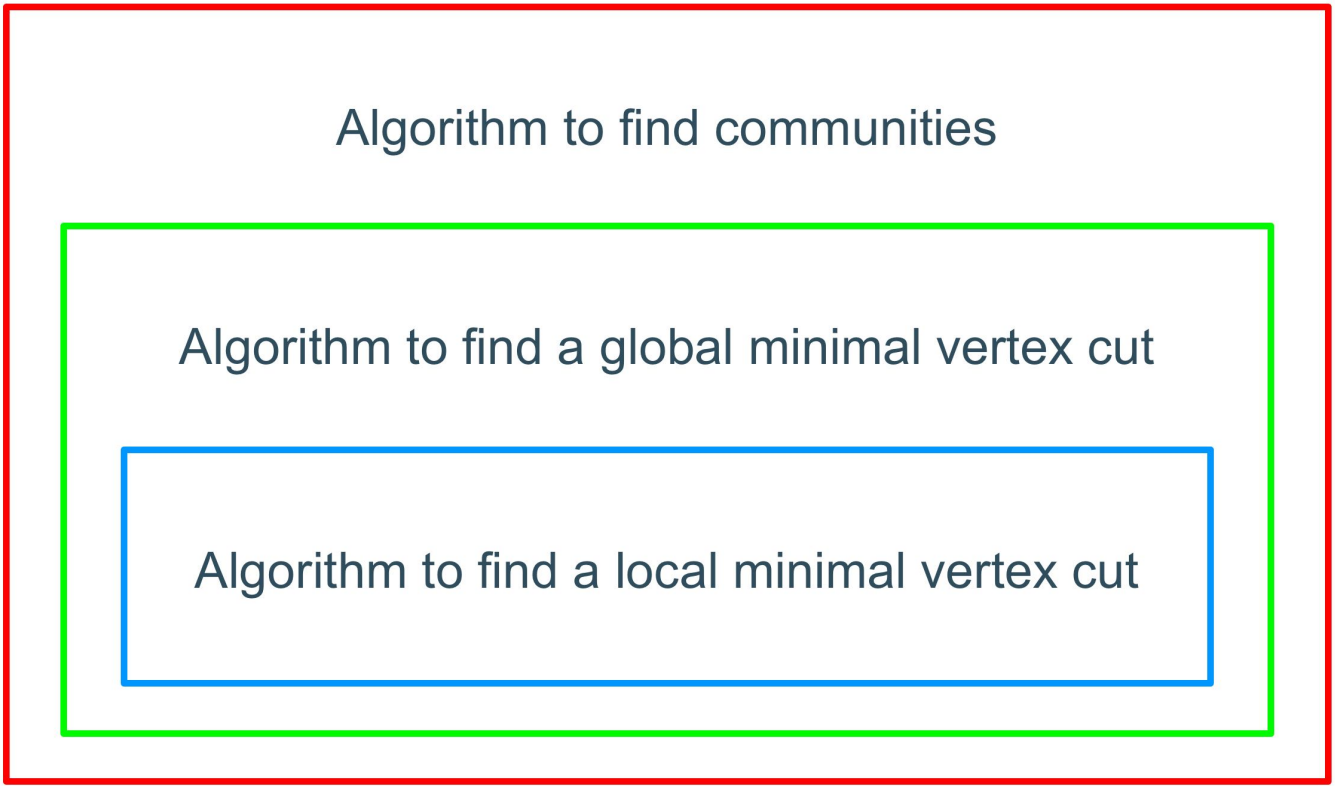
Research question 2:

a scalable algorithm

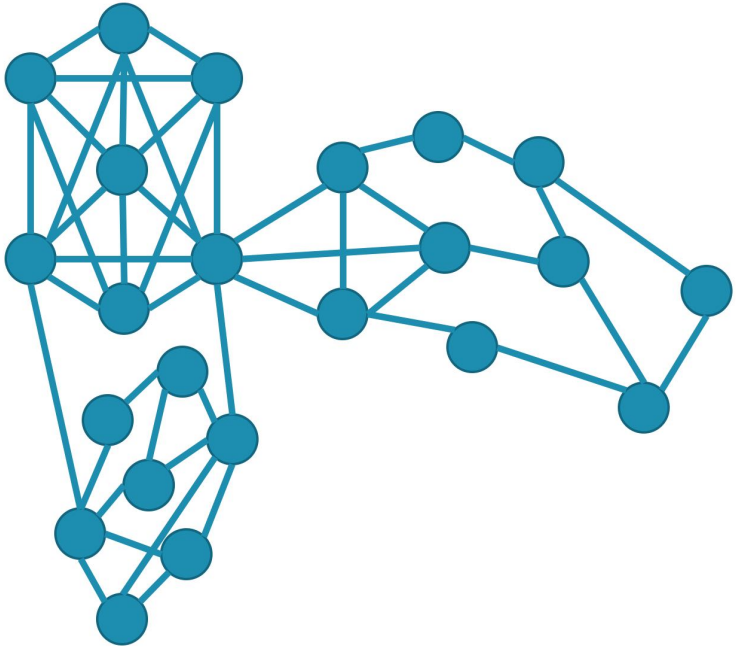
Algorithm to find communities

Algorithm to find communities

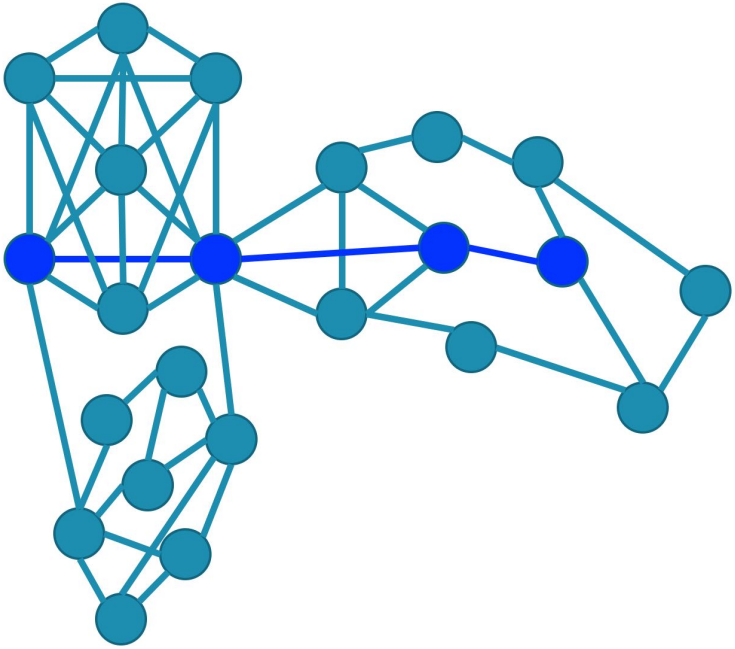
Algorithm to find a global minimal vertex cut


Algorithm to find communities
Algorithm to find a global minimal vertex cut
Algorithm to find a local minimal vertex cut

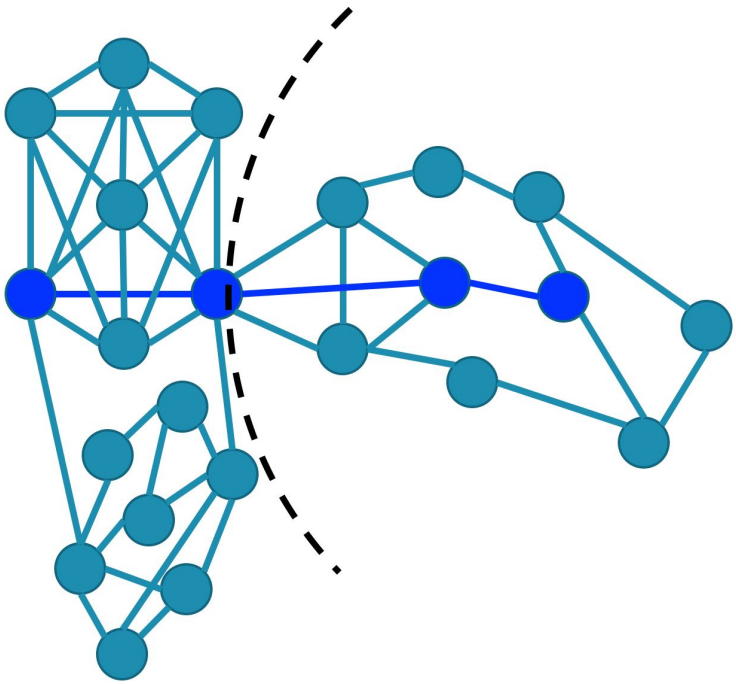
Local minimal flow-free vertex cut



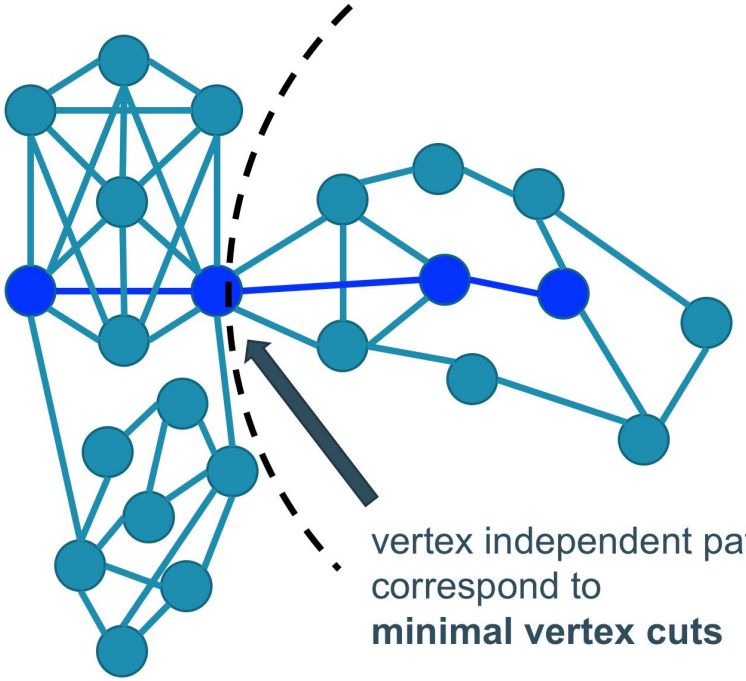
Local minimal flow-free vertex cut



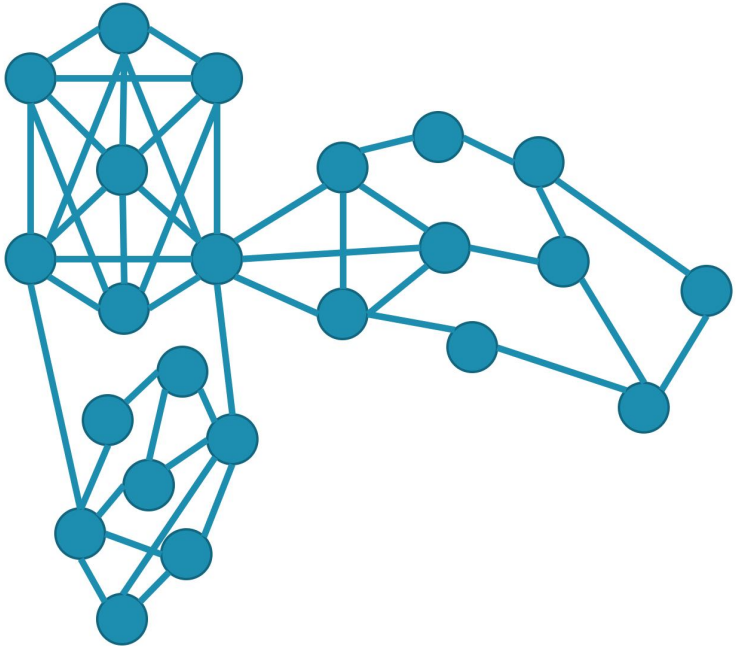
Local minimal flow-free vertex cut



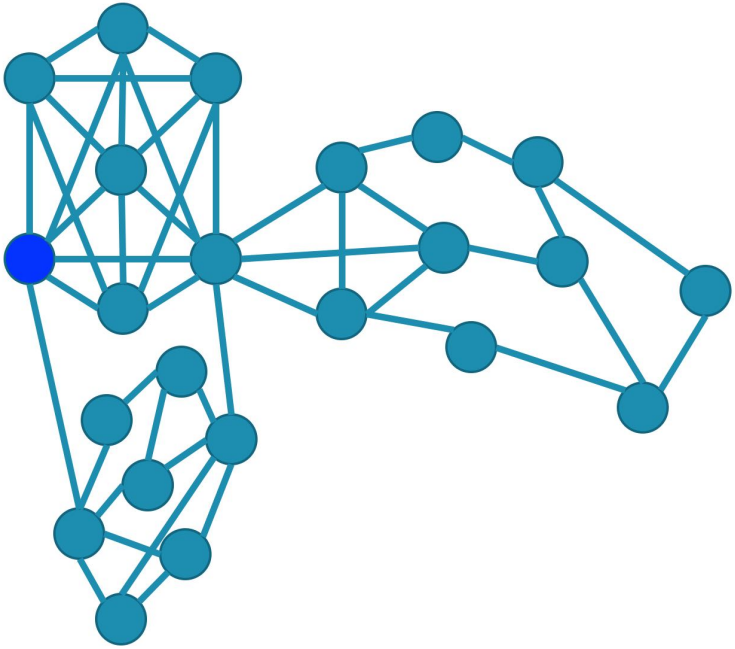
Local minimal flow-free vertex cut



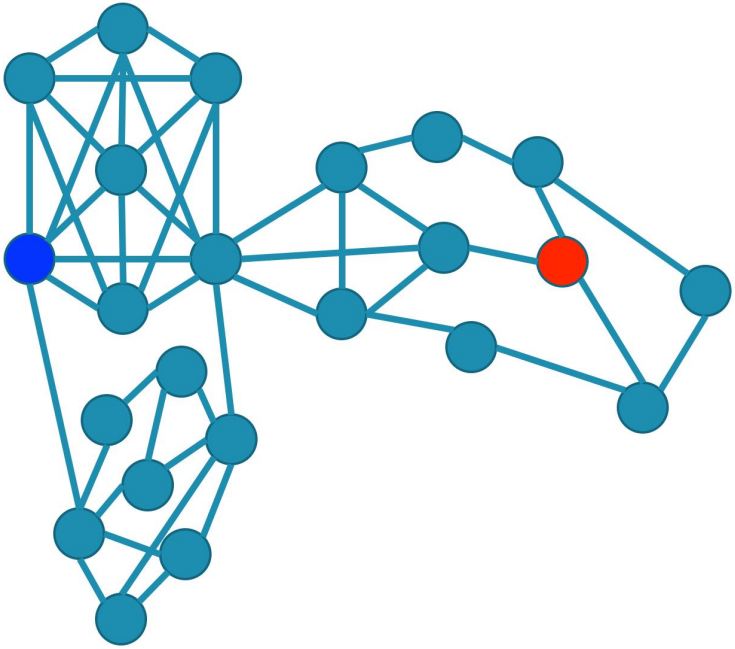
Local minimal flow-free vertex cut



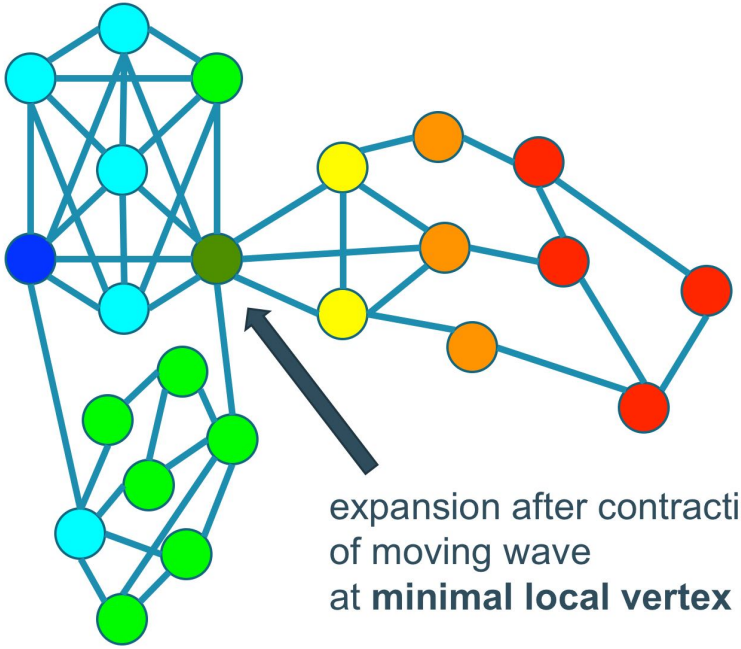
Local minimal flow-free vertex cut



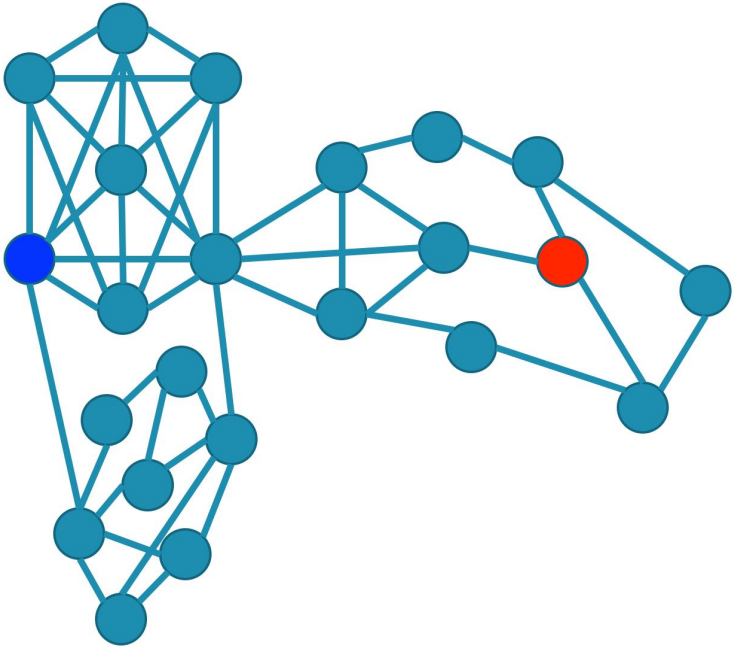
Local minimal flow-free vertex cut



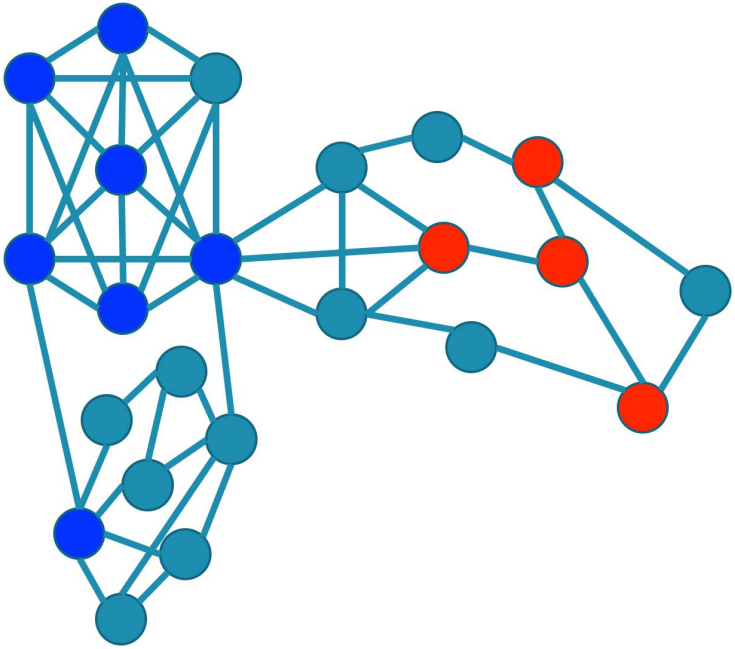
Local minimal flow-free vertex cut



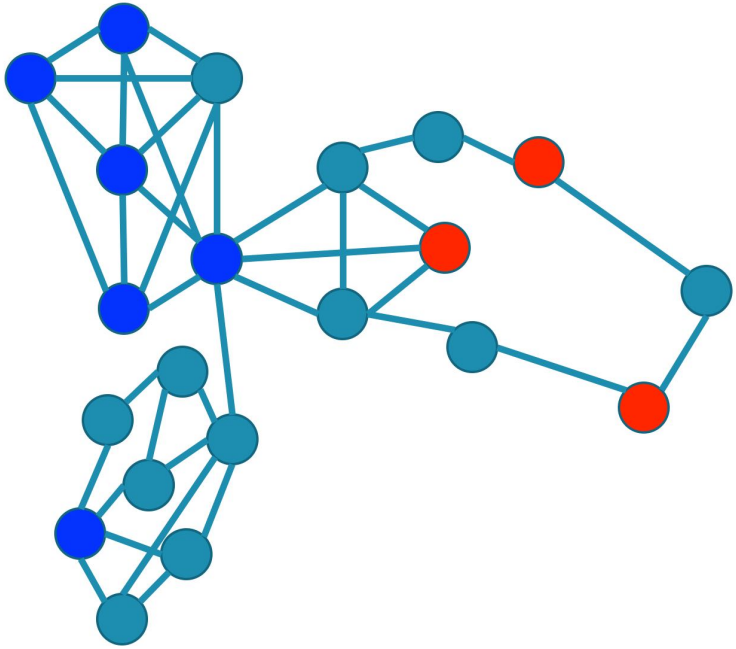
Local minimal flow-free vertex cut



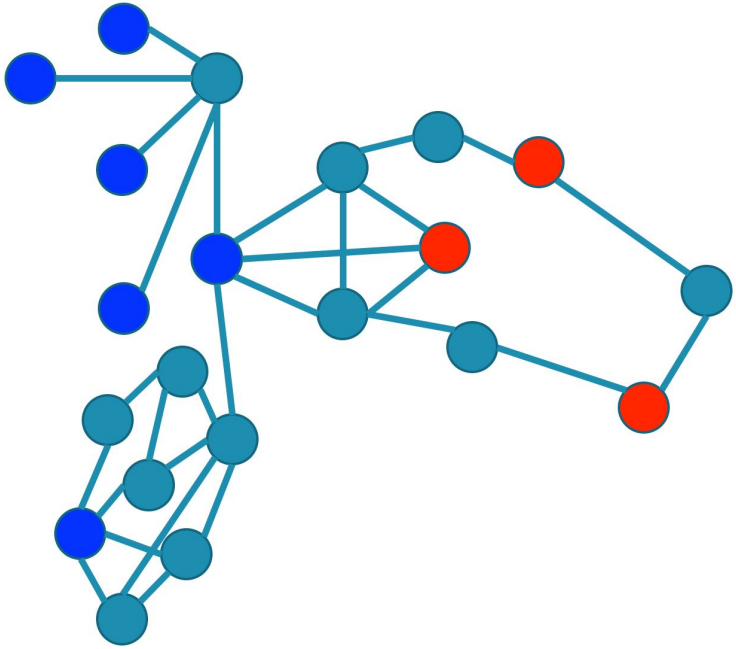
Local minimal flow-free vertex cut



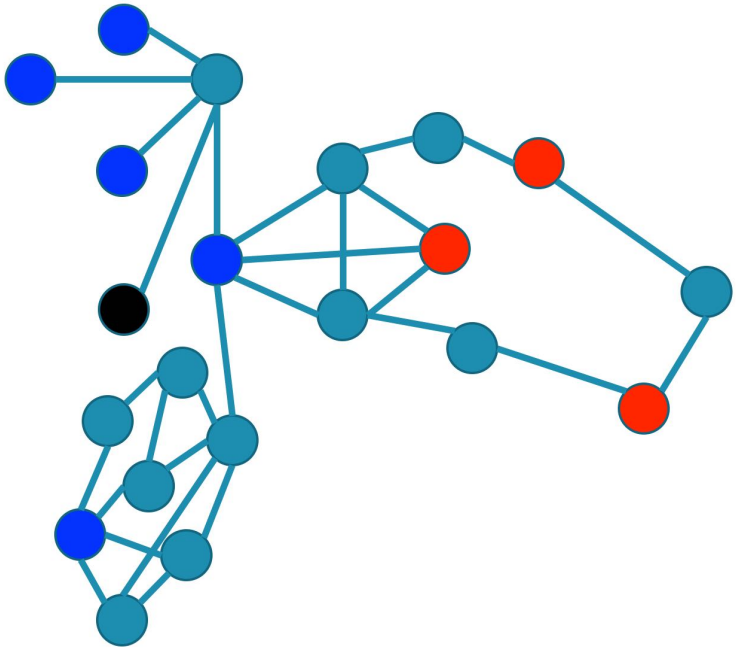
Local minimal flow-free vertex cut



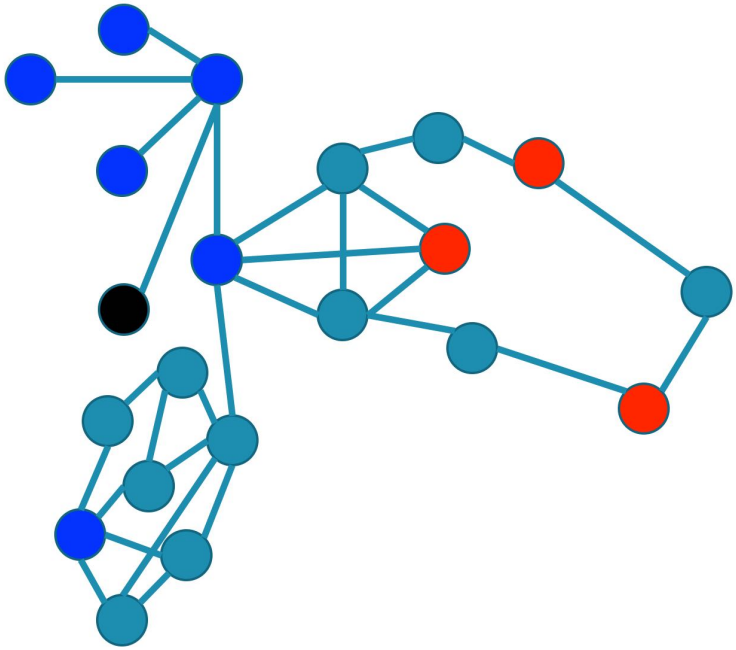
Local minimal flow-free vertex cut



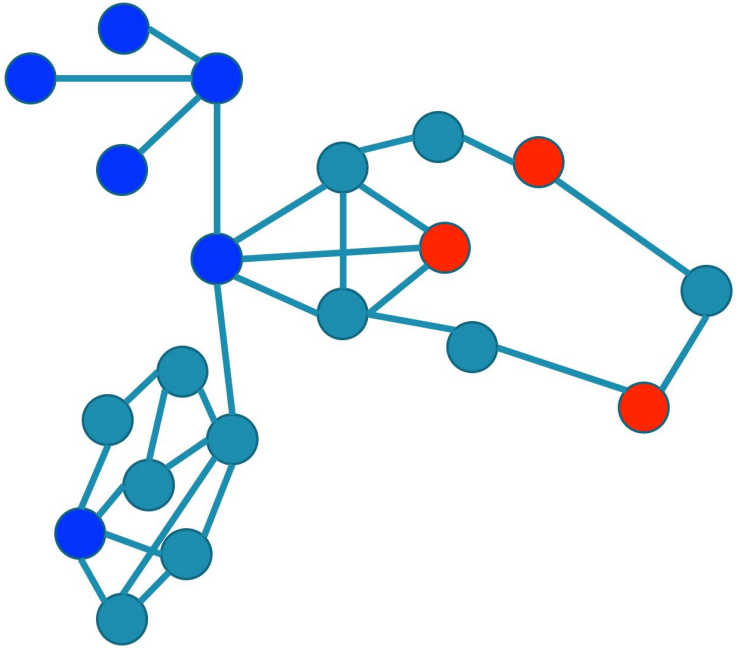
Local minimal flow-free vertex cut



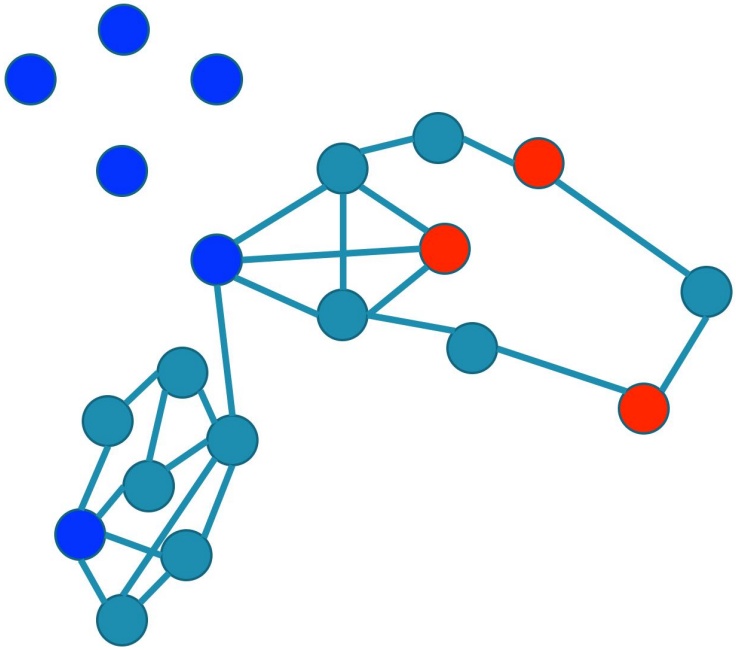
Local minimal flow-free vertex cut



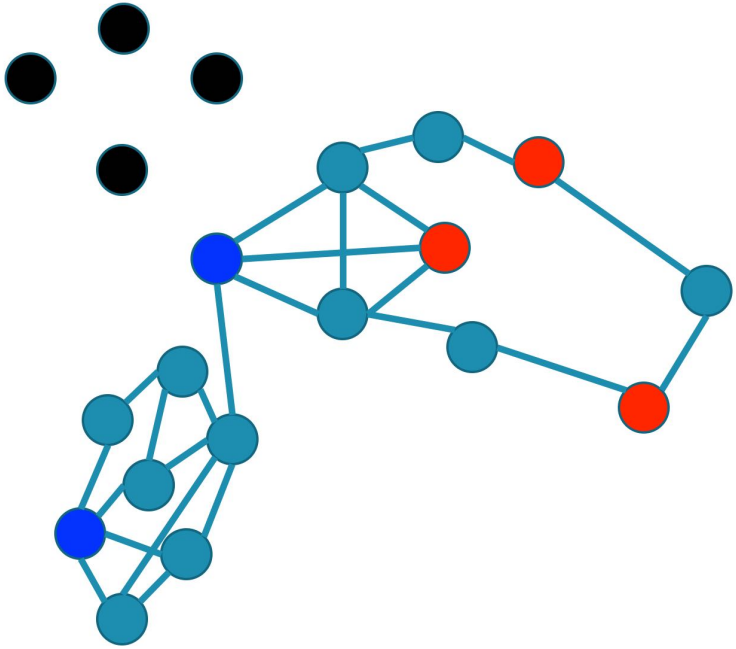
Local minimal flow-free vertex cut



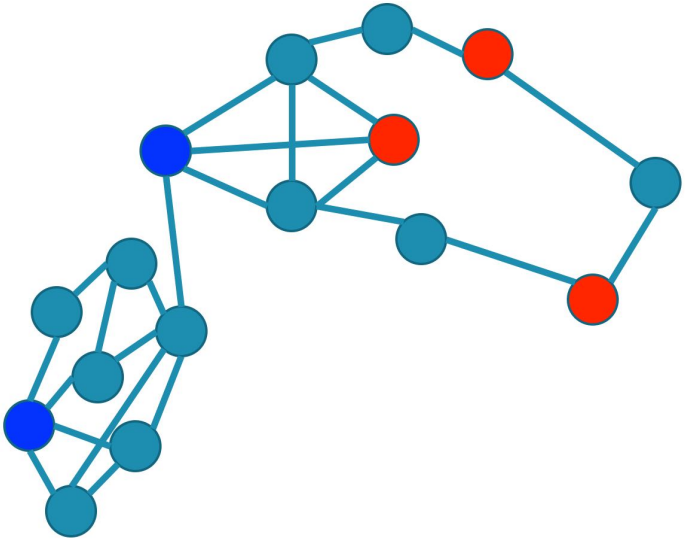
Local minimal flow-free vertex cut



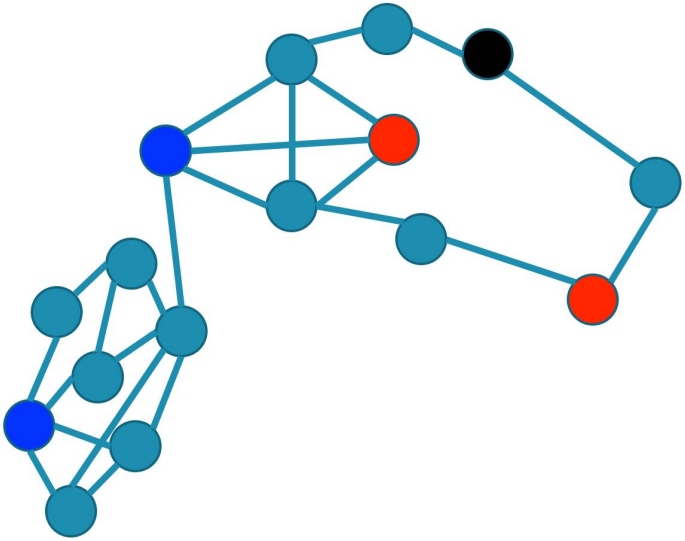
Local minimal flow-free vertex cut



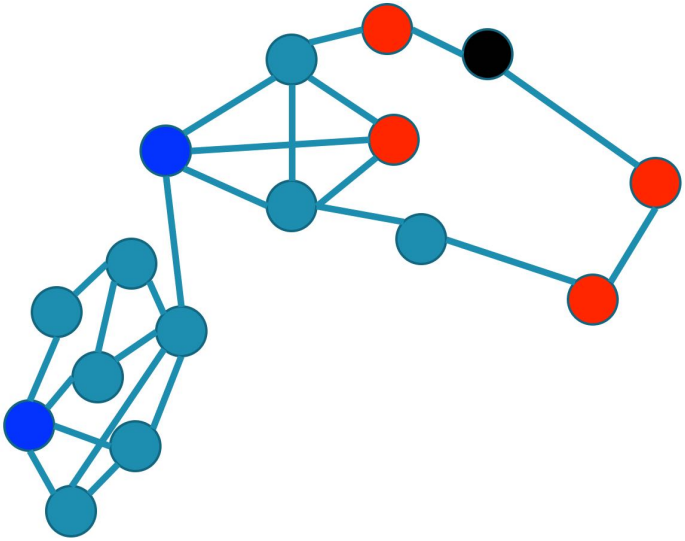
Local minimal flow-free vertex cut



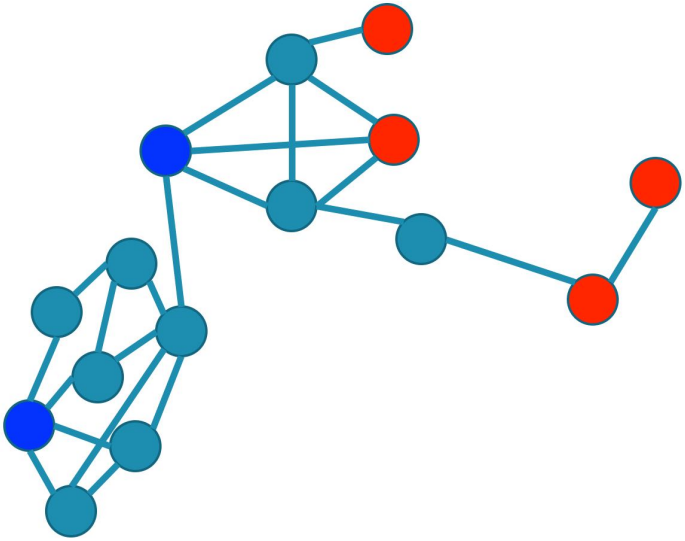
Local minimal flow-free vertex cut



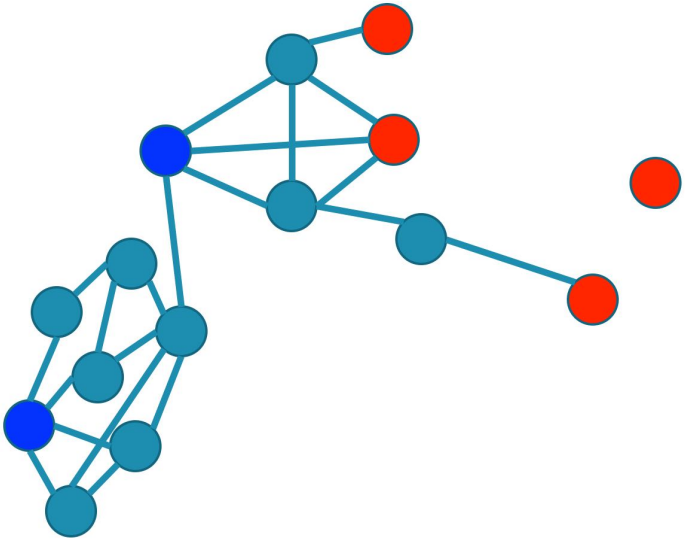
Local minimal flow-free vertex cut



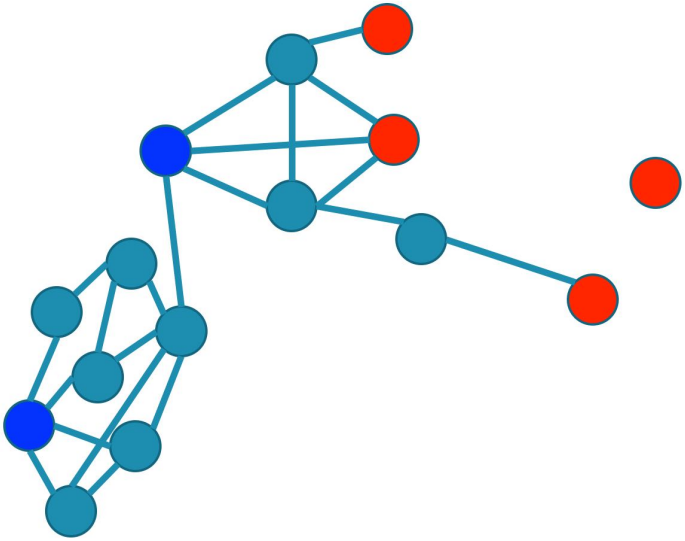
Local minimal flow-free vertex cut



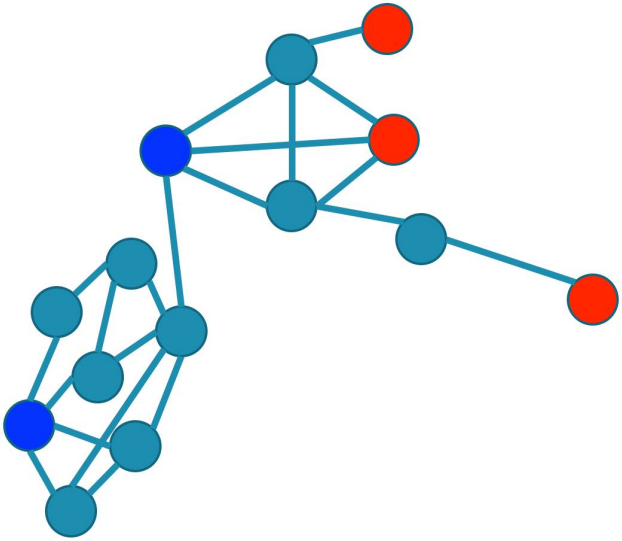
Local minimal flow-free vertex cut



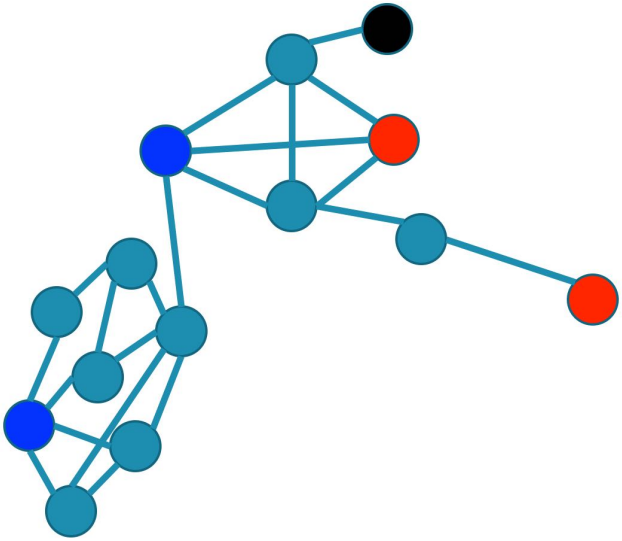
Local minimal flow-free vertex cut



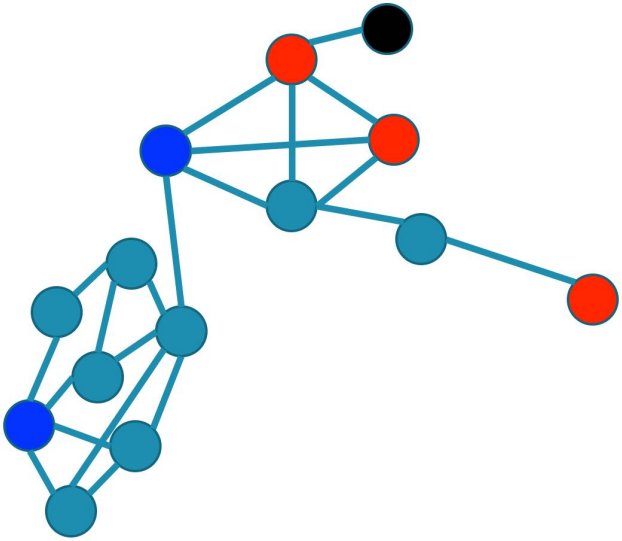
Local minimal flow-free vertex cut



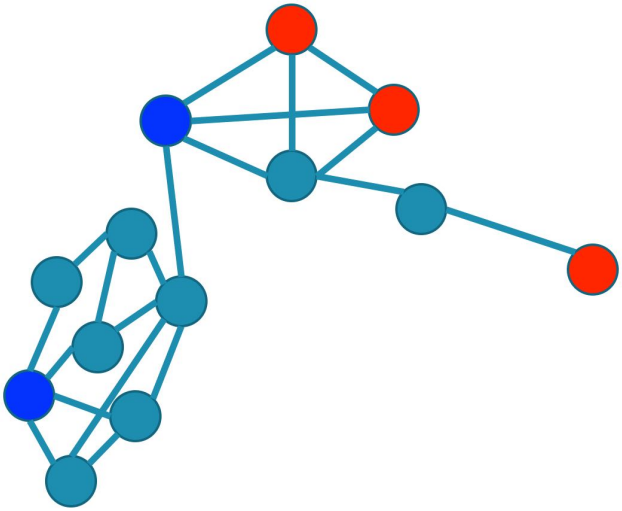
Local minimal flow-free vertex cut



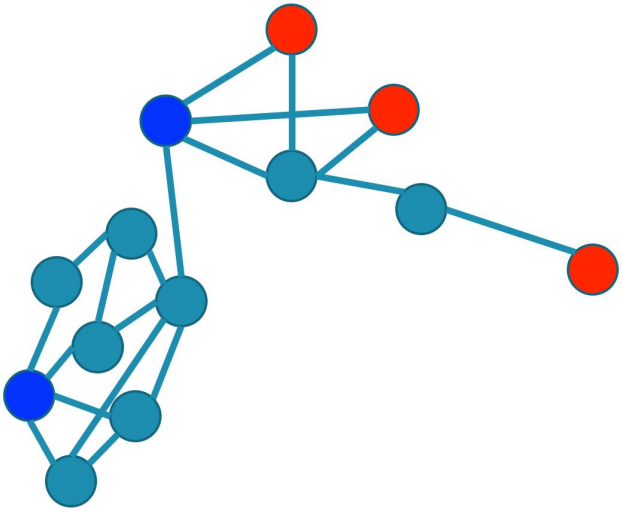
Local minimal flow-free vertex cut



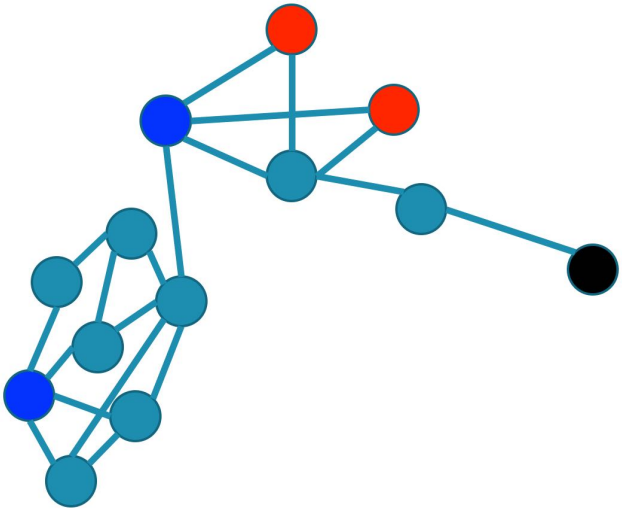
Local minimal flow-free vertex cut



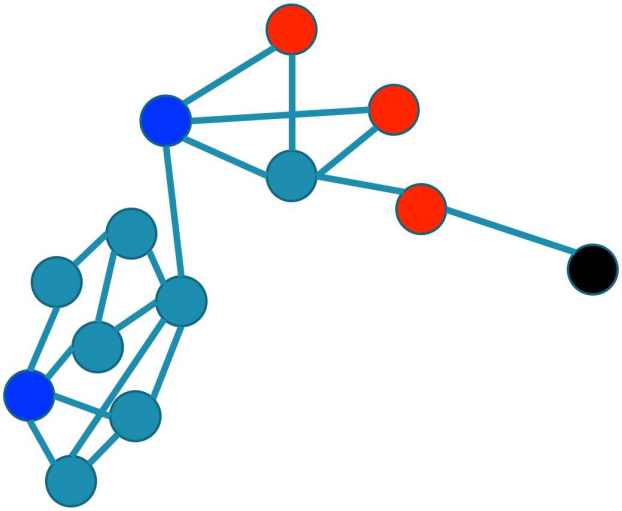
Local minimal flow-free vertex cut



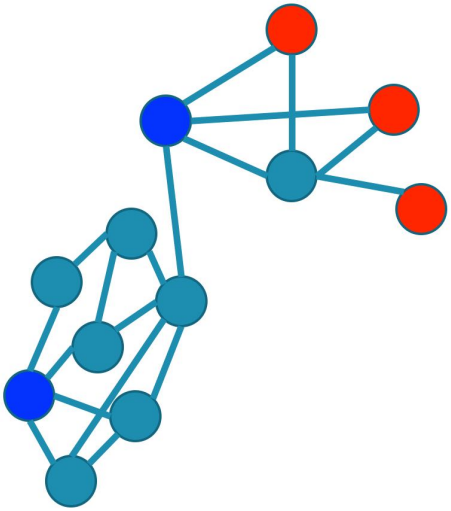
Local minimal flow-free vertex cut



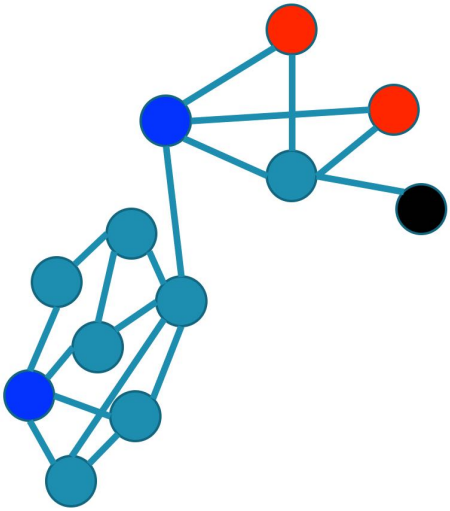
Local minimal flow-free vertex cut



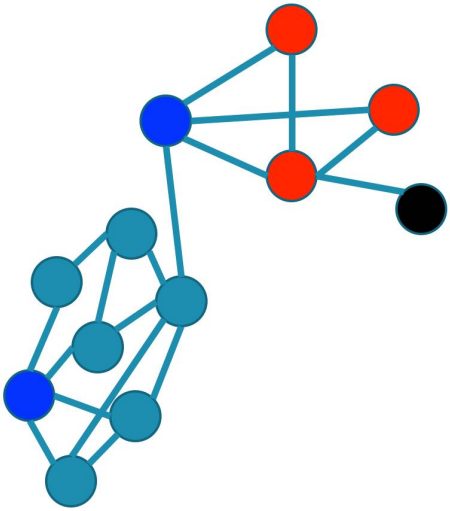
Local minimal flow-free vertex cut



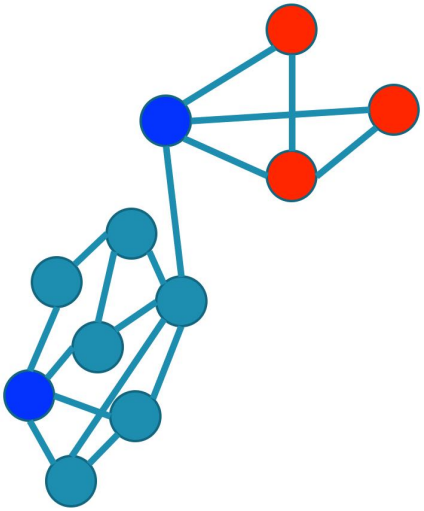
Local minimal flow-free vertex cut



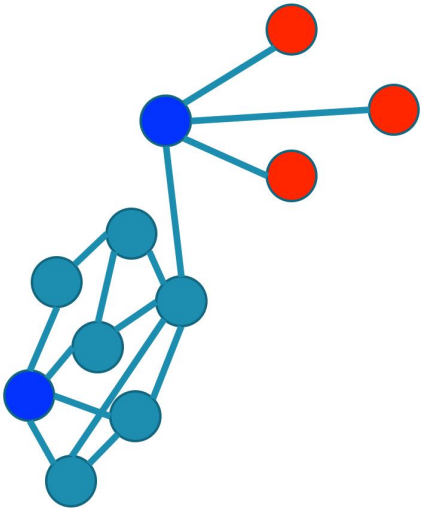
Local minimal flow-free vertex cut



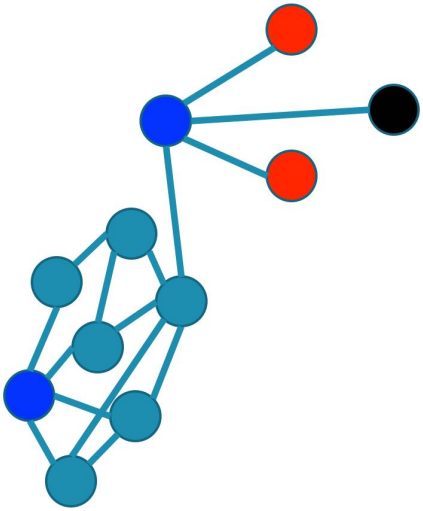
Local minimal flow-free vertex cut



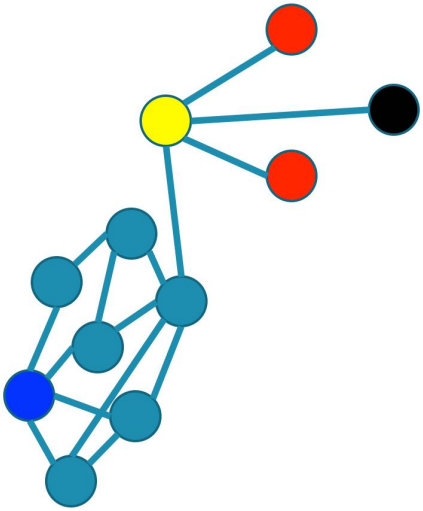
Local minimal flow-free vertex cut



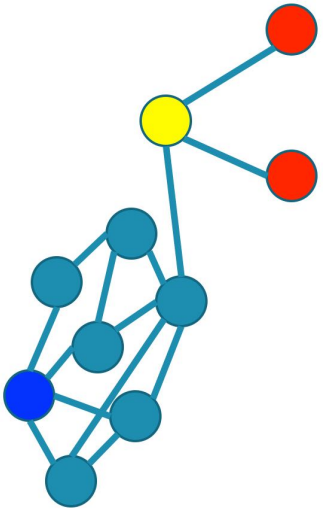
Local minimal flow-free vertex cut



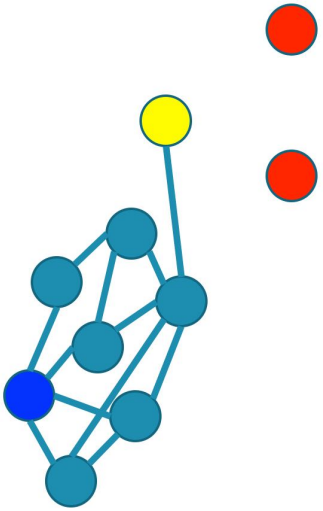
Local minimal flow-free vertex cut



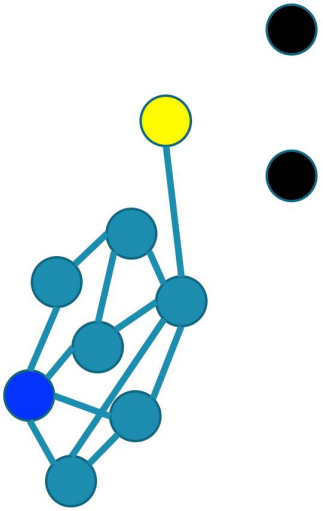
Local minimal flow-free vertex cut



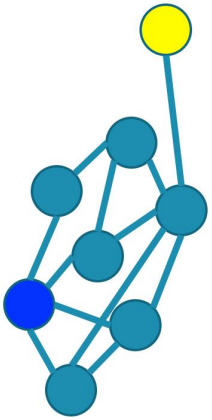
Local minimal flow-free vertex cut



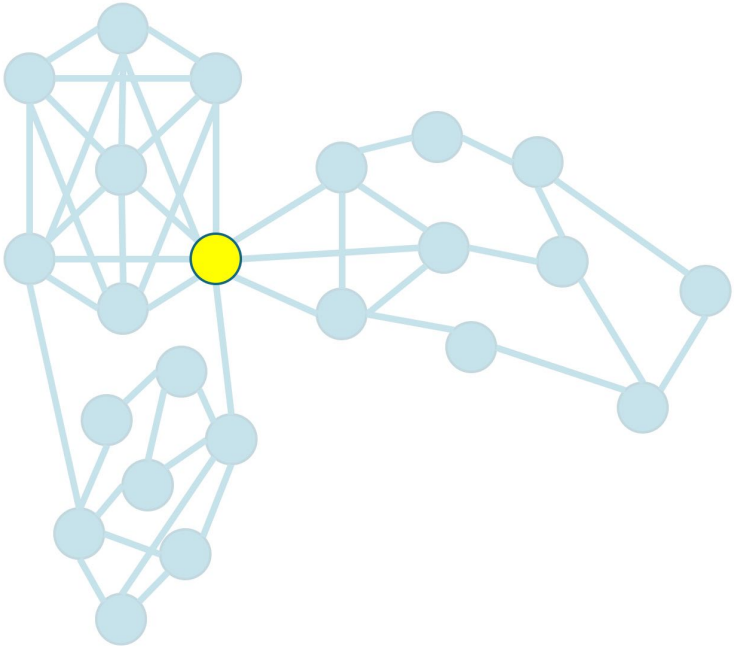
Local minimal flow-free vertex cut



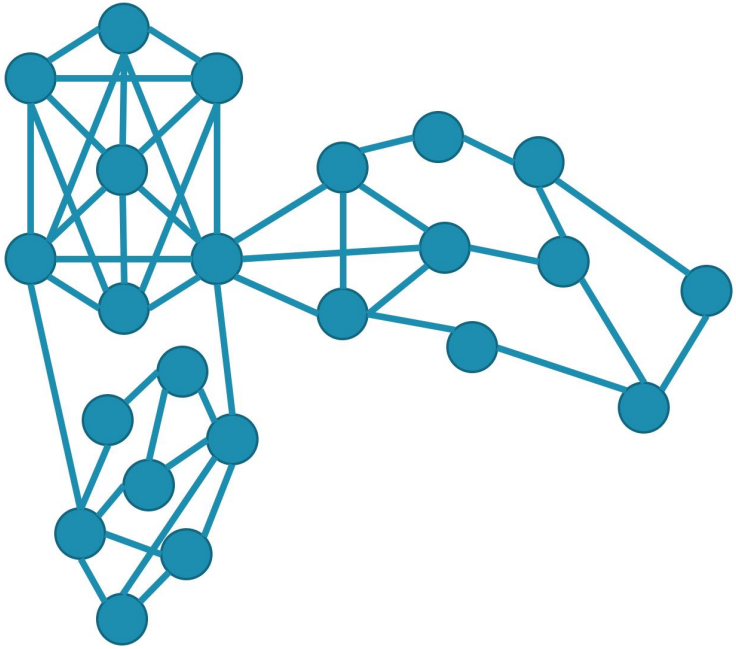
Local minimal flow-free vertex cut



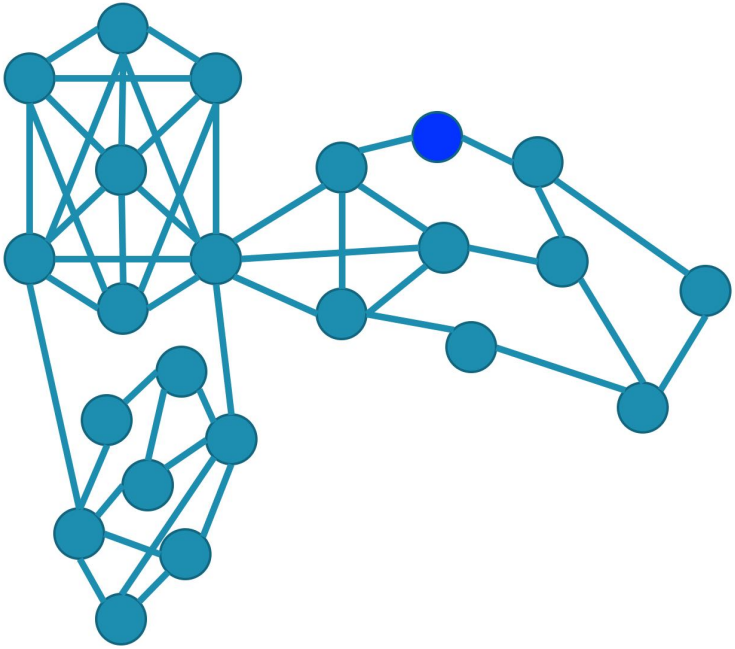
Local minimal flow-free vertex cut



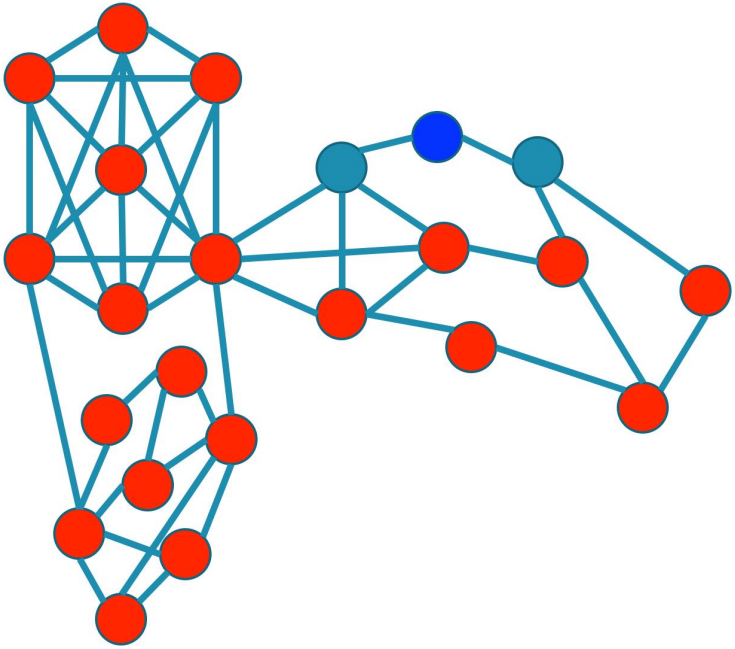
Global minimal vertex cut



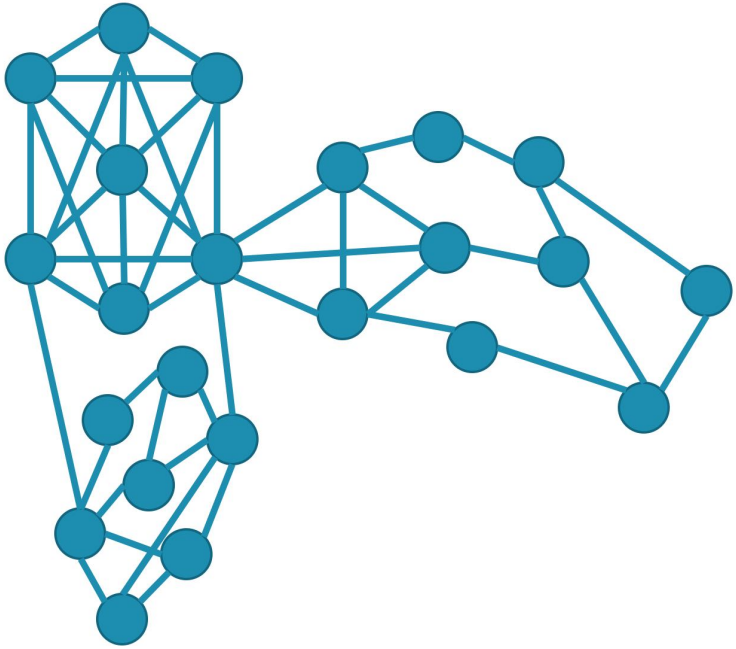
Global minimal vertex cut



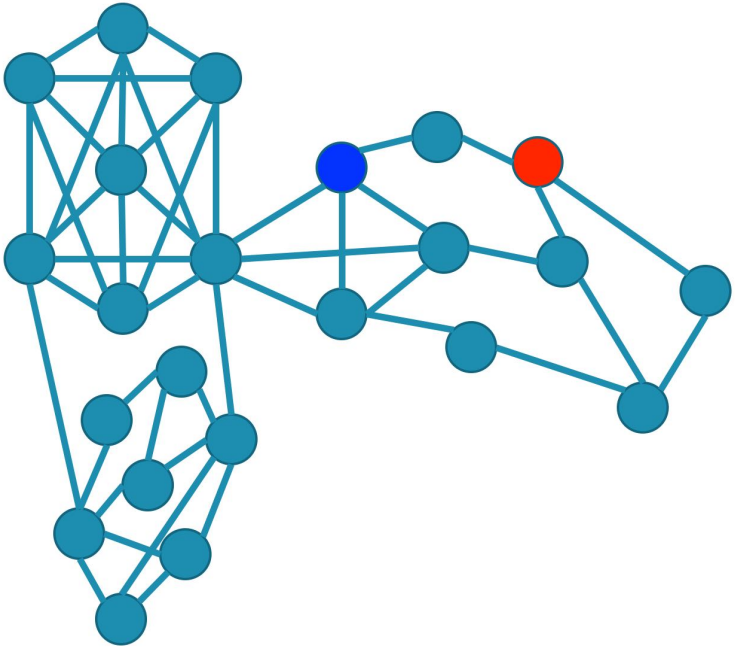
Global minimal vertex cut



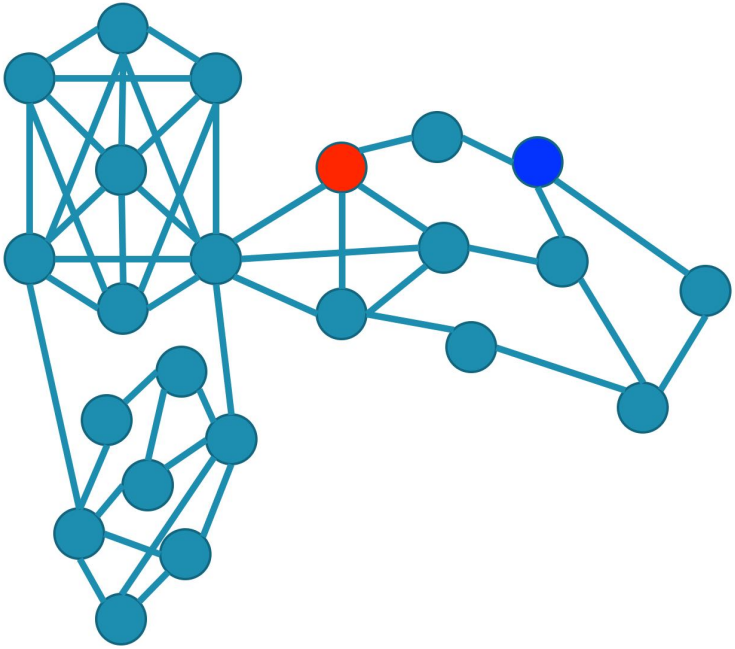
Global minimal vertex cut



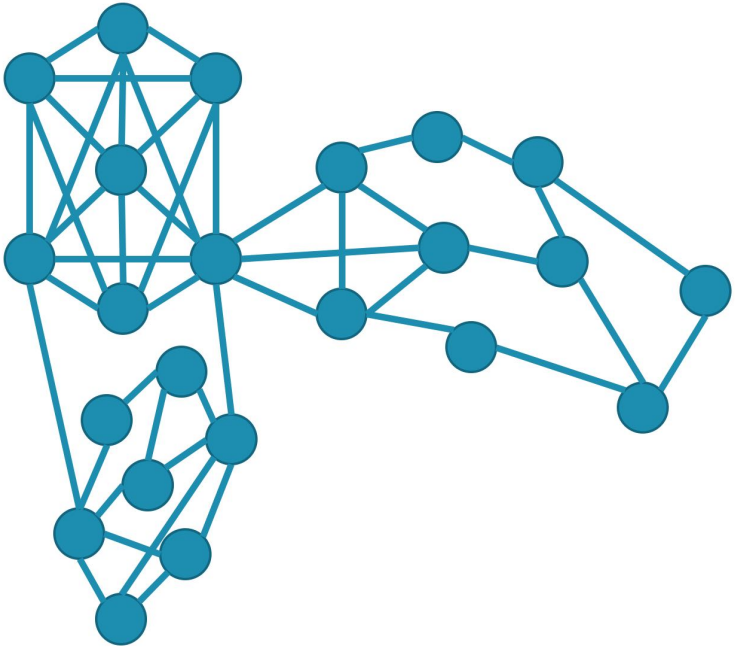
Global minimal vertex cut



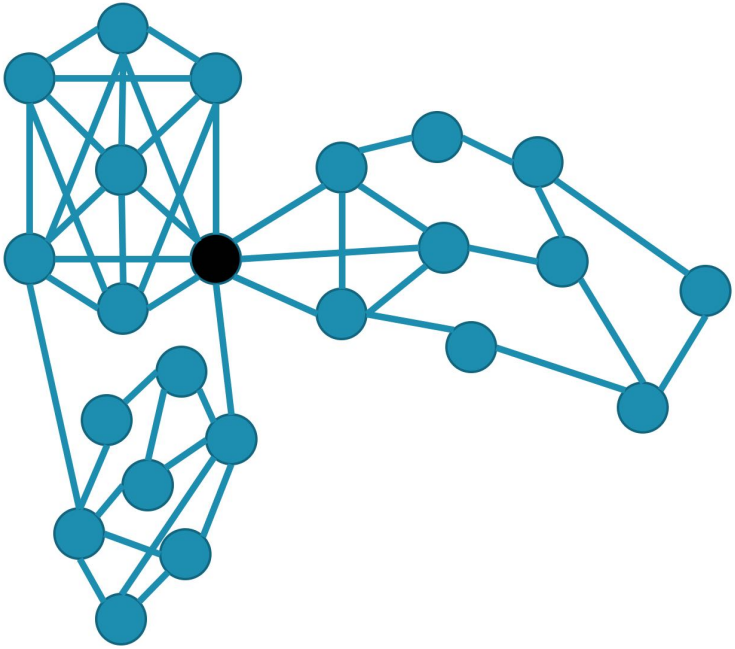
Global minimal vertex cut



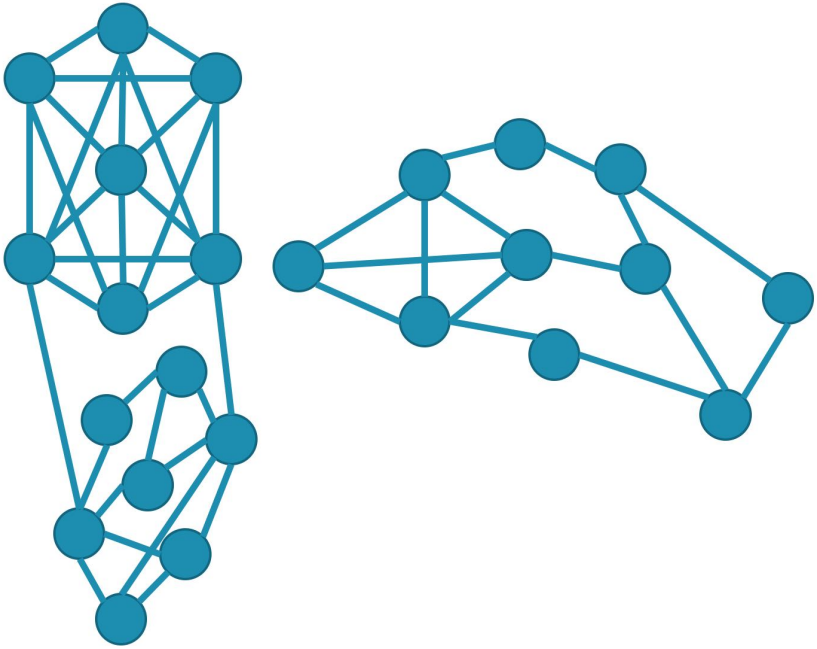
Find communities (Moody & White)



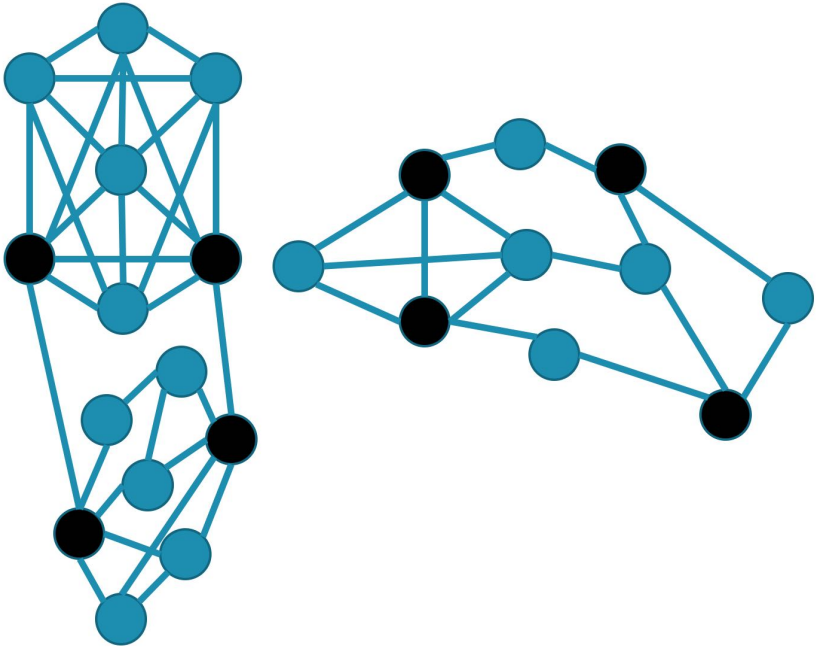
Find communities (Moody & White)



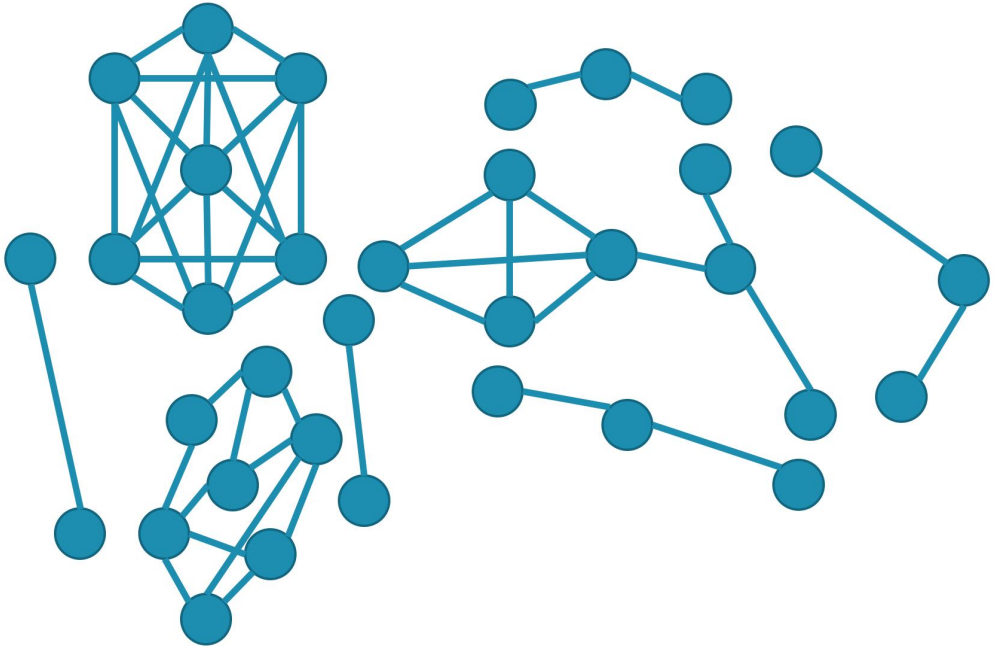
Find communities (Moody & White)



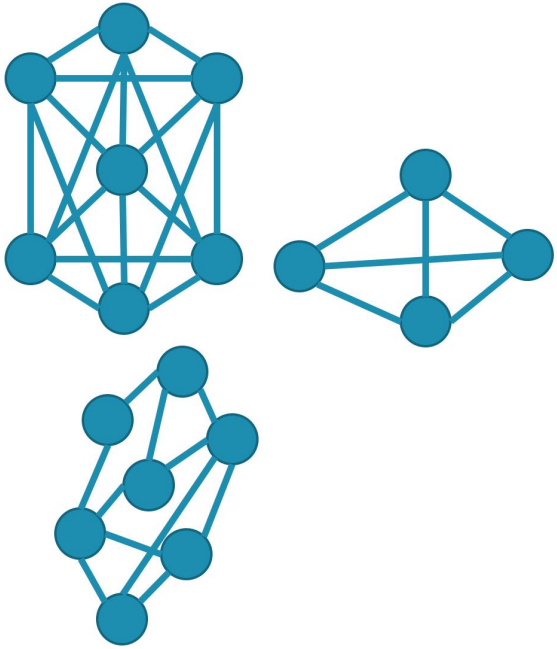
Find communities (Moody & White)



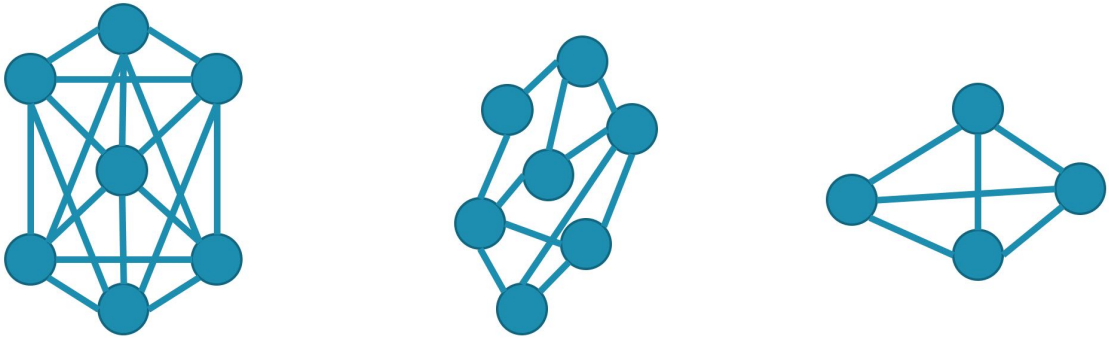
Find communities (Moody & White)

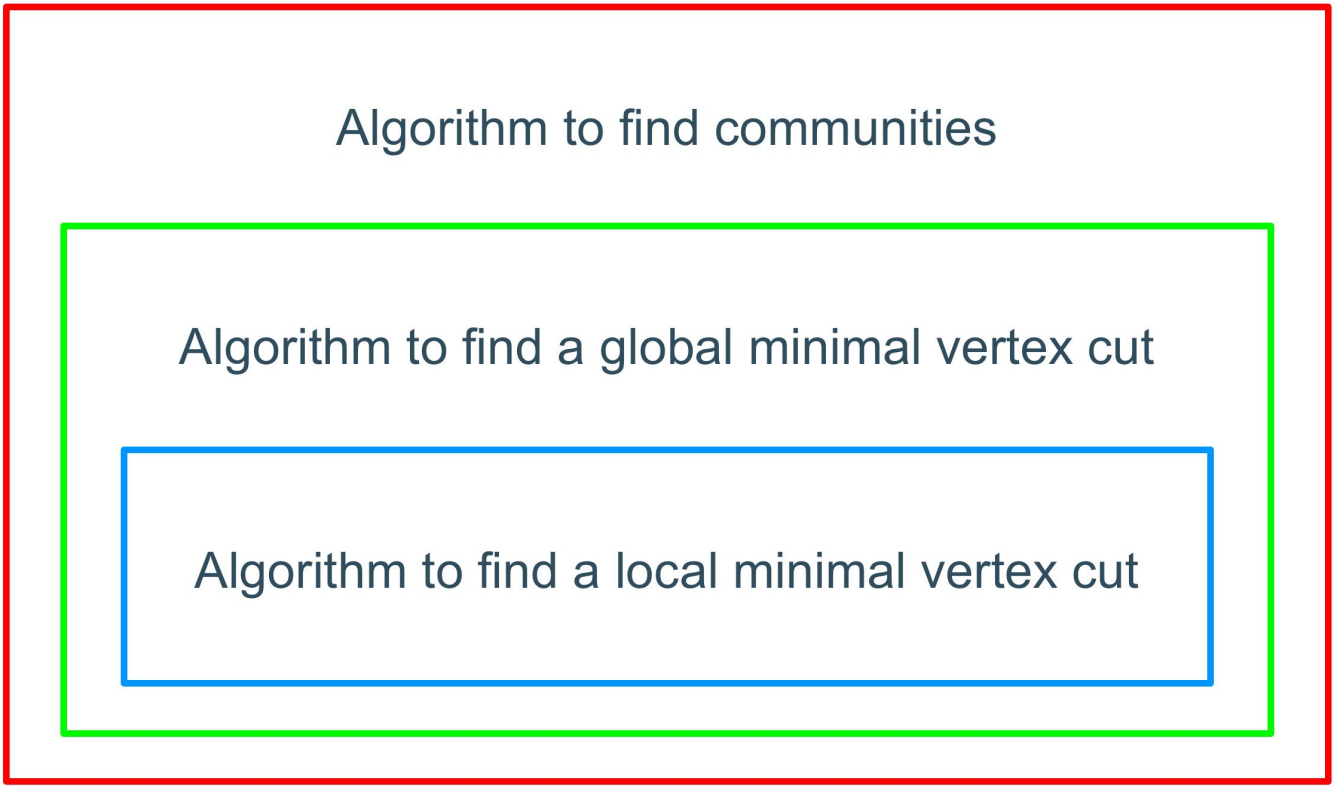


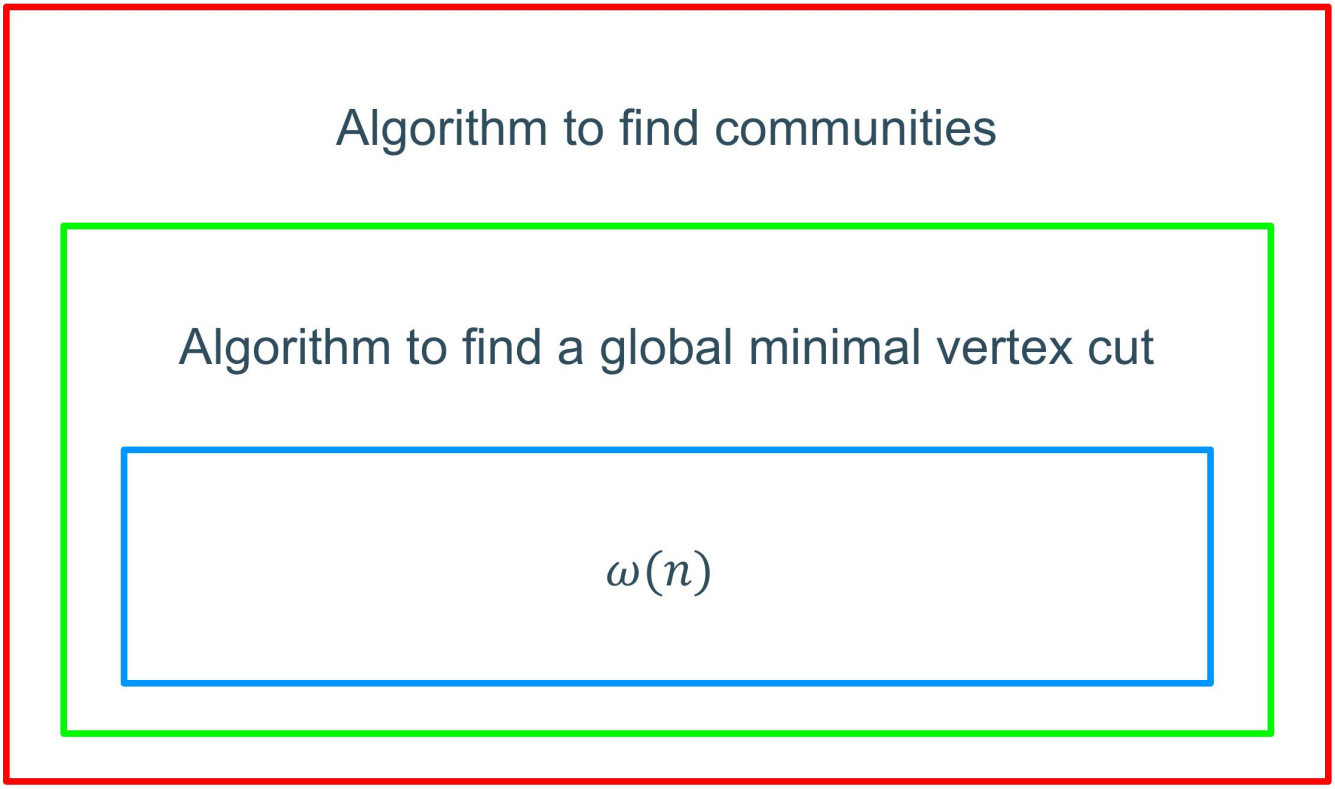
Find communities (Moody & White)

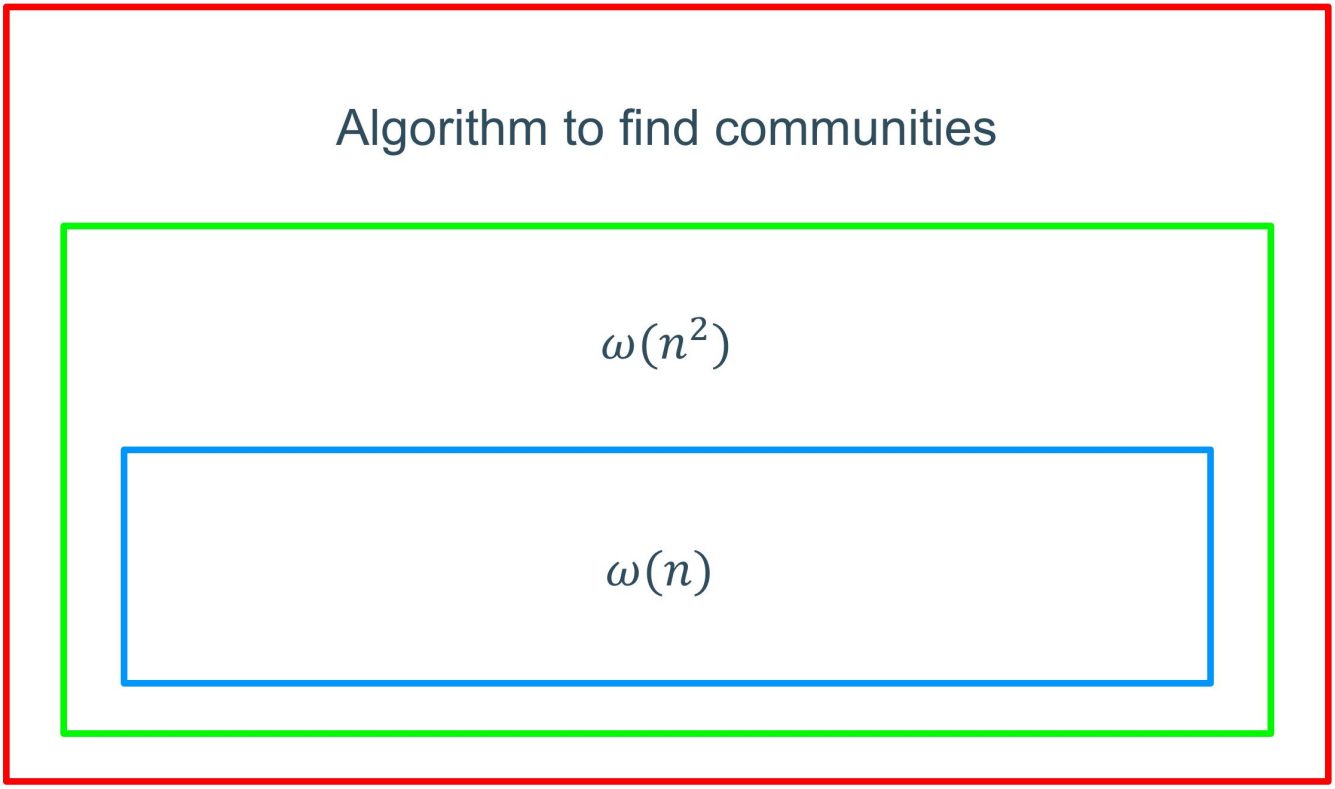


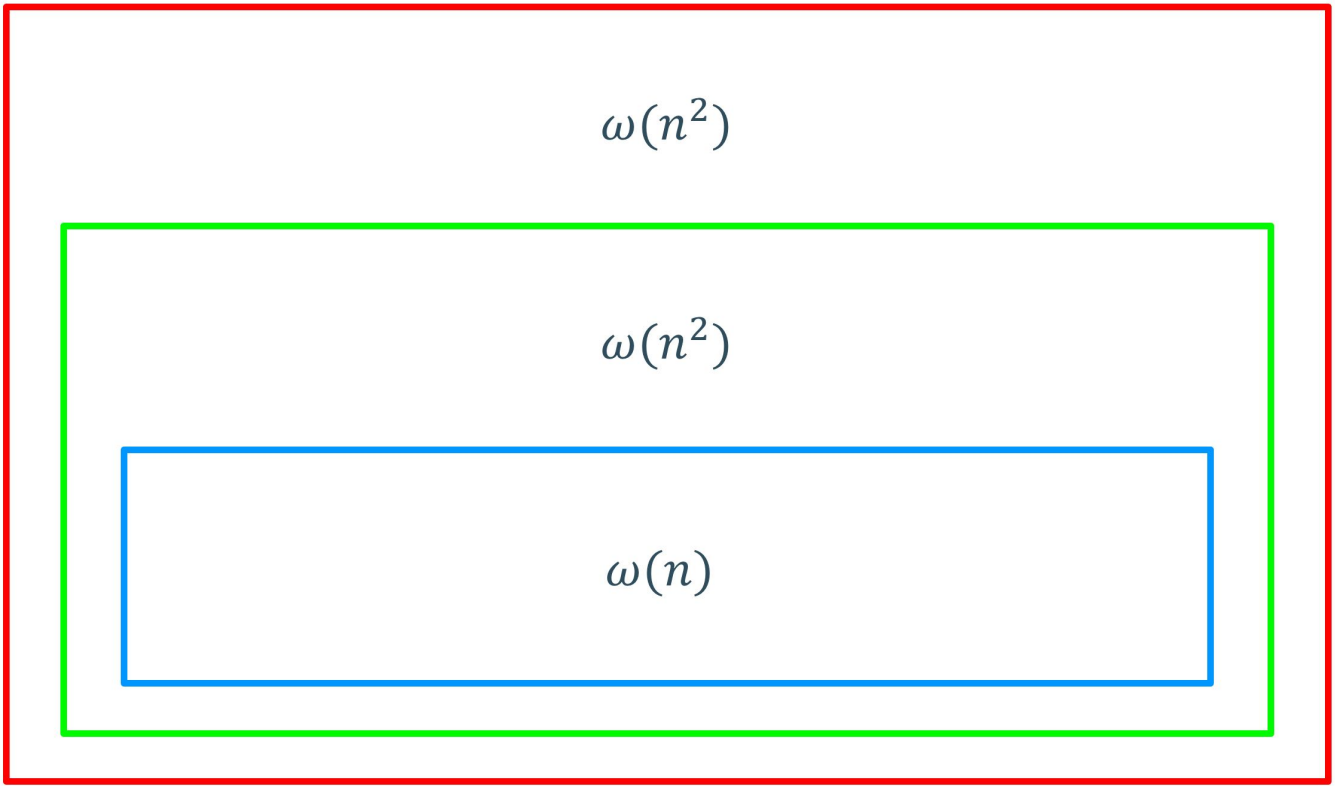
Find communities (Moody & White)




Algorithm to find communities
Algorithm to find a global minimal vertex cut
Algorithm to find a local minimal vertex cut


Algorithm to find communities
Algorithm to find a global minimal vertex cut
$\omega(n)$


Algorithm to find communities
$\omega(n^2)$
$\omega(n)$


$\omega(n^2)$
$\omega(n^2)$
$\omega(n)$

Thank you!

Any questions?

Source code at Github:
https://github.com/MCMXCVII/scale-independent-communities-flow-free-vertex-cut